StemExpert

I

Intermediate Handboek

Leerstof: uitleg, uitgewerkte voorbeelden, proeven, naslag en verbanden.

14-16 jaar **40 hoofdstukken** **schaal 3–7**

Brecht Corbeel · schoolium.me

StemExpert Intermediate — Handboek

Auteur: Brecht Corbeel
Uitgave: schoolium.me
Reeks: StemExpert — STEM in drie niveaus (Fundamental, Intermediate, Expert)

Voor wie: Voor wie Fundamental beheerst (tweede graad secundair, STEM-richtingen). Van kwalitatief naar kwantitatief: rekenen met modellen, meetfouten, Python en C++, en de eerste regelkringen.

Wiskunde op dit niveau: algebra, lineaire, kwadratische en exponentiële functies, rechthoekige driehoeken, vectoren

Habitat-laag: De klimaat- en energiemodule: eigen zonnestroom met batterij, een waterkringloop met hydroponie, een geregeld klimaat en datalogging.

Dit boek hoort bij een handboek, een werkboek en een oplossingenboek per niveau, en bij de offline website van StemExpert. De StemExpert-schaal (1 tot 9) loopt over de drie niveaus heen. Druk af zonder schaling (100 %).

Inhoud

I1 · Onderzoeken met data

I1 Meetfouten en nauwkeurigheid 5
I2 Experimenten ontwerpen en rapporteren 29
I3 Verbanden in data 46

I2 · Wiskunde als taal van STEM

I4 Functies als modellen 71
I5 Groei en verval: exponentiële modellen 95
I6 Goniometrie en vectoren 118

I3 · Mechanica

I7 Beweging beschrijven 143
I8 De wetten van Newton 167
I9 Arbeid, energie en vermogen 192
I10 Moment en evenwicht 213
I11 Druk en hydraulica 241

I4 · Chemie en materialen

I12 Atomen en het periodiek systeem 269
I13 Chemische bindingen en stofeigenschappen 294
I14 Reacties en stoichiometrie 316
I15 Zuren, basen en pH 340
I16 Redox en batterijen 365
I17 Polymeren en composieten 388

I5 · Energie en warmte

I18 Warmte berekenen 411
I19 Warmtetransport en isolatie 435
I20 Zonne-energie dimensioneren 457

I6 · Elektronica

I21 De wetten van Kirchhoff 479
I22 De spanningsdeler en sensoren 501
I23 Halfgeleiders: diode, led en transistor 523
I24 Condensatoren en de RC-tijd 550
I25 Van analoog naar digitaal 574
I26 Motoren en actuatoren 596

I7 · Programmeren

I27 Python: de basis 618
I28 Functies, lijsten en data in Python 651
I29 Microcontrollers programmeren in C++ 683
I30 Datalogging en communicatie 717

I8 · Regelen en robotica

I31 Regelsystemen 758
I32 Toestandsmachines 782
I33 Een mobiele robot 806

I9 · Leven en biotechnologie

I34 Enzymen en stofwisseling 828

I35 Micro-organismen en biotechnologie 853

I36 Hydroponie en voedselproductie 883

I10 · Ontwerpen en data

I37 Parametrisch ontwerpen 906

I38 Duurzaam ontwerpen 934

I39 Data visualiseren en eerste stappen in AI 956

I11 · Habitat I: energie, water en klimaat

I40 Integratieproject: de klimaatmodule 982

I INTERMEDIATE · DEEL I1 · ONDERZOEKEN MET DATA

HANDBOEK · HOOFDSTUK 1

Meetfouten en nauwkeurigheid

Waarom elke meting een marge heeft, en hoe je die eerlijk in een getal zet

Geen enkele meting is perfect: niet die van jouw lat, niet die van de sensoren in de meetkas en niet die van een ruimtetelescoop. In dit hoofdstuk leer je fouten herkennen, ze uitdrukken als waarde ± fout, rekenen met meetwaarden zonder schijnprecisie, en een meetreeks samenvatten met het gemiddelde en de standaardafwijking. Daarmee beantwoord je een echte vraag: hoe ver kan je een goedkope sensor vertrouwen?

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · ± 7 lestijden

O Onderzoeken en meten · W Wiskundig modelleren

Drie thermometers naast elkaar, drie verschillende antwoorden. Welke vertrouw je?

Na dit hoofdstuk

- Je kan systematische en toevallige fouten herkennen, hun oorzaak benoemen en zeggen hoe je ze verkleint.
- Je kan uitleggen wat nauwkeurig en precies betekenen, en waarom resolutie iets anders is dan nauwkeurigheid.
- Je kan een meetresultaat correct noteren als (waarde ± fout) eenheid, correct afgerond.
- Je kan de absolute en de relatieve fout berekenen en ermee instrumenten en metingen vergelijken.
- Je kan beduidende cijfers tellen en de regels toepassen als je met meetwaarden rekent.
- Je kan de fout op een berekende grootheid schatten met de regels voor som, verschil, product, quotiënt en macht.
- Je kan gemiddelde, bereik, mediaan en standaardafwijking berekenen en uitleggen wat ze zeggen.
- Je kan een uitschieter opsporen en onderbouwd beslissen wat je ermee doet.

UITLEG · 1.1

Wat je al weet, en wat nu komt

1/18

In Fundamental leerde je meten als een vakman. Een meetresultaat is een **grootheid** met een **getal** en een **eenheid** (hoofdstuk F3). Je koos het juiste instrument, las recht van voren af om parallax te

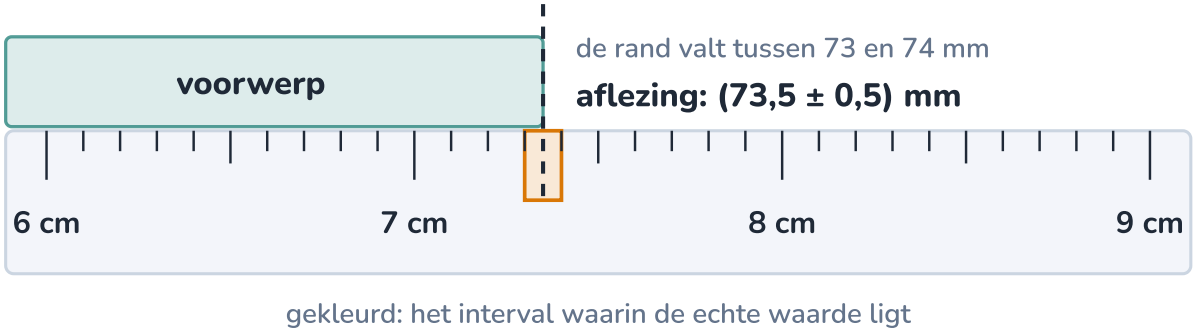
vermijden en wist dat een lat met millimeterverdeling niet preciezer meet dan een halve millimeter. In hoofdstuk F2 herhaalde je een proef en nam je het gemiddelde, omdat één meting niets bewijst.

Nu maak je die onzekerheid **zichtbaar in een getal**. Niet 'ongeveer 73 mm', maar $l = (73{,}5 \pm 0{,}5)\ \mathrm{mm}$. Je leert in dit hoofdstuk

- welke soorten fouten er bestaan en hoe je ze opspoort;
- hoe je een fout absoluut en relatief uitdrukt en correct noteert;
- hoeveel cijfers een resultaat verdient, ook na een berekening;
- hoe fouten doorwerken als je met meetwaarden rekent;
- hoe je een meetreeks samenvat met gemiddelde, bereik en standaardafwijking, en wat je met een uitschieter doet.

Een meetfout is geen vergissing

In de wetenschap betekent *fout* (beter: *onzekerheid*) niet dat je iets verkeerd deed. Elke meting heeft een marge, ook die van het beste labo ter wereld. Een *vergissing* is iets anders: een verkeerd overgeschreven cijfer, een verkeerde eenheid, een verkeerd ingestelde meter. Een vergissing spoor je op en verbeter je; een meetfout schat je en vermeld je.



De rand valt tussen 73 en 74 mm. Je schat 73,5 mm en noteert de afleesfout erbij: een halve schaalverdeling.

UITLEG · 1.2

Twee soorten fouten: systematisch en toevallig

2/18

Zet twee temperatuursensoren samen in een waterbad dat met een geijkte thermometer op precies 20,0 °C gehouden wordt, en lees ze zestien keer af. De twee sensoren falen op een heel verschillende manier (grafiek).

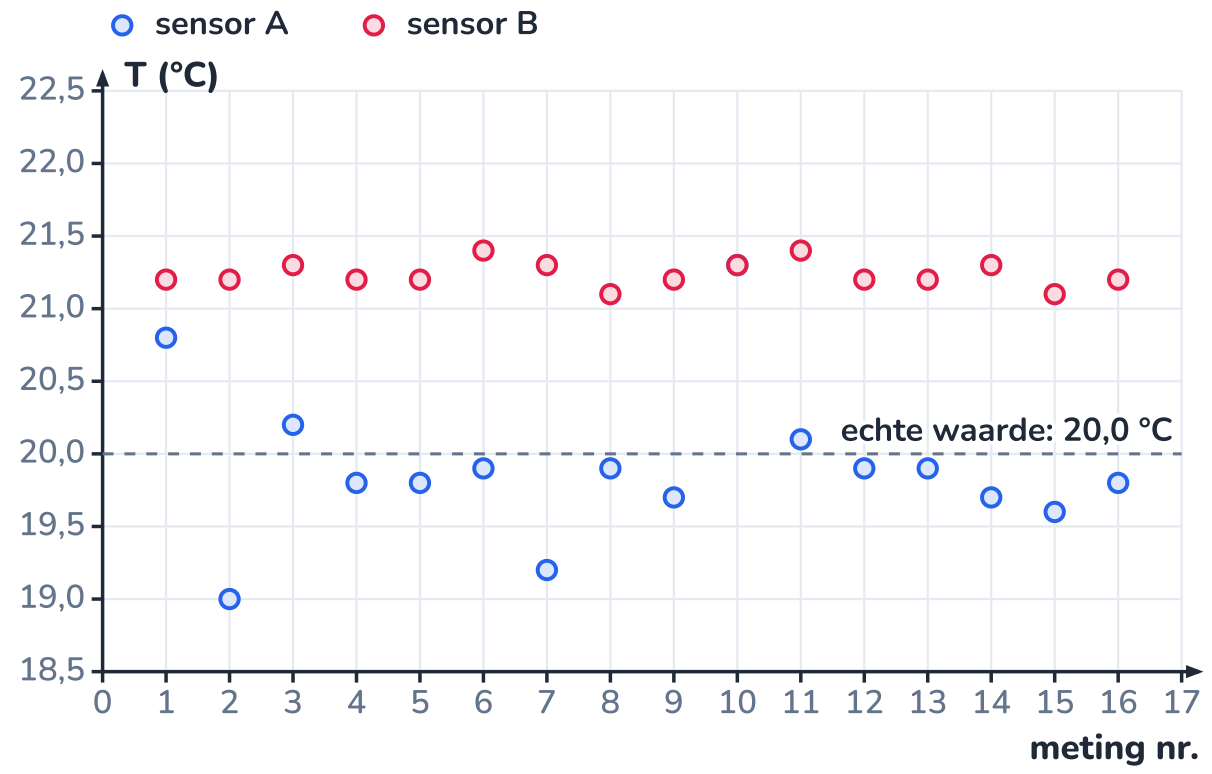
Sensor A springt heen en weer rond 20 °C: de ene keer 19,0 °C, een andere keer 21,3 °C. Dat is een **toevallige fout**: ze is bij elke meting anders, soms te groot en soms te klein, en je kan ze niet voorspellen. Oorzaken: elektrische ruis, kleine stromingen in het water, een aflezing die net tussen twee streepjes valt, je reactietijd bij een stopwatch.

Sensor B is opvallend rustig, maar zit er telkens ongeveer 1,2 °C naast, altijd in dezelfde richting. Dat is een **systematische fout**: ze duwt elke meting evenveel dezelfde kant op. Oorzaken: een instrument dat niet op nul staat, een slecht geijkte sensor, parallax die je altijd van dezelfde kant maakt, een sensor in de zon die vooral zijn eigen omhulsel meet.

	toevallige fout	systematische fout
richting	wisselt: soms +, soms −	altijd dezelfde kant
herken je door	te herhalen: de metingen spreiden	te vergelijken met een referentie of een andere methode
verklein je door	meer metingen, een fijner instrument, veel slingeringen samen timen	ijken, het nulpunt instellen, de methode verbeteren
verdwijnt door te middelen?	grotendeels	nee: het gemiddelde zit er even ver naast

Herhalen lost alles op

Het gemiddelde van honderd metingen met sensor B blijft 1,2 °C te hoog. Herhalen verkleint alleen de toevallige fout. Een systematische fout vind je enkel door te vergelijken met iets wat je vertrouwt: een geijkte thermometer, smeltend ijs in water (0 °C), een andere meetmethode.



Sensor A spreidt rond de echte waarde (gemiddeld 19,9 °C); sensor B ligt er strak naast (gemiddeld 21,2 °C).

VERHAAL · 1.3

Een spiegel die perfect verkeerd was

3/18

In april 1990 bracht een spaceshuttle de Hubble-ruimtetelescoop in een baan om de aarde. Het hart ervan is een spiegel van 2,4 meter doorsnede, gepolijst met een precisie die toen ongezien was. Binnen enkele weken bleek dat de beelden wazig waren: het licht van een ster werd uitgesmeerd tot een vlekje met een waas errond.

De oorzaak was een **systematische fout**. Tijdens het polijsten werd de vorm van de spiegel gecontroleerd met een optisch meettoestel, een zogenoemde nulcorrector. Eén lens in dat toestel stond ongeveer 1,3 mm verkeerd. Het bedrijf polijstte de spiegel dus uiterst nauwgezet naar een verkeerde

vorm: aan de rand was hij ongeveer 2,2 micrometer te vlak, zowat een vijftigste van de dikte van een haar. Elke controle met hetzelfde toestel bevestigde het resultaat, want het toestel zelf was de fout.

Er waren aanwijzingen. Andere, eenvoudigere testen tijdens de bouw wezen op een afwijking, maar ze werden opzijgeschoven omdat ze minder precies waren dan de nulcorrector. Achteraf bleken net die minder precieze metingen juist te zijn.

In december 1993 plaatsten astronauten tijdens ruimtewandelingen corrigerende optiek in de telescoop, een soort bril. Omdat de fout zo precies gekend was, kon de correctie even precies zijn. Sindsdien levert Hubble de scherpe beelden waarvoor hij beroemd is.

De les voor elke meter

Precisie bewijst geen juistheid. Een meting die zichzelf telkens herhaalt, kan telkens even fout zijn. Controleer een instrument daarom met een **onafhankelijke** referentie, en neem een afwijkend resultaat ernstig, ook als het van een minder fijn instrument komt.

Een telescoopspiegel wordt tot op een fractie van een micrometer gepolijst: precies, maar is hij ook juist?

UITLEG · 1.4

Nauwkeurig tegenover precies

4/18

De twee soorten fouten horen bij twee eigenschappen van een meetreeks. Een reeks is **precies** als de metingen dicht bij elkaar liggen: de toevallige fout is klein. Ze is **nauwkeurig** als ze *gemiddeld* dicht bij de echte waarde ligt: de systematische fout is klein. Op een dartbord zie je het verschil meteen.

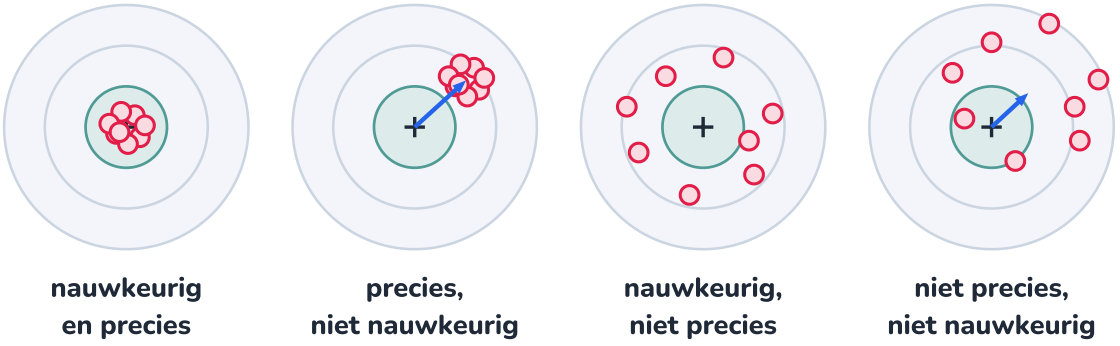
Sensor B uit het vorige blad is precies maar niet nauwkeurig, net als de spiegel van Hubble. Sensor A is gemiddeld nauwkeurig, maar niet precies. Je wil beide. Precisie haal je met een goed instrument en herhaalde metingen, nauwkeurigheid met ijken tegenover een referentie.

Meer cijfers betekent nauwkeuriger

De DS18B20, een veelgebruikte digitale temperatuursensor, geeft de temperatuur in stapjes van 0,0625 °C. Toch belooft de fabrikant maar een nauwkeurigheid van ± 0,5 °C (tussen −10 °C en +85 °C). De **resolutie**, de kleinste stap die het toestel toont, zegt niets over de **nauwkeurigheid**, hoe ver het er gemiddeld naast kan zitten. Een scherm met vier decimalen kan vier decimalen lang fout zijn.

Woorden in de norm

In de officiële meettaal (de ISO-normen over meetmethoden) heet 'gemiddeld dicht bij de echte waarde' *juistheid*, en betekent *nauwkeurigheid* juist én precies samen. Lees je 'juistheid' in een handleiding, dan weet je wat bedoeld wordt.



Het kruisje is de echte waarde, elke rode cirkel een meting. De spreiding toont de toevallige fout, de verschuiving van het zwaartepunt (blauwe pijl) de systematische.

UITLEG · 1.5

Absolute en relatieve fout

5/18

Een meetresultaat noteer je als **waarde ± fout**, met de eenheid erachter:

EEN MEETRESULTAAT

$$x = x_{\text{gemeten}} \pm \Delta x$$

Δx is de **absolute fout**, in dezelfde eenheid als de meting. Je bedoelt: de echte waarde ligt tussen $x - \Delta x$ en $x + \Delta x$. Voor een afleesfout is dat vrijwel zeker; voor een fout uit de standaardafwijking (verderop) in ongeveer twee op de drie gevallen.

Hoe groot is Δx? Dat hangt af van waar de fout vandaan komt:

- **Analoge schaal** (lat, maatcilinder, wijzer): een halve kleinste schaalverdeling, tenzij je slechter kan aflezen (een trillende wijzer, een dikke meniscus).
- **Digitaal toestel**: de specificatie van de fabrikant, vaak geschreven als ± (een percentage van de aflezing + een aantal eenheden van het laatste cijfer). Staat er niets, neem dan minstens één eenheid van het laatste cijfer.
- **Herhaalde metingen**: de spreiding van de reeks, maar nooit kleiner dan de fout van het instrument.

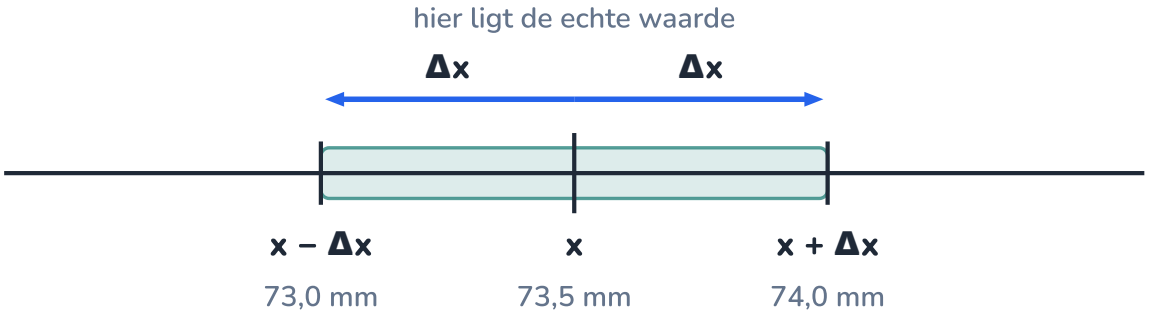
Of een fout groot is, hangt af van wat je meet. Een halve millimeter op de lengte van een kas is niets; op de dikte van een draad is het enorm. Daarom bestaat de **relatieve fout**:

RELATIEVE FOUT

$$\text{relatieve fout} = \frac{\Delta x}{|x|}$$

Zonder eenheid, vaak in procent. Zo vergelijk je de kwaliteit van metingen van verschillende grootheden of met verschillende instrumenten.

instrument	kleinste stap	typische fout
lat met mm-verdeling	1 mm	± 0,5 mm
rolmeter	1 mm	± 0,5 mm tot ± 2 mm (lint dat doorhangt, haakje aan het begin)
schuifmaat met nonius	0,05 mm	± 0,05 mm
digitale balans	0,1 g	± 0,1 g (of volgens specificatie)
maatcilinder van 100 mL	1 mL	± 0,5 mL
stopwatch, met de hand bediend	0,01 s	± 0,1 à 0,2 s: je reactie, niet de klok
digitale thermometer	0,1 °C	volgens specificatie, vaak ± 0,5 °C of meer



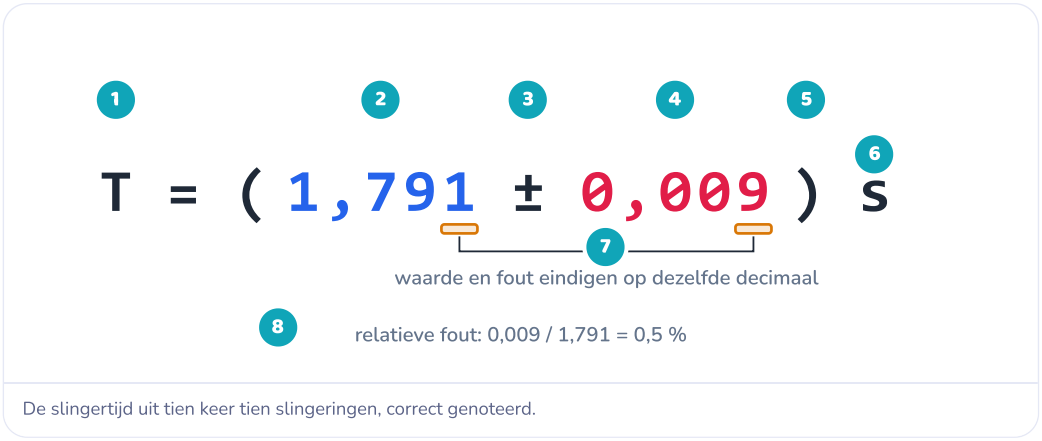
Een meetresultaat is geen punt maar een interval: de echte waarde ligt ergens in de gekleurde band.

ONTLEED · 1.6

Een meetresultaat ontleed

6/18

Zo noteert een onderzoeker het resultaat van de slingerproef verderop in dit hoofdstuk. Elk teken heeft een reden. Wie één onderdeel weglaat, laat de lezer raden.



De slingertijd uit tien keer tien slingeringen, correct genoteerd.

1 Grootheid
Het symbool van wat je meet: hier T, de slingertijd (periode). Een getal zonder grootheid kan niemand controleren.

2 Waarde
De beste schatting, meestal het gemiddelde van je metingen. Ze krijgt evenveel decimalen als de fout: niet meer (schijnprecisie), niet minder (informatie weggooien).

3 Plus of min
Het interval: de echte waarde ligt tussen 1,782 s en 1,800 s. Een meetresultaat is een gebied, geen punt.

4 Fout
De absolute fout, afgerond op één beduidend cijfer (twee als dat cijfer een 1 is). Hier werd $0{,}0085\,\mathrm{s}$ afgerond tot $0{,}009\,\mathrm{s}$.

5 Haakjes
Waarde en fout hebben dezelfde eenheid; de haakjes tonen dat de eenheid voor beide geldt. Zonder haakjes staat er een kaal getal 1,791 plus 0,009 seconde.

6 Eenheid
Altijd, als SI-eenheid of met een voorvoegsel. In een berekening reken je de eenheid mee.

7 Dezelfde decimaal
Waarde en fout eindigen op dezelfde decimaal, hier de duizendsten. Dat is de gouden regel van het afronden.

8 Relatieve fout
Zet ze erbij als je metingen wil vergelijken: $0{,}009/1{,}791 = 0{,}005 = 0{,}5\,\%$.

UITGEWERKT VOORBEELD · 1.7

Klein meten is moeilijk meten

7/18

Voor de meetkas noteert een leerling drie maten. Bereken telkens de relatieve fout en beoordeel of het instrument geschikt is.

- (a) de lengte van de plantenbak: $120{,}0\,\mathrm{cm}$, met een rolmeter, $\pm 0{,}2\,\mathrm{cm}$;
- (b) de dikte van de polycarbonaatplaat van de wand: $4{,}00\,\mathrm{mm}$, met een schuifmaat, $\pm 0{,}05\,\mathrm{mm}$;
- (c) de diameter van een gaatje in de druppelslang: met een lat 'ongeveer 1 mm' ($\pm 0{,}5\,\mathrm{mm}$), met een schuifmaat $1{,}20\,\mathrm{mm}$ ($\pm 0{,}05\,\mathrm{mm}$).

Voor kleine maten is een schuifmaat nodig: met een lat is de relatieve fout tientallen procenten.

GEGEVEN

- de drie metingen en hun absolute fout

GEVRAAGD

- de relatieve fout van elke meting
- is het instrument geschikt?

OPLOSSING

1 (a) $\dfrac{\Delta l}{l} = \dfrac{0{,}2\,\text{cm}}{120{,}0\,\text{cm}} = 0{,}0017 = 0{,}17\,\%$.

2 (b) $\dfrac{0{,}05\,\text{mm}}{4{,}00\,\text{mm}} = 0{,}0125 \approx 1{,}3\,\%$.

3 (c) Met de lat: $\dfrac{0{,}5\,\text{mm}}{1\,\text{mm}} = 0{,}5 = 50\,\%$. Met de schuifmaat: $\dfrac{0{,}05\,\text{mm}}{1{,}20\,\text{mm}} = 0{,}042 \approx 4\,\%$.

4 Beoordeel. De rolmeter is ruim goed genoeg voor de bak. Voor het gaatje is de lat onbruikbaar: bij 50 % fout weet je niet of het gaatje 0,5 mm of 1,5 mm is. En dat maakt uit: het water dat door een gaatje stroomt, hangt af van de oppervlakte van de opening, dus van het kwadraat van de diameter.

ANTWOORD

0,17 %, 1,3 % en 50 % (lat) of 4 % (schuifmaat). Hoe kleiner het voorwerp, hoe fijner het instrument moet zijn om dezelfde relatieve fout te halen.

Klopt dit? Een relatieve fout heeft geen eenheid: mm gedeeld door mm. Krijg je toch een eenheid, dan deelde je twee verschillende eenheden door elkaar (cm door mm) en moet je eerst omzetten.

UITLEG · 1.8

Beduidende cijfers

Staat er geen fout bij een getal, dan vertellen de cijfers zelf hoe precies het is. $12{,}0\,\text{cm}$ zegt iets anders dan $12\,\text{cm}$: bij het eerste heb je tot op de millimeter gemeten. De cijfers die zo informatie dragen, heten **beduidende cijfers**.

1. Alle cijfers die geen nul zijn, tellen: $3{,}47$ heeft er 3.
2. Nullen **tussen** andere cijfers tellen: $3{,}040$ heeft er 4.
3. Nullen **vooraan** tellen nooit: ze zetten alleen de komma op zijn plaats. $0{,}00520$ heeft er 3.
4. Nullen **achteraan na de komma** tellen wel: de laatste 0 van $0{,}00520$ is gemeten.
5. Nullen achteraan in een getal zonder komma zijn dubbelzinnig: is $1200\,\text{g}$ tot op de gram of tot op 100 g gemeten? Schrijf daarom $1{,}20 \cdot 10^3\,\text{g}$ (3 beduidende cijfers) of $1{,}2 \cdot 10^3\,\text{g}$ (2).

Rekenen met meetwaarden volgt twee vuistregels:

- **Vermenigvuldigen en delen**: het resultaat krijgt evenveel beduidende cijfers als de factor met de *minste* beduidende cijfers: $12{,}5\,\text{cm} \times 3{,}3\,\text{cm} = 41{,}25\,\text{cm}^2 \to 41\,\text{cm}^2$.
- **Optellen en aftrekken**: het resultaat krijgt evenveel *decimalen* als de term met de *minste* decimalen: $15{,}42\,\text{g} + 3{,}1\,\text{g} = 18{,}52\,\text{g} \to 18{,}5\,\text{g}$.

Rond pas **op het einde** af: reken verder met alle cijfers van je rekenmachine, anders stapel je afrondingsfouten op. Exacte getallen beperken niets: de 2 in $2\pi r$, het aantal metingen, de 100 in $1\,\text{m} = 100\,\text{cm}$.

> ⚠ **De fout gaat voor**
>
> Beduidende cijfers zijn een vuistregel voor als je de fout niet kent. Ken je de fout wel, dan bepaalt de fout hoeveel cijfers je noteert: rond de fout af op één beduidend cijfer (twee als het eerste cijfer een 1 is) en rond de waarde af op dezelfde decimaal.

$0{,}00520$	$3{,}040$	$1{,}20 \cdot 10^3$
3 beduidende cijfers	**4 beduidende cijfers**	**3 beduidende cijfers**
nullen vooraan tellen niet	nul tussenin en achteraan telt	macht van tien: geen twijfel

Blauw: beduidende cijfers. Grijs: nullen die alleen de komma op zijn plaats zetten, en de macht van tien.

UITLEG · 1.9

Hoe fouten doorwerken in een berekening

Vaak meet je een grootheid niet rechtstreeks maar bereken je ze: een oppervlakte uit twee lengtes, een dichtheid uit een massa en een volume, een debiet uit een volume en een tijd. Elke gemeten waarde brengt haar fout mee. Hoe groot wordt de fout op het resultaat?

De zekere manier: reken met de uitersten. Een bak meet $l = (42{,}0 \pm 0{,}1)\,\text{cm}$ bij $b = (28{,}5 \pm 0{,}1)\,\text{cm}$. De grootst mogelijke oppervlakte is $42{,}1 \times 28{,}6 = 1204{,}06\,\text{cm}^2$, de kleinst mogelijke $41{,}9 \times 28{,}4 = 1189{,}96\,\text{cm}^2$. De halve breedte van dat interval is $7{,}05\,\text{cm}^2$, dus $A = (1197 \pm 7)\,\text{cm}^2$.

De snelle manier: regels. Voor kleine fouten geven ze (bijna) hetzelfde als de uitersten.

OPTELLEN EN AFTREKKEN

$$\Delta(a+b) = \Delta(a-b) = \Delta a + \Delta b$$

De absolute fouten tellen op, ook bij een verschil.

VERMENIGVULDIGEN EN DELEN

$$\frac{\Delta(a \cdot b)}{|a \cdot b|} \approx \frac{\Delta a}{|a|} + \frac{\Delta b}{|b|}$$

De relatieve fouten tellen op; voor a/b geldt dezelfde regel. Bij een macht wordt de relatieve fout n keer zo groot: $\frac{\Delta(a^n)}{|a^n|} \approx n \cdot \frac{\Delta a}{|a|}$.

Voor de bak: $\frac{0{,}1}{42{,}0} + \frac{0{,}1}{28{,}5} = 0{,}24\,\% + 0{,}35\,\% = 0{,}59\,\%$, en $0{,}59\,\%$ van $1197\,\mathrm{cm}^2$ is $7\,\mathrm{cm}^2$: hetzelfde resultaat met minder rekenwerk. De omtrek is een som: $2(l+b) = 141{,}0\,\mathrm{cm}$, met fout $2 \times (0{,}1 + 0{,}1) = 0{,}4\,\mathrm{cm}$.

Aftrekken maakt de fout kleiner

Bij een verschil tellen de fouten op, ook al wordt het resultaat kleiner. Meet je het volume van een steen als verschil van twee aflezingen op een maatcilinder, 57,5 mL − 50,0 mL = 7,5 mL, met elk ± 0,5 mL, dan is de fout 1,0 mL: 13 % van het resultaat, terwijl elke aflezing apart maar 1 % fout had.

In Expert: slimmer optellen

Toevallige fouten zijn zelden allemaal tegelijk maximaal en in dezelfde richting. In hoofdstuk E6 leer je ze kwadratisch op te tellen, wat een kleinere, realistischere fout geeft. De regels hier geven de veilige bovengrens.

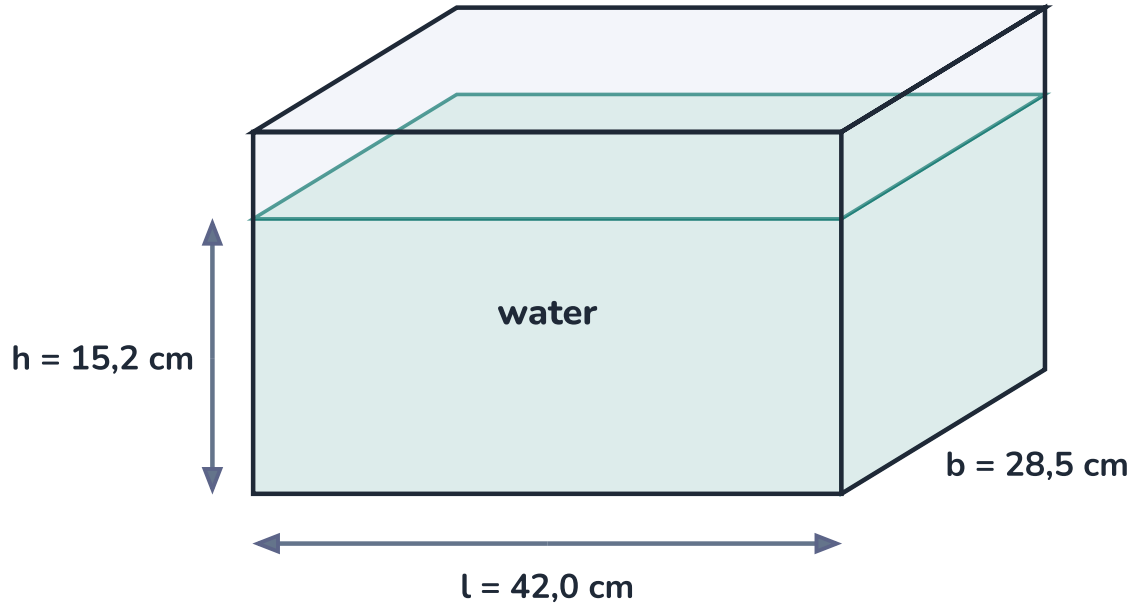
De verschillen zijn overdreven getekend. De gekleurde rand tussen de kleinste en de grootste rechthoek is de onzekerheid op de oppervlakte.

UITGEWERKT VOORBEELD · 1.10

Hoeveel water zit er in het reservoir?

10/18

Het waterreservoir van de meetkas is een rechthoekige bak. De binnenmaten, gemeten met een rolmeter: lengte $42{,}0\,\text{cm}$ en breedte $28{,}5\,\text{cm}$, elk $\pm 0{,}1\,\text{cm}$. Het water staat $15{,}2\,\text{cm}$ hoog; omdat het water tegen de wand opkruipt en de bodem niet helemaal vlak is, schat je die fout op $\pm 0{,}2\,\text{cm}$. Hoeveel liter water zit erin?



Drie gemeten lengtes, elk met hun eigen fout: welke weegt het zwaarst door in het volume?

GEGEVEN

- $l = (42{,}0 \pm 0{,}1)\,\text{cm}$
- $b = (28{,}5 \pm 0{,}1)\,\text{cm}$
- $h = (15{,}2 \pm 0{,}2)\,\text{cm}$

GEVRAAGD

- $V \pm \Delta V$, in liter

OPLOSSING

1. Waarde: $V = l \cdot b \cdot h = 42{,}0 \times 28{,}5 \times 15{,}2 = 18\,194{,}4\,\text{cm}^3$. Nog niet afronden.
2. Relatieve fouten: $\frac{0{,}1}{42{,}0} = 0{,}238\,\%$, $\frac{0{,}1}{28{,}5} = 0{,}351\,\%$ en $\frac{0{,}2}{15{,}2} = 1{,}316\,\%$. Een product, dus optellen: $1{,}905\,\%$.
3. Absolute fout: $\Delta V = 0{,}01905 \times 18\,194\,\text{cm}^3 \approx 347\,\text{cm}^3 = 0{,}347\,\text{L}$.
4. Afronden: de fout op één beduidend cijfer, $0{,}3\,\text{L}$; de waarde op dezelfde decimaal, $18{,}194\,\text{L} \to 18{,}2\,\text{L}$.
5. Controle met de uitersten: $42{,}1 \times 28{,}6 \times 15{,}4 = 18\,543\,\text{cm}^3$ en $41{,}9 \times 28{,}4 \times 15{,}0 = 17\,849\,\text{cm}^3$. De halve breedte is $347\,\text{cm}^3$: klopt.

ANTWOORD

$V = (18{,}2 \pm 0{,}3)\,\text{L}$, een relatieve fout van ongeveer 2 %. De waterhoogte levert $1{,}316/1{,}905 \approx 69\,\%$ van die fout: wie beter wil meten, meet eerst de hoogte beter (op meerdere plaatsen, of met een peilstok met schaalverdeling).

Klopt dit? Ook de vuistregel van de beduidende cijfers geeft hier 18,2 L (drie factoren met drie beduidende cijfers). En de orde van grootte: een bak van ongeveer 40 × 30 × 15 cm bevat 18 dm³, dus 18 L.

Gemiddelde, bereik en standaardafwijking

Herhaal je een meting, dan krijg je een reeks getallen. Drie getallen vatten die reeks samen. Als voorbeeld: tien keer de tijd van één slingering van dezelfde slinger, met de hand gemeten (de proef verderop): 1,74 · 1,64 · 1,84 · 1,85 · 1,77 · 1,82 · 1,92 · 1,87 · 1,85 · 1,85 (in s).

Het **gemiddelde** $\bar{x}$ is de beste schatting van de echte waarde. Hier $\bar{x} = 18{,}15\,\text{s}/10 = 1{,}815\,\text{s}$.

GEMIDDELDE

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{\sum x_i}{n}$$

Het **bereik** is het verschil tussen de grootste en de kleinste meting: $1{,}92 - 1{,}64 = 0{,}28\,\text{s}$. Snel berekend, maar het hangt maar van twee metingen af, en net die twee zijn vaak de vreemdste.

De **standaardafwijking** s gebruikt alle metingen. Je bepaalt hoe ver elke meting van het gemiddelde ligt, kwadrateert die afwijkingen (zodat plus en min elkaar niet opheffen), neemt er een soort gemiddelde van en trekt de wortel (zodat de eenheid weer klopt):

STANDAARDAFWIJKING VAN EEN MEETREEKS

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

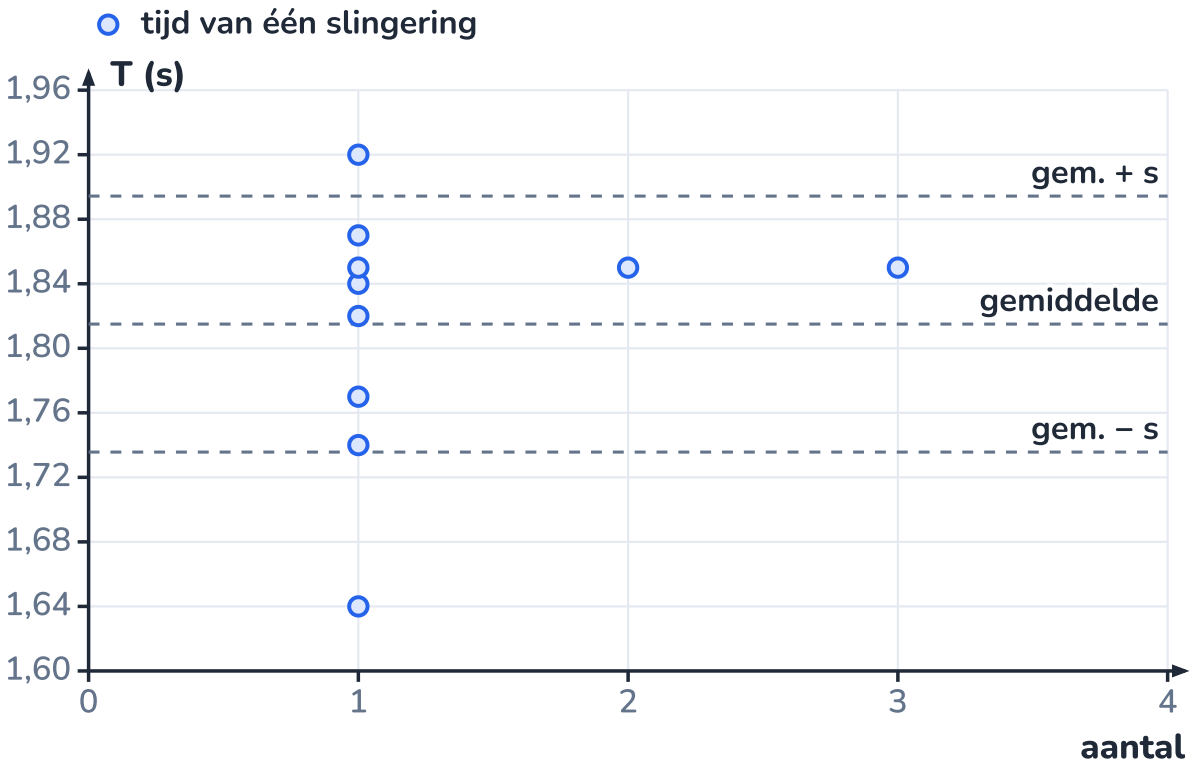
Voor de slingertijden: $s = 0{,}079\,\text{s}$.

Heeft een reeks alleen toevallige fouten, dan geldt als vuistregel: ongeveer twee derde van de metingen ligt binnen één s van het gemiddelde, ongeveer 95 % binnen twee s. Hier liggen 8 van de 10 metingen tussen 1,736 s en 1,894 s.

In dit hoofdstuk noteer je een herhaalde meting als $\bar{x} \pm s$: $T = (1{,}82 \pm 0{,}08)\,\text{s}$. Dat is een voorzichtige keuze: s zegt hoe ver *één* meting er typisch naast zit. Het gemiddelde van tien metingen is zelf preciezer. In Expert (hoofdstuk E6) leer je die kleinere fout te gebruiken, de standaardfout $s/\sqrt{n}$.

Waarom delen door n − 1?

Het gemiddelde komt uit dezelfde metingen en ligt daardoor altijd een beetje 'in het midden' van je reeks: de afwijkingen tot $\bar{x}$ zijn gemiddeld iets kleiner dan die tot de echte waarde. Delen door $n - 1$ in plaats van n corrigeert dat. Op een rekenmachine heet deze versie s_x of σ_{n-1}, in een rekenblad STDEV.S. De versie met n (σ_n, STDEV.P) hoort bij een volledige populatie, niet bij een meetreeks.



Tien slingertijden, gestapeld per waarde (horizontaal: hoe vaak die waarde voorkwam). De stippellijnen tonen het gemiddelde en één standaardafwijking erboven en eronder.

UITGEWERKT VOORBEELD · 1.12

De standaardafwijking met de hand: het pompje

12/18

Hoeveel water levert het pompje van de meetkas per minuut? Een leerling meet zes keer hoe lang het pompje erover doet om een maatbeker tot de streep van 1,00 L te vullen: 41,8 s; 43,1 s; 42,4 s; 42,0 s; 43,5 s; 42,6 s. De maatbeker heeft een fout van ± 0,02 L. Bereken gemiddelde, bereik en standaardafwijking, en daarmee het debiet.

GEGEVEN

- zes tijden (s)
- $V = (1{,}00 \pm 0{,}02)\,\mathrm{L}$

GEVRAAGD

- $\bar{t}$, bereik en s
- debiet Q in L/min, met fout

OPLOSSING

1. Gemiddelde: $\bar{t} = \dfrac{41{,}8 + 43{,}1 + 42{,}4 + 42{,}0 + 43{,}5 + 42{,}6}{6} = \dfrac{255{,}4}{6} = 42{,}57\,\mathrm{s}$.

2. Bereik: $43{,}5 - 41{,}8 = 1{,}7\,\mathrm{s}$.

3 Afwijkingen en hun kwadraten in een tabel. Controle: de som van de afwijkingen is op afronding na nul.

T (s)	$T - \bar{T}$ (s)	$(T - \bar{T})^2$ (s²)
41,8	−0,77	0,5929
43,1	0,53	0,2809
42,4	−0,17	0,0289
42,0	−0,57	0,3249
43,5	0,93	0,8649
42,6	0,03	0,0009
som	−0,02	**2,0934**

4 $s = \sqrt{\dfrac{2{,}0934\,\mathrm{s}^2}{6-1}} = \sqrt{0{,}4187\,\mathrm{s}^2} = 0{,}647\,\mathrm{s}$, dus $\bar{t} = (42{,}6 \pm 0{,}6)\,\mathrm{s}$.

5 Debiet: $Q = \dfrac{V}{t} = \dfrac{1{,}00\,\mathrm{L}}{42{,}57\,\mathrm{s}} = 0{,}02349\,\mathrm{L/s} = 1{,}409\,\mathrm{L/min}$.

6 Fout: een quotiënt, dus relatieve fouten optellen: $\dfrac{0{,}02}{1{,}00} + \dfrac{0{,}647}{42{,}57} = 2{,}0\,\% + 1{,}5\,\% = 3{,}5\,\%$. $\Delta Q = 0{,}035 \times 1{,}409 = 0{,}049\,\mathrm{L/min}$.

ANTWOORD

$\bar{t} = (42{,}6 \pm 0{,}6)\,\mathrm{s}$ en $Q = (1{,}41 \pm 0{,}05)\,\mathrm{L/min}$.

Klopt dit? Voor zes metingen verwacht je een bereik van ongeveer 2,5 keer s; hier is het $1{,}7/0{,}647 = 2{,}6$. Controleer s ook met de statistiekfunctie van je rekenmachine (s_x) of met STDEV.S in een rekenblad.

PROEF · 1.13

Eén slingering of tien?

13/18

ONDERZOEKSVRAAG

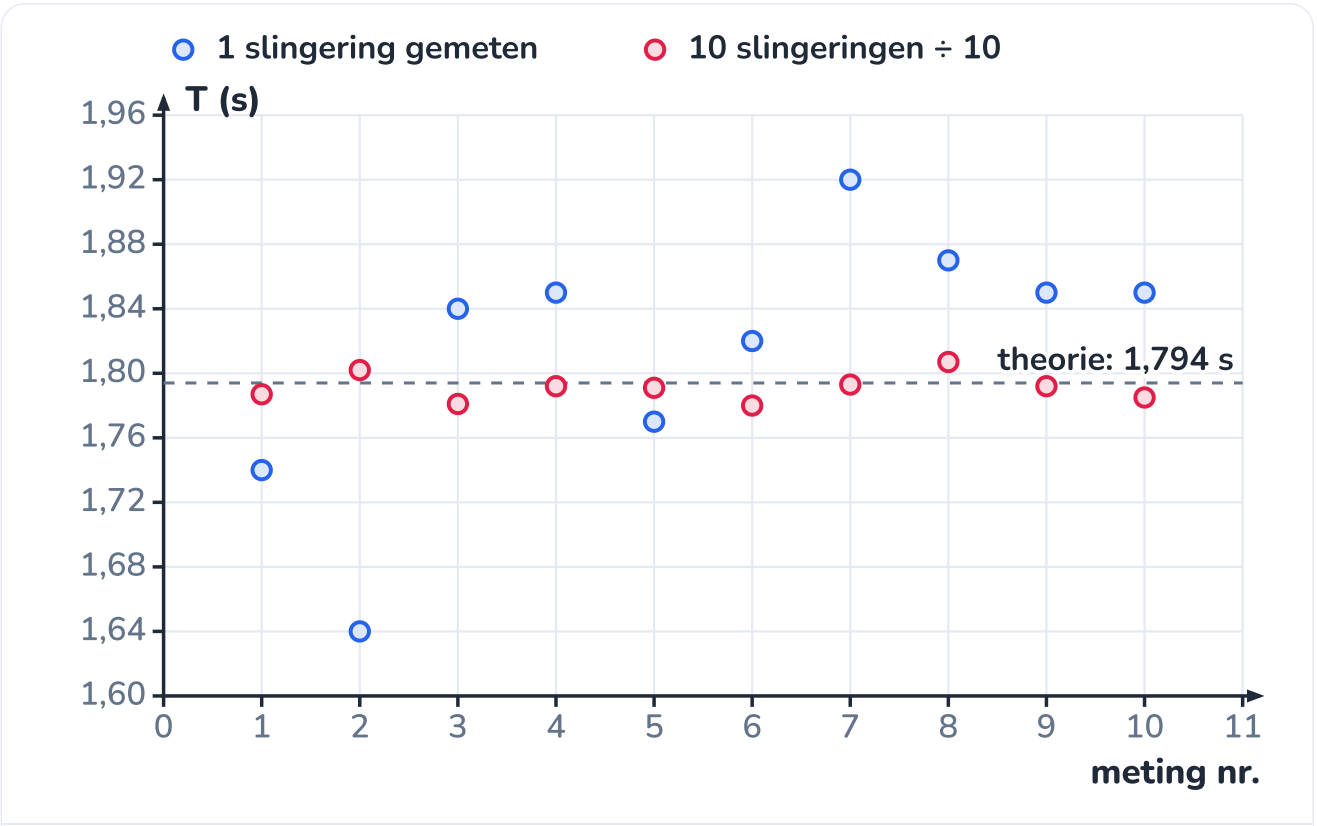
Hoe meet je de slingertijd preciezer, met dezelfde stopwatch en dezelfde handen?

MATERIAAL

statief met klem · touw van ongeveer 1 m en een slingergewicht (moer of metalen kogel) · rolmeter · stopwatch of smartphone · rekenmachine of rekenblad

Veiligheid

Laat het gewicht niet ver uitzwaaien en houd je hoofd buiten de baan van de slinger.



Blauw: één slingering, met de hand getimed. Rood: tien slingeringen gedeeld door tien. Zelfde handen, tien keer minder spreiding.

WERKWIJZE

1. Hang het gewicht zo op dat de afstand van het ophangpunt tot het midden van het gewicht 0,80 m is.
2. Trek het gewicht een kleine hoek opzij (hoogstens 10°) en laat los zonder duw.
3. Meet tien keer de tijd van één volledige slingering (heen en terug). Start en stop telkens als het gewicht door het laagste punt gaat: daar is het het snelst, dus het moment het scherpst.
4. Meet daarna tien keer de tijd van tien opeenvolgende slingeringen en deel elke tijd door tien.
5. Bereken voor beide reeksen het gemiddelde en de standaardafwijking.

WAARNEMING

nr.	1 slingering (s)	10 slingeringen (s)	÷ 10 (s)
1	1,74	17,87	1,787
2	1,64	18,02	1,802
3	1,84	17,81	1,781
4	1,85	17,92	1,792
5	1,77	17,91	1,791
6	1,82	17,80	1,780
7	1,92	17,93	1,793
8	1,87	18,07	1,807
9	1,85	17,92	1,792
10	1,85	17,85	1,785
gem.	**1,815**	**17,910**	**1,791**
s	**0,079**	**0,085**	**0,0085**

Met één slingering: $T = (1{,}82 \pm 0{,}08)\,\text{s}$, een relatieve fout van 4,4 %. Uit tien slingeringen: $T = (1{,}791 \pm 0{,}009)\,\text{s}$, een relatieve fout van 0,5 %.

VERKLARING

De grootste fout zit niet in de stopwatch maar in je reactie bij starten en stoppen: telkens ongeveer een tiende van een seconde. Die fout is even groot, of je nu één of tien slingeringen timet. Deel je de tijd van tien slingeringen door tien, dan deel je ook die fout door tien: s daalt van 0,079 s naar 0,0085 s.

De theorie voorspelt voor een slinger van 0,80 m een slingertijd van $T = 2\pi\sqrt{L/g} = 1{,}794\,\text{s}$. Beide resultaten zijn daarmee in overeenstemming, maar alleen het tweede is precies genoeg om er iets mee te doen, bijvoorbeeld g bepalen (hoofdstuk 3). Een systematische fout, zoals altijd iets te laat stoppen, verdwijnt niet door tien slingeringen: die deel je ook door tien, maar ze blijft dezelfde kant op wijzen.

UITLEG · 1.14

Herhalen, en wat met een uitschieter?

14/18

Herhalen levert drie dingen op: je **ziet de spreiding** (de toevallige fout), het **gemiddelde** is betrouwbaarder dan één meting, en je **ziet vergissingen**.

Een **uitschieter** is een meting die opvallend ver van de andere ligt. Hij kan drie dingen betekenen: een vergissing (verkeerd afgelezen, verkeerd genoteerd), een storing (los contact, trilling, een pomp die aansloeg) of een echt verschijnsel dat je niet verwachtte. Dat laatste is soms het interessantste resultaat van je hele onderzoek. Daarom schrap je nooit zomaar.

Een werkwijze die in de meeste schoollabo's volstaat:

1. Zoek in je labojournaal of er bij die meting iets bijzonders gebeurde.
2. Bereken gemiddelde en standaardafwijking van de *andere* metingen.
3. Ligt de verdachte waarde meer dan drie standaardafwijkingen van dat gemiddelde, dan is ze een uitschieter. Met een gevonden oorzaak mag je ze weglaten; zonder oorzaak meet je liever opnieuw.
4. Laat je een meting weg, **vermeld dat** in je verslag, met de reden.

Een uitschieter ligt altijd meer dan 2s van het gemiddelde

Niet als je hem zelf meerekent: dan maakt hij s groter en verstopt hij zich. Neem vier keer 10 en één keer 20. Het gemiddelde is 12, de standaardafwijking 4,47, en de 20 ligt maar $8/4{,}47 = 1{,}79$ standaardafwijkingen van het gemiddelde. Met vijf metingen kan een waarde wiskundig nooit verder dan 1,79 s liggen. Daarom reken je $\bar{x}$ en s zonder de verdachte waarde.

De mediaan

Sorteer de metingen en neem de middelste (bij een even aantal: het gemiddelde van de middelste twee). Eén uitschieter verschuift de mediaan nauwelijks, het gemiddelde wel. In een rekenblad: MEDIAAN (Engels: MEDIAN).

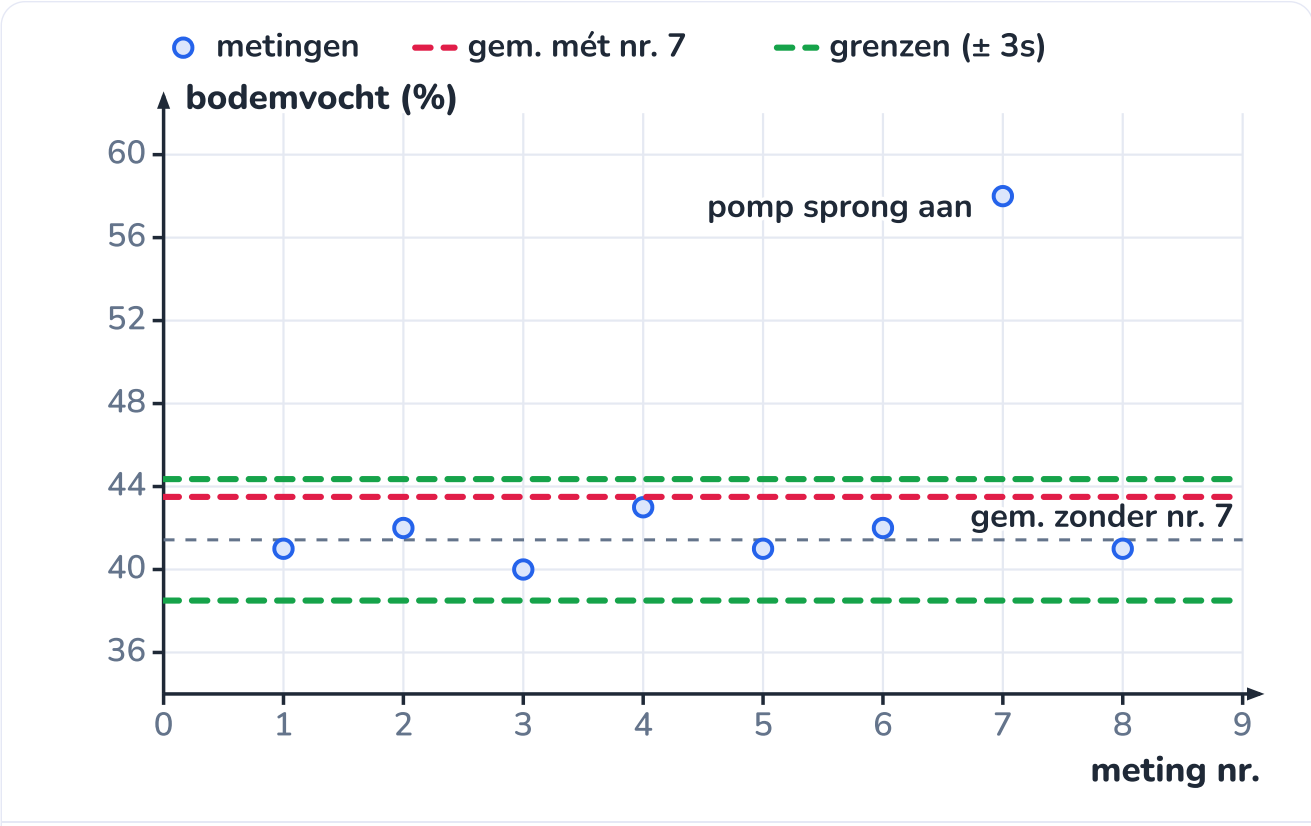
Eerst een oorzaak zoeken, dan pas rekenen: een uitschieter is ook informatie.

UITGEWERKT VOORBEELD · 1.15

De bodemvochtsensor die even hapert

15/18

De bodemvochtsensor in de meetkas wordt acht keer uitgelezen, met tien seconden ertussen: 41, 42, 40, 43, 41, 42, 58 en 41 (in %). In het logboek staat bij meting 7: 'pompje sprong net aan'. Welk bodemvocht rapporteer je?



De groene grenzen liggen drie standaardafwijkingen boven en onder het gemiddelde van de zeven andere metingen. Meting 7 valt er ver buiten.

GEGEVEN

- acht metingen (%)
- een notitie in het logboek bij meting 7

GEVRAAGD

- het bodemvocht als waarde ± fout
- wat doe je met meting 7?

OPLOSSING

1. Met alle acht metingen: $\bar{x} = 43{,}5\,\%$, $s = 5{,}9\,\%$, mediaan $41{,}5\,\%$.
2. Zonder meting 7: $\bar{x} = 41{,}43\,\%$ en $s = 0{,}98\,\%$.
3. Afstand van 58 tot dat gemiddelde: $\dfrac{58 - 41{,}43}{0{,}98} = 17$ standaardafwijkingen, ver voorbij de grens van 3.
4. Oorzaak: het druppelwater kwam op dat moment vlak bij de sensor in de grond. De meting is niet fout, maar ze meet iets anders dan wat je zoekt: de vochtigheid van de bak vóór het water geven.
5. Merk op: het gemiddelde mét de uitschieter (43,5 %) ligt boven zeven van de acht metingen en beschrijft de reeks slecht. De mediaan (41,5 %) had je niet misleid.

ANTWOORD

Bodemvocht $= (41 \pm 1)\,\%$, uit zeven metingen. Meting 7 is weggelaten omdat de pomp op dat moment water gaf; dat staat in het verslag.

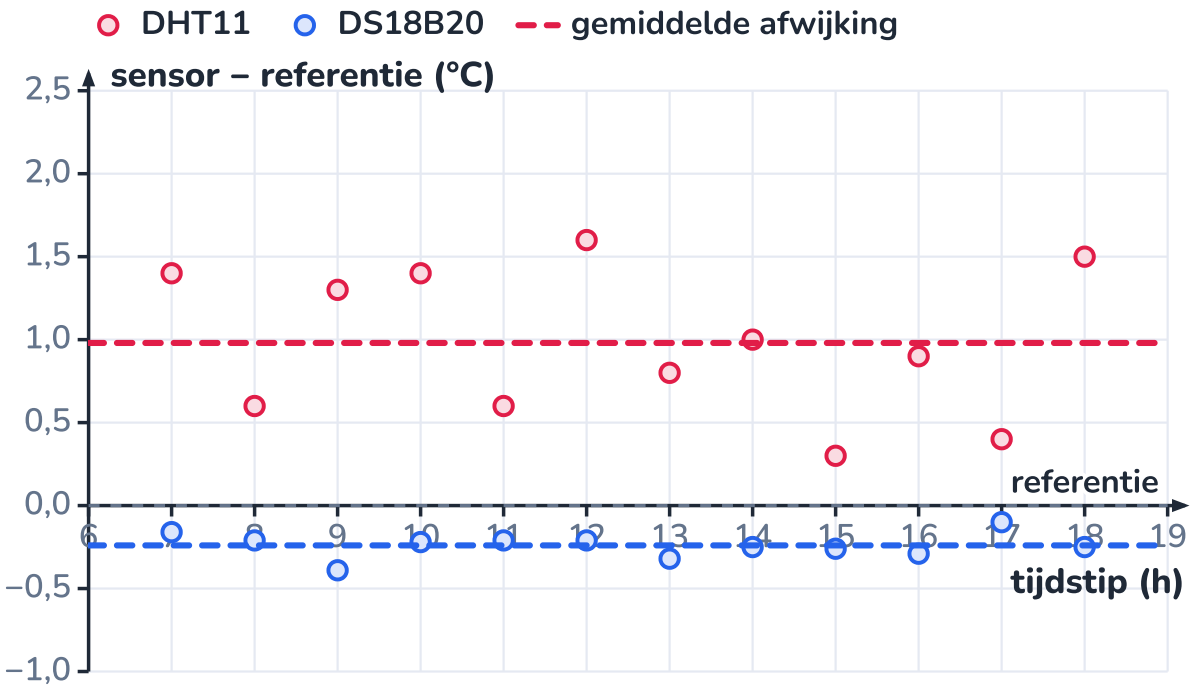
Klopt dit? Kijk naar de grafiek: zeven punten liggen in een strook van drie procentpunten, één punt ligt 15 procentpunten boven de hoogste van de andere. Wat je met het oog ziet, moet je berekening bevestigen.

UITGEWERKT VOORBEELD · 1.16

Habitat: hoe betrouwbaar is een goedkope sensor?

In de meetkas hangen twee goedkope temperatuursensoren naast een geijkte referentiethermometer (fout ± 0,1 °C): een DHT11, die hele graden toont, en een DS18B20, die stapjes van 0,0625 °C toont. Een dag lang lees je ze elk uur af. Hoe groot is de systematische en de toevallige fout van elke sensor, en welke is goed genoeg om de klimaatmodule tot op 1 °C te regelen?

tijdstip	referentie (°C)	DHT11 (°C)	DS18B20 (°C)
7 h	12,6	14	12,44
8 h	13,4	14	13,19
9 h	14,7	16	14,31
10 h	16,6	18	16,38
11 h	19,4	20	19,19
12 h	22,4	24	22,19
13 h	25,2	26	24,88
14 h	27,0	28	26,75
15 h	26,7	27	26,44
16 h	25,1	26	24,81
17 h	22,6	23	22,50
18 h	19,5	21	19,25



De DHT11 ligt gemiddeld 1,0 °C te hoog en spreidt; de DS18B20 ligt 0,24 °C te laag maar heel regelmatig.

GEGEVEN

GEVRAAGD

- twaalf gelijktijdige aflezingen van de referentie en de twee sensoren

- systematische en toevallige fout van elke sensor
- welke sensor is goed genoeg?

OPLOSSING

1. Idee: de referentie is (bijna) de echte waarde. Het verschil *sensor – referentie* is dus de fout van de sensor bij die meting.
2. Verschillen voor de DHT11: 1,4; 0,6; 1,3; 1,4; 0,6; 1,6; 0,8; 1,0; 0,3; 0,9; 0,4; 1,5 °C. Voor de DS18B20: −0,16; −0,21; −0,39; −0,22; −0,21; −0,21; −0,32; −0,25; −0,26; −0,29; −0,10; −0,25 °C.
3. Het **gemiddelde** verschil is de systematische fout: DHT11 $+0{,}98\,^\circ\mathrm{C}$, DS18B20 $-0{,}24\,^\circ\mathrm{C}$.
4. De **standaardafwijking** van de verschillen is de toevallige fout: DHT11 $0{,}45\,^\circ\mathrm{C}$, DS18B20 $0{,}07\,^\circ\mathrm{C}$.
5. Hoe ver kan één aflezing ernaast zitten? Ongeveer 95 % valt binnen systematische fout + $2s$: DHT11 $0{,}98 + 0{,}90 \approx 1{,}9\,^\circ\mathrm{C}$; DS18B20 $0{,}24 + 0{,}15 \approx 0{,}4\,^\circ\mathrm{C}$.
6. Trek je de systematische fout af (een **correctie**: DHT11-waarde min 1,0 °C), dan blijft voor de DHT11 nog $2s \approx 0{,}9\,^\circ\mathrm{C}$ over.

ANTWOORD

DHT11: systematisch $+1{,}0\,^\circ\mathrm{C}$, toevallig $0{,}45\,^\circ\mathrm{C}$. DS18B20: systematisch $-0{,}24\,^\circ\mathrm{C}$, toevallig $0{,}07\,^\circ\mathrm{C}$. Voor een regeling tot op 1 °C volstaat de DS18B20 zonder correctie; de DHT11 pas na correctie, en dan nog krap.

Klopt dit? Beide resultaten passen bij wat de fabrikanten beloven: ± 2 °C voor de DHT11, ± 0,5 °C voor de DS18B20. Let op het geldigheidsgebied: je vergeleek alleen tussen 12,6 en 27,0 °C, op één dag. Of de afwijking bij 5 °C of 35 °C even groot is, weet je niet. Dat onderzoek je in hoofdstuk 3 met een kalibratierechte.

NASLAG · 1.17

Overzicht: meetfouten in formules en regels

17/18

meetresultaat

$$x = x_{\text{gemeten}} \pm \Delta x$$

symbool	betekenis	eenheid
Δx	absolute fout	eenheid van x

relatieve fout

$$\text{relatieve fout} = \frac{\Delta x}{|x|}$$

symbool	betekenis	eenheid
$\Delta x/\lvert x\rvert$	fout als deel van de waarde	geen (vaak in %)

gemiddelde

$$\bar{x} = \frac{\sum x_i}{n}$$

symbool	betekenis	eenheid
n	aantal metingen	–
x_i	de i-de meting	eenheid van x

standaardafwijking

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$$

symbool	betekenis	eenheid
s	spreiding van de meetreeks	eenheid van x

fout op een som of verschil

$$\Delta(a \pm b) = \Delta a + \Delta b$$

fout op een product of quotiënt

$$\frac{\Delta(a \cdot b)}{|a \cdot b|} \approx \frac{\Delta(a/b)}{|a/b|} \approx \frac{\Delta a}{|a|} + \frac{\Delta b}{|b|}$$

fout op een macht

$$\frac{\Delta(a^n)}{|a^n|} \approx n \cdot \frac{\Delta a}{|a|}$$

regel	voorbeeld
fout: 1 beduidend cijfer (2 als het eerste een 1 is)	0,0347 → 0,03; 0,0153 → 0,015
waarde: afronden op dezelfde decimaal als de fout	(18,194 ± 0,347) L → (18,2 ± 0,3) L
× en ÷: minste aantal beduidende cijfers	12,5 × 3,3 = 41
+ en −: minste aantal decimalen	15,42 + 3,1 = 18,5
pas op het einde afronden	tussenresultaten met alle cijfers bewaren
uitschieter: toets tegen de andere metingen	verder dan 3 s van hun gemiddelde én oorzaak gevonden

wat	rekenblad (Nederlands)	rekenblad (Engels)	rekenmachine
gemiddelde	`=GEMIDDELDE(B2:B11)`	`=AVERAGE(B2:B11)`	$\bar{x}$
standaardafwijking	`=STDEV.S(B2:B11)`	`=STDEV.S(B2:B11)`	s_x of σ_{n-1}
mediaan	`=MEDIAAN(B2:B11)`	`=MEDIAN(B2:B11)`	(sorteren)
bereik	`=MAX(B2:B11)-MIN(B2:B11)`	idem	max − min
aantal metingen	`=AANTAL(B2:B11)`	`=COUNT(B2:B11)`	n

In een Nederlandstalig rekenblad scheid je argumenten met een puntkomma, bijvoorbeeld `=AFRONDEN(B2;2)`; in een Engelstalig met een komma: `=ROUND(B2,2)`.

HET GEHEEL · 1.18

Waar meetfouten terugkomen

18/18

Dit hoofdstuk opent Intermediate met de vraag die onder elke meting ligt: hoe zeker ben je? Elk volgend hoofdstuk dat meet, rekent of een sensor uitleest, bouwt erop verder. Je leert hier de taal (waarde ± fout, systematisch en toevallig, $\bar{x}$ en s) waarin je vanaf nu elk resultaat noteert.

F2 Onderzoeken: van vraag tot besluit

Herhalen en het gemiddelde nemen: hier krijgt de spreiding een getal, de standaardafwijking.

F3 Meten: grootheden en eenheden

Instrumenten, afleesfouten en zinvol afronden: hier worden het regels voor afronden en beduidende cijfers.

I2 Experimenten ontwerpen en rapporteren

Hoeveel herhalingen heb je nodig? Dat hangt af van de spreiding die je hier leerde meten.

I3 Verbanden in data

Een sensor kalibreren met een rechte haalt de systematische fout weg over het hele meetbereik.

I22 De spanningsdeler en sensoren

Een sensor uitlezen en ijken tegenover een referentie, nu met een microcontroller.

I28 Functies, lijsten en data in Python

Gemiddelde en standaardafwijking van een hele week sensordata, berekend in Python.

E6 Statistiek en meetonzekerheid

De normale verdeling, de standaardfout $s/\sqrt{n}$ en fouten die kwadratisch optellen.

E40 Sensorfusie en schatten

Ruis en onzekerheid van meerdere sensoren slim combineren tot één betrouwbare meting.

Habitat

De meetkas uit Fundamental meet met goedkope sensoren. Voor de klimaat- en energiemodule moet je weten hoe ver je die metingen kan vertrouwen: een verwarming die bij 18 °C inschakelt met een sensor die 1 °C te hoog meet, houdt de kas op 17 °C. In opdracht 1.14 licht je twee sensoren door en stel je een correctie voor; in hoofdstuk 3 maak je daar een kalibratierechte van.

SAMENVATTING

In het kort

- Elke meting heeft een onzekerheid. Je noteert ze als $(x \pm \Delta x)$ eenheid, met de fout op één beduidend cijfer (twee als het een 1 is) en de waarde op dezelfde decimaal.
- Een **toevallige fout** wisselt van teken en wordt kleiner door te herhalen; een **systematische fout** duwt altijd dezelfde kant op en vind je alleen door te vergelijken met een referentie.
- **Precies** = weinig spreiding; **nauwkeurig** = gemiddeld dicht bij de echte waarde. Resolutie (aantal cijfers) is geen nauwkeurigheid.
- De relatieve fout $\Delta x/|x|$ vergelijkt metingen van verschillende grootte; kleine voorwerpen vragen fijne instrumenten.
- Bij optellen en aftrekken tellen de absolute fouten op, bij vermenigvuldigen en delen de relatieve; bij een macht a^n wordt de relatieve fout n keer zo groot.
- Een meetreeks vat je samen met het gemiddelde $\bar{x}$ en de standaardafwijking s (delen door $n-1$); het bereik en de mediaan zijn snelle aanvullingen.
- Een uitschieter toets je tegen $\bar{x}$ en s van de *andere* metingen, en je laat hem alleen weg met een oorzaak, die je vermeldt.

BEGRIPPEN

Wat je nu kent

meetfout (onzekerheid)

Het verschil dat er kan zijn tussen de gemeten en de echte waarde. Geen vergissing: elke meting heeft er een.

systematische fout

Fout die elke meting evenveel dezelfde kant op duwt (een nulpunt dat niet klopt, een slecht geijkte sensor). Verdwijnt niet door te herhalen.

toevallige fout

Fout die bij elke meting anders is, soms te groot en soms te klein. Wordt kleiner door te herhalen en te middelen.

nauwkeurig

Gemiddeld dicht bij de echte waarde: de systematische fout is klein (in normtaal: juist).

precies

Herhaalde metingen liggen dicht bij elkaar: de toevallige fout is klein.

resolutie

De kleinste stap die een instrument kan tonen. Zegt niets over hoe juist die stap is.

absolute fout

De fout in dezelfde eenheid als de meting, genoteerd als Δx in $x \pm \Delta x$.

relatieve fout

De absolute fout gedeeld door de waarde, $\Delta x / |x|$, vaak in procent. Heeft geen eenheid.

beduidende cijfers

De cijfers van een getal die informatie over de meting dragen; nullen vooraan tellen niet mee.

gemiddelde

De som van de metingen gedeeld door hun aantal: de beste schatting van de echte waarde.

bereik

Het verschil tussen de grootste en de kleinste meting van een reeks.

standaardafwijking

Maat voor de spreiding van een meetreeks: $s = \sqrt{\sum (x_i - \bar{x})^2 / (n-1)}$.

mediaan

De middelste waarde van een gesorteerde reeks; weinig gevoelig voor uitschieters.

uitschieter

Een meting die opvallend ver van de andere ligt; eerst de oorzaak zoeken, dan beslissen.

HANDBOEK · HOOFDSTUK 2

Experimenten ontwerpen en rapporteren

Een eerlijke proef ontwerpen, alles noteren en er een verslag over schrijven dat een ander kan controleren

Een goed experiment begint lang voor de eerste meting: bij een scherpe vraag, een toetsbare hypothese en een opzet die toeval, vooroordelen en verstorende factoren buitensluit. En het eindigt pas als iemand anders je werk kan nalezen en overdoen. In dit hoofdstuk leer je proeven ontwerpen met een controlegroep, voldoende herhalingen, loting en blind meten; je houdt een labojournaal bij, schrijft een wetenschappelijk verslag en vermeldt je bronnen correct.

7× uitleg · 2× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · ± 7 lestijden

O Onderzoeken en meten · D Ontwerpen en maken

Een eerlijke vergelijking van twee lichtkleuren: elk vak een eigen lamp, elke pot een nummer, elke meting in het journaal.

Na dit hoofdstuk

- Je kan een onderzoeksvraag scherp formuleren en er een toetsbare hypothese en een meetbare voorspelling bij schrijven.
- Je kan de onafhankelijke, afhankelijke en gecontroleerde variabelen van een proef benoemen en een verstorende variabele herkennen.
- Je kan uitleggen waarom een controlegroep, herhalingen, loting en blind meten nodig zijn, en ze in een proefopzet gebruiken.
- Je kan met een vuistregel schatten hoeveel herhalingen een proef nodig heeft, en herkent pseudoreplicatie.
- Je kan een labojournaal bijhouden dat een ander kan controleren.
- Je kan een wetenschappelijk verslag opbouwen volgens inleiding, methode, resultaten en discussie.
- Je kan bronnen zoeken, beoordelen met vijf vragen, en correct vermelden in de tekst en in een bronnenlijst (APA).

UITLEG · 2.1

Wat je al weet, en wat nu komt

1/14

In Fundamental (hoofdstuk F2) leerde je de onderzoekscyclus: een vraag, een hypothese, een proef, waarnemingen en een besluit. Je onderscheidde de **onafhankelijke variabele** (die jij verandert), de **afhankelijke variabele** (die je meet) en de **gecontroleerde variabelen** (die je gelijk houdt), want alleen

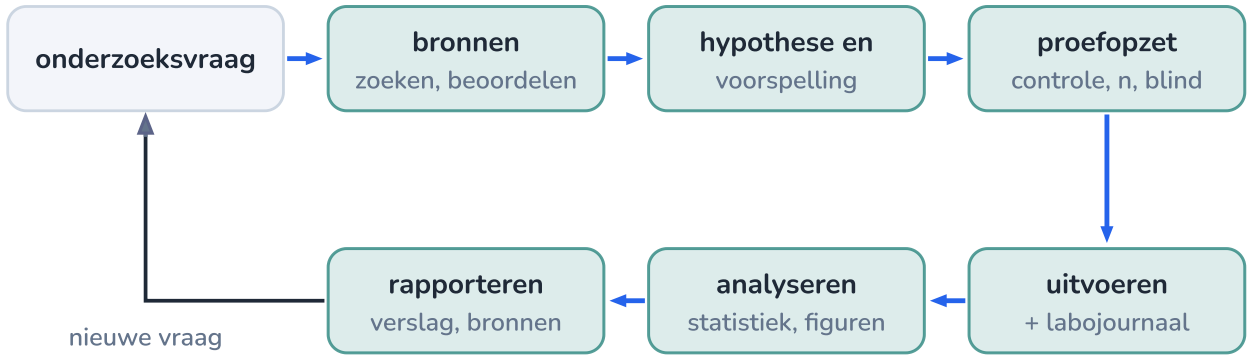
dan is een proef eerlijk. Je herhaalde metingen en nam het gemiddelde, en je hield waarneming en interpretatie uit elkaar.

Op Intermediate verdiep je elke stap. Je formuleert de vraag zo scherp dat ze **meetbaar** wordt, je maakt onderscheid tussen een **hypothese** en een **voorspelling**, je ontwerpt een proef met een **controlegroep**, genoeg **herhalingen**, **loting** en **blind meten**, je houdt een **labojournaal** bij, en je rapporteert in de vorm die onderzoekers wereldwijd gebruiken, met **bronnen** die een ander kan terugvinden. Uit hoofdstuk 1 neem je mee: elke meting heeft een fout, en een verschil betekent pas iets als het groter is dan de spreiding.

Waarom zoveel moeite?

Omdat je jezelf het makkelijkst voor de gek houdt. Wie een resultaat verwacht, ziet het sneller, meet net iets anders of vergeet de proef die mislukte. Een goed proefontwerp is een reeks afspraken die je tegen je eigen verwachting beschermt.

de onderzoekscyclus · gekleurd: wat je op Intermediate verdiept



De cyclus uit Fundamental, met de stappen die je nu verdiept. Het verslag leidt vaak tot een nieuwe vraag.

VERHAAL · 2.2

Twaalf zieke matrozen

2/14

Op lange zeereizen in de achttiende eeuw stierven meer matrozen aan **scheurbuik** dan in gevechten: hun tandvlees bloedde, wonden genazen niet meer, ze werden doodmoe. Er deden tientallen 'remedies' de ronde, en iedere scheepsarts zwoer bij de zijne.

In mei 1747 deed de Schotse scheepsarts James Lind op het oorlogsschip HMS Salisbury iets nieuws. Hij koos twaalf matrozen met scheurbuik die zo goed mogelijk op elkaar leken, liet ze in hetzelfde deel van het schip liggen en gaf ze hetzelfde basisvoedsel. Daarna verdeelde hij ze in zes paren. Elk paar kreeg een andere behandeling: cider, verdund zwavelzuur, azijn, zeewater, een kruidenpasta, of elke dag twee sinaasappelen en een citroen.

Het paar met de citrusvruchten herstelde veruit het snelst: na zes dagen was een van beiden weer geschikt voor dienst. Lind publiceerde zijn resultaten in 1753. Toch duurde het tot 1795 voor de Britse marine citroensap standaard meegaf, en pas in de twintigste eeuw werd duidelijk waarom het werkte: scheurbuik is een tekort aan vitamine C.

Wat Lind goed deed, en wat beter kon

Goed: vergelijkbare patiënten, dezelfde omstandigheden, één verschil per groep, en meerdere behandelingen naast elkaar. **Beter:** twee patiënten per groep is heel weinig; Lind liet het toeval niet beslissen wie welke behandeling kreeg; en iedereen wist wie wat kreeg. Loting, voldoende herhalingen en blind meten zijn precies de onderwerpen van dit hoofdstuk.

Zes behandelingen naast elkaar, telkens twee patiënten: een van de eerste vergelijkende proeven in de geneeskunde.

UITLEG · 2.3

Van onderwerp naar vraag, hypothese en voorspelling

3/14

Een onderwerp ('licht en planten') is nog geen onderzoek. Je trechtert het in drie stappen.

1. Een onderzoeksvraag is specifiek (één onafhankelijke en één afhankelijke variabele), **meetbaar** en **haalbaar** met jouw tijd en materiaal. Handige vormen: 'Wat is het effect van X op Y?' of 'Hoe hangt Y af van X?'. Meetbaar maken heet **operationaliseren**: 'groei' wordt 'verse massa van het bovengrondse deel na 28 dagen, in gram'.

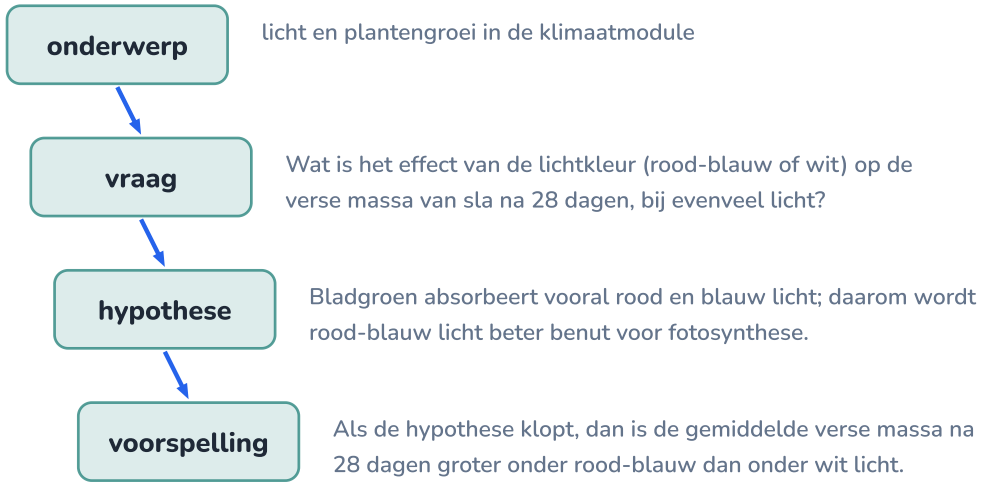
te vaag	onderzoekbaar
Is rood licht goed voor planten?	Wat is het effect van de lichtkleur (rood-blauw of wit) op de verse massa van sla na 28 dagen, bij evenveel licht?
Isoleert karton?	Hoe hangt de afkoelsnelheid van 200 mL water van 60 °C af van het aantal lagen karton rond de beker?

te vaag	onderzoekbaar
Werkt onze sensor goed?	Hoe groot is de systematische fout van de DHT11 tegenover een referentiethermometer tussen 10 en 30 °C?

2. Een hypothese is een toetsbare *verklaring*: wat je denkt dat er gebeurt en waarom. Ze steunt op wat je al weet of in bronnen vond. Een goede hypothese kan fout blijken: 'planten groeien zoals ze willen' kan je met geen enkele meting tegenspreken, en is dus geen hypothese.

3. Een voorspelling zegt wat je *zal meten* als de hypothese klopt, in de vorm 'als ..., dan ...'. De voorspelling maakt de hypothese toetsbaar: komt ze niet uit, dan moet de hypothese bijgesteld worden.

> **Een proef bewijst een hypothese**
>
> Een proef kan een hypothese **ondersteunen** of **tegenspreken**, maar niet bewijzen: er kan altijd een andere verklaring zijn voor hetzelfde resultaat. Daarom schrijf je 'de resultaten ondersteunen de hypothese', niet 'de hypothese is bewezen'. Een voorspelling die niet uitkomt, is geen mislukte proef: ze is een resultaat.



Elke stap is smaller en scherper dan de vorige. Pas bij de voorspelling weet je precies wat je moet meten.

UITLEG · 2.4

Eerlijk vergelijken: controlegroep en loting

4/14

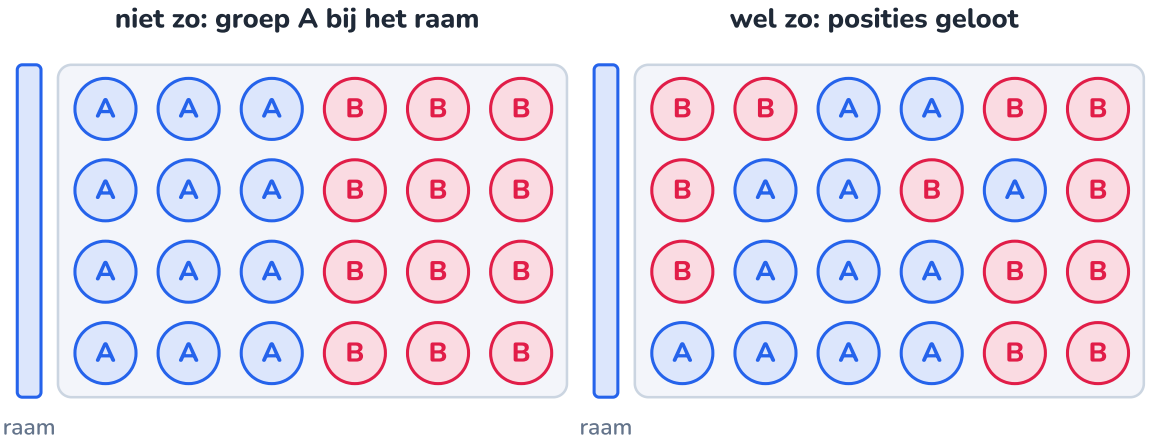
Wil je weten wat een behandeling doet, dan heb je iets nodig om mee te vergelijken. De **controlegroep** krijgt de gewone behandeling (wit licht, gewone voeding) of geen behandeling (0 g/L zout), en is voor de rest identiek aan de **proefgroep**. Zonder controlegroep weet je niet of een verandering door jouw behandeling komt of door iets anders dat intussen gebeurde: het weer, de seizoenen, de leeftijd van de planten.

Een controlegroep is iets anders dan gecontroleerde variabelen. Gecontroleerde variabelen houd je gelijk in álle groepen (temperatuur, pot, ras). Maar je kan nooit alles gelijk houden: in een kas is het bij het raam lichter en kouder, bij de deur tochtiger. Daarom **loot** je (**randomiseren**): het toeval beslist welke pot welke behandeling krijgt en waar hij staat. Zo verdelen onbekende verschillen zich eerlijk over de groepen, in plaats van samen te vallen met je behandeling.

Loten kan met een dobbelsteen, briefjes uit een pot of een rekenblad: zet naast elke pot `=ASELECT()` (Engels: `=RAND()`), sorteer op die kolom, en geef de eerste helft behandeling A.

Verstorende variabele

Een **verstorende variabele** verandert samen met je onafhankelijke variabele en kan het resultaat verklaren. Staan alle planten van groep A bij het raam (linkse figuur), dan kan je een verschil niet meer toeschrijven aan de behandeling: licht en temperatuur verschillen mee. Een ander voorbeeld: rood-blauwe en witte ledlampen hebben een ander rendement en kunnen bij evenveel licht verschillend warm worden. Meet daarom ook de temperatuur in elk vak.



Links valt de behandeling samen met de afstand tot het raam; rechts beslist het lot, zodat licht, kou en tocht zich over beide groepen verdelen.

UITGEWERKT VOORBEELD · 2.5

Een slaproef ontwerpen

5/14

Voor de klimaatmodule wil je weten of het zin heeft om rood-blauwe groeiledlampen te kopen in plaats van witte. Je hebt een rek met zes vakjes, elk met een eigen ledpaneel, en 24 jonge slaplantjes van hetzelfde ras. Ontwerp de proef.

GEGEVEN

- zes vakjes met elk een ledpaneel (drie rood-blauw, drie wit)
- 24 slaplantjes, hetzelfde ras en dezelfde leeftijd
- balans ($\pm 0{,}1$ g), PAR-meter (meet de hoeveelheid bruikbaar licht)

GEVRAAGD

- onderzoeksvraag, hypothese, voorspelling
- variabelen, controlegroep, loting, meting

OPLOSSING

1. **Vraag:** Wat is het effect van de lichtkleur (rood-blauw of wit) op de verse massa van sla na 28 dagen, bij evenveel bruikbaar licht?
2. **Hypothese:** bladgroen absorbeert vooral rood en blauw licht, dus rood-blauw licht wordt beter benut. **Voorspelling:** als dat klopt, dan is de gemiddelde verse massa na 28 dagen onder rood-blauw groter dan onder wit.
3. **Onafhankelijke variabele:** de lichtkleur. **Afhankelijke:** de verse massa van het bovengrondse deel na 28 dagen (g). **Gecontroleerd:** de hoeveelheid bruikbaar licht (elk paneel afgesteld op $200\ \mu\text{mol}/(\text{m}^2 \cdot \text{s})$ met de PAR-meter), 16 uur licht per dag, ras, potgrond, potmaat, water, voeding. **Bewaken:** de temperatuur in elk vak.
4. **Eerlijk licht.** Niet 'lampen van hetzelfde vermogen', maar 'evenveel licht op de planten': een lamp met een beter rendement geeft bij hetzelfde vermogen meer licht, en dan vergelijk je de hoeveelheid, niet de kleur.

5 **Controlegroep:** wit licht, de gewone keuze. **Loting:** welke drie vakjes rood-blauw krijgen en welke plant in welk vakje komt (vier per vakje), met een rekenblad. **Herhalingen:** drie lampen per kleur, twaalf planten per kleur.

6 **Meten:** na 28 dagen afknippen op het grondoppervlak en wegen. De planten krijgen een code; wie weegt, weet niet welke code bij welke kleur hoort.

ANTWOORD

Een proef met twee groepen van twaalf planten onder elk drie lampen, evenveel bruikbaar licht, geloot, blind gewogen, met wit licht als controle en de temperatuur als bewaakte verstorende variabele.

Klopt dit? Test je ontwerp met de vraag: 'Als ik een verschil vind, kan ik dan nog een andere oorzaak bedenken dan de lichtkleur?' Elke oorzaak die je kan bedenken, moet het ontwerp uitsluiten of meten.

UITLEG · 2.6

Hoeveel herhalingen?

6/14

Geen twee planten, schaaltjes of proefpersonen zijn gelijk. Die natuurlijke spreiding, samen met je meetfout (hoofdstuk 1), maakt dat één plant per groep niets zegt. De grafiek toont een gesimuleerde proef waarin voeding B de planten *echt* 13 g zwaarder maakt. Met één plant per groep zou je het omgekeerde besluiten; pas vanaf een tiental planten liggen de gemiddelden stabiel in de juiste volgorde.

Hoeveel herhalingen je nodig hebt, hangt af van twee getallen: de **spreiding** s binnen een groep (uit een vooronderzoek of uit eerdere proeven) en het kleinste **verschil** δ dat je wil kunnen aantonen. Een bruikbare vuistregel:

VUISTREGEL VOOR DE STEEKPROEFGROOTTE

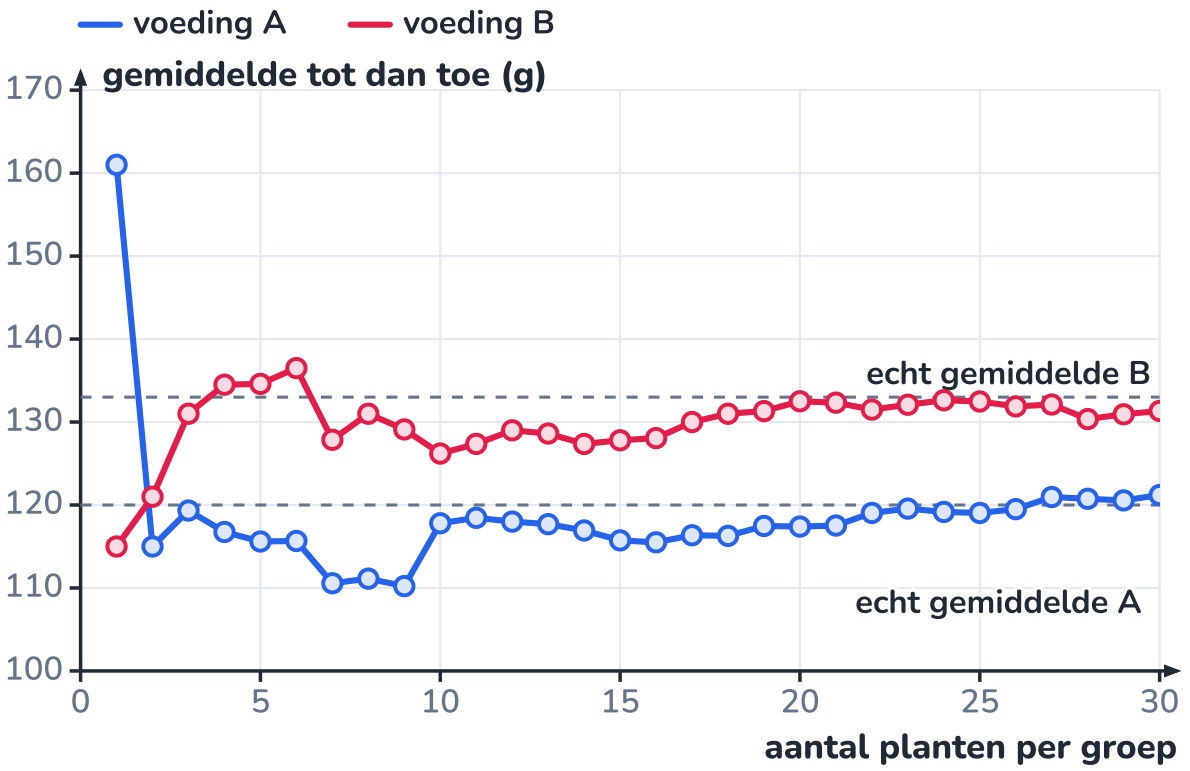
$$n \approx 16 \cdot \left(\frac{s}{\delta}\right)^2 \quad \text{per groep}$$

Met zoveel herhalingen vind je een echt verschil δ in ongeveer 8 op de 10 proeven terug. Een half zo groot verschil vraagt vier keer zoveel herhalingen. In Expert (hoofdstuk E6) leer je waar die 16 vandaan komt.

Voor de sla: een vooronderzoek gaf $s \approx 18\,\text{g}$. Wil je een verschil van 15 g kunnen zien, dan is $n \approx 16 \cdot (18/15)^2 = 23$ planten per groep nodig. Met twaalf planten per groep zie je betrouwbaar alleen verschillen vanaf ongeveer $4s/\sqrt{12} \approx 21\,\text{g}$.

Pseudoreplicatie: tellen wat niet onafhankelijk is

Twaalf planten onder *één* lamp zijn twaalf metingen, maar maar één herhaling van 'die lamp': een lamp die toevallig iets feller brandt, geeft alle twaalf hetzelfde voordeel. Hetzelfde geldt voor tien metingen aan dezelfde plant. Tel als herhaling alleen wat de behandeling onafhankelijk kreeg: daarom gebruikt de slaproef drie lampen per kleur.



Gesimuleerd: 30 planten per groep, echt verschil 13 g. Bij kleine aantallen springen de gemiddelden alle kanten op, zelfs in de verkeerde volgorde.

UITLEG · 2.7

Blind meten en verwachtingen

7/14

Rond 1900 werd in Berlijn een paard beroemd: 'Kluge Hans', de slimme Hans, die rekensommen oploste door met zijn hoef te tikken. Zijn eigenaar geloofde er zelf in. In 1907 onderzocht de psycholoog Oskar Pfungst het paard. Wist de vraagsteller het antwoord niet, of kon Hans hem niet zien, dan faalde het paard. Hans rekende niet: hij las onbewuste signalen in de houding en het gezicht van de vraagsteller, die ontspande zodra het juiste aantal tikken bereikt was.

Onderzoekers zijn niet anders. Wie weet welke plant de 'goede' behandeling kreeg, knipt die net iets lager af, telt een twijfelachtig kiemworteltje net wel mee, of beoordeelt een blad net iets groener. Niet uit oneerlijkheid, maar omdat verwachtingen onze waarneming kleuren. De remedie:

techniek	hoe	beschermt tegen
codes	elke proefeenheid krijgt een nummer; de sleutel ligt bij iemand anders	verwachting van wie meet
blind (enkelblind)	wie meet of beoordeelt, weet niet welke behandeling er bij hoort	verwachting van wie meet
dubbelblind	ook de proefpersoon weet het niet (bv. met een placebo, een nepbehandeling)	ook het placebo-effect bij mensen
operationele definitie vooraf	'gekiemd = worteltje van minstens 2 mm', vastgelegd vóór de proef	grenzen verschuiven tijdens het meten

Ook als het meettoestel objectief is

Een balans heeft geen verwachtingen, maar de hand die de plant afknipt, de aarde eraf klopt en beslist of een geel blad nog meetelt, wel. Blind meten kost weinig: een rol plakband en een klasgenoot die de sleutel bewaart.

Wegen op code: wie de balans afleest, weet niet welke plant onder welke lamp stond.

PROEF · 2.8

Kiemt tuinkers minder in zout water?

8/14

ONDERZOEKSVRAAG

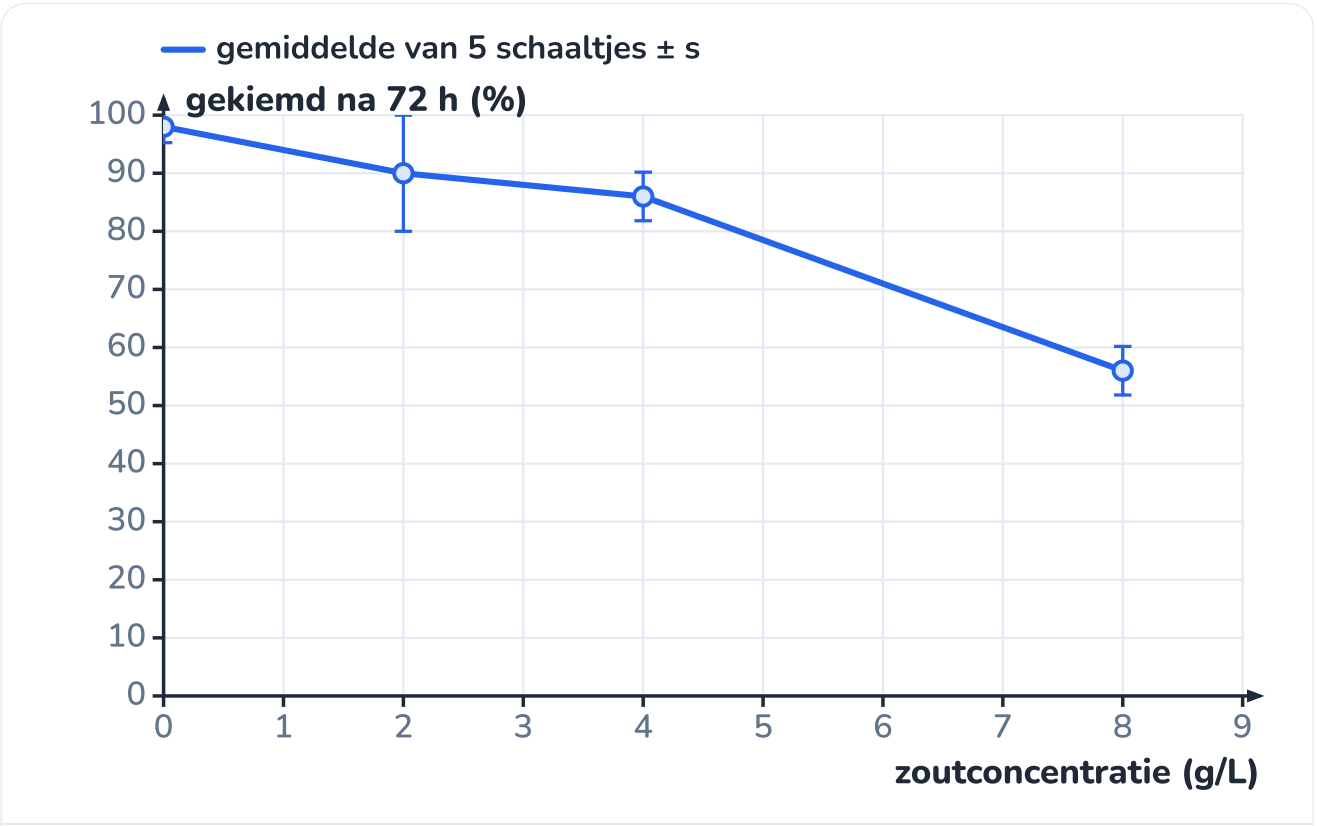
Wat is het effect van de keukenzoutconcentratie van het gietwater (0, 2, 4 en 8 g/L) op het percentage gekiemde tuinkerszaadjes na 72 uur?

MATERIAAL

- 20 petrischalen met filtreerpapier
- 400 tuinkerszaadjes uit één zakje
- keukenzout, balans, maatcilinder, 4 flessen
- pipet of spuitje van 5 mL
- plakband en stift voor codes
- loep en lat

Veiligheid

Keukenzout is onschadelijk; was je handen na het werken met zaden die met schimmelwerend middel behandeld kunnen zijn.



Elk punt is het gemiddelde van vijf schaaltjes; de balkjes tonen één standaardafwijking.

WERKWIJZE

1. Maak vier oplossingen: 0, 2, 4 en 8 g keukenzout per liter kraantjeswater. De fles met 0 g/L is de controle.
2. Laat een klasgenoot de 20 schalen loten over de vier concentraties (vijf schalen per concentratie) en er een code K1 tot K20 op kleven. Hij bewaart de sleutel.
3. Leg in elke schaal filtreerpapier, bevochtig met 4 mL van de juiste oplossing en leg er 20 zaadjes op.
4. Zet de schalen in een geloot patroon op dezelfde vensterbank, uit de volle zon. Noteer alles in je labojournaal.
5. Tel na 72 uur, zonder de sleutel te kennen, de gekiemde zaadjes. Leg vooraf vast: gekiemd = worteltje van minstens 2 mm.
6. Open de sleutel pas na het tellen en bereken per concentratie het gemiddelde percentage en de standaardafwijking.

WAARNEMING

NaCl (g/L)	gekiemd per schaaltje (van 20)	gemiddeld (%)	s (%)
0	20 · 19 · 19 · 20 · 20	98	2,7
2	16 · 20 · 20 · 16 · 18	90	10,0
4	17 · 17 · 16 · 18 · 18	86	4,2
8	11 · 11 · 12 · 10 · 12	56	4,2

Tussen 0 en 4 g/L daalt de kieming weinig; bij 8 g/L kiemt nog iets meer dan de helft. De spreiding tussen schaaltjes van dezelfde concentratie is 3 tot 10 procentpunten.

VERKLARING

Zout water trekt water uit het zaadje: water verplaatst zich door osmose naar de kant met de hoogste zoutconcentratie. Hoe zouter het gietwater, hoe moeilijker het zaadje genoeg water opneemt om te kiemen. In deze proef is het verschil tussen 0 en 8 g/L (ongeveer 40 procentpunten) veel groter dan de spreiding: dat is een duidelijk effect. Het verschil tussen 2 en 4 g/L (4 procentpunten) is kleiner dan de spreiding: daarover kan deze proef niets besluiten.

Kijk naar het ontwerp: een controle (0 g/L), vijf onafhankelijke schalen per concentratie, loting van posities, codes en blind tellen met een vooraf vastgelegde definitie. Ter vergelijking: zeewater bevat ongeveer 35 g zout per liter.

ONTLEED · 2.9

Een bladzijde uit een labojournaal

9/14

Het **labojournaal** is het geheugen van je onderzoek: alles wat je deed, mat en zag, op het moment zelf. In een onderzoekslabo is het een officieel document: wie een resultaat publiceert, moet het kunnen tonen. De regels zijn eenvoudig. Een gebonden schrift met genummerde bladzijden (of een digitaal journaal dat elke wijziging bewaart). Schrijf in pen, meteen, met datum en uur. Noteer ruwe gegevens met eenheden, niet alleen berekende. Gom of verwijder nooit: haal door met één lijn, zodat het leesbaar blijft, en schrijf de correctie en de reden erbij. En noteer ook wat misloopt: juist daar zitten vaak de verklaringen.

Drie dagen kiemproef op één bladzijde: elke notitie heeft een datum, elke meting een eenheid, elke correctie een reden.

1. **Datum, uur en fase**
Elke notitie begint met datum en uur. Zo kan je later reconstrueren wat wanneer gebeurde, bijvoorbeeld hoeveel uur de zaadjes echt kiemden.

2. **Genummerde bladzijde**
Genummerde bladzijden tonen dat er niets uitgescheurd is. Verwijs ernaar vanuit je verslag.

3. **Doel en materiaal**
Kort: wat wil je weten, met welk materiaal. Het lotnummer van de zaadjes lijkt overbodig, tot een volgende proef met een ander zakje andere resultaten geeft.

4. **Codes en sleutel**
De codes staan in het journaal; de sleutel bewaart iemand anders. Zo is blind tellen mogelijk en toch alles terug te vinden.

5. **Schets**
Een snelle schets van de opstelling zegt meer dan een alinea: waar stond wat, bij welke temperatuur.

6. **Afwijking van het plan**
Iets liep anders dan gepland: noteer wat, wanneer en wat je deed. In het verslag vermeld je het in de methode of de discussie.

7. **Operationele definitie**
Wat telt als gekiemd, vastgelegd vóór het tellen. Een volgende onderzoeker kan zo exact hetzelfde meten.

8. **Correctie, leesbaar**
Doorgehaald met één lijn, de nieuwe waarde ernaast, met de reden en de initialen. Nooit gommen of overschrijven: de oorspronkelijke notitie blijft zichtbaar.

9. **Onverwachte waarneming**
Schimmel in schaal K12: misschien verklaart die later een afwijkend resultaat. Een foto krijgt een nummer dat naar deze bladzijde verwijst.

10. **Volgende stap en namen**
Wat moet er nu gebeuren, en wie werkte mee? Bij een vraag over de metingen weet iedereen wie hij moet aanspreken.

UITLEG · 2.10

Het wetenschappelijk verslag

10/14

Onderzoekers over de hele wereld rapporteren in dezelfde volgorde, vaak afgekort als **IMRaD**: *Introduction, Methods, Results and Discussion*. Wie die vorm kent, vindt in elk artikel meteen wat hij zoekt. Het verslag heeft de vorm van een zandloper: breed beginnen, smal worden bij je eigen proef, en weer breed eindigen.

deel	wat staat erin	tijd en toon
titel	het onderwerp in één zin, met de variabelen	concreet, geen vraag of slogan
samenvatting	vraag, methode, belangrijkste resultaat en besluit, in 100 à 200 woorden	als laatste geschreven
inleiding	achtergrond met bronnen, waarom het ertoe doet, onderzoeksvraag, hypothese, voorspelling	tegenwoordige tijd voor wat bekend is
methode	materiaal (met meetfouten), opstelling, werkwijze, variabelen, aantal herhalingen, hoe je loot en blind meet	verleden tijd; zo volledig dat een ander het kan overdoen
resultaten	verwerkte gegevens in tabellen en figuren, met gemiddelde ± s; wat opvalt, zonder uitleg	verleden tijd; feiten, geen meningen
discussie	klopt de voorspelling? verklaring, vergelijking met bronnen, fouten en grenzen, vervolgonderzoek	voorzichtig: 'wijst op', 'past bij'
besluit	het antwoord op de onderzoeksvraag, in enkele zinnen	kort en precies
bronnen	elke gebruikte bron, in één vaste stijl (bv. APA)	alfabetisch op eerste auteur

Figuren en tabellen krijgen een nummer en een titel, zodat je ernaar kan verwijzen ('figuur 2 toont …'). In de natuurwetenschappen staat de titel meestal boven een tabel en onder een figuur; in de APA-stijl

staan nummer en titel bij beide erboven. Volg de afspraak van je school. Elke as en elke kolom heeft een grootheid en een eenheid, elk resultaat een fout.

Wetenschappelijke integriteit

Drie dingen zijn in onderzoek nooit toegestaan: gegevens **verzinnen**, gegevens **vervalsen** (weglaten wat niet past, waarden bijsturen) en **plagiaat** (andermans werk als het jouwe voorstellen). Een eerlijk verslag met een tegenvallend resultaat is meer waard dan een mooi verslag dat niet klopt.

De zandloper: de inleiding vernauwt van het onderwerp naar jouw vraag, de discussie verbreedt van jouw resultaat naar wat het betekent.

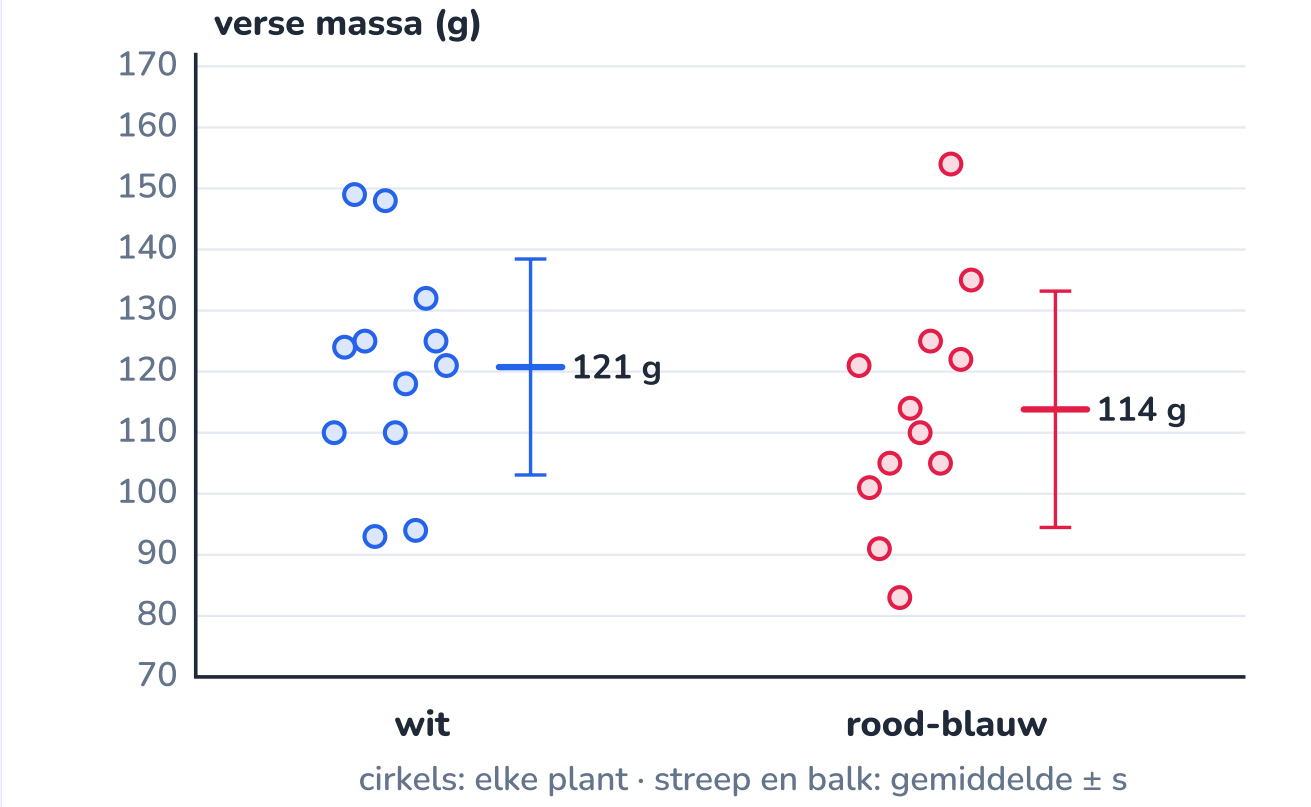
UITGEWERKT VOORBEELD · 2.11

Van ruwe gegevens naar resultaten en discussie

11/14

De slaproef is afgelopen. Na 28 dagen woog je (blind) de verse massa van elke plant. Verwerk de gegevens en schrijf de paragrafen *Resultaten* en *Discussie*.

plant	1	2	3	4	5	6	7	8	9	10	11	12
wit (g)	110	118	149	132	93	121	110	124	94	125	125	148
rood-blauw (g)	121	125	91	154	83	135	110	101	105	105	122	114



Figuur 1 van het verslag: de planten van beide groepen overlappen sterk; het verschil tussen de gemiddelden is klein tegenover de spreiding.

GEGEVEN

- 12 massa's per groep, in g, balans $\pm 0{,}1\,\text{g}$
- voorspelling: rood-blauw zwaarder
- vooraf berekend: met 12 planten per groep zie je verschillen vanaf ongeveer 21 g

GEVRAAGD

- gemiddelde en standaardafwijking per groep; een figuur
- een resultatenparagraaf en een discussieparagraaf

OPLOSSING

1. Verwerken. Wit: som 1449 g, $\bar{m} = 120{,}75\,\text{g}$, $s = 17{,}7\,\text{g}$. Rood-blauw: som 1366 g, $\bar{m} = 113{,}83\,\text{g}$, $s = 19{,}3\,\text{g}$. Verschil: $-6{,}9\,\text{g}$ (rood-blauw lichter).

2. Figuur: toon elke plant apart, met gemiddelde ± s per groep (zie figuur). Zo ziet de lezer de spreiding zelf, niet alleen twee staven.

3. **Resultaten** (voorbeeldtekst): 'Na 28 dagen was de verse massa onder wit licht $(121 \pm 18)\,\text{g}$ en onder rood-blauw licht $(114 \pm 19)\,\text{g}$ (gemiddelde ± standaardafwijking, $n = 12$ per groep, figuur 1). De massa's liepen uiteen van 83 tot 154 g: de spreiding binnen elke groep was veel groter dan het verschil tussen de gemiddelden. De temperatuur in de vakken verschilde hoogstens 0,6 °C.'

4. **Discussie** (voorbeeldtekst): 'De voorspelling kwam niet uit: de planten onder rood-blauw licht waren gemiddeld niet zwaarder, eerder 7 g lichter. Dat verschil is kleiner dan de ongeveer 21 g die deze proef betrouwbaar kan aantonen; we kunnen dus niet besluiten dat een van beide kleuren beter is. Een mogelijke verklaring is dat wit ledlicht ook veel rood en blauw bevat, en dat planten groen licht toch deels gebruiken. Bij evenveel bruikbaar licht lijkt de kleur hier weinig uit te maken. Om een verschil van 7 g aan te tonen, zijn volgens de vuistregel ongeveer 110 planten per groep nodig.'

ANTWOORD

De proef ondersteunt de hypothese niet. Het besluit is niet 'mislukt', maar: bij gelijke hoeveelheid licht is geen verschil aantoonbaar, en voor de klimaatmodule zijn duurdere rood-blauwe lampen op basis van deze proef niet te verantwoorden.

Klopt dit? Klopt de vuistregel? $16 \cdot (18{,}5/7)^2 \approx 112$ planten per groep, met $18{,}5\ \text{g}$ als gemiddelde spreiding van beide groepen. En de afronding: de fout op één of twee beduidende cijfers (18 en 19 g), de gemiddelden op dezelfde positie (121 en 114 g).

UITLEG · 2.12

Bronnen zoeken en beoordelen

12/14

Een inleiding en een discussie steunen op wat anderen al vonden. Maar niet elke bron is even betrouwbaar. Een **primaire bron** bevat de oorspronkelijke metingen van de auteurs zelf. Een artikel in een wetenschappelijk tijdschrift is voor publicatie meestal gecontroleerd door onafhankelijke vakgenoten: **peer review**. Dat is geen garantie (ook gecontroleerde artikels bevatten fouten), maar wel een eerste zeef.

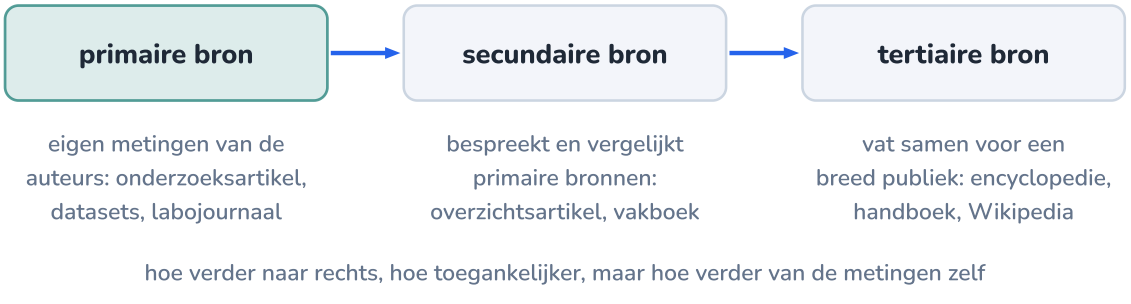
Zoeken. Begin breed in een handboek of encyclopedie om de juiste vaktermen te leren, en zoek daarmee verder in de bibliotheek, in Google Scholar of in databanken van tijdschriften. Zoek ook in het Engels: de meeste onderzoeksliteratuur is Engelstalig. Volg de bronnenlijst van een goede bron naar de oorspronkelijke artikels. Een Wikipedia-pagina is een prima startpunt, maar citeer liever de bronnen waarnaar ze verwijst.

Beoordelen. Stel bij elke bron vijf vragen. Een Amerikaanse universiteitsbibliotheek vatte ze samen als de CRAAP-test; in het Nederlands:

criterium	vraag die je stelt	alarmsignaal
Actueel	Van wanneer is het? Is er nieuwer onderzoek?	geen datum; verouderde cijfers bij een snel veranderend onderwerp
Relevant	Gaat het over mijn vraag, op mijn niveau?	ander gewas, ander klimaat, andere schaal
Auteur	Wie schreef het? Wat is zijn deskundigheid?	geen auteur of organisatie; geen contactgegevens
Accuraat	Staan er bronnen, methodes en data bij? Kan je het controleren?	stellige beweringen zonder bewijs; geen bronnen
Doel	Wil de bron informeren, overtuigen of verkopen?	de auteur verkoopt wat hij aanprijst

AI-chatbots zijn geen bron

Een taalmodel formuleert vlot een antwoord, maar kan feiten en zelfs bronnen verzinnen die echt lijken. Gebruik het hoogstens om zoektermen te vinden, en controleer elke bewering in een bron die je zelf gelezen hebt. Vermeld in je verslag als je een AI-hulpmiddel gebruikte en waarvoor.



Hoe dichter bij de metingen, hoe beter je kan controleren wat er echt gevonden werd.

NASLAG · 2.13

Bronnen vermelden in APA-stijl

13/14

Elke bron die je gebruikt, vermeld je twee keer: **kort in de tekst**, op de plaats waar je de informatie gebruikt, en **volledig in de bronnenlijst** achteraan, alfabetisch op de familienaam van de eerste auteur. De meest gebruikte stijl in wetenschap en hoger onderwijs is die van de American Psychological Association (APA, zevende editie). Hieronder de vormen die je het meest nodig hebt, met echte voorbeelden. Titels van boeken en tijdschriften staan cursief.

in de tekst	tussen haakjes	in de zin
één auteur	(Ohm, 1827)	Ohm (1827) toonde aan dat …
twee auteurs	(Watson & Crick, 1953)	Watson en Crick (1953) stelden voor dat …
drie of meer	(Feynman et al., 1963)	Feynman et al. (1963) leggen uit dat …
organisatie	(Bureau International des Poids et Mesures [BIPM], 2019), daarna (BIPM, 2019)	Het BIPM (2019) definieert …
letterlijk citaat	met aanhalingstekens en bladzijde: (Watson & Crick, 1953, p. 737)	

soort bron	vorm en voorbeeld in de bronnenlijst
artikel in een tijdschrift	Auteur, A. A., & Auteur, B. B. (jaar). Titel van het artikel. *Naam van het Tijdschrift, volume*(nummer), bladzijden. https://doi.org/… Watson, J. D., & Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature, 171*(4356), 737–738. https://doi.org/10.1038/171737a0
boek	Auteur, A. A. (jaar). *Titel van het boek* (editie of deel). Uitgever. Feynman, R. P., Leighton, R. B., & Sands, M. (1963). *The Feynman lectures on physics* (Vol. 1). Addison-Wesley.
rapport van een organisatie	Organisatie. (jaar). *Titel* (editie). Uitgever, als die anders is dan de auteur. Bureau International des Poids et Mesures. (2019). *The International System of Units (SI)* (9th ed.).
webpagina	Auteur of organisatie. (jaar, dag maand). *Titel van de pagina*. Naam van de website. URL Geen datum: (z.d.). Verandert de pagina voortdurend, voeg dan toe: Geraadpleegd op dag maand jaar, van URL.
AI-hulpmiddel	Maker. (jaar). *Naam van het model* (versie) [Groot taalmodel]. URL Alleen als je beschrijft hoe je het gebruikte; feiten haal je uit een gecontroleerde bron.

controlelijst voor je verslag
Staat elke bron uit de tekst in de lijst, en elke bron uit de lijst in de tekst?
Zijn parafrases in je eigen woorden, en staan letterlijke citaten tussen aanhalingstekens met bladzijde?
Heeft elke figuur en tabel een nummer, een titel en een verwijzing in de tekst?
Heeft elke as en kolom een grootheid en een eenheid, elk resultaat een fout (hoofdstuk 1)?
Kan een klasgenoot je proef overdoen met alleen je methode?

HET GEHEEL · 2.14

Waar proefontwerp terugkomt

14/14

Elk onderzoek in de rest van StemExpert gebruikt wat je hier leerde: de vraag scherp stellen, eerlijk vergelijken, genoeg herhalen, alles noteren en rapporteren. In Expert krijgt het een statistische onderbouwing: dan reken je uit hoe zeker een verschil is, in plaats van het te schatten met een vuistregel.

F2 **Onderzoeken: van vraag tot besluit**

I1 **Meetfouten en nauwkeurigheid**

I3 **Verbanden in data**

De onderzoekscyclus en de eerlijke proef: hier aangevuld met controlegroep, loting en blind meten.

Gemiddelde en standaardafwijking: de spreiding bepaalt hoeveel herhalingen je nodig hebt.

Na het meten volgt het zoeken naar een verband: een rechte, een omgekeerd evenredig verband.

I34 Enzymen en stofwisseling

Een enzymproef opzetten is een schoolvoorbeeld van één variabele veranderen met een controle.

I39 Data visualiseren en eerste stappen in AI

Correlatie is geen oorzakelijk verband: waarom alleen een gecontroleerde proef oorzaak en gevolg aantoont.

I40 Integratieproject: de klimaatmodule

Het technisch rapport en de verdediging van het integratieproject volgen de opbouw van dit hoofdstuk.

E6 Statistiek en meetonzekerheid

De t-toets en de p-waarde: hoe zeker is een verschil tussen twee groepen, en waar komt de 16 in de vuistregel vandaan?

E48 Technisch communiceren en projectmanagement

Het technisch rapport, planning en teamwerk op het niveau van ingenieurs.

Habitat

De klimaatmodule wordt een systeem van tientallen keuzes: lampkleur, voeding, ventilatie, isolatie. Elke keuze die je met een proef onderbouwt, moet een eerlijke proef zijn. In opdracht 2.14 schrijf je het onderzoeksplan voor één zo'n keuze, met vraag, hypothese, controlegroep, herhalingen, loting, blind meten en bronnen.

SAMENVATTING

In het kort

- Een onderzoeksvraag is specifiek, meetbaar en haalbaar; **operationaliseren** maakt een vaag begrip meetbaar.
- Een **hypothese** is een toetsbare verklaring; een **voorspelling** zegt wat je zal meten als ze klopt ('als ..., dan ...'). Een proef ondersteunt of weerlegt, maar bewijst niet.
- Een **controlegroep** geeft een vergelijkingspunt; **loten** verdeelt onbekende verschillen eerlijk; een **verstorende variabele** die samenvalt met je behandeling maakt een besluit onmogelijk.
- Het aantal herhalingen volgt uit spreiding en gezocht verschil: $n \approx 16\,(s/\delta)^2$ per groep. Tel alleen onafhankelijke herhalingen (geen pseudoreplicatie).
- **Blind meten** met codes en vooraf vastgelegde definities beschermt tegen je eigen verwachtingen.
- Het **labojournaal**: gedateerd, in pen, ruwe gegevens met eenheden, correcties doorhalen met reden, afwijkingen noteren.
- Het verslag volgt inleiding, methode, resultaten en discussie; bronnen vermeld je in de tekst en in een lijst (APA), na ze beoordeeld te hebben.

BEGRIPPEN

Wat je nu kent

hypothese

Een toetsbare verklaring voor een verschijnsel: wat je denkt dat er gebeurt, en waarom.

voorspelling

Wat je concreet zal meten als de hypothese klopt, geformuleerd als 'als ..., dan ...'.

operationele definitie

Een precieze afspraak over hoe je een grootheid meet, zodat iedereen hetzelfde meet (bv. 'gekiemd = worteltje van minstens 2

controlegroep

De groep die de gewone behandeling of geen behandeling krijgt, als vergelijkingspunt voor de proefgroep.

mm').

verstorende variabele

Een variabele die samen met de onafhankelijke variabele verandert en zo het resultaat kan verklaren.

loten (randomiseren)

Het toeval laten beslissen welke proefeenheid welke behandeling krijgt, zodat onbekende verschillen zich eerlijk verdelen.

blind meten

Meten zonder te weten welke behandeling een proefeenheid kreeg; dubbelblind als ook de proefpersoon het niet weet.

steekproefgrootte

Het aantal onafhankelijke herhalingen per groep; hoe kleiner het verschil dat je zoekt tegenover de spreiding, hoe meer je er nodig hebt.

pseudoreplicatie

Herhalingen tellen die niet onafhankelijk zijn, zoals twaalf planten onder één lamp als twaalf herhalingen van 'die lamp'.

labojournaal

Het doorlopende, gedateerde verslag van alles wat je in een onderzoek deed, mat en vaststelde.

peer review

Controle van een wetenschappelijk artikel door onafhankelijke vakgenoten voor het gepubliceerd wordt.

primaire bron

Een bron met de oorspronkelijke metingen of waarnemingen van de auteurs zelf.

plagiaat

Tekst, ideeën, figuren of data van iemand anders gebruiken zonder de bron te vermelden.

I INTERMEDIATE · DEEL I1 · ONDERZOEKEN MET DATA

HANDBOEK · HOOFDSTUK 3

Verbanden in data

De best passende rechte, de richtingscoëfficiënt als grootheid, en krommen die je rechttrekt

Metingen worden pas kennis als je er een verband in ziet. In dit hoofdstuk leer je door een wolk van meetpunten de best passende rechte te trekken, op het oog en met een rekenblad, en je leest uit de helling een echte grootheid af: een dichtheid, een opwarmsnelheid, een veerconstante. Je herkent omgekeerd evenredige verbanden, trekt krommen recht om er een rechte van te maken, en kalibreert een sensor van de meetkas met een rechte, in een rekenblad en in een paar regels Python.

7× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 8 lestijden

O Onderzoeken en meten · W Wiskundig modelleren · A Data en AI

IJswater, kraantjeswater, warm water: drie referentiepunten, één rechte, en de sensor spreekt de waarheid.

Na dit hoofdstuk

- Je kan de richtingscoëfficiënt en het snijpunt van een rechte bepalen en ze interpreteren als grootheden met een eenheid.
- Je kan door meetpunten de best passende rechte tekenen en er met de steilste en de minst steile rechte een fout op de helling bij schatten.
- Je kan uitleggen hoe de methode van de kleinste kwadraten een rechte kiest, en a en b met de formules berekenen.
- Je kan residuen en R^2 gebruiken om te beoordelen of een lineair model past.
- Je kan een omgekeerd evenredig verband herkennen aan een tabel, een formule en een grafiek.
- Je kan een kromme rechttrekken door de juiste grootheden op de assen te zetten, en uit de rechte een grootheid bepalen.
- Je kan in een rekenblad formules met relatieve en absolute celverwijzingen maken, een spreidingsdiagram tekenen en een trendlijn laten berekenen.
- Je kan een sensor kalibreren met een rechte en de kalibratieformule omvormen tot een bruikbare omrekening.

UITLEG · 3.1

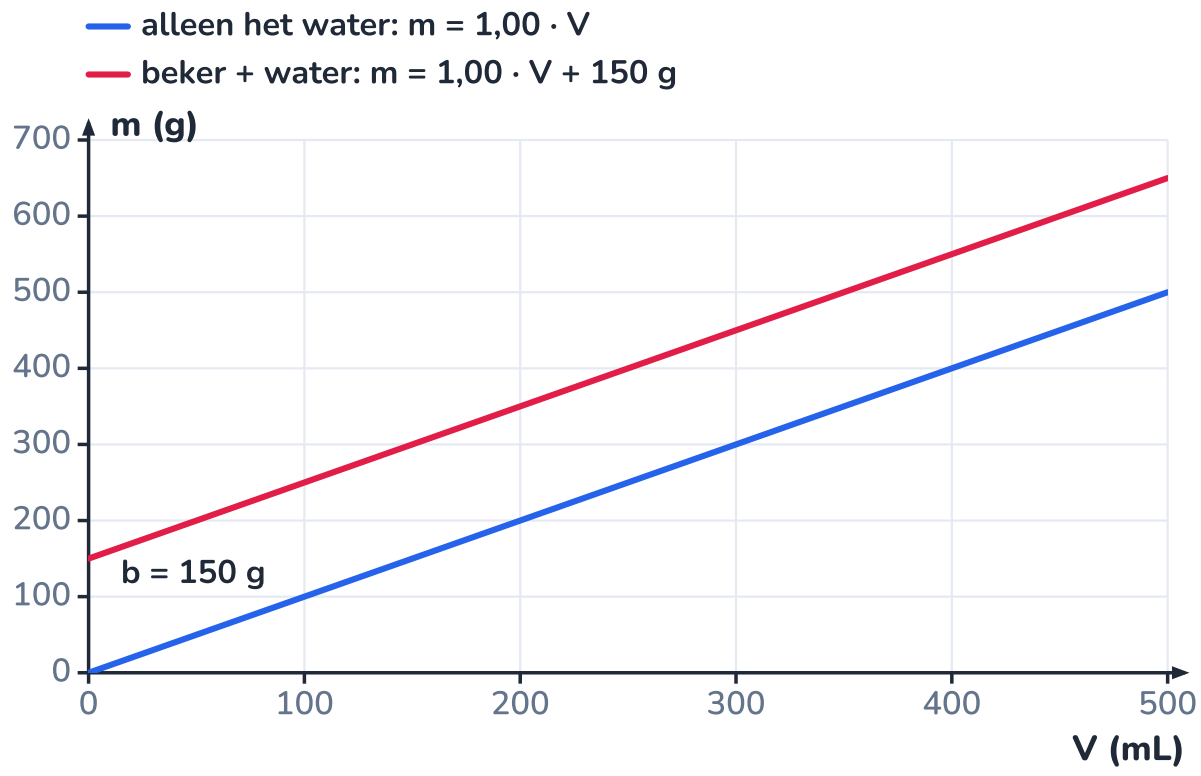
Wat je al weet, en wat nu komt

1/18

In Fundamental (hoofdstuk F4) zette je meetresultaten in een tabel met grootheden en eenheden in de kop, en tekende je een grafiek met gekozen schaal en een trendlijn. Je leerde het **recht evenredig verband** herkennen: de punten liggen op een rechte door de oorsprong, en de verhouding y/x is constant. Die constante, de evenredigheidsconstante k in $y = k \cdot x$, had een betekenis: in hoofdstuk F19 was ze de weerstand. En je wist dat extrapoleren, voorbij je metingen voorspellen, gevaarlijk kan zijn.

Maar veel verbanden gaan niet door de oorsprong. Weeg je een bekerglas met water, dan meet je ook de beker: de rechte schuift 150 g omhoog. Dat is een **lineair verband** $y = a \cdot x + b$. In dit hoofdstuk leer je

- de **richtingscoëfficiënt** a en het **snijpunt** b bepalen en er een fysische betekenis aan geven, met eenheid;
- de **best passende rechte** door punten met meetfouten trekken, op het oog en met de methode van de kleinste kwadraten;
- het **omgekeerd evenredig verband** $y = k/x$ herkennen;
- krommen **rechttrekken** (lineariseren), zodat je ook daar een rechte kan gebruiken;
- dat alles in een **rekenblad** en, voor wie wil, in een paar regels **Python**.



Dezelfde helling (1,00 g per mL), maar de rechte met de beker begint 150 g hoger: lineair, niet recht evenredig.

VERHAAL · 3.2

Een helling met de leeftijd van het heelal erin

2/18

In 1927 publiceerde de Leuvense priester en natuurkundige **Georges Lemaître** een artikel in de *Annales de la Société scientifique de Bruxelles*. Hij toonde dat de vergelijkingen van Einstein een uitdijend heelal

toelaten, en dat verre sterrenstelsels dan van ons weg bewegen met een snelheid die evenredig is met hun afstand. Het artikel verscheen in het Frans, in een weinig gelezen tijdschrift, en kreeg eerst weinig aandacht.

Twee jaar later zette de Amerikaanse sterrenkundige **Edwin Hubble** de snelheid van een reeks sterrenstelsels uit tegen hun afstand. De punten lagen ruwweg op een rechte door de oorsprong: $v = H_0 \cdot d$. De richtingscoëfficiënt H_0 kreeg zijn naam. Hubble vond ongeveer 500 km/s per megaparsec (een megaparsec is ongeveer 3,26 miljoen lichtjaar).

Kijk naar de eenheid: kilometer per seconde, gedeeld door een afstand. Dat is 'per seconde', en het omgekeerde $1/H_0$ is dus een **tijd**: ruwweg hoe lang het heelal al uitdijt. Met Hubbles helling kwam dat uit op ongeveer 2 miljard jaar, jonger dan de ouderdom die geologen met radioactiviteit aan gesteenten op aarde toekenden. Die tegenspraak bleef jaren bestaan, tot bleek dat Hubbles afstanden **systematisch** te klein waren (hoofdstuk 1): de 'standaardsterren' waarmee hij afstanden schatte, waren verkeerd geijkt.

Vandaag meten sterrenkundigen een helling van ongeveer 70 km/s per megaparsec. Dat geeft $1/H_0 \approx 14$ miljard jaar, in dezelfde orde als de leeftijd van het heelal volgens de beste metingen: 13,8 miljard jaar. In 2018 raadde de Internationale Astronomische Unie aan om voortaan van de wet van **Hubble-Lemaître** te spreken.

> **Wat je uit dit verhaal meeneemt**
>
> Een richtingscoëfficiënt is geen kaal getal: ze heeft een eenheid en een betekenis, soms een verrassende. En een helling is maar zo goed als de metingen waarop ze steunt: een systematische fout op de afstanden gaf een heelal dat zeven keer te jong leek.

Uit zulke fotoplaten haalden sterrenkundigen snelheden en afstanden van sterrenstelsels, en daaruit één helling.

UITLEG · 3.3

De rechte y = a · x + b en wat a en b betekenen

Een zwart watervat van 20 liter staat in de zon in de kas. Om het halfuur meet je de temperatuur (grafiek). De punten liggen mooi op een rechte, niet door de oorsprong:

LINEAIR VERBAND

$$y = a \cdot x + b \qquad a = \frac{\Delta y}{\Delta x}$$

a is de **richtingscoëfficiënt** (helling), b het **snijpunt met de y-as**, de waarde van y bij $x = 0$. Is $b = 0$, dan is het verband recht evenredig.

De richtingscoëfficiënt bepalen. Kies twee punten *op de rechte* (niet twee meetpunten), zo ver mogelijk uit elkaar, want dan weegt een afleesfout het minst. Op de rechte lees je $(1{,}0\,\text{h};\ 15{,}7\,^\circ\text{C})$ en $(4{,}0\,\text{h};\ 25{,}3\,^\circ\text{C})$:

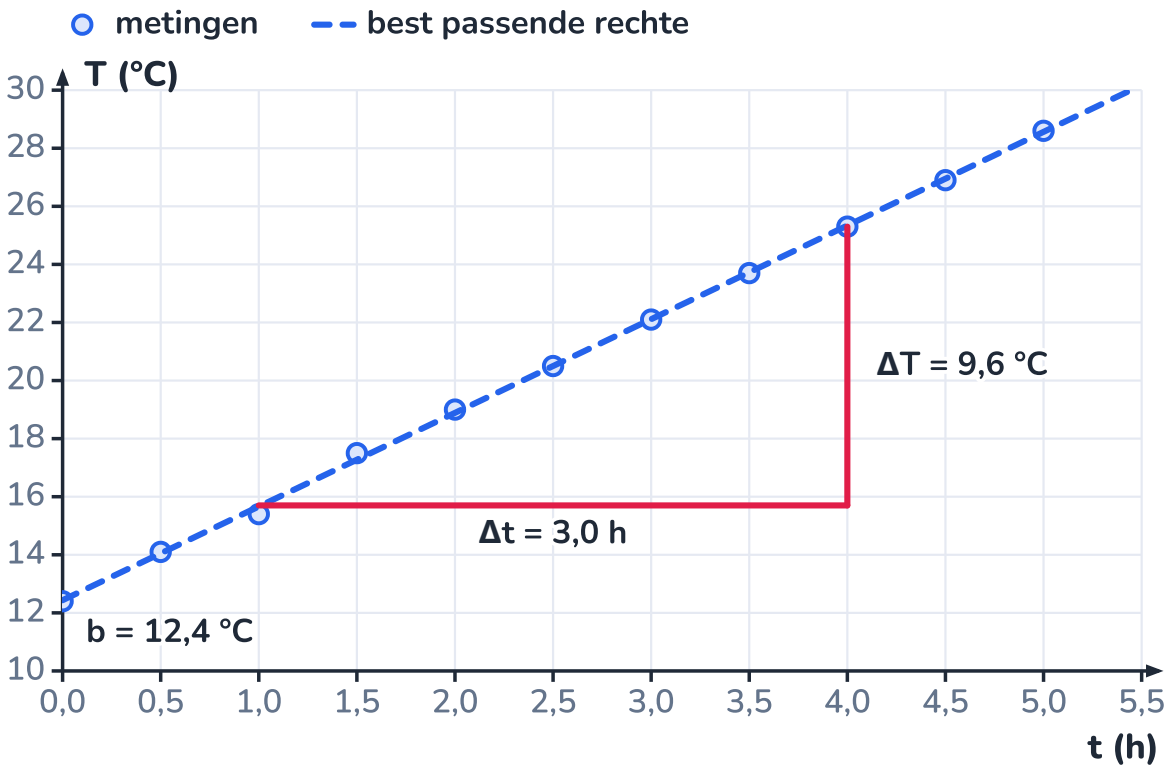
$$a = \frac{25{,}3\,^\circ\text{C} - 15{,}7\,^\circ\text{C}}{4{,}0\,\text{h} - 1{,}0\,\text{h}} = \frac{9{,}6\,^\circ\text{C}}{3{,}0\,\text{h}} = 3{,}2\,^\circ\text{C/h}$$

De eenheid van a is de eenheid van y gedeeld door die van x. Hier is a de **opwarmsnelheid**: het vat wordt 3,2 °C warmer per uur. Het snijpunt $b = 12{,}4\,^\circ\text{C}$ is de **begintemperatuur**. Het model: $T = 3{,}2\,^\circ\text{C/h} \cdot t + 12{,}4\,^\circ\text{C}$.

Een model heeft een geldigheidsgebied. Binnen je metingen (0 tot 5 h) mag je interpoleren: na 2 h 15 min is het vat ongeveer 19,6 °C. Extrapoleren naar 12 h geeft 51 °C: onzin, want 's middags verliest het vat steeds meer warmte en 's avonds gaat de zon onder.

Een richtingscoëfficiënt is gewoon een getal

3,2 °C per uur en 3,2 °C per minuut schelen een factor 60. Zonder eenheid is een helling betekenisloos. Een negatieve helling is net zo betekenisvol: een waterstand die daalt met 0,8 cm per dag heeft $a = -0{,}8\,\text{cm/dag}$.



De hellingsdriehoek staat op de rechte, niet op twee meetpunten. Hoe groter de driehoek, hoe nauwkeuriger je afleest.

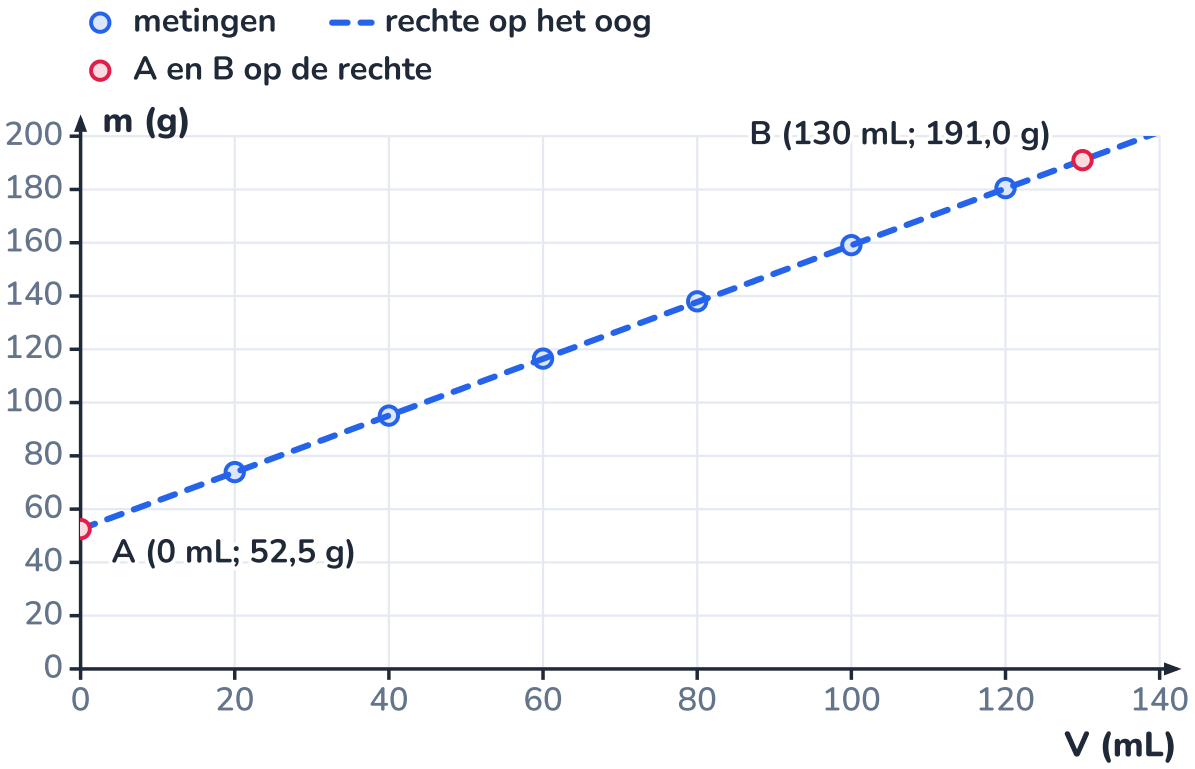
UITGEWERKT VOORBEELD · 3.4

Een dichtheid uit de helling

4/18

Hoe dicht is de geconcentreerde stamoplossing van de hydroponievoeding? Je zet een leeg bekerglas op een balans, giet er telkens 20 mL oplossing bij (gemeten met een maatcilinder) en noteert de massa van beker plus oplossing.

V (mL)	20	40	60	80	100	120
m (g)	73,9	95,1	116,5	138,0	159,1	180,5



De helling van de rechte is de dichtheid, het snijpunt de massa van de lege beker.

GEGEVEN

- zes metingen van V (mL) en m (g)
- balans $\pm 0{,}1\ \text{g}$, maatcilinder $\pm 0{,}5\ \text{mL}$

GEVRAAGD

- de dichtheid ρ van de oplossing
- de betekenis van het snijpunt

OPLOSSING

1. Het model: de totale massa is de massa van de oplossing plus die van de beker, $m = \rho \cdot V + m_{\text{beker}}$. Dat is $y = a \cdot x + b$ met $a = \rho$ en $b = m_{\text{beker}}$.
2. Zet de punten in een grafiek met V horizontaal en m verticaal, en trek op het oog de rechte die zo goed mogelijk tussen alle punten loopt (zie grafiek).
3. Kies twee punten op de rechte, ver uit elkaar: A $(0\ \text{mL};\ 52{,}5\ \text{g})$ en B $(130\ \text{mL};\ 191{,}0\ \text{g})$.
4. $a = \dfrac{191{,}0\ \text{g} - 52{,}5\ \text{g}}{130\ \text{mL} - 0\ \text{mL}} = \dfrac{138{,}5\ \text{g}}{130\ \text{mL}} = 1{,}065\ \text{g/mL}$.
5. Het snijpunt $b = 52{,}5\ \text{g}$ is de massa van het lege bekerglas.

ANTWOORD

$\rho \approx 1{,}07\ \text{g/mL} = 1{,}07\ \text{g/cm}^3$ **(op het oog, tot op ongeveer 1 %). Het bekerglas weegt ongeveer 52,5 g.**

Klopt dit? Water heeft een dichtheid van $0{,}998\ \text{g/mL}$ bij 20 °C, zeewater ongeveer $1{,}025\ \text{g/mL}$: een geconcentreerde zoutoplossing mag zwaarder zijn. En het lege bekerglas wegen geeft 52,6 g: het snijpunt klopt. Merk op: je hoefde geen enkele meting door haar volume te delen. De rechte haalt de beker er vanzelf uit.

UITLEG · 3.5

De best passende rechte op het oog, en haar onzekerheid

Meetpunten liggen nooit precies op een rechte: elke meting heeft een fout (hoofdstuk 1). De **best passende rechte** loopt zo dicht mogelijk langs alle punten tegelijk. Op het oog trek je ze met vijf regels:

1. Verbind de punten niet: een zigzaglijn beschrijft de meetfouten, niet het verband.
2. Leg de lat zo dat er ongeveer evenveel punten boven als onder de rechte liggen, verspreid over het hele bereik (niet alle bovenste links en alle onderste rechts).
3. Laat de rechte door het **zwaartepunt** $(\bar{x}, \bar{y})$ van de punten gaan: de best passende rechte gaat daar altijd door.
4. Dwing de rechte niet door de oorsprong, tenzij de natuurkunde dat eist én de punten het toelaten. Een snijpunt dat niet nul is, kan juist informatie zijn (de beker, de rustlengte van een veer).
5. Negeer een uitschieter alleen met een reden (hoofdstuk 1), en vermeld het.

Hoe zeker is de helling? Teken de foutbalken van de metingen. Trek dan de **steilste** en de **minst steile** rechte die nog door alle foutbalken gaan. Het verschil tussen hun hellingen is een maat voor de onzekerheid:

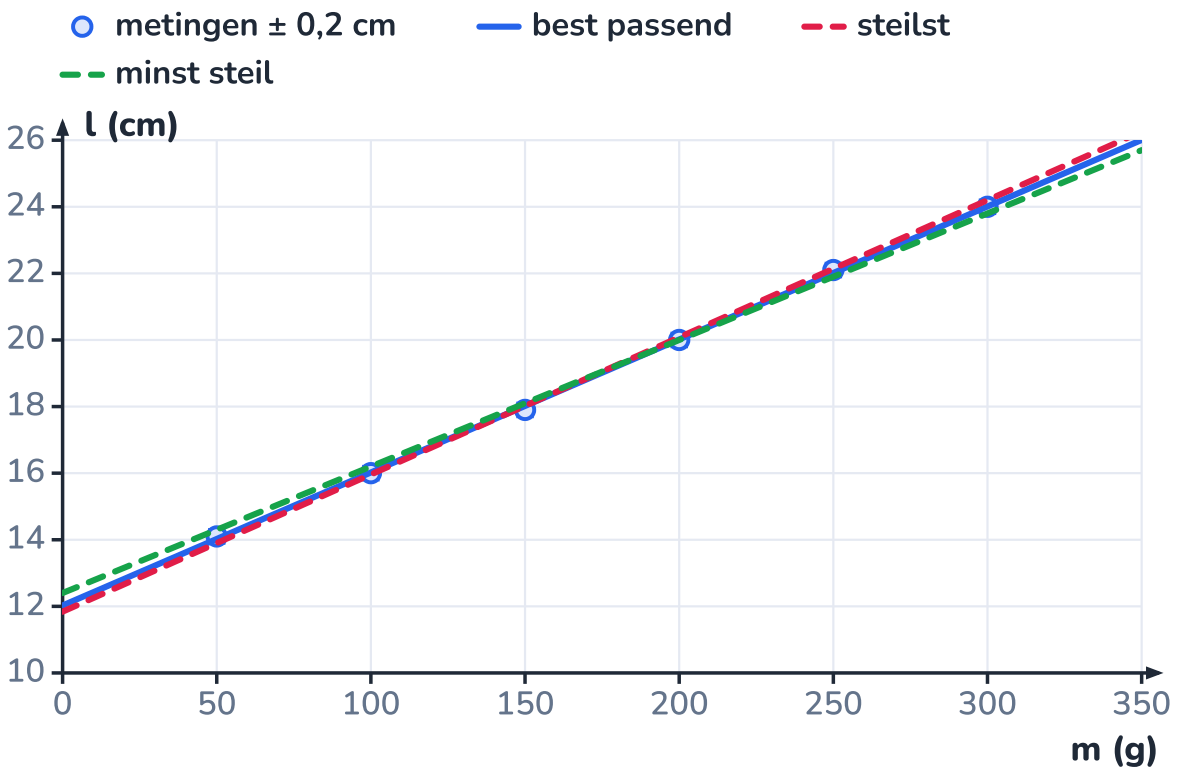
FOUT OP DE HELLING

$$\Delta a \approx \frac{a_{\text{steilst}} - a_{\text{minst steil}}}{2}$$

Voor de veer: de best passende rechte heeft $a = 0{,}0399\,\text{cm/g}$. De steilste rechte gaat door (50 g; 13,9 cm) en (300 g; 24,2 cm): $a = 10{,}3/250 = 0{,}0412\,\text{cm/g}$. De minst steile door (50 g; 14,3 cm) en (300 g; 23,8 cm): $a = 9{,}5/250 = 0{,}0380\,\text{cm/g}$. Dus $a = (0{,}0399 \pm 0{,}0016)\,\text{cm/g}$: de veer rekt 4,0 cm uit per 100 g, met een fout van 4 %. Het snijpunt, 12,0 cm, is de lengte van de veer zonder massa.

Van helling naar veerconstante

Per kilogram rekt de veer $0{,}0399\,\text{cm/g} = 0{,}399\,\text{m/kg}$ uit. Een kilogram weegt 9,81 N, dus de veerconstante is $k = 9{,}81\,\text{N/kg} / 0{,}399\,\text{m/kg} = 24{,}6\,\text{N/m}$, met dezelfde relatieve fout van 4 %: $k = (24{,}6 \pm 1{,}0)\,\text{N/m}$ (wet van Hooke, F11).



De steilste en de minst steile rechte gaan net door alle foutbalken; de best passende rechte ligt ertussen.

UITLEG · 3.6

Hoe een rekenblad de rechte kiest: de kleinste kwadraten

6/18

Op het oog trekt iedereen een net iets andere rechte. Een rekenblad gebruikt een vaste regel. Voor elk meetpunt bepaalt het het **residu**: hoeveel het punt boven (+) of onder (–) de rechte ligt, gemeten in de y-richting. De **methode van de kleinste kwadraten** kiest de rechte waarvoor de som van de kwadraten van alle residuen zo klein mogelijk is.

Waarom kwadraten? Plus en min heffen elkaar dan niet op, grote afwijkingen wegen zwaarder dan kleine, en er is precies één oplossing, met eenvoudige formules:

KLEINSTE-KWADRATENRECHTE

$$a = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \qquad b = \bar{y} - a \cdot \bar{x}$$

Uit de formule voor b volgt dat de rechte altijd door het zwaartepunt $(\bar{x}, \bar{y})$ gaat.

Hoe goed past de rechte? Een rekenblad toont vaak R^2 (de determinatiecoëfficiënt): het deel van de spreiding van de y-waarden dat de rechte verklaart. $R^2 = 1$ betekent dat alle punten op de rechte liggen, $R^2 = 0$ dat de rechte niets verklaart:

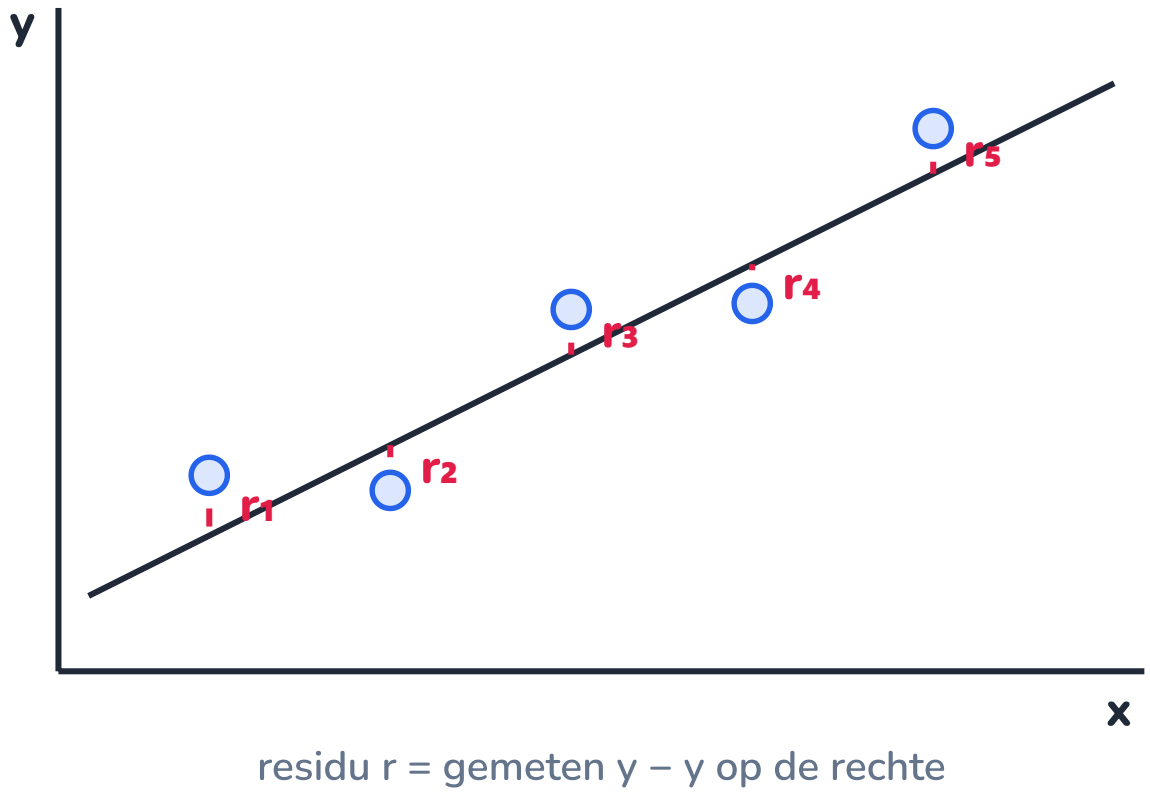
DETERMINATIECOËFFICIËNT

$$R^2 = 1 - \frac{\sum(\text{residu})^2}{\sum(y_i - \bar{y})^2}$$

Beter dan R^2 is kijken naar de **residuen** zelf. Zet ze uit tegen x. Liggen ze willekeurig boven en onder nul, dan beschrijft de rechte het verband en zijn de afwijkingen gewone meetfouten. Vormen ze een patroon, een boog of een trend, dan is het model niet goed.

R^2 bijna 1, dus het verband is lineair

Een stopafstand groeit met het kwadraat van de snelheid, en toch geeft een rechte door tien metingen tussen 30 en 120 km/h nog $R^2 = 0{,}987$. De residuen verraden het: eerst positief, dan negatief, dan weer positief, een boog (opdracht 3.14). R^2 zegt hoe dicht de punten bij de rechte liggen, niet of een rechte het juiste model is.



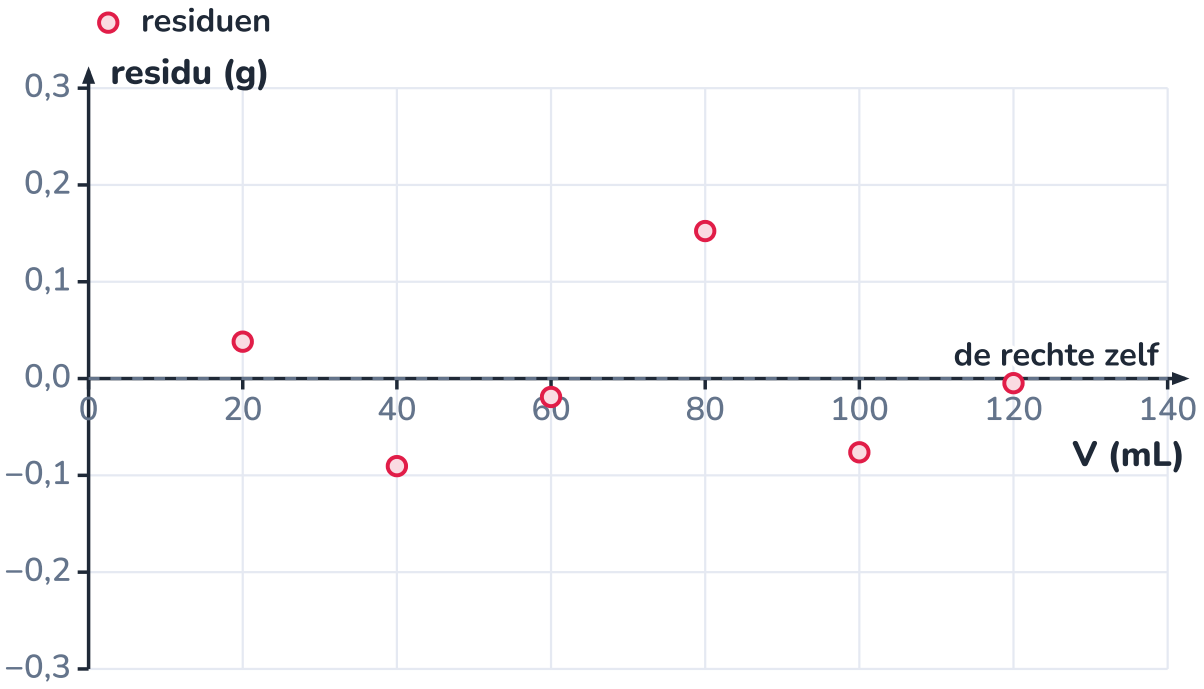
Schematisch: de residuen zijn de verticale afstanden tot de rechte. De kleinste-kwadratenrechte maakt $r_1^2 + r_2^2 + \ldots$ zo klein mogelijk.

UITGEWERKT VOORBEELD · 3.7

De kleinste kwadraten met de hand

7/18

Bereken de kleinste-kwadratenrechte door de metingen van de stamoplossing (vorig voorbeeld). Hoe goed was de rechte op het oog?



De residuen liggen zonder patroon rond nul en zijn allemaal kleiner dan 0,2 g: een rechte is hier het juiste model.

GEGEVEN

- zes punten (V, m) uit het voorbeeld 'Een dichtheid uit de helling'

GEVRAAGD

- a en b volgens de kleinste kwadraten
- R^2 en de residuen

OPLOSSING

1 Gemiddelden: $\bar{V} = 420/6 = 70\,\text{mL}$ en $\bar{m} = 763{,}1/6 = 127{,}183\,\text{g}$.

2 Maak een tabel met de afwijkingen, hun producten en de kwadraten van de x-afwijkingen.

V (ML)	M (G)	$V-\bar{V}$	$M-\bar{M}$	$(V-\bar{V})(M-\bar{M})$	$(V-\bar{V})^2$
20	73,9	−50	−53,283	2664,17	2500
40	95,1	−30	−32,083	962,50	900
60	116,5	−10	−10,683	106,83	100
80	138,0	10	10,817	108,17	100
100	159,1	30	31,917	957,50	900
120	180,5	50	53,317	2665,83	2500
gem. 70	**gem. 127,183**	0	0	**7465,00**	**7000**

3 $a = \dfrac{7465{,}00\,\text{g}\cdot\text{mL}}{7000\,\text{mL}^2} = 1{,}06643\,\text{g/mL}$.

4 $b = \bar{m} - a\cdot\bar{V} = 127{,}183 - 1{,}06643\times 70 = 52{,}53\,\text{g}$.

5 Residuen, gemeten min berekend: $+0{,}04$; $-0{,}09$; $-0{,}02$; $+0{,}15$; $-0{,}08$; $-0{,}01\,\text{g}$. Ze liggen willekeurig rond nul en zijn kleiner dan wat de maatcilinder toelaat ($0{,}5\,\text{mL}$ komt overeen met ongeveer $0{,}5\,\text{g}$). $R^2 = 0{,}999995$.

ANTWOORD

$m = 1{,}066\,\mathrm{g/mL} \cdot V + 52{,}5\,\mathrm{g}$, **dus** $\rho = 1{,}066\,\mathrm{g/mL}$. **De rechte op het oog gaf** $1{,}065\,\mathrm{g/mL}$**: een verschil van 0,1 %, veel kleiner dan de meetfouten.**

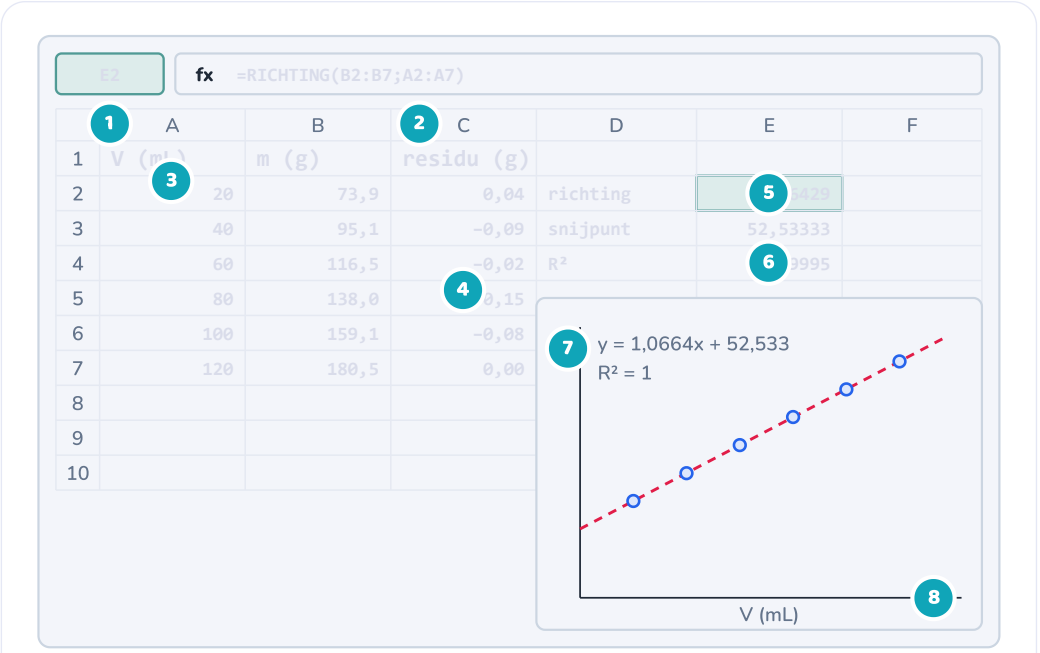
Klopt dit? Controleer met een rekenblad: `=RICHTING(B2:B7;A2:A7)` geeft 1,066429 en `=SNIJPUNT(B2:B7;A2:A7)` 52,53333. Hoeveel van die cijfers beduidend zijn, beslis jij (hoofdstuk 1): de spreiding van de punten laat hoogstens vier beduidende cijfers voor a toe.

ONTLEED · 3.8

Een rekenblad ontleed

8/18

Een rekenblad (Excel, LibreOffice Calc, Google Spreadsheets) doet het rekenwerk van het vorige voorbeeld in een oogwenk. Maar het rekent alleen wat je het vraagt, en het toont meer cijfers dan je meting verdient. Dit is het blad van de stamoplossing.



Gegevens in kolommen, berekeningen in cellen, en een spreidingsdiagram met trendlijn dat over de cellen zweeft.

1 Naamvak
Toont welke cel geselecteerd is: kolom E, rij 2. Elke cel heeft zo'n adres, en formules verwijzen naar adressen, niet naar getallen.

2 Formulebalk
Hier staat wat de cel *berekent*. In de cel zelf zie je alleen het resultaat. Een formule begint altijd met =.

3 Kop met eenheid
Rij 1 bevat de grootheid met de eenheid, net als in een tabel op papier. In de cellen eronder staan alleen getallen, anders kan het rekenblad er niet mee rekenen.

4 Een kolom met een formule
C2 bevat `=B2-($E$2*A2+$E$3)`: het residu. Naar beneden gekopieerd wordt B2 telkens B3, B4 … (relatieve verwijzing), maar `$E$2` en `$E$3` blijven staan (absolute verwijzing, met dollartekens).

5 Richting en snijpunt
RICHTING en SNIJPUNT rekenen de kleinste-kwadratenrechte uit met de formules van het vorige blad. Eerst het bereik van de y-waarden, dan dat van de x-waarden.

6 Zes cijfers, niet zes beduidende
0,999995 en 1,066429: het rekenblad toont zoveel cijfers als de kolom toelaat. Rond zelf af volgens de fout (hoofdstuk 1).

7 Trendlijn met vergelijking
Het rekenblad schrijft x en y, niet V en m. Vertaal naar je eigen grootheden, met eenheden. 'R² = 1' is afgerond: de echte waarde is 0,999995.

8 Spreidingsdiagram
Kies altijd een spreidingsdiagram (xy-diagram): dan staat elk punt op zijn echte x-waarde. Een lijndiagram zet de punten op gelijke afstanden, ook als je metingen dat niet zijn.

UITLEG · 3.9

Werken met een rekenblad

9/18

Een stappenplan dat voor elke meetreeks werkt:

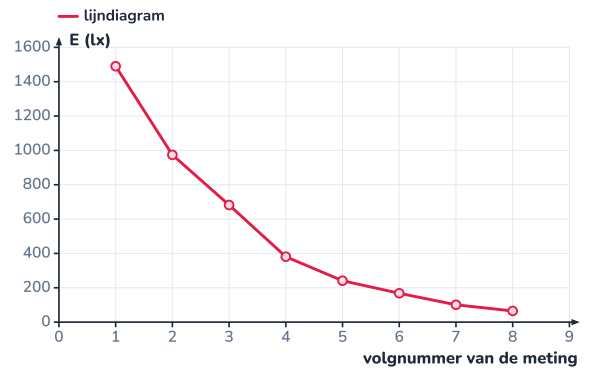
1. **Invoeren.** Eén grootheid per kolom, in rij 1 de naam met de eenheid, daaronder alleen getallen. Let op het decimaalteken: een Nederlandstalig rekenblad verwacht een komma.
2. **Berekenen.** Een nieuwe kolom met een formule, bijvoorbeeld `=1/A2` of `=B2^2`, en die naar beneden doortrekken. Moet een formule altijd naar dezelfde cel verwijzen (een constante, een helling), zet dan dollartekens: `$E$2`. In Excel wissel je met F4 tussen relatief en absoluut.
3. **Grafiek.** Selecteer de twee kolommen en kies een **spreidingsdiagram** (xy-diagram) met alleen punten. Geef de assen een titel met grootheid en eenheid.

4. **Trendlijn.** Voeg een lineaire trendlijn toe en laat de vergelijking en R^2 tonen. Vraagt de natuurkunde een rechte door de oorsprong, dan kan je het snijpunt op nul vastzetten; doe dat alleen met een reden.
5. **Controleren.** Bereken met RICHTING en SNIJPUNT dezelfde getallen in cellen, voeg een kolom met residuen toe, en rond af volgens de fout.

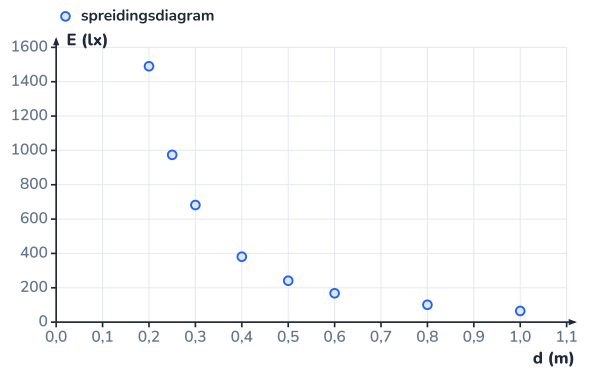
wat	Nederlandstalig rekenblad	Engelstalig rekenblad
richtingscoëfficiënt	`=RICHTING(B2:B7;A2:A7)`	`=SLOPE(B2:B7,A2:A7)`
snijpunt met de y-as	`=SNIJPUNT(B2:B7;A2:A7)`	`=INTERCEPT(B2:B7,A2:A7)`
R^2	`=R.KWADRAAT(B2:B7;A2:A7)`	`=RSQ(B2:B7,A2:A7)`
gemiddelde	`=GEMIDDELDE(A2:A7)`	`=AVERAGE(A2:A7)`
kwadraat, omgekeerde, wortel	`=A2^2` · `=1/A2` · `=WORTEL(A2)`	`=A2^2` · `=1/A2` · `=SQRT(A2)`

Let op de volgorde: eerst de *y*-waarden, dan de *x*-waarden. In een Nederlandstalig rekenblad scheidt een puntkomma de argumenten, in een Engelstalig een komma.

> ⚠ **Lijndiagram of spreidingsdiagram**
>
> Een **lijndiagram** behandelt de x-waarden als namen en zet ze op gelijke afstanden. Voor metingen op ongelijke afstanden (0,20 m, 0,25 m, 0,30 m, 0,40 m … 1,00 m) vervormt het de kromme volledig. Gebruik voor meetreeksen altijd een **spreidingsdiagram**.



Fout: in een lijndiagram staan de metingen op gelijke afstanden, de x-as is een volgnummer.



Juist: in een spreidingsdiagram staat elk punt op zijn echte afstand.

UITLEG · 3.10

Omgekeerd evenredig

10/18

Hoe sterker de pomp, hoe sneller het reservoir van de kas vol is. Meet je de vultijd t bij verschillende debieten Q, dan krijg je een dalende kromme (grafiek). Verdubbel je het debiet van 3 naar 6 L/min, dan halveert de vultijd van ongeveer 20 naar 10 minuten. Dat is een **omgekeerd evenredig verband**:

> **OMGEKEERD EVENREDIG**
>
> $$y = \frac{k}{x} \quad \Leftrightarrow \quad x \cdot y = k$$
>
> Het product van beide grootheden is constant. De grafiek is een **hyperbool**: ze komt steeds dichter bij de assen, zonder ze te raken.

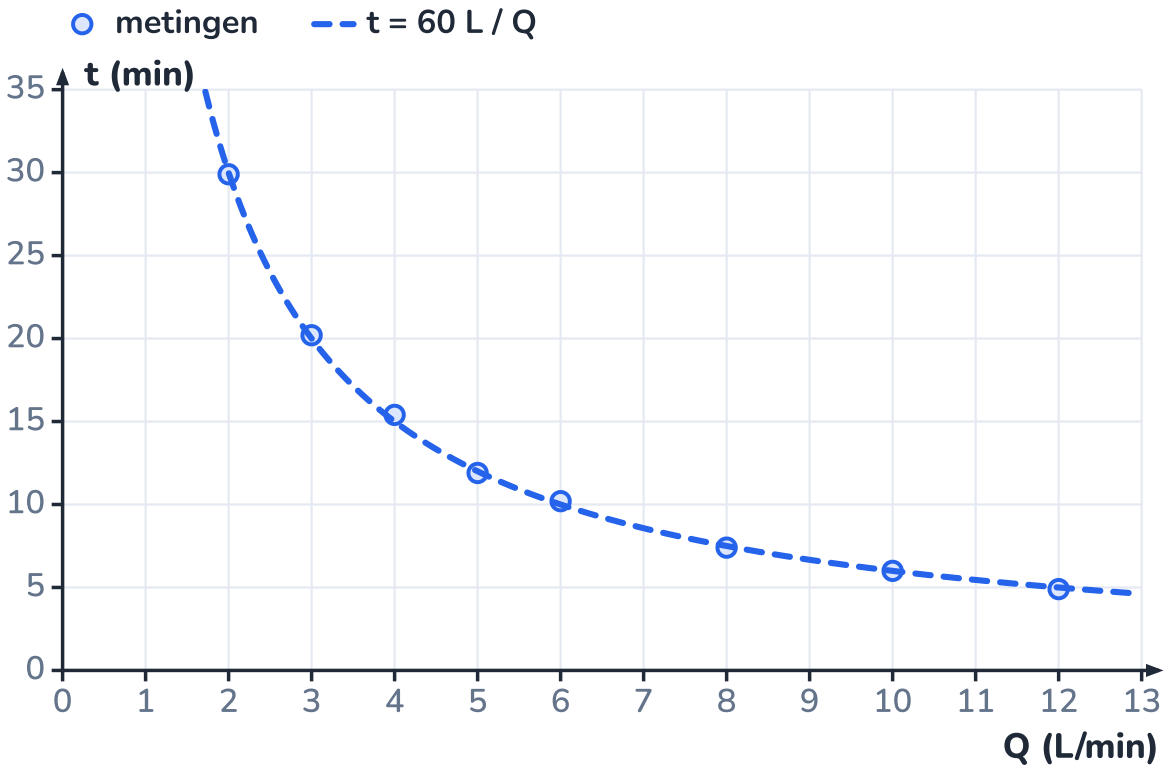
Voor de pomp is $t = V/Q$, met V het volume van het reservoir: $k = V$. Andere voorbeelden: de stroom door een weerstand bij vaste spanning, $I = U/R$ (hoofdstuk F19); de reistijd bij een vaste afstand, $t =$

s/v; de druk van een ingesloten gas bij vaste temperatuur, $p \cdot V =$ constant (hoofdstuk E14).

Herkennen doe je op drie manieren: in een tabel is het product $x \cdot y$ (binnen de meetfout) constant; in de formule staat x in de noemer; in de grafiek zie je een hyperbool. Die laatste is de zwakste test: veel dalende krommen lijken op een hyperbool.

Dalend is omgekeerd evenredig

Een waterstand die daalt met 0,8 cm per dag ($h = 30 - 0{,}8\,t$) daalt ook, maar het product $t \cdot h$ is niet constant: bij $t = 0$ is het nul. Een dalende rechte is lineair, geen omgekeerd evenredig verband. En ook een kromme die daalt, zoals een afkoelende thee (hoofdstuk 5), is niet vanzelf een hyperbool.



De vultijd daalt steeds trager bij grotere debieten: de kromme nadert de as, maar raakt ze nooit.

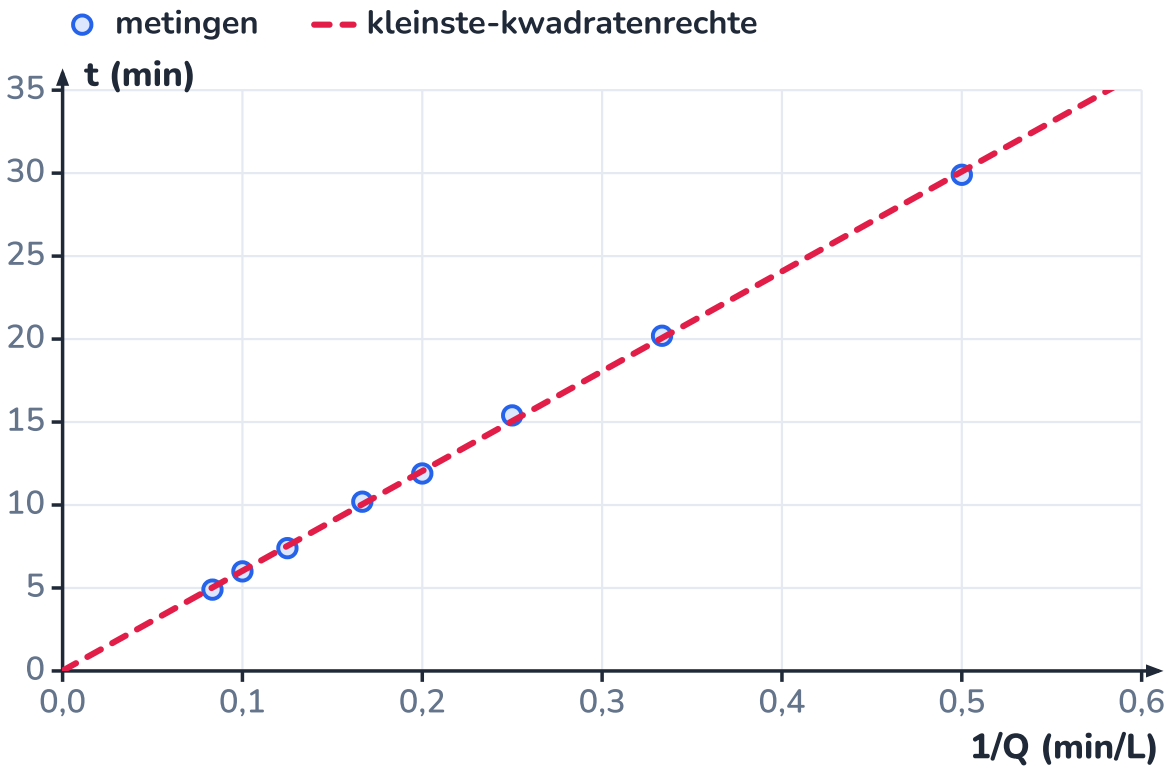
UITGEWERKT VOORBEELD · 3.11

Is het echt omgekeerd evenredig? De pomp

11/18

Bij acht instellingen van de pomp werd de vultijd van het lege reservoir gemeten. Toon aan dat het verband omgekeerd evenredig is en bepaal het volume van het reservoir. Hoe lang duurt het vullen bij 7 L/min?

Q (L/min)	2	3	4	5	6	8	10	12
t (min)	29,9	20,2	15,4	11,9	10,2	7,4	6,0	4,9
Q · t (L)	59,8	60,6	61,6	59,5	61,2	59,2	60,0	58,8
1/Q (min/L)	0,500	0,333	0,250	0,200	0,167	0,125	0,100	0,083



Uit de hyperbool wordt een rechte door de oorsprong; de helling is het volume van het reservoir.

GEGEVEN

- acht debieten Q en vultijden t

GEVRAAGD

- toets: omgekeerd evenredig?
- volume V van het reservoir
- t bij $Q = 7\,\mathrm{L/min}$

OPLOSSING

1. Productentest: $Q \cdot t$ ligt tussen 58,8 L en 61,6 L, gemiddeld $60{,}1\,\mathrm{L}$ met $s = 1{,}0\,\mathrm{L}$. Er is geen trend: klein en groot debiet geven hetzelfde product, binnen de meetfout.
2. Rechttrekken: als $t = V/Q = V \cdot (1/Q)$, dan geeft t tegen $1/Q$ een rechte door de oorsprong met helling V. De punten liggen inderdaad op een rechte (grafiek).
3. Kleinste kwadraten door de punten $(1/Q,\ t)$: $a = 60{,}15\,\mathrm{L}$ en $b = 0{,}02\,\mathrm{min}$, praktisch nul, zoals het model voorspelt.
4. De eenheid van de helling: min gedeeld door min/L geeft L. De helling is het volume: $V = 60\,\mathrm{L}$.
5. Voorspelling (interpoleren, binnen het gemeten bereik): $t = 60{,}15\,\mathrm{L} / 7\,\mathrm{L/min} = 8{,}6\,\mathrm{min}$.

ANTWOORD

Het verband is omgekeerd evenredig, het reservoir bevat $60\,\mathrm{L}$, en bij 7 L/min duurt vullen $8{,}6\,\mathrm{min}$.

Klopt dit? Het snijpunt van de rechte ligt vrijwel in de oorsprong: een extra constante tijd (bijvoorbeeld een leiding die eerst vol moet lopen) zou als snijpunt verschijnen. Hier is die er niet.

UITLEG · 3.12

Linearisen: een kromme rechttrekken

12/18

Een rechte is makkelijk: je ziet meteen of punten erop liggen, je trekt ze op het oog of met de kleinste kwadraten, en haar helling heeft een betekenis. Een kromme is lastig: is dit een hyperbool, een parabool of iets anders? Daarom **linearisee**r je: je zet op de assen de grootheden die het verwachte verband tot een rechte maken.

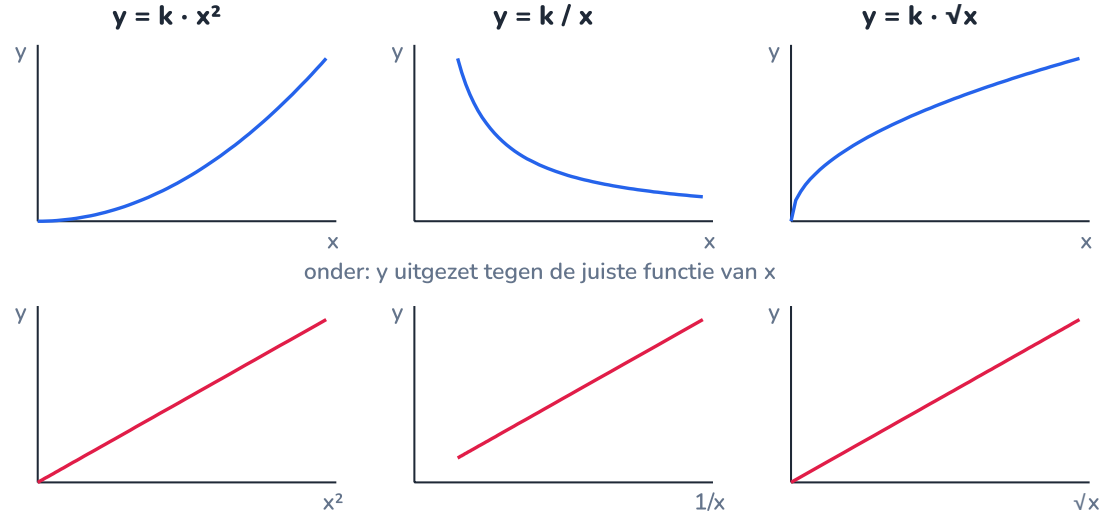
Het recept: schrijf het model in de vorm $y = a \cdot X + b$, waarbij X een *functie* van je gemeten grootheid is (een kwadraat, een omgekeerde, een wortel). Bereken X in een extra kolom en zet y uit tegen X.

model	horizontaal	verticaal	helling
$t = V/Q$ (pomp)	$1/Q$	t	V
$s = \frac{1}{2} g\, t^2$ (vrije val)	t^2	s	$g/2$
$T = 2\pi\sqrt{L/g}$ (slinger)	L	T^2	$4\pi^2/g$
$E = k/d^2$ (licht van een kleine lamp)	$1/d^2$	E	k
$I = U/R$ (vaste spanning)	$1/R$	I	U

Linearisen is ook een **test**. Liggen de punten na de omzetting op een rechte, dan past het model. Buigen ze nog, of vormen de residuen een patroon, dan klopt het model niet en zoek je verder. Bij exponentiële groei en verval (hoofdstuk 5) trek je de kromme recht met een logaritme.

Welke kant op?

Bij de slinger kan je ook T tegen $\sqrt{L}$ zetten. Beide geven een rechte, maar T^2 tegen L is handiger: L is wat je instelt en meet, en het snijpunt krijgt een betekenis (volgend voorbeeld).



Boven drie krommen, onder dezelfde gegevens met y uitgezet tegen x², 1/x en √x: telkens een rechte door de oorsprong.

UITGEWERKT VOORBEELD · 3.13

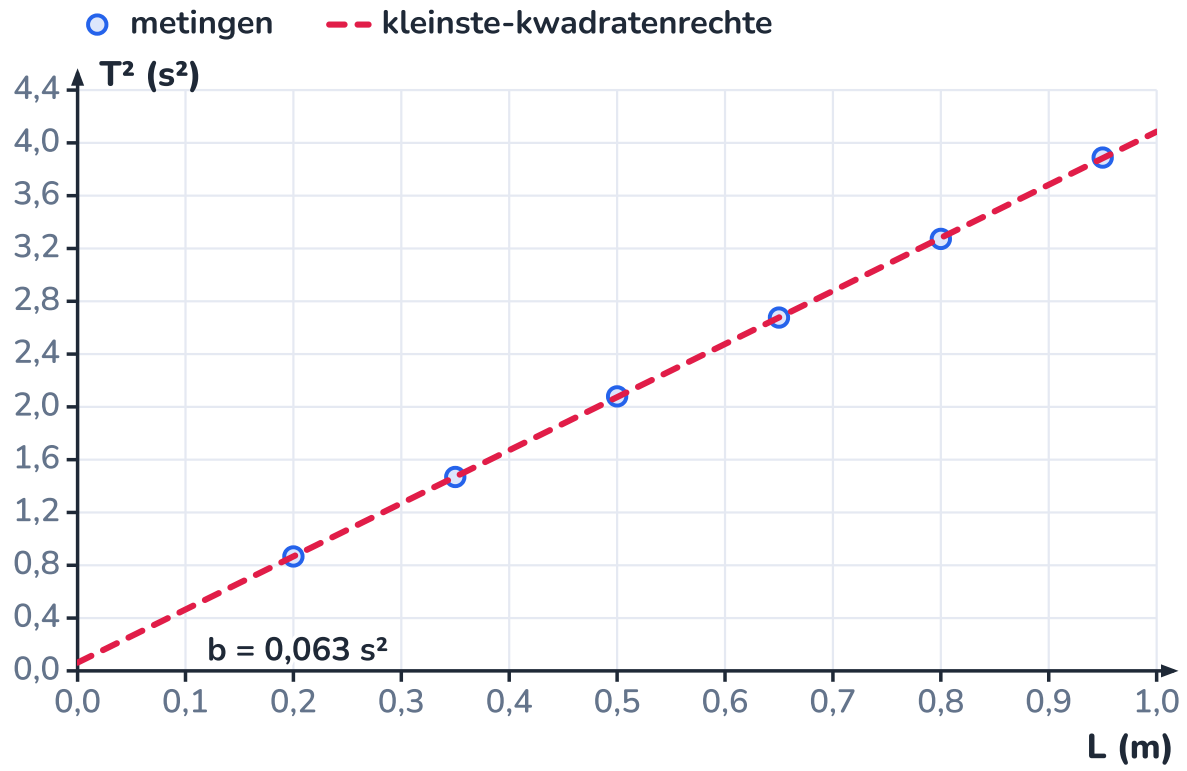
De valversnelling uit een slinger

13/18

Een leerling meet de slingertijd T (uit tien slingeringen, zoals in hoofdstuk 1) bij zes lengtes L. Ze meet L van het ophangpunt tot de *bovenkant* van de kogel. Bepaal de valversnelling g. Wat betekent het

snijpunt?

L (m)	0,200	0,350	0,500	0,650	0,800	0,950
T (s)	0,931	1,212	1,442	1,636	1,809	1,972
T² (s²)	0,8668	1,4689	2,0794	2,6765	3,2725	3,8888



T² tegen L is een rechte; ze snijdt de verticale as net boven nul, omdat L tot de bovenkant van de kogel gemeten werd.

GEGEVEN

- zes lengtes L (m) en slingertijden T (s)
- model: $T = 2\pi\sqrt{L/g}$

GEVRAAGD

- g uit de helling
- de betekenis van het snijpunt

OPLOSSING

1. Lineariseren: kwadrateer het model. $T^2 = \dfrac{4\pi^2}{g} \cdot L$. Dat is $y = a \cdot x$ met $y = T^2$, $x = L$ en $a = 4\pi^2/g$.
2. Bereken T^2 in een extra rij (tabel) en zet T^2 uit tegen L: de punten liggen op een rechte.
3. Kleinste kwadraten: $a = 4{,}022\,\mathrm{s^2/m}$ en $b = 0{,}063\,\mathrm{s^2}$, $R^2 = 0{,}99998$.
4. $g = \dfrac{4\pi^2}{a} = \dfrac{39{,}48}{4{,}022\,\mathrm{s^2/m}} = 9{,}81\,\mathrm{m/s^2}$.
5. Het snijpunt is niet nul. De rechte snijdt de L-as bij $L = -b/a = -0{,}063/4{,}022 = -0{,}0156\,\mathrm{m}$: de slinger gedraagt zich alsof hij 1,6 cm langer is dan gemeten. Dat is de straal van de kogel: de echte slingerlengte loopt tot het *midden* van de kogel.

ANTWOORD

$g = 9{,}81\,\mathrm{m/s^2}$. **Het snijpunt verraadt een systematische fout in de lengtemeting: L werd ongeveer 1,6 cm te kort gemeten, de straal van de kogel (1,5 cm gemeten met een schuifmaat).**

Klopt dit? Had de leerling de rechte door de oorsprong gedwongen, dan was de systematische fout in de helling terechtgekomen en had ze een verkeerde g gevonden. Een vrij snijpunt beschermt de helling tegen een constante fout in x.

PROEF · 3.14

Hoe hangt de verlichtingssterkte af van de afstand tot een lamp?

14/18

ONDERZOEKSVRAAG

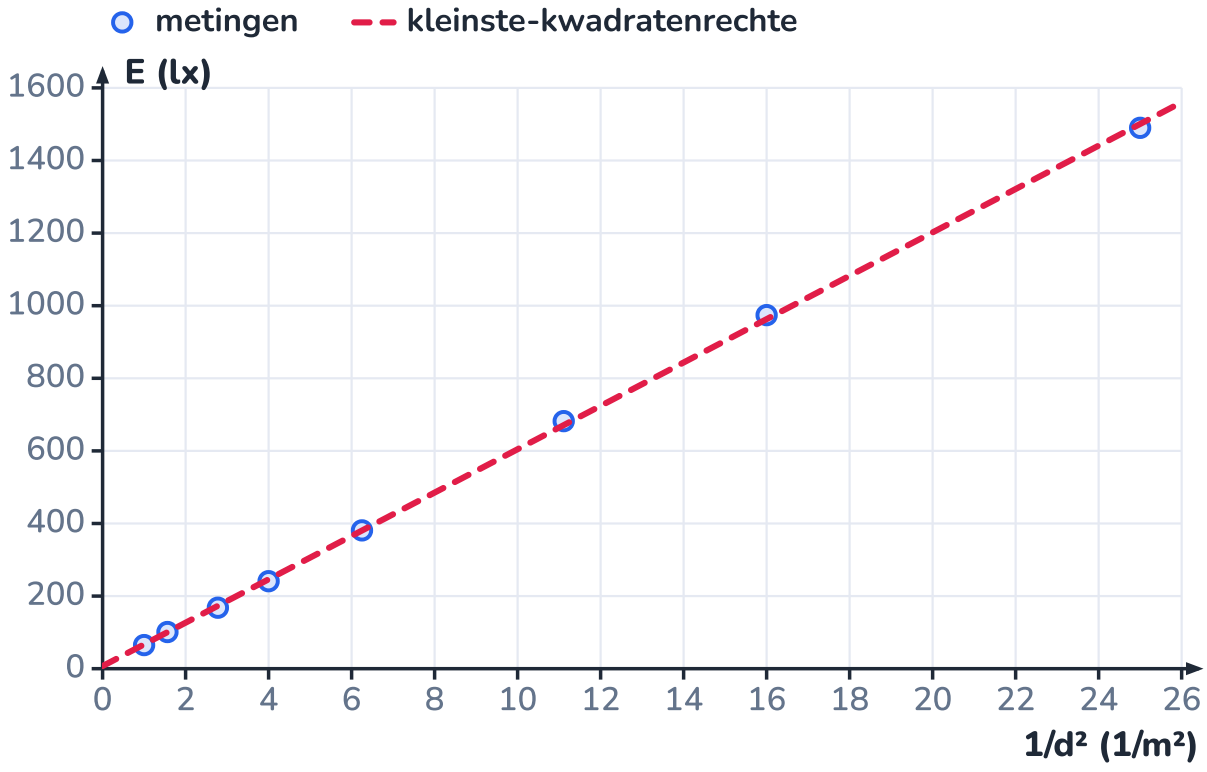
Hoe hangt de verlichtingssterkte E (in lux) af van de afstand d tot een kleine lamp, en hoe voorspel je dat voor een groeilamp in de kas?

MATERIAAL

- een klein ledlampje of zaklamp zonder reflector
- smartphone met een app die de lichtsensor uitleest (bijvoorbeeld phyphox)
- rolmeter, plakband
- een donker lokaal

Veiligheid

Kijk niet recht in een fel ledlampje van dichtbij.



Na het rechttrekken liggen alle punten op een rechte; het snijpunt (7 lx) is het achtergrondlicht.

WERKWIJZE

1. Verduister het lokaal. Leg de smartphone plat op tafel en zoek waar de lichtsensor zit (meestal bovenaan, naast de camera).
2. Houd de lamp recht boven de sensor. Meet de afstand d van de lamp tot de sensor, niet tot de telefoon.
3. Meet E bij acht afstanden van 0,20 m tot 1,00 m. Meet ook E met de lamp uit.

4 Bereken $1/d^2$ en $E \cdot d^2$ in een rekenblad en zet E uit tegen $1/d^2$ in een spreidingsdiagram.

WAARNEMING

d (m)	0,20	0,25	0,30	0,40	0,50	0,60	0,80	1,00
E (lx)	1490	974	682	381	241	168	101	6
1/d² (1/m²)	25,00	16,00	11,11	6,25	4,00	2,78	1,56	1
E · d² (lx·m²)	59,6	60,9	61,4	61,0	60,2	60,5	64,6	6

Het product $E \cdot d^2$ is niet constant: het stijgt van ongeveer 60 naar 65 lx·m² bij grote afstanden. E tegen $1/d^2$ geeft wel een mooie rechte, met $a = 59{,}7\ \mathrm{lx \cdot m^2}$ en $b = 7{,}2\ \mathrm{lx}$. Met de lamp uit meet de telefoon ongeveer 7 lx.

VERKLARING

Het licht van een kleine lamp verspreidt zich over een steeds groter boloppervlak, $4\pi d^2$: op dubbele afstand valt dezelfde hoeveelheid licht op een vier keer groter oppervlak. Dus $E = k/d^2$, de **kwadratenwet**. Het snijpunt van de rechte is het licht dat er altijd is: het achtergrondlicht in het lokaal (7 lx, zoals gemeten met de lamp uit). Het maakt het product $E \cdot d^2$ groter bij grote afstanden, waar de lamp zelf zwak is. De rechte haalt het er netjes uit.

Geldigheidsgebied: de kwadratenwet geldt voor een lichtbron die klein is tegenover de afstand. Een breed ledpaneel boven de planten is dat niet: daar daalt E trager met de afstand. Toch blijft de les voor de kas: een kleine lamp twee keer zo hoog hangen, geeft de planten maar een kwart van het licht.

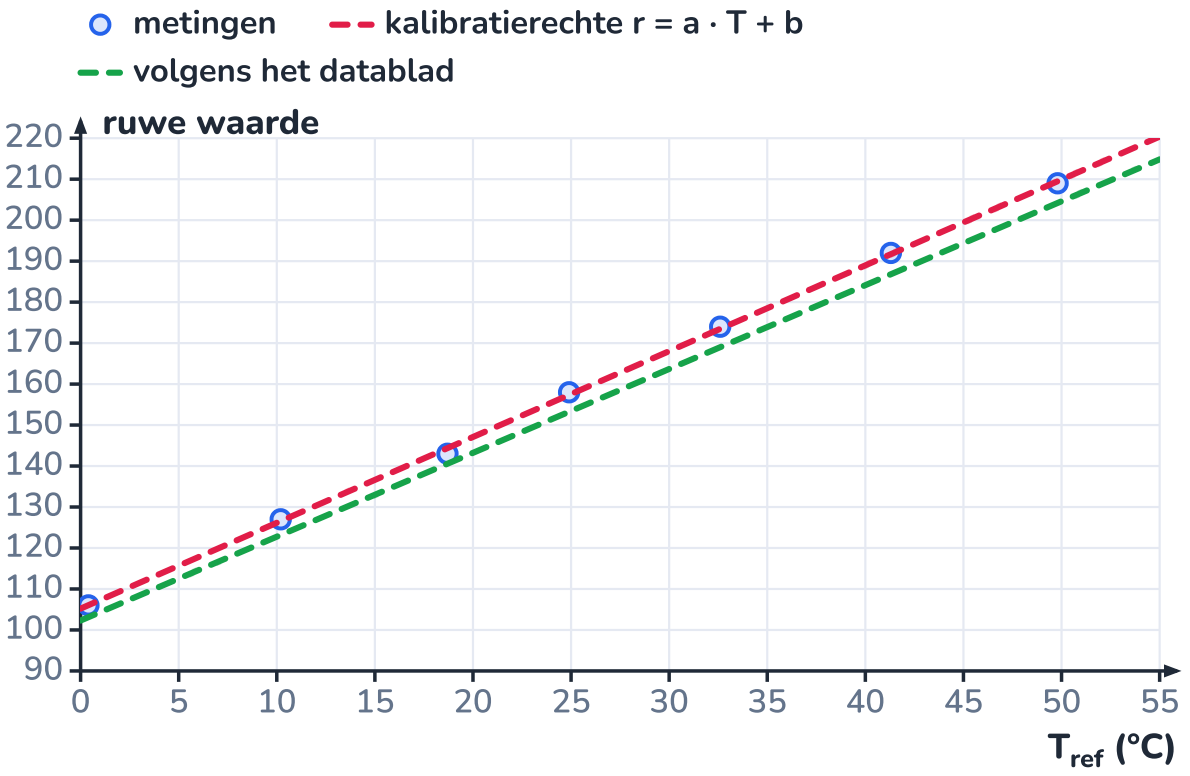
UITGEWERKT VOORBEELD · 3.15

Habitat: een temperatuursensor kalibreren

15/18

De klimaatmodule krijgt een goedkope analoge temperatuursensor die de microcontroller inleest als een getal tussen 0 en 1023 (de 'ruwe waarde'). Volgens het datablad hoort bij elke °C een vaste stap, maar elke sensor wijkt een beetje af. Je zet de sensor samen met een referentiethermometer in ijswater, kraantjeswater en warm water op verschillende temperaturen. Stel de kalibratieformule op die de microcontroller moet gebruiken.

T_{ref} (°C)	0,4	10,2	18,7	24,9	32,6	41,3	49,8
ruwe waarde	106	127	143	158	174	192	209



De eigen kalibratierechte (rood) ligt boven en is steiler dan de rechte uit het datablad (groen).

GEGEVEN

- zeven paren (referentietemperatuur, ruwe waarde)
- referentiethermometer $\pm 0{,}1\,^\circ\mathrm{C}$

GEVRAAGD

- de kalibratierechte
- een formule die een ruwe waarde omzet naar °C
- hoe nauwkeurig is de gekalibreerde sensor?

OPLOSSING

1 Kies de assen: de referentie is (bijna) foutloos, de ruwe waarde heeft ruis. Zet daarom de ruwe waarde r verticaal en T_{ref} horizontaal: $r = a \cdot T + b$.

2 Kleinste kwadraten (rekenblad of Python, volgend blad): $a = 2{,}094$ tellen per °C, $b = 105{,}2$ tellen, $R^2 = 0{,}9996$. De helling is de **gevoeligheid** van de sensor, het snijpunt de ruwe waarde bij 0 °C.

3 Omvormen naar de formule die je nodig hebt: $T = \dfrac{r - b}{a} = \dfrac{r - 105{,}2}{2{,}094}$, of $T = 0{,}4775 \cdot r - 50{,}24$ (in °C).

4 Controle met de residuen, omgerekend naar °C: tussen $-0{,}65$ en $+0{,}31\,^\circ\mathrm{C}$, zonder patroon. Eén stap van de ruwe waarde komt overeen met $1/2{,}094 = 0{,}48\,^\circ\mathrm{C}$: de sensor kan niet fijner meten dan dat.

5 Vergelijk met het datablad ($r = 204{,}6 \cdot (0{,}5 + 0{,}01 \cdot T)$, dus $a = 2{,}046$ en $b = 102{,}3$): bij $r = 158$ geeft die formule 27,2 °C, terwijl de referentie 24,9 °C aanwees. Het verschil loopt op van +1,4 °C bij 0 °C tot +2,4 °C bij 50 °C: zowel het snijpunt als de helling wijken af.

ANTWOORD

Kalibratieformule: $T = 0{,}4775 \cdot r - 50{,}24$ (°C), geldig van 0 tot 50 °C. De gekalibreerde sensor zit binnen ongeveer $\pm 0{,}7\,^\circ\mathrm{C}$ van de referentie, beperkt door zijn stappen van 0,48 °C.

Klopt dit? Een correctie met één getal, zoals in hoofdstuk 1, volstaat hier niet: de afwijking is niet overal even groot. Een rechte corrigeert tegelijk de verschuiving (snijpunt) en de gevoeligheid (helling). Buiten 0 tot 50 °C heb je geen metingen: daar is de formule een extrapolatie.

CODE · 3.16

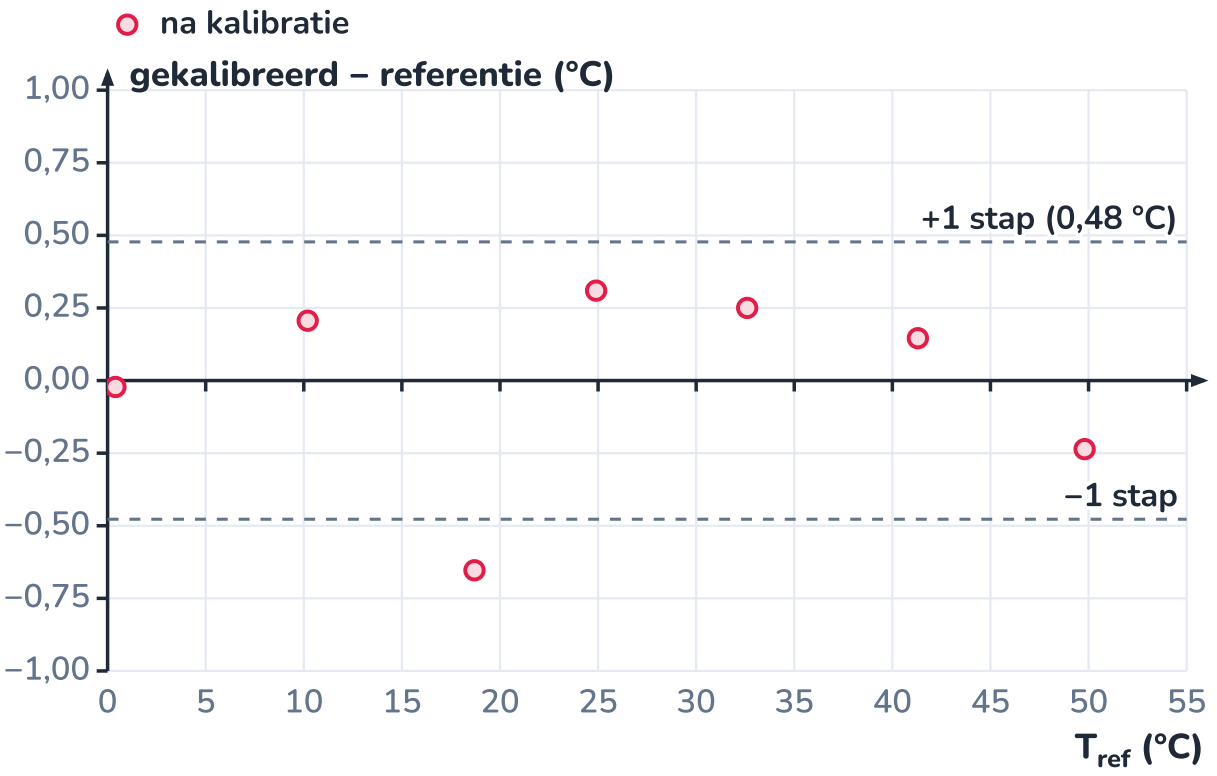
De kalibratie in Python

16/18

Wat het rekenblad met RICHTING en SNIJPUNT doet, kan je ook in een paar regels Python schrijven, met de formules van de kleinste kwadraten. Het voordeel: hetzelfde programma kalibreert morgen tien sensoren, en de functie `temperatuur()` kan je later bijna letterlijk overnemen in de software van de klimaatmodule. Hoe lijsten, lussen en functies werken, leer je in hoofdstuk 28; hier volstaat het om de formules te herkennen.

Python **kalibratie.py** 27 regels

```python
# kalibratie.py - kalibratierechte van een temperatuursensor (kleinste kwadraten)
T_ref = [0.4, 10.2, 18.7, 24.9, 32.6, 41.3, 49.8]   # referentiethermometer (°C)
ruw = [106, 127, 143, 158, 174, 192, 209]           # ruwe waarde van de sensor (ADC)

n = len(T_ref)
x_gem = sum(T_ref) / n
y_gem = sum(ruw) / n
teller = sum((x - x_gem) * (y - y_gem) for x, y in zip(T_ref, ruw))
noemer = sum((x - x_gem) ** 2 for x in T_ref)
a = teller / noemer          # gevoeligheid: tellen per °C
b = y_gem - a * x_gem        # ruwe waarde bij 0 °C

# R²: welk deel van de spreiding van de ruwe waarden verklaart de rechte?
ss_res = sum((y - (a * x + b)) ** 2 for x, y in zip(T_ref, ruw))
ss_tot = sum((y - y_gem) ** 2 for y in ruw)
r2 = 1 - ss_res / ss_tot
print(f"a = {a:.4f} tellen/°C   b = {b:.2f} tellen   R² = {r2:.5f}")


def temperatuur(r):
    """Zet een ruwe sensorwaarde om naar °C met de omgekeerde kalibratierechte."""
    return (r - b) / a


for x, y in zip(T_ref, ruw):
    t = temperatuur(y)
    print(f"ref {x:5.1f} °C   ruw {y:4d}   gekalibreerd {t:5.1f} °C   verschil {t - x:+.2f} °C")
```



Na kalibratie blijven de afwijkingen binnen ongeveer één stap van de sensor (0,48 °C), zonder patroon.

Uitvoer (gecontroleerd door het programma te draaien):

```
a = 2.0941 tellen/°C   b = 105.21 tellen   R² = 0.99961
ref   0.4 °C   ruw  106   gekalibreerd   0.4 °C   verschil -0.02 °C
ref  10.2 °C   ruw  127   gekalibreerd  10.4 °C   verschil +0.21 °C
ref  18.7 °C   ruw  143   gekalibreerd  18.0 °C   verschil -0.65 °C
ref  24.9 °C   ruw  158   gekalibreerd  25.2 °C   verschil +0.31 °C
ref  32.6 °C   ruw  174   gekalibreerd  32.9 °C   verschil +0.25 °C
ref  41.3 °C   ruw  192   gekalibreerd  41.4 °C   verschil +0.15 °C
ref  49.8 °C   ruw  209   gekalibreerd  49.6 °C   verschil -0.24 °C
```

Python toont een punt als decimaalteken. De getallen zijn dezelfde als in het vorige blad. `zip()` overloopt twee lijsten paar per paar; `sum(... for ...)` telt de termen van een som op, precies zoals het $\sum$-teken in de formule. Wie numpy kent, krijgt a en b ook met `numpy.polyfit(T_ref, ruw, 1)`.

NASLAG · 3.17

Overzicht: rechten, verbanden en rekenblad

17/18

lineair verband

$$y = a \cdot x + b$$

symbool	betekenis	eenheid
a	richtingscoëfficiënt	eenheid y / eenheid x
b	snijpunt met de y-as	eenheid y

helling uit twee punten op de rechte

$$a = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

kleinste kwadraten: helling

$$a = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

kleinste kwadraten: snijpunt

$$b = \bar{y} - a \cdot \bar{x}$$

determinatiecoëfficiënt

$$R^2 = 1 - \frac{\sum(\text{residu})^2}{\sum(y_i - \bar{y})^2}$$

fout op de helling

$$\Delta a \approx \frac{a_{\text{steilst}} - a_{\text{minst steil}}}{2}$$

omgekeerd evenredig

$$y = \frac{k}{x} \Leftrightarrow x \cdot y = k$$

verband	tabel	grafiek	rechttrekken
recht evenredig $y = kx$	y/x constant	rechte door de oorsprong	(al recht)
lineair $y = ax + b$	gelijke stappen in x geven gelijke stappen in y	rechte	(al recht)
omgekeerd evenredig $y = k/x$	$x \cdot y$ constant	hyperbool	y tegen $1/x$
kwadratisch $y = kx^2$	y/x^2 constant	halve parabool	y tegen x^2
kwadratenwet $y = k/x^2$	$x^2 \cdot y$ constant	snel dalende kromme	y tegen $1/x^2$
wortel $y = k\sqrt{x}$	y^2/x constant	afvlakkende kromme	y tegen $\sqrt{x}$ of y^2 tegen x

wat	Nederlandstalig rekenblad	Engelstalig rekenblad
richtingscoëfficiënt	`=RICHTING(B2:B7;A2:A7)`	`=SLOPE(B2:B7,A2:A7)`
snijpunt met de y-as	`=SNIJPUNT(B2:B7;A2:A7)`	`=INTERCEPT(B2:B7,A2:A7)`
R^2	`=R.KWADRAAT(B2:B7;A2:A7)`	`=RSQ(B2:B7,A2:A7)`
gemiddelde	`=GEMIDDELDE(A2:A7)`	`=AVERAGE(A2:A7)`
kwadraat, omgekeerde, wortel	`=A2^2` · `=1/A2` · `=WORTEL(A2)`	`=A2^2` · `=1/A2` · `=SQRT(A2)`

Let op de volgorde: eerst de *y*-waarden, dan de *x*-waarden. In een Nederlandstalig rekenblad scheidt een puntkomma de argumenten, in een Engelstalig een komma.

controlelijst voor een grafiek met een rechte
spreidingsdiagram, geen lijndiagram; assen met grootheid en eenheid
punten niet verbinden; foutbalken als de fout bekend is
helling uit twee punten op de rechte, ver uit elkaar, of met de kleinste kwadraten
helling en snijpunt met eenheid en betekenis, afgerond volgens de fout
residuen bekeken: willekeurig rond nul?
geldigheidsgebied vermeld: alleen binnen het gemeten bereik voorspellen

HET GEHEEL · 3.18

Waar verbanden in data terugkomen

18/18

De rechte is het werkpaard van elk labo: bijna elk verband dat je in StemExpert tegenkomt, maak je ooit recht om er een grootheid uit te halen. Dit hoofdstuk sluit het deel 'Onderzoeken met data' af: meten met een fout (hoofdstuk 1), eerlijk ontwerpen (hoofdstuk 2), en nu het verband vinden. In het volgende deel wordt de rechte een van de vele functies die je als model gebruikt.

F4 Tabellen, grafieken en verbanden

I1 Meetfouten en nauwkeurigheid

I4 Functies als modellen

Recht evenredig, trendlijn, interpoleren en extrapoleren: hier uitgebreid tot rechten met snijpunt en de kleinste kwadraten.

Foutbalken, uitschieters en afronden bepalen hoe zeker een helling is.

De lineaire functie als model naast de kwadratische, met domein en geldigheidsgebied.

I5 Groei en verval: exponentiële modellen

Groei en verval rechttrekken met een logaritmische schaal.

I7 Beweging beschrijven

De helling van een (x,t)-grafiek is de snelheid, die van een (v,t)-grafiek de versnelling.

I22 De spanningsdeler en sensoren

Een sensor met een microcontroller inlezen en kalibreren met een referentie.

I28 Functies, lijsten en data in Python

Lijsten, functies en CSV-bestanden in Python: de kalibratie van dit hoofdstuk voor een hele week data.

E1 Afgeleiden: verandering meten

De helling van een kromme in één punt: de afgeleide als richtingscoëfficiënt van de raaklijn.

E41 Van data naar model: regressie

Kleinste kwadraten met meerdere variabelen, overfitting en het analyseren van residuen.

Habitat

Elke sensor van de klimaatmodule wordt voor gebruik gekalibreerd met een rechte, zoals de temperatuursensor in dit hoofdstuk. In opdracht 3.11 kalibreer je de bodemvochtsensor; de formules die je vindt, komen later in de software van de module (hoofdstukken 22, 29 en 30).

SAMENVATTING

In het kort

- Een lineair verband $y = a \cdot x + b$ heeft een richtingscoëfficiënt $a = \Delta y / \Delta x$ en een snijpunt b, allebei grootheden met een eenheid en een betekenis (dichtheid, opwarmsnelheid, beginwaarde).
- Bepaal de helling uit twee punten op de rechte, ver uit elkaar. De steilste en de minst steile rechte door de foutbalken geven de fout op de helling.
- De methode van de kleinste kwadraten kiest de rechte met de kleinste som van de gekwadrateerde residuen; ze gaat door $(\bar{x}, \bar{y})$.
- R^2 zegt hoe dicht de punten bij de rechte liggen, niet of een rechte het juiste model is: kijk naar de residuen.
- Omgekeerd evenredig: $y = k/x$, het product $x \cdot y$ is constant, de grafiek is een hyperbool. Dalend is niet hetzelfde als omgekeerd evenredig.
- Lineariseren: zet y uit tegen een functie van x ($1/x$, x^2, $1/x^2$ ...) zodat het model een rechte wordt; een snijpunt dat niet nul is, verraadt vaak een systematische fout of een achtergrond.
- In een rekenblad: één grootheid per kolom met eenheid in de kop, formules met relatieve en absolute verwijzingen (met dollartekens), altijd een spreidingsdiagram, RICHTING en SNIJPUNT.
- Een sensor kalibreer je met een rechte tegenover een referentie en een omgevormde formule $T = (r - b)/a$, geldig binnen het gemeten bereik.

BEGRIPPEN

Wat je nu kent

lineair verband

Verband van de vorm $y = a \cdot x + b$; de grafiek is een rechte. Recht evenredig is het speciale geval $b = 0$.

richtingscoëfficiënt

De helling $a = \Delta y / \Delta x$ van een rechte, met als eenheid de eenheid van y gedeeld door die van x.

snijpunt met de y-as

De waarde b van y bij $x = 0$, met de eenheid van y.

best passende rechte

De rechte die zo dicht mogelijk bij alle meetpunten ligt; met de methode van de kleinste kwadraten wordt ze uniek bepaald.

residu

Het verschil tussen een gemeten waarde en de waarde op de rechte (of het model) bij dezelfde x.

kleinste kwadraten

Methode die de rechte kiest waarvoor de som van de kwadraten van de residuen het kleinst is.

R² (determinatiecoëfficiënt)

Het deel van de spreiding in y dat de rechte verklaart; 1 = alle punten op de rechte. Bewijst geen lineair verband.

omgekeerd evenredig

Verband $y = k/x$: het product $x \cdot y$ is constant; verdubbel je x, dan halveert y. De grafiek is een hyperbool.

lineariseren

Grootheden op de assen zetten (bv. $1/x$ of x^2) zodat een verwacht verband een rechte wordt.

kalibratierechte

Rechte die de ruwe waarde van een sensor koppelt aan de echte waarde, bepaald met een referentie.

spreidingsdiagram

Grafiek met punten op hun echte x- en y-waarde (xy-grafiek); het juiste diagram voor meetreeksen.

HANDBOEK · HOOFDSTUK 4

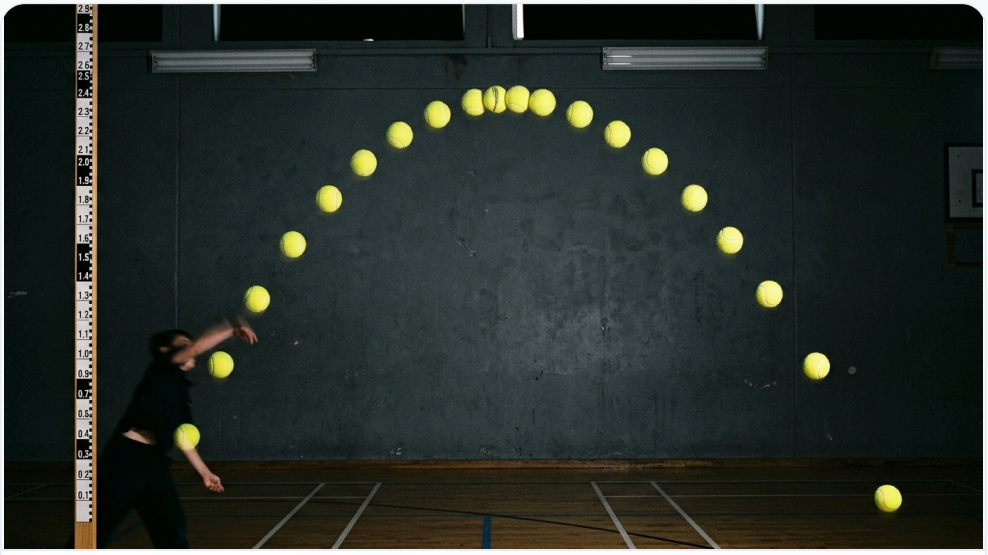
Functies als modellen

Rechten en parabolen die kosten, afstanden en remwegen voorspellen, en weten waar ze ophouden

Een formule die een stopafstand, een kostprijs of de hoogte van een bal voorspelt, is een **model**: een functie die bij elke invoer één uitvoer geeft. In dit hoofdstuk leer je functies lezen als tabel, grafiek en formule, werk je met de twee werkpaarden van de techniek, de lineaire en de kwadratische functie, en geef je betekenis aan hun top, nulpunten en snijpunten. Je kiest een model dat bij je data past, en je leert de belangrijkste vraag bij elk model stellen: tot waar mag ik het geloven?

7× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 9 lestijden

W Wiskundig modelleren · K Krachten, beweging en constructies



Gelijke tijdstappen, ongelijke afstanden: de baan van een bal is een parabool, en de top is waar hij het traagst gaat.

Na dit hoofdstuk

- Je kan uitleggen wat een functie is en een functie voorstellen als tabel, grafiek en formule.
- Je kan het domein en het bereik van een functie in een context bepalen, met eenheden, en onderscheiden van wat de formule wiskundig toelaat.
- Je kan een lineaire functie opstellen uit een beschrijving of uit twee punten, en richtingscoëfficiënt en snijpunt interpreteren als grootheden.
- Je kan van een kwadratische functie de top, de symmetrieas en de nulpunten berekenen en ze in de context betekenis geven.
- Je kan snijpunten van twee functies algebraïsch en grafisch bepalen en uitleggen wat ze betekenen (kantelpunt, inhalen).
- Je kan met eerste en tweede verschillen nagaan of data lineair of kwadratisch zijn, en een model kiezen dat bij data én fysica past.
- Je kan het geldigheidsgebied van een model benoemen en inschatten wanneer extrapoleren misloopt.
- Je kan in Python een functie schrijven, een tabel maken en een lineair en een kwadratisch model fitten.

UITLEG · 4.1

Wat je al weet, en wat nu komt

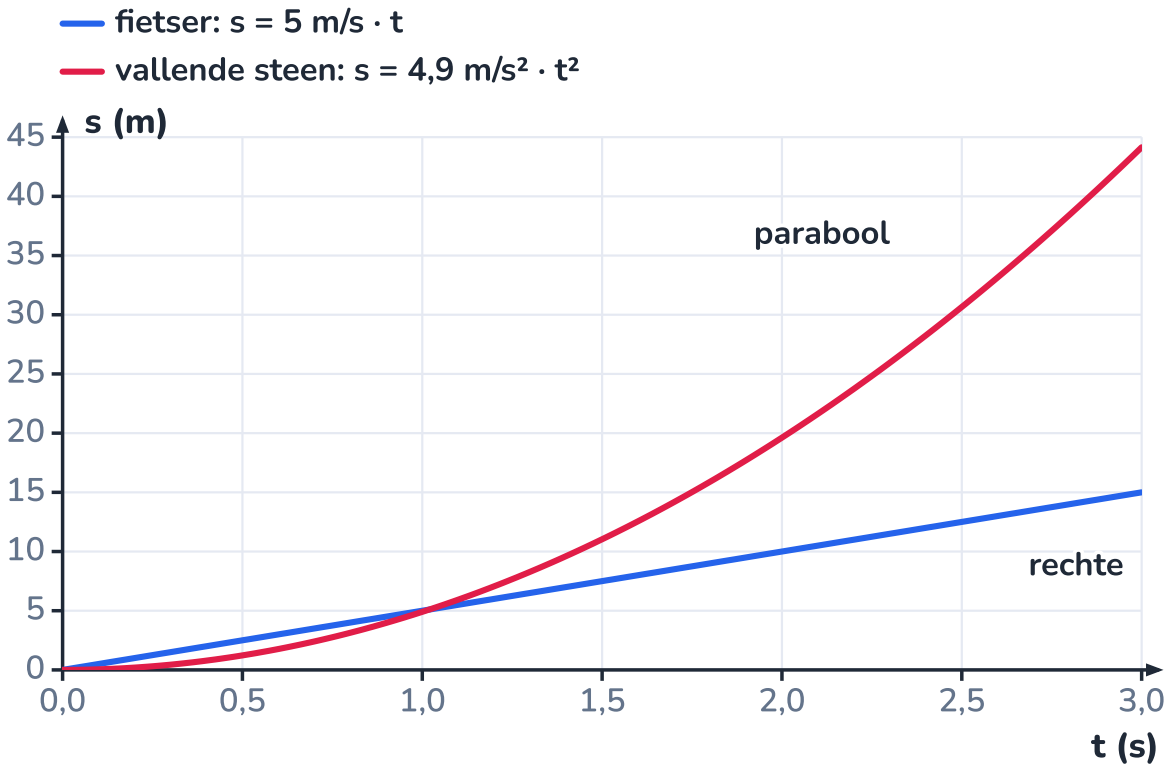
In Fundamental (hoofdstuk F4) zette je metingen in een tabel en een grafiek, en herkende je het **recht evenredig verband** $y = k \cdot x$. In hoofdstuk F14 las je een (s,t)-grafiek: een rechte betekent een constante snelheid, en hoe steiler, hoe sneller. In hoofdstuk 3 werd de rechte algemener, $y = a \cdot x + b$, en leerde je de helling en het snijpunt lezen als grootheden met een eenheid. Je trok krommen recht door y uit te zetten tegen x^2 of $1/x$.

Nu draai je het om. In plaats van een verband uit metingen te halen, gebruik je een verband om te **voorspellen**: wat kost een systeem met 30 plantplaatsen, hoe hoog komt een bal, bij welke snelheid stop je nog op tijd? Zo'n voorspellend verband is een **functie**. In dit hoofdstuk

- leer je wat een functie is, en wat haar **domein** en **bereik** zijn in een echte situatie;
- werk je met de **lineaire functie** (vaste start, vaste stap) en de **kwadratische functie** (de parabool van vallen, remmen en oppervlakte);
- geef je betekenis aan **top**, **nulpunten** en **snijpunten**;
- kies je een model dat bij data past, en zoek je zijn **geldigheidsgebied**.

Eén letter, veel betekenissen

In de wiskunde heet de invoer vaak x en de uitvoer y. In de techniek krijgen ze de naam van hun grootheid: t voor tijd, h voor hoogte, v voor snelheid, K voor kosten. De wiskunde is dezelfde; de eenheden maken het een model.



De fietser legt elke seconde 5 m af: een rechte. De steen valt elke seconde verder dan de vorige: een parabool.

VERHAAL · 4.2

Galilei en de oneven getallen

Rond 1604 wilde **Galileo Galilei**, hoogleraar wiskunde in Padua, weten hoe een vallend voorwerp versnelt. Een vrije val duurt te kort om met de klokken van toen te meten. Daarom liet hij een bronzen kogel rollen door een gladde groef in een lange houten balk die schuin stond: dezelfde soort beweging, maar trager. De tijd mat hij door water uit een vat te laten lopen zolang de kogel rolde, en dat water te wegen.

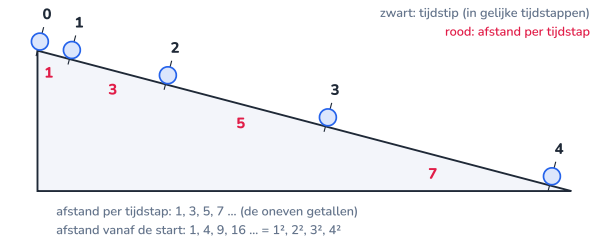
Zijn vondst was verrassend eenvoudig. Deel de tijd op in gelijke stukken. In het eerste stuk legt de kogel een afstand 1 af, in het tweede 3, in het derde 5, in het vierde 7: de **oneven getallen**. Tel je ze op, dan krijg je de afstand vanaf de start: 1, 4, 9, 16, de **kwadraten** van 1, 2, 3, 4. De afgelegde weg is evenredig met het kwadraat van de tijd, $s = k \cdot t^2$, en dat gold voor elke helling die hij probeerde, en dus ook voor de vrije val, de steilste helling van allemaal.

Galilei publiceerde dit pas in 1638, in zijn boek *Discorsi*. Hij was toen 74 en stond onder huisarrest na zijn proces over het wereldbeeld; het boek werd daarom gedrukt in Leiden, in de Nederlanden. Het boek werd een van de grondslagen van de natuurkunde: niet omdat het een theorie gaf, maar omdat het toonde dat je een beweging kan meten, in getallen kan gieten en in een wiskundige functie kan vangen.

Wat je uit dit verhaal meeneemt

De oneven getallen zijn de **eerste verschillen** van de kwadraten: 4 − 1 = 3, 9 − 4 = 5 … en hun eigen verschillen (de **tweede verschillen**) zijn allemaal 2. Constante tweede verschillen zijn de vingerafdruk van een kwadratische functie. Je gebruikt die test verderop om een model te kiezen.

Een reconstructie van de opstelling: een kogel in een groef, en een waterklok die de tijd meet in gram water.



Na gelijke tijdstappen staat de knikker op afstand 1, 4, 9, 16: tussen twee stappen legt hij 1, 3, 5, 7 af.

UITLEG · 4.3

Functie, domein en bereik

Een **functie** is een voorschrift dat aan elke toegelaten invoer **precies één** uitvoer koppelt. Je schrijft $y = f(x)$: 'y is f van x'. De invoer x is de **onafhankelijke** veranderlijke (die kies je, of die verloopt vanzelf, zoals de tijd), de uitvoer y de **afhankelijke**. Dezelfde functie kan je op drie manieren tonen:

- een **tabel**: enkele invoerwaarden met hun uitvoer, handig om af te lezen;
- een **grafiek**: het verloop in één oogopslag, met stijgen, dalen en extreme waarden;
- een **formule** (voorschrift): het compactst, en de enige vorm waarmee je exact kan rekenen.

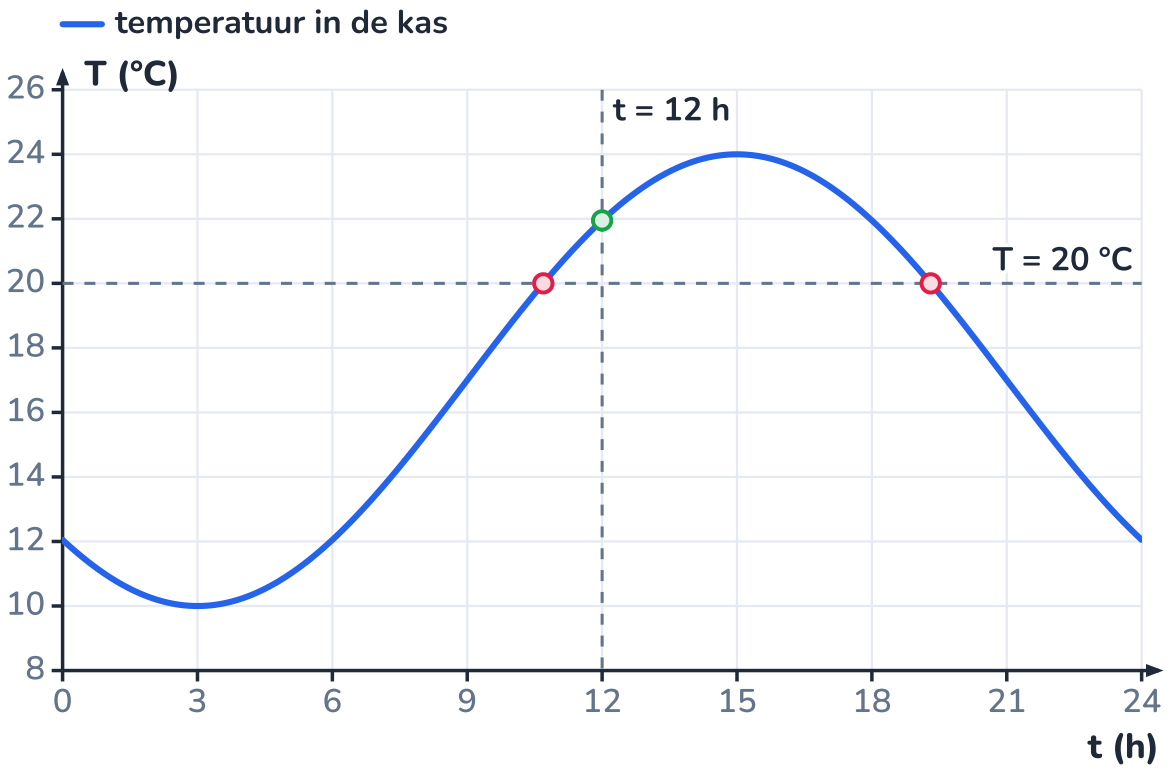
De temperatuur in de kas als functie van de tijd is een functie: op elk tijdstip is er één temperatuur. Omgekeerd niet: de temperatuur 20 °C komt twee keer voor, rond 10.40 u en rond 19.20 u. In de grafiek

zie je dat met de **verticale-lijntest**: een grafiek stelt een functie van x voor als elke verticale lijn hem hoogstens één keer snijdt.

Het **domein** is de verzameling invoerwaarden waarvoor je de functie gebruikt; het **bereik** de verzameling uitvoerwaarden die ze dan aanneemt. Bij de kas: domein $0 \leq t \leq 24\,\text{h}$, bereik $10 \leq T \leq 24\,^\circ\text{C}$. In een model heeft het domein twee grenzen:

- **wiskundig**: wat de formule toelaat. Je mag niet delen door nul, en geen vierkantswortel nemen van een negatief getal.
- **fysisch**: wat in de werkelijkheid zin heeft. Een tijd vóór het loslaten, een negatief aantal plantplaatsen, een bal onder de grond. En vaak ook: waar het model getoetst is.

> ⚠ **Punten of een lijn?**
> Kost een plantplaats € 11, dan is de kost voor n plaatsen $K = 11n$. Het domein is $\{0, 1, 2, \dots\}$: natuurlijke getallen. De grafiek is een rij **losse punten** die op een rechte liggen, geen doorlopende lijn. Een halve plantplaats bestaat niet.



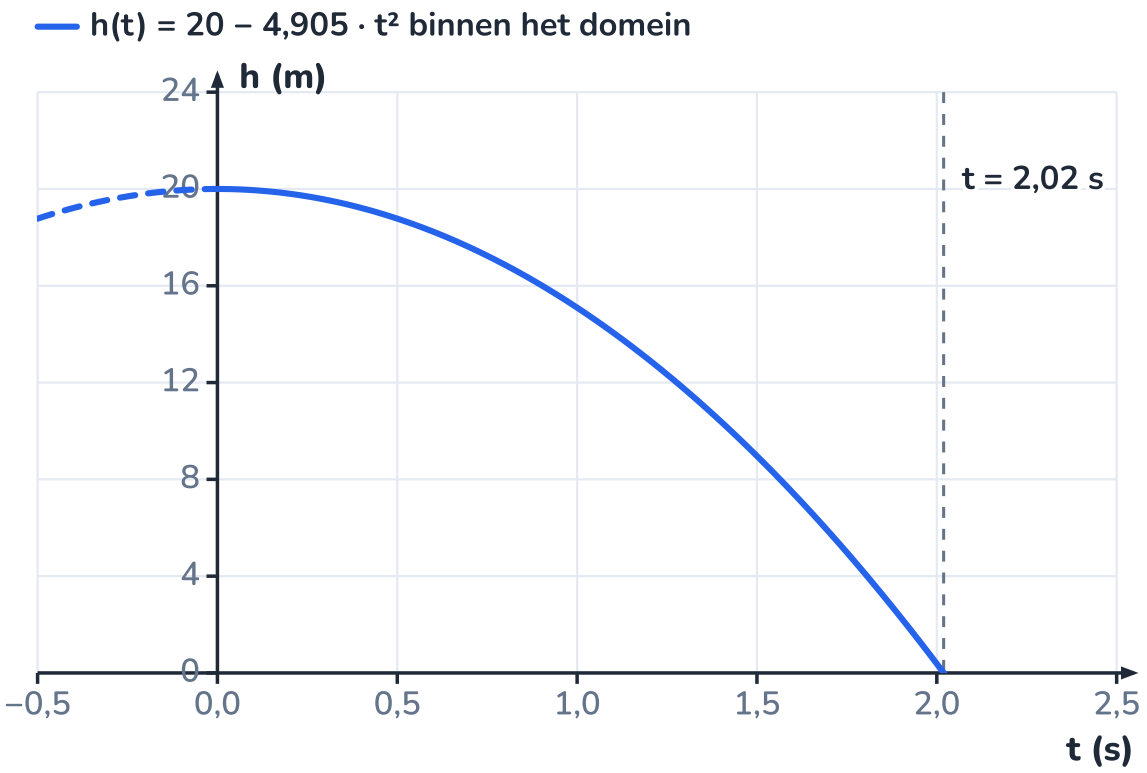
Elke verticale lijn snijdt de kromme één keer: T is een functie van t. De horizontale lijn bij 20 °C snijdt twee keer: t is geen functie van T.

UITGEWERKT VOORBEELD · 4.4

Een steen valt van een brug: domein en bereik

4/18

Je laat een steen los vanaf een brug, 20 m boven het water. Zonder luchtweerstand is zijn hoogte boven het water $h(t) = 20 - 4{,}905 \cdot t^2$ (h in m, t in s, $t = 0$ bij het loslaten). Bepaal het domein en het bereik van dit model, en leg uit wat de formule buiten het domein 'voorspelt'.



Volle lijn: het model binnen zijn domein. Stippellijn: wat de formule zou zeggen als je ze blind gelooft.

GEGEVEN

- $h(t) = 20 - 4{,}905\,t^2$
- $4{,}905\,\mathrm{m/s^2} = \frac{1}{2}g$ met $g = 9{,}81\,\mathrm{m/s^2}$

GEVRAAGD

- domein en bereik in de context
- betekenis van de formule buiten het domein

OPLOSSING

1. Wiskundig mag je elke t invullen: een kwadraat bestaat altijd. De grenzen komen dus uit de fysica.
2. Begin: de beweging start bij het loslaten, dus $t \geq 0$.
3. Einde: de steen raakt het water als $h = 0$: $4{,}905\,t^2 = 20$, dus $t^2 = 4{,}077$ en $t = \sqrt{4{,}077} = 2{,}02\,\mathrm{s}$. De negatieve oplossing $-2{,}02\,\mathrm{s}$ valt weg: vóór het loslaten viel er niets.
4. Domein: $0 \leq t \leq 2{,}02\,\mathrm{s}$. Op dat domein daalt h van 20 m naar 0 m, dus het bereik is $0 \leq h \leq 20\,\mathrm{m}$.
5. Buiten het domein rekent de formule gewoon door: bij $t = 2{,}5\,\mathrm{s}$ geeft ze $h = -10{,}7\,\mathrm{m}$, een steen die ongehinderd onder water verder versnelt. En bij $t = -0{,}5\,\mathrm{s}$ zou de steen al gevallen zijn, maar omhoog: de parabool loopt naar links weer omlaag. Het model weet niets van de hand die de steen vasthield of van het water.

ANTWOORD

Domein $[0;\ 2{,}02]\,\mathrm{s}$, bereik $[0;\ 20]\,\mathrm{m}$. Buiten het domein geeft de formule getallen, maar geen werkelijkheid.

Klopt dit? Een valtijd van ongeveer 2 s voor 20 m klopt met de ervaring (probeer het met een steentje van een brug en tel). Luchtweerstand maakt de echte valtijd voor een steen maar enkele procenten langer; voor een blad papier zou het model volledig falen.

De lineaire functie: vaste start, vaste stap

Veel situaties hebben een **vaste start** en een **vaste stap**: een abonnement met een vast bedrag plus een prijs per eenheid, een fietser die vanaf een startpunt met constante snelheid rijdt, een reservoir dat met een constant debiet leegloopt. Dat zijn lineaire functies:

LINEAIRE FUNCTIE

$$f(x) = a\,x + b$$

b is de waarde bij $x = 0$ (de vaste start), a de verandering van y per eenheid x (de vaste stap, de richtingscoëfficiënt). Eenheid van a: eenheid van y gedeeld door die van x.

Het kenmerk in een tabel: bij gelijke stappen in x zijn de **eerste verschillen** van y constant. Heb je twee punten (x_1, y_1) en (x_2, y_2), dan ligt de rechte vast:

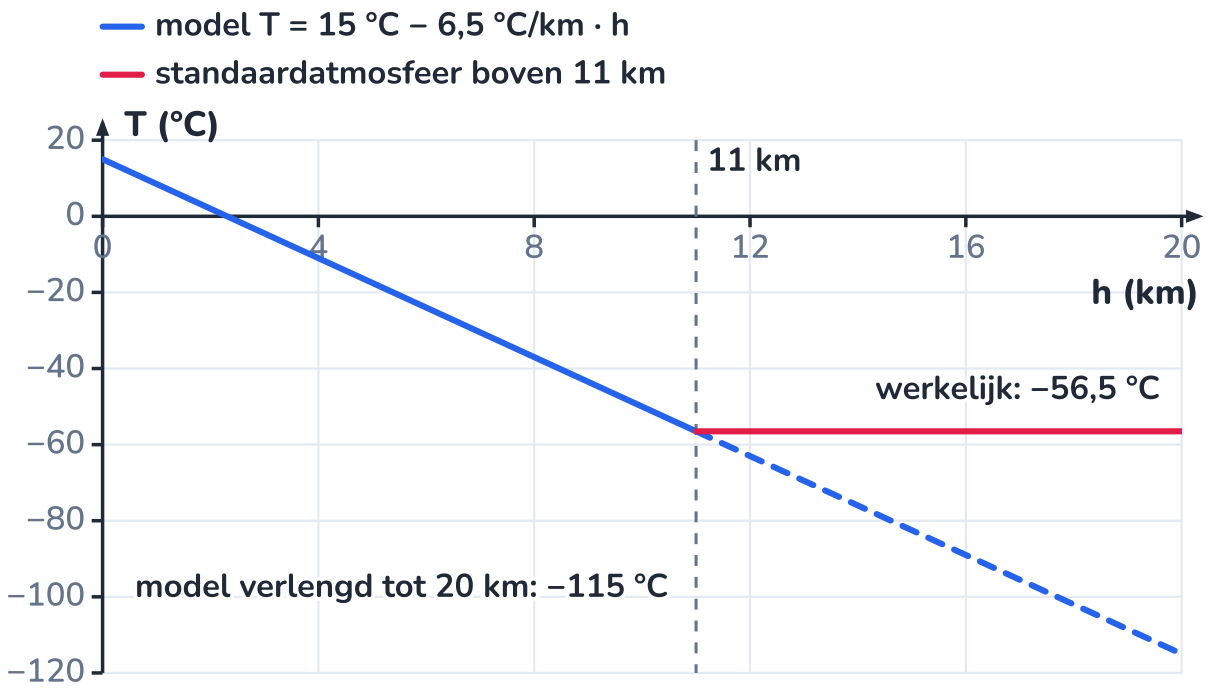
$$a = \frac{y_2 - y_1}{x_2 - x_1} \qquad \text{en} \qquad y - y_1 = a\,(x - x_1)$$

context	model	a	b
kosten	$K(n) = 11\,n + 260$	€ 11 per plantplaats	€ 260 vaste kost
afstand (ERB)	$x(t) = 6\,t + 150$	6 m/s, de snelheid	150 m, de startpositie
temperatuurschaal	$T_F = 1{,}8\,T_C + 32$	1,8 °F per °C	32 °F bij 0 °C
atmosfeer	$T(h) = -6{,}5\,h + 15$	−6,5 °C per km	15 °C op zeeniveau

Het laatste model is de **standaardatmosfeer** die de luchtvaart gebruikt: de lucht koelt 6,5 °C af per kilometer hoogte. Dat geldt tot ongeveer 11 km. Daarboven, in de stratosfeer, blijft de temperatuur tot 20 km gelijk op −56,5 °C. Het model 'verlengen' naar 20 km geeft −115 °C: 58 °C te koud. Een rechte gaat eeuwig door, de werkelijkheid niet.

Lineair betekent recht evenredig

Recht evenredig is het speciale geval $b = 0$: dan verdubbelt y als x verdubbelt. Bij $K = 11n + 260$ kosten 20 plantplaatsen € 480, en 40 plaatsen € 700, niet het dubbele. Wat wel verdubbelt, is de *toename*: van 0 naar 20 plaatsen komt er € 220 bij, van 0 naar 40 € 440.



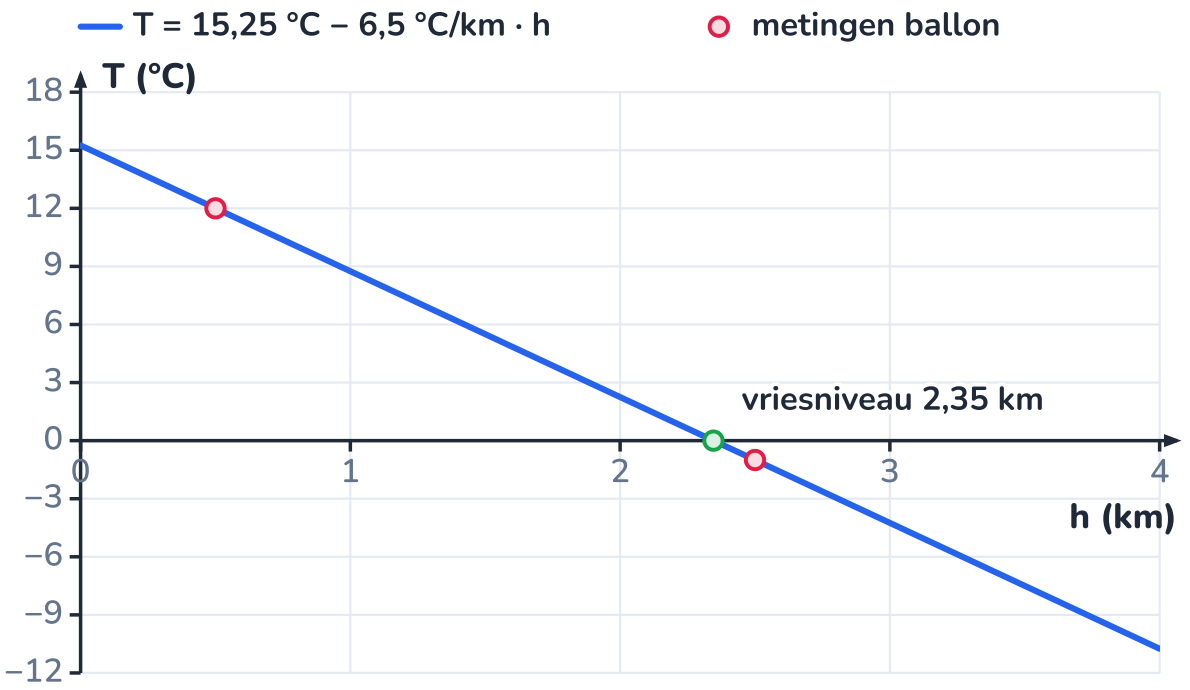
Tot 11 km beschrijft de rechte de standaardatmosfeer; daarboven buigt de werkelijkheid af en wordt de verlengde rechte onzin.

UITGEWERKT VOORBEELD · 4.6

Hoe hoog ligt het vriesniveau?

6/18

Een weerballon meet op 500 m hoogte 12 °C en op 2500 m hoogte −1 °C. Stel een lineair model $T(h)$ op, geef de betekenis van de parameters, en bereken op welke hoogte het vriest.



Twee metingen leggen de rechte vast; het nulpunt is het vriesniveau.

GEGEVEN

- $(h_1, T_1) = (0{,}5\,\text{km};\ 12\,^\circ\text{C})$
- $(h_2, T_2) = (2{,}5\,\text{km};\ -1\,^\circ\text{C})$

GEVRAAGD

- $T(h) = a\,h + b$
- de hoogte waar $T = 0\,^\circ\text{C}$

OPLOSSING

1. Richtingscoëfficiënt: $a = \dfrac{-1 - 12}{2{,}5 - 0{,}5} = \dfrac{-13\,^\circ\text{C}}{2{,}0\,\text{km}} = -6{,}5\,^\circ\text{C/km}$. Per kilometer hoger is het 6,5 °C kouder.
2. Snijpunt: vul een punt in. $12 = -6{,}5 \cdot 0{,}5 + b$, dus $b = 12 + 3{,}25 = 15{,}25\,^\circ\text{C}$: de temperatuur die het model aan de grond geeft.
3. Model: $T(h) = -6{,}5\,^\circ\text{C/km} \cdot h + 15{,}25\,^\circ\text{C}$. Controle met het tweede punt: $-6{,}5 \cdot 2{,}5 + 15{,}25 = -1{,}0$. Klopt.
4. Nulpunt: $-6{,}5\,h + 15{,}25 = 0$ geeft $h = \dfrac{15{,}25}{6{,}5} = 2{,}35\,\text{km}$.

ANTWOORD

$T(h) = 15{,}25 - 6{,}5\,h$ **(T in °C, h in km). Het vriesniveau ligt op ongeveer 2350 m. Het nulpunt van de functie is hier een echte grootheid, die weerdiensten dagelijks vermelden.**

Klopt dit? Dat het vriesniveau tussen de twee meethoogtes ligt, is een interpolatie: betrouwbaar. De waarde aan de grond (15,25 °C) is een kleine extrapolatie: bij een windstille nacht kan het aan de grond juist kouder zijn dan hoger (een temperatuurinversie), en dan faalt het model onderaan.

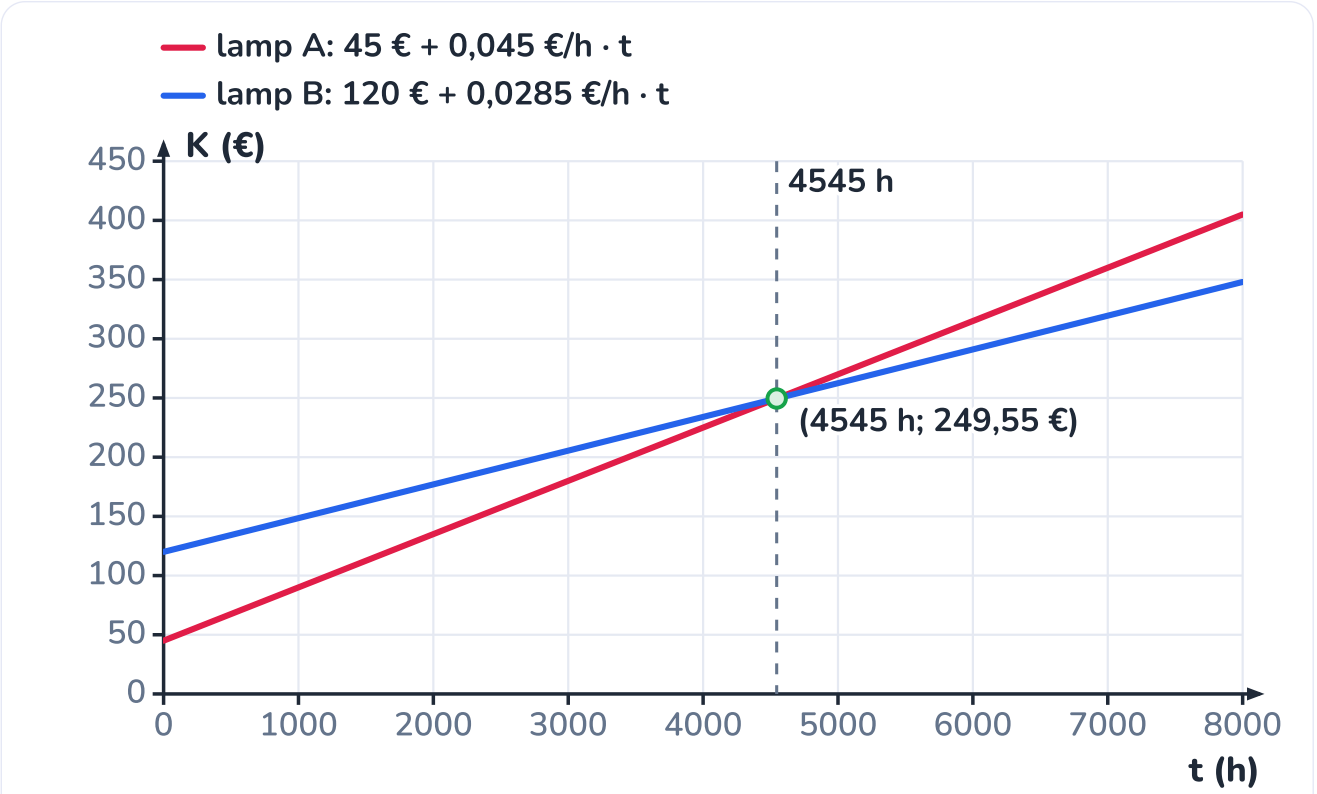
Een radiosonde meet op weg naar boven elke paar seconden temperatuur, druk en vochtigheid: zo kennen weerdiensten het vriesniveau.

UITGEWERKT VOORBEELD · 4.7

Twee groeilampen: waar de rechten elkaar snijden

7/18

Voor de kweekruimte van de klimaatmodule twijfel je tussen twee groeilampen met dezelfde lichtopbrengst. Lamp A kost € 45 en neemt 150 W op. Lamp B kost € 120 en neemt 95 W op. Reken met € 0,30 per kWh. Na hoeveel branduren is lamp B in totaal goedkoper? Hoeveel dagen is dat als de lamp 16 uur per dag brandt?



Links van het snijpunt ligt de rode rechte (A) lager, rechts de blauwe (B).

GEGEVEN

- A: € 45, 150 W; B: € 120, 95 W
- energieprijs € 0,30/kWh; 16 h per dag

GEVRAAGD

- $K_A(t)$ en $K_B(t)$
- het snijpunt en zijn betekenis

OPLOSSING

1. Energiekost per branduur: A neemt $0{,}150\,\text{kW}$ op, dus per uur $0{,}150\,\text{kWh}$ × € 0,30 per kWh = € 0,045. B: $0{,}095 \cdot 0{,}30 = 0{,}0285$, dus € 0,0285 per uur.
2. Twee lineaire modellen, met t in branduren: $K_A(t) = 0{,}045\,t + 45$ en $K_B(t) = 0{,}0285\,t + 120$ (in €).
3. Snijpunt: stel ze gelijk. $0{,}045\,t + 45 = 0{,}0285\,t + 120$, dus $0{,}0165\,t = 75$ en $t = 4545\,\text{h}$. De kost is dan $0{,}045 \cdot 4545 + 45 = 249{,}55$ euro voor elke lamp.
4. Omrekenen: $4545\,\text{h} / 16\,\text{h/dag} = 284$ dagen, ruim negen maanden.
5. Interpretatie: vóór het snijpunt is A goedkoper (lagere start), erna B (kleinere helling). Het snijpunt is het **kantelpunt**: de *terugverdientijd* van de duurdere, zuinigere lamp.

ANTWOORD

Na 4545 branduren, ongeveer 284 dagen bij 16 uur per dag, zijn beide lampen € 249,55 gekost. Daarna bespaart B € 0,0165 per uur, ongeveer € 96 per jaar.

Klopt dit? Eenheden: € gedeeld door €/h geeft h. En de grootte-orde: het verschil in aankoop (€ 75) gedeeld door het verschil in kost per uur (€ 0,0165) moet het snijpunt geven: $75 / 0{,}0165 = 4545$. Het model geldt zolang de energieprijs gelijk blijft en de lampen niet vroeger stuk gaan: allebei aannames.

De kwadratische functie: de parabool

Zodra een grootheid met het **kwadraat** van een andere groeit, krijg je een kwadratische functie. Drie contexten die je in dit hoofdstuk tegenkomt:

- **valbeweging**: $h(t) = h_0 + v_0\,t - \tfrac{1}{2}g\,t^2$;
- **remweg**: $d = \dfrac{v^2}{2\,a_{\text{rem}}}$, twee keer zo snel is vier keer zo ver;
- **oppervlakte**: een rechthoek met breedte x en lengte $4{,}8 - 2x$ heeft $A = x\,(4{,}8 - 2x)$.

KWADRATISCHE FUNCTIE

$$f(x) = a\,x^2 + b\,x + c \qquad (a \neq 0)$$

De grafiek is een **parabool**: een dalparabool (opening naar boven) als $a > 0$, een bergparabool als $a < 0$. $c = f(0)$ is het snijpunt met de verticale as.

De top. Een parabool is symmetrisch rond een verticale lijn, de **symmetrieas**. De top ligt erop, en is het laagste of hoogste punt:

TOP VAN DE PARABOOL

$$x_{\text{top}} = -\frac{b}{2a} \qquad y_{\text{top}} = f(x_{\text{top}})$$

De nulpunten. Waar is $f(x) = 0$? Bereken eerst de **discriminant** $D = b^2 - 4ac$. Is $D > 0$, dan zijn er twee nulpunten; is $D = 0$, dan raakt de top de as (één nulpunt); is $D < 0$, dan zijn er geen. De nulpunten liggen symmetrisch rond de top:

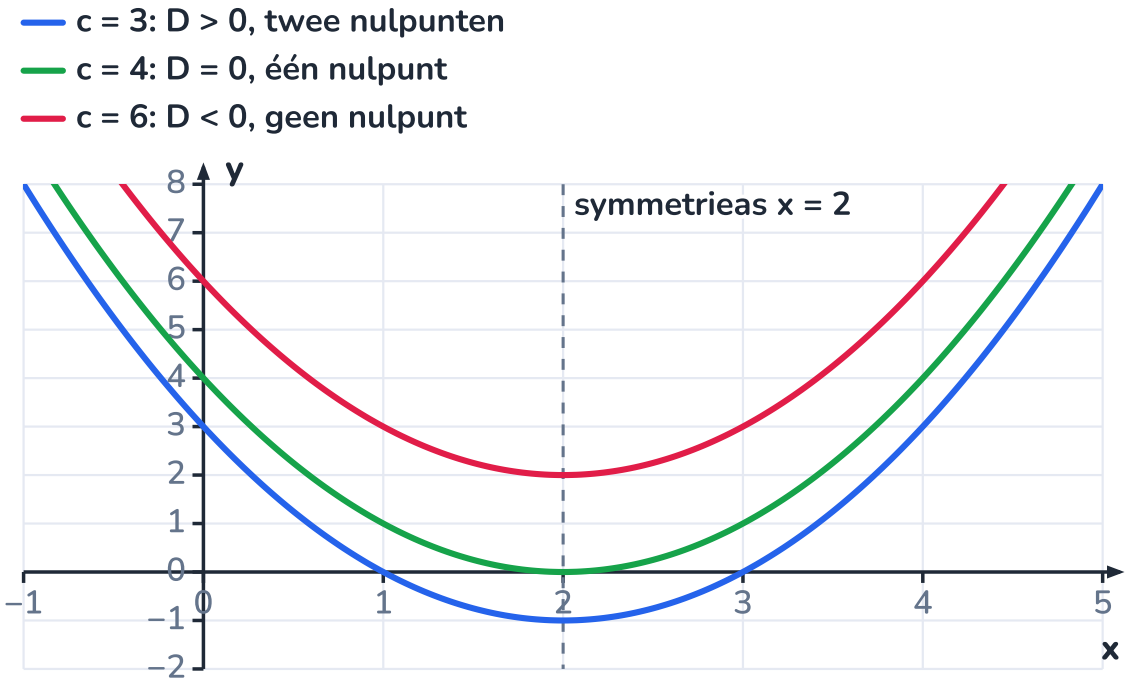
ABC-FORMULE

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a} \qquad D = b^2 - 4ac$$

Een kwadratische functie kan je ook schrijven in **topvorm** $f(x) = a\,(x - p)^2 + q$, met top (p, q), of, als ze nulpunten x_1 en x_2 heeft, in **productvorm** $f(x) = a\,(x - x_1)(x - x_2)$. Welke vorm je kiest, hangt af van wat je wil weten.

Gelijke tweede verschillen

Neem $f(x) = 3x^2$: de waarden bij $x = 0, 1, 2, 3$ zijn 0, 3, 12, 27; de eerste verschillen 3, 9, 15; de tweede verschillen 6, 6. Dat is altijd $2a$ bij stappen van 1 in x: bij Galilei was $a = 1$ en het tweede verschil 2.



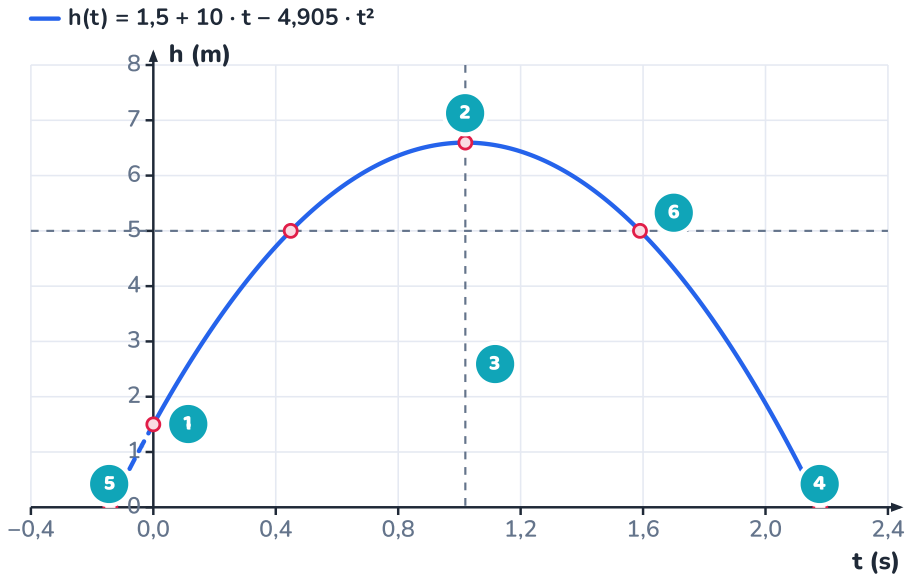
Zelfde a en b, dus dezelfde symmetrieas; c schuift de parabool op en neer, en de discriminant telt de nulpunten.

ONTLEED · 4.9

Een worp ontleed

9/18

Je gooit een bal recht omhoog. Hij verlaat je hand op 1,5 m hoogte met een snelheid van 10 m/s. Zonder luchtweerstand is zijn hoogte $h(t) = 1{,}5 + 10\,t - 4{,}905\,t^2$ (h in m, t in s). Elk bijzonder punt van de parabool heeft een betekenis. Wijs de nummers aan.



Volle lijn: het domein van de worp. Stippellijnen: de formule vóór het loslaten en na de landing.

1 Snijpunt met de verticale as

$h(0) = c = 1{,}5\,\text{m}$: de hoogte van je hand bij het loslaten. Het getal c in de formule is altijd de beginwaarde.

2 De top

$t_{\text{top}} = -\dfrac{b}{2a} = -\dfrac{10}{2 \cdot (-4{,}905)} = 1{,}02\,\text{s}$ en $h_{\text{top}} = h(1{,}02) = 6{,}60\,\text{m}$. Op dat ogenblik is de snelheid nul: de bal keert om. Het hoogste punt van de bal is de top van de functie.

3 Symmetrieas

Van 1,5 m omhoog naar de top duurt even lang (1,02 s) als van de top terug naar 1,5 m. Zonder luchtweerstand is de beweging omkeerbaar: de bal komt met dezelfde snelheid terug in je hand.

4 Het nulpunt dat telt

$D = 10^2 - 4 \cdot (-4{,}905) \cdot 1{,}5 = 129{,}4$, dus $t = \dfrac{-10 - \sqrt{129{,}4}}{-9{,}81} = 2{,}18\,\text{s}$: de bal raakt de grond (als je hem niet vangt). Dat is het einde van het domein.

5 Het nulpunt dat niet telt

De andere oplossing, $t = -0{,}14\,\text{s}$, ligt buiten het domein. Toch heeft ze een betekenis: een bal die 0,14 s eerder vanaf de grond vertrok met de juiste snelheid, zou dezelfde baan volgen.

6 Snijpunten met h = 5 m

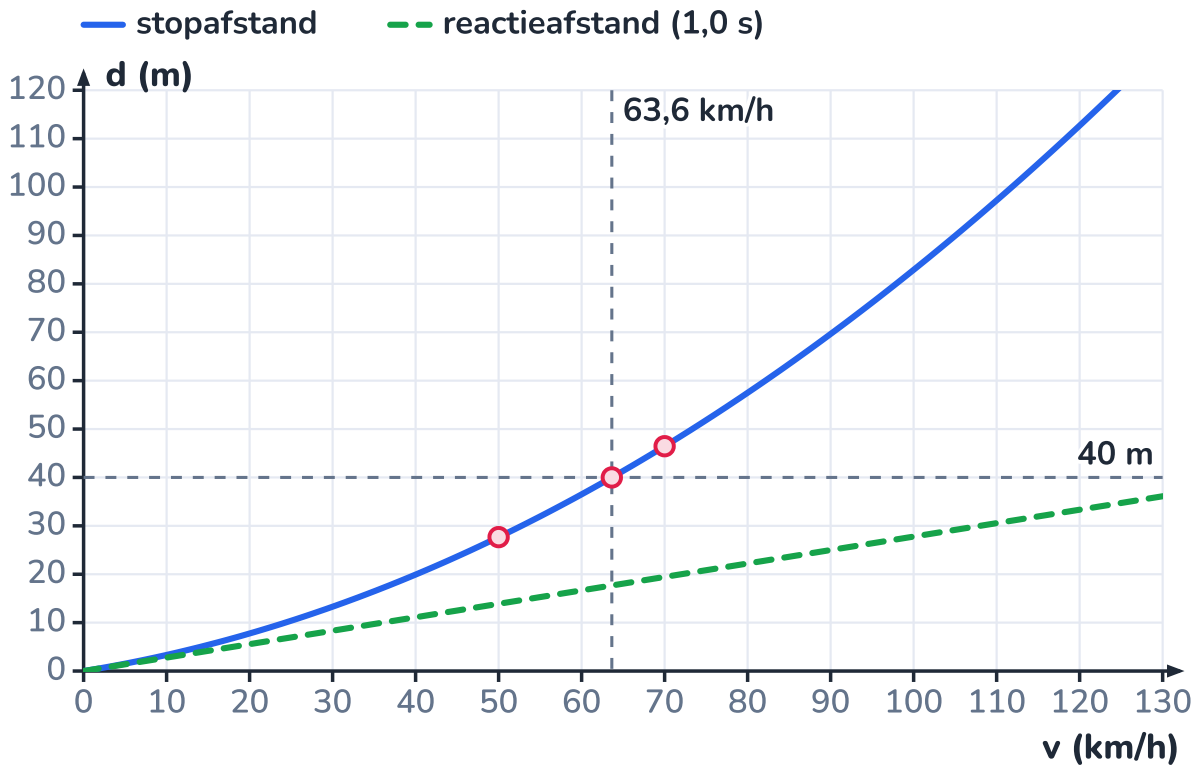
Los $1{,}5 + 10\,t - 4{,}905\,t^2 = 5$ op: $t = 0{,}45\,\text{s}$ (stijgend) en $t = 1{,}59\,\text{s}$ (dalend). De bal is 1,14 s boven 5 m. Een horizontale lijn snijdt een parabool twee, één of nul keer.

UITGEWERKT VOORBEELD · 4.10

Stopafstand: een parabool op de weg

10/18

Een bestuurder heeft een reactietijd van 1,0 s; zijn auto remt op een droge weg met een vertraging van 7,0 m/s² (de waarden uit opdracht 3.14). Bereken de stopafstand bij 50 km/h en bij 70 km/h. Hoe snel mag hij hoogstens rijden om binnen 40 m te stoppen?



De reactieafstand groeit lineair met v, de stopafstand kwadratisch.

GEGEVEN

- $t_r = 1{,}0\,\text{s}$, $a_{\text{rem}} = 7{,}0\,\text{m/s}^2$
- $v = 50\,\text{km/h}$ en $70\,\text{km/h}$; $d_{\max} = 40\,\text{m}$

GEVRAAGD

- $d(50)$, $d(70)$
- $v_{\max}$ voor $d = 40\,\text{m}$

OPLOSSING

1 Model: tijdens de reactietijd rijdt de auto aan constante snelheid ($v \cdot t_r$), daarna remt hij tot stilstand ($v^2/(2a_{\text{rem}})$, hoofdstuk 7). Samen: $d(v) = v\,t_r + \dfrac{v^2}{2\,a_{\text{rem}}} = 1{,}0\,v + \dfrac{v^2}{14}$, met v in m/s.

2 50 km/h = $50/3{,}6 = 13{,}9\,\text{m/s}$: $d = 13{,}9 + 13{,}9^2/14 = 13{,}9 + 13{,}8 = 27{,}7\,\text{m}$.

3 70 km/h = $19{,}4\,\text{m/s}$: $d = 19{,}4 + 27{,}0 = 46{,}5\,\text{m}$. 40 % sneller geeft 68 % meer stopafstand: het kwadraat van de remweg doet zich gelden.

4 Omgekeerd: $\dfrac{v^2}{14} + v - 40 = 0$. Met $a = 1/14$, $b = 1$, $c = -40$: $D = 1 + 4 \cdot \frac{1}{14} \cdot 40 = 12{,}43$ en $v = \dfrac{-1 \pm \sqrt{12{,}43}}{2/14} = 7 \cdot (-1 \pm 3{,}525)$.

5 Dat geeft $v = 17{,}7\,\text{m/s}$ of $v = -31{,}7\,\text{m/s}$. Een negatieve snelheid valt buiten het domein: $v_{\max} = 17{,}7\,\text{m/s} = 63{,}6\,\text{km/h}$.

ANTWOORD

Stopafstand 27,7 m bij 50 km/h en 46,5 m bij 70 km/h. Wie binnen 40 m wil stoppen, mag hoogstens 64 km/h rijden.

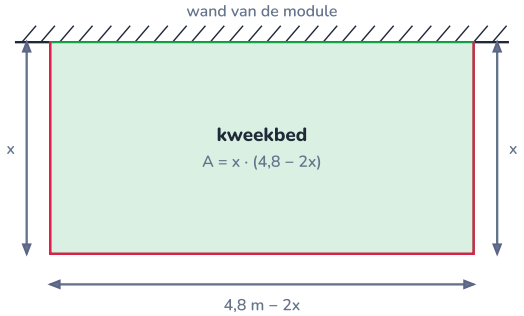
Klopt dit? Bij 50 km/h zijn reactie- en remafstand bijna even groot; bij hogere snelheid wint het kwadratische deel. Het model geldt voor een droge weg en een aandachtige bestuurder: op een natte weg is de vertraging kleiner, met een telefoon in de hand de reactietijd groter (opdracht 4.7).

UITGEWERKT VOORBEELD · 4.11

Habitat: het grootste kweekbed

11/18

In de klimaatmodule komt een rechthoekig kweekbed tegen een wand. Je hebt 4,8 m aluminiumprofiel voor de drie vrije zijden. Welke afmetingen geven de grootste oppervlakte? En welke afmetingen geven precies 2,5 m²?



rood: 4,8 m aluminiumprofiel op drie zijden

De wand spaart één zijde uit; het profiel loopt langs de andere drie.

GEGEVEN

- profiel voor drie zijden: $4{,}8\,\text{m}$
- de wand vormt de vierde zijde

GEVRAAGD

- $A(x)$ met domein
- de grootste oppervlakte
- x voor $A = 2{,}5\,\text{m}^2$

OPLOSSING

1 Noem de diepte (loodrecht op de wand) x. Twee zijden zijn x, de zijde langs de wand is wat overblijft: $4{,}8 - 2x$. Oppervlakte: $A(x) = x\,(4{,}8 - 2x) = -2x^2 + 4{,}8\,x$ (in m², x in m).

2 Domein: beide afmetingen positief: $x > 0$ en $4{,}8 - 2x > 0$, dus $0 < x < 2{,}4\,\text{m}$.

3 Bergparabool ($a = -2 < 0$), dus de top is een maximum: $x_{\text{top}} = -\dfrac{4{,}8}{2 \cdot (-2)} = 1{,}2\,\text{m}$. Lengte langs de wand: $4{,}8 - 2{,}4 = 2{,}4\,\text{m}$. $A = 1{,}2 \cdot 2{,}4 = 2{,}88\,\text{m}^2$.

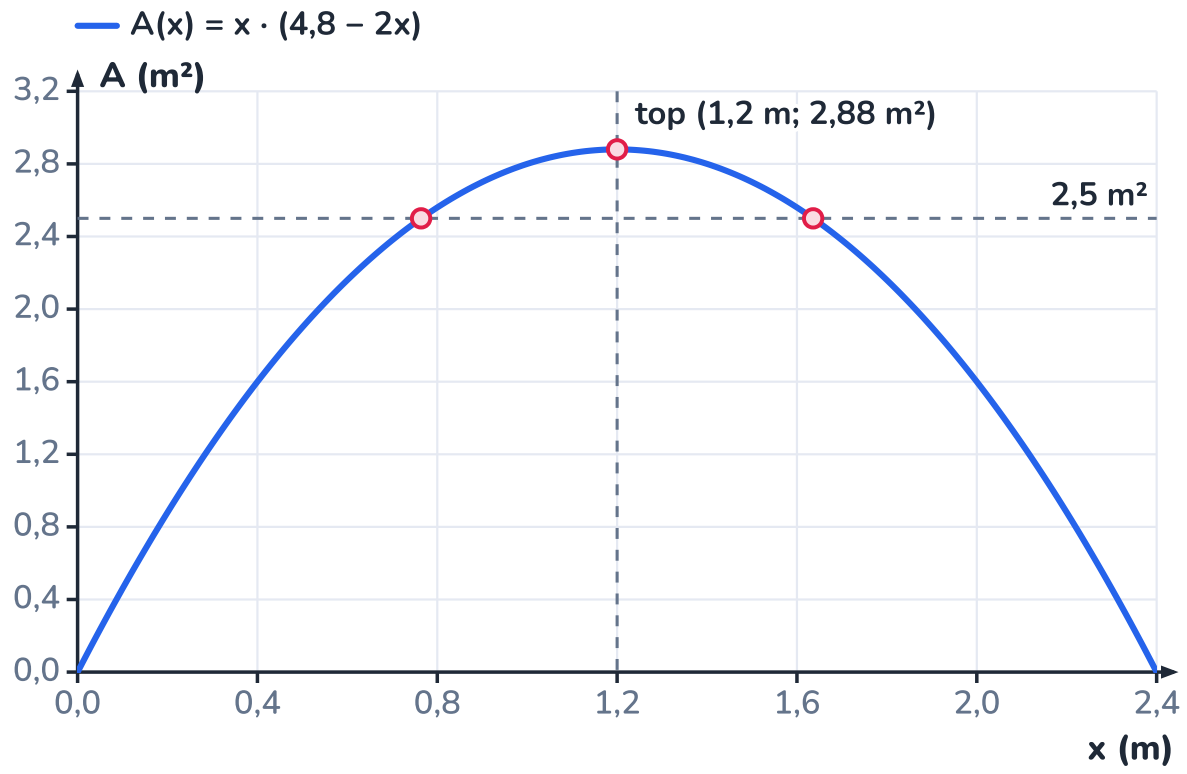
4 Voor $A = 2{,}5$: $-2x^2 + 4{,}8\,x - 2{,}5 = 0$. $D = 4{,}8^2 - 4 \cdot 2 \cdot 2{,}5 = 3{,}04$, $x = \dfrac{4{,}8 \pm 1{,}744}{4}$: $x = 0{,}76\,\text{m}$ of $x = 1{,}64\,\text{m}$.

5 Twee vormen dus: 0,76 m diep en 3,27 m lang, of 1,64 m diep en 1,53 m lang. Beide liggen in het domein.

ANTWOORD

Het grootste bed is 1,2 m diep en 2,4 m lang, 2,88 m². Een bed van 2,5 m² kan ondiep en lang (0,76 m × 3,27 m) of diep en kort (1,64 m × 1,53 m).

Klopt dit? Symmetrie: de twee oplossingen voor 2,5 m² liggen even ver links en rechts van de top (1,2 ± 0,44 m). Praktisch kies je niet blind de top: een bed dieper dan ongeveer 1,2 m is vanaf één kant niet meer te bereiken. Hier valt het wiskundige optimum toevallig samen met een bruikbare maat.



De oppervlakte als functie van de diepte: een bergparabool met de top bij 1,2 m.

UITLEG · 4.12

Snijpunten: waar twee modellen elkaar ontmoeten

12/18

Twee functies snijden elkaar waar ze **dezelfde uitvoer** geven. Algebraïsch stel je de voorschriften gelijk en los je op; grafisch lees je het punt af. Het snijpunt krijgt pas betekenis in de context: een kantelpunt tussen twee keuzes, het moment van inhalen, de snelheid waarbij een grens overschreden wordt.

Voorbeeld. Een auto staat stil aan een verkeerslicht. Op het moment dat het groen wordt, rijdt een fietser hem voorbij aan 6 m/s. De auto trekt op met 2,0 m/s². Posities vanaf het licht: fietser $x_f = 6\,t$

(lineair), auto $x_a = \frac{1}{2} \cdot 2{,}0 \cdot t^2 = t^2$ (kwadratisch, hoofdstuk 7). Gelijkstellen:

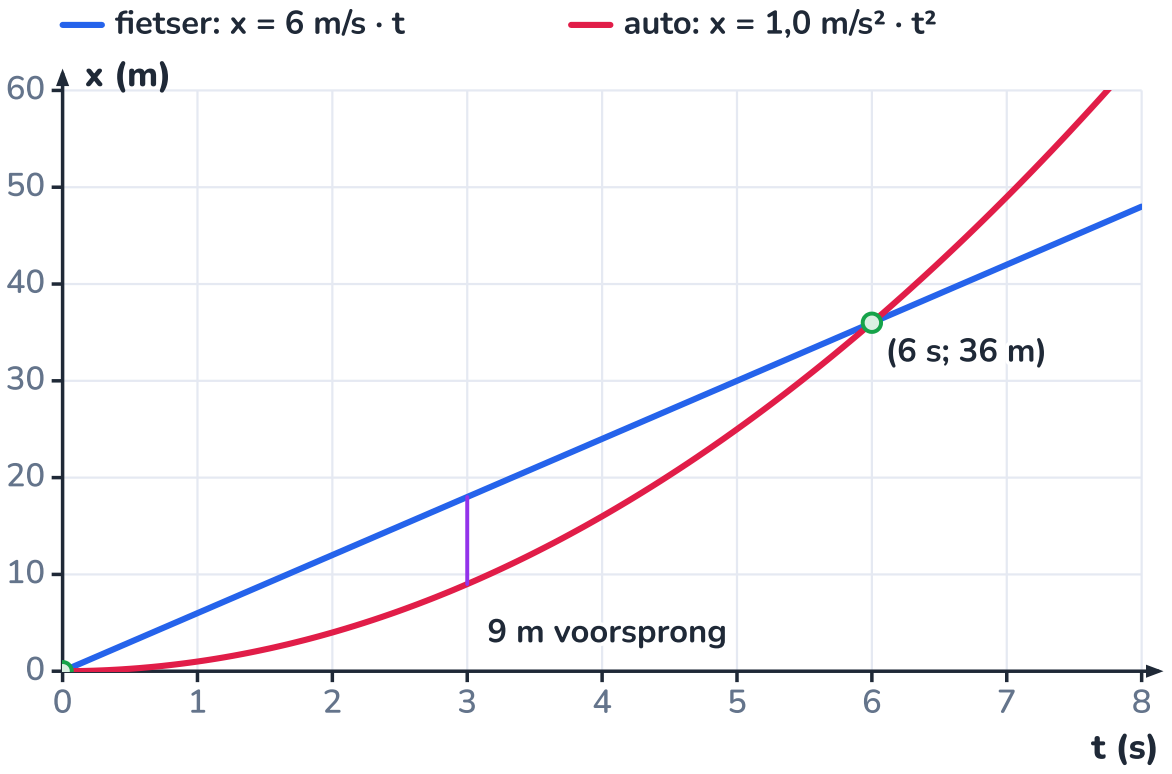
$$t^2 = 6\,t \;\Leftrightarrow\; t^2 - 6\,t = 0 \;\Leftrightarrow\; t\,(t-6) = 0 \;\Leftrightarrow\; t = 0 \text{ of } t = 6\,\text{s}$$

Twee snijpunten, twee betekenissen: bij $t = 0$ staan ze naast elkaar aan het licht, bij $t = 6\,\text{s}$ haalt de auto de fietser in, 36 m verder. De auto rijdt dan $2{,}0 \cdot 6 = 12\,\text{m/s}$ (43 km/h), het dubbele van de fietser.

Het **verschil** van twee functies is zelf een functie: de voorsprong van de fietser is $6\,t - t^2$, een bergparabool met top bij $t = 3\,\text{s}$: 9 m voorsprong. Het snijpunt van twee functies is altijd een nulpunt van hun verschil. Zo worden alle snijpuntproblemen nulpuntproblemen.

Een snijpunt is een ontmoeting

In een (x,t)-grafiek betekent een snijpunt dat twee voorwerpen op hetzelfde moment op dezelfde plaats zijn. In een kostengrafiek betekent het alleen dat twee keuzes even duur zijn. En twee grafieken met verschillende grootheden op de assen (een temperatuur en een kost) mag je niet laten snijden: dat snijpunt is een toeval van de schaal.



Snijpunten bij 0 s en 6 s; de verticale afstand tussen beide krommen is de voorsprong, maximaal 9 m bij 3 s.

UITLEG · 4.13

Welk model past bij de data?

13/18

Een model kiezen begint niet bij de trendlijnknop van een rekenblad. Drie vragen, in deze volgorde:

1. **Wat zegt de fysica?** Een stopafstand bevat een remweg met v^2; een kost met een vast en een variabel deel is lineair. Een model met een reden is sterker dan een model dat toevallig past.
2. **Wat zeggen de verschillen?** Meet je bij gelijke stappen in x, bereken dan de eerste verschillen Δy. Zijn ze (binnen de meetfout) constant: lineair. Zijn de tweede verschillen constant: kwadratisch. Is de *verhouding* van opeenvolgende waarden constant: exponentieel (hoofdstuk 5).

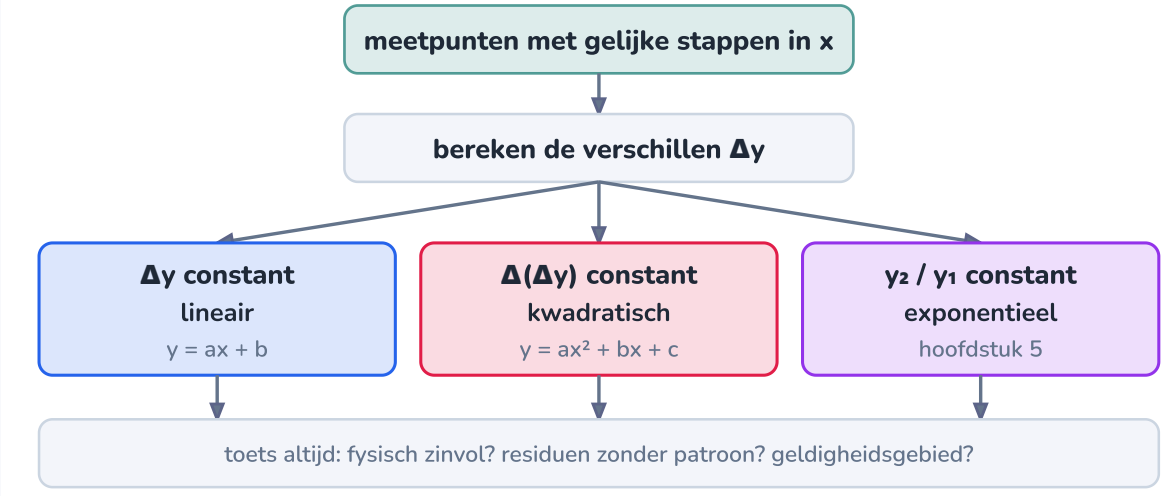
3. Wat zeggen de residuen? Fit het model en bekijk de residuen (hoofdstuk 3). Liggen ze willekeurig rond nul, dan past het. Vormen ze een boog, dan mist het model iets.

Drie meetreeksen als voorbeeld:

x	0	1	2	3	4	5
A: kaarslengte (cm), x in stappen van 10 min	20,0	18,6	17,3	15,9	14,6	13,2
eerste verschillen		−1,4	−1,3	−1,4	−1,3	−1,4
B: remafstand fiets (m) bij 10, 15, 20, 25, 30 km/h	1,29	2,89	5,14	8,04	11,57	
eerste verschillen		1,60	2,25	2,90	3,53	
tweede verschillen			0,65	0,65	0,63	
C: bacteriën per mL (×1000), x in uren	100	160	256	410	655	
eerste verschillen		60	96	154	245	
tweede verschillen			36	58	91	
verhouding y_2/y_1		1,60	1,60	1,60	1,60	

A: eerste verschillen ongeveer −1,35 cm per 10 min: lineair, een kaars die gelijkmatig opbrandt. B: eerste verschillen groeien, tweede verschillen constant (0,64 m): kwadratisch, zoals een remafstand hoort te zijn. C: de tweede verschillen groeien ook, maar elke waarde is 1,60 keer de vorige: geen parabool, maar exponentiële groei.

> ⚠ **Meer parameters is niet beter**
>
> Door drie punten gaat altijd een parabool, door zeven punten altijd een veelterm van graad zes. Zo'n model volgt de meetfouten, niet het verband, en voorspelt daarbuiten onzin (opdracht 4.14). Kies het eenvoudigste model dat de fysica toelaat en waarvan de residuen geen patroon tonen.



Verschillen zijn een snelle eerste test; de fysica en de residuen hebben het laatste woord.

PROEF · 4.14

Een knikker in een goot

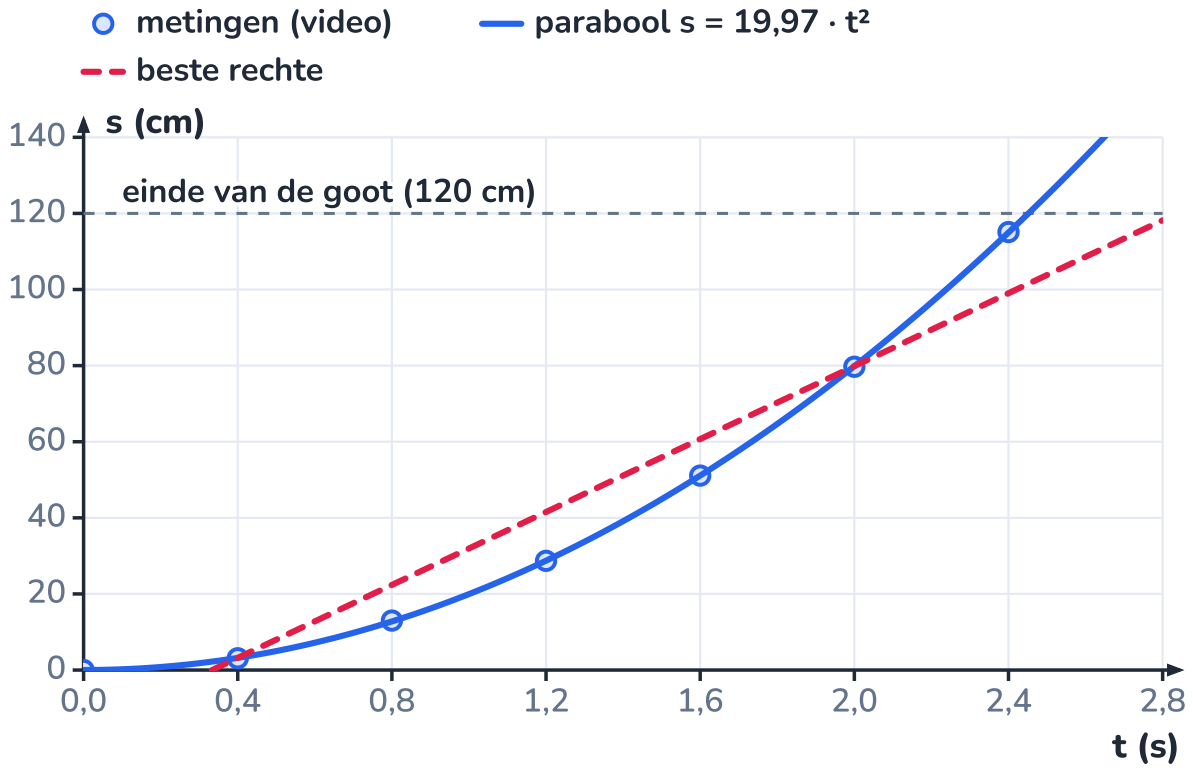
14/18

ONDERZOEKSVRAAG

Hoe hangt de afgelegde weg van een knikker in een licht hellende goot af van de tijd? Is het verband lineair of kwadratisch, en wat is de versnelling?

MATERIAAL

- pvc-goot of kabelgoot van ongeveer 1,2 m
- knikker
- rolmeter, stroken plakband om de 10 cm
- boeken of een blokje om de goot 4 à 5 cm op te tillen
- smartphone op een statief (video)



De parabool gaat door alle punten; de beste rechte mist systematisch: te hoog in het midden, te laag aan de uiteinden.

WERKWIJZE

1. Leg de goot schuin (één uiteinde 4 à 5 cm hoger) en plak een meetlint of stroken plakband om de 10 cm langs de goot.
2. Film de hele goot van opzij, met de camera stil en loodrecht op de goot. Houd de knikker bij de nulstreep en laat hem los zonder te duwen.
3. Speel de video beeld per beeld af. Noteer de positie van de knikker om de 0,4 s (bij 30 beelden per seconde: elk twaalfde beeld).
4. Bereken de eerste en de tweede verschillen van de posities.
5. Zet s uit tegen t en tegen t^2. Bepaal het model.

WAARNEMING

t (s)	0,0	0,4	0,8	1,2	1,6	2,0	2,4
s (cm)	0,0	3,1	13,0	28,7	51,1	79,7	115,1
eerste verschil		3,1	9,9	15,7	22,4	28,6	35,4
tweede verschil			6,8	5,8	6,7	6,2	6,8

De eerste verschillen groeien telkens met ongeveer 6,5 cm; de tweede verschillen liggen tussen 5,8 en 6,8 cm, zonder trend. s tegen t^2 geeft een rechte door de oorsprong.

VERKLARING

Constante tweede verschillen: het verband is kwadratisch, $s = k \cdot t^2$, zoals bij Galilei. Voor $s = \frac{1}{2} a\, t^2$ is het tweede verschil bij een tijdstap Δt gelijk aan $a\,(\Delta t)^2$. Het gemiddelde tweede verschil is 6,46 cm, dus $a = 6{,}46\,\text{cm}/(0{,}4\,\text{s})^2 = 40\,\text{cm/s}^2 = 0{,}40\,\text{m/s}^2$.

Geldigheidsgebied: het model geldt zolang de knikker in de goot rolt, dus tot 1,2 m. Bij een steilere helling gaat de knikker slippen en klopt de eenvoudige verhouding niet meer; bij een heel lange goot gaan wrijving en luchtweerstand meetellen.

CODE · 4.15

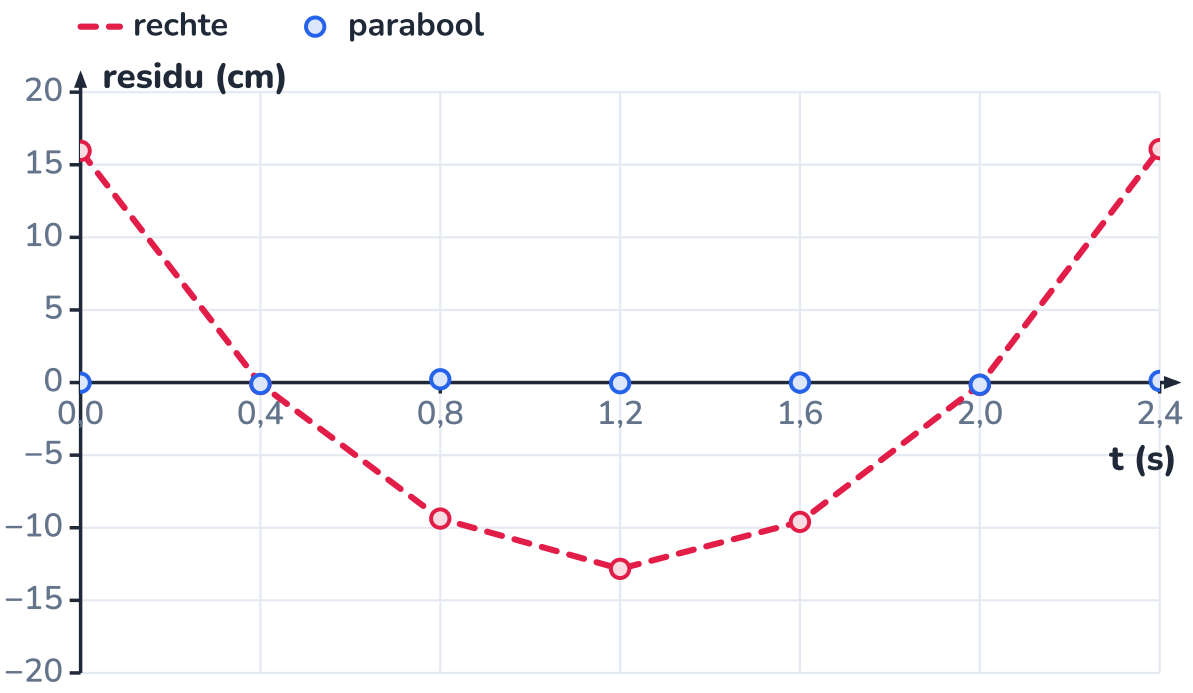
Twee modellen fitten in Python

15/18

In hoofdstuk 3 rekende je de kleinste-kwadratenrechte met de hand uit. De bibliotheek numpy doet hetzelfde voor elke veelterm: `np.polyfit(t, s, 1)` geeft de beste rechte, `np.polyfit(t, s, 2)` de beste parabool. Het programma vergelijkt beide op de metingen van de knikker, via hun residuen.

Python **modellen.py** 17 regels

```python
# modellen.py - een lineair en een kwadratisch model fitten op de knikkermetingen
import numpy as np

t = np.array([0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4])          # tijd (s)
s = np.array([0.0, 3.1, 13.0, 28.7, 51.1, 79.7, 115.1])    # afgelegde weg (cm)

for graad in (1, 2):
    coef = np.polyfit(t, s, graad)       # kleinste kwadraten; hoogste macht eerst
    model = np.poly1d(coef)
    residu = s - model(t)
    print(f"graad {graad}: coefficienten", "  ".join(f"{c:.2f}" for c in coef))
    print("   residuen (cm):", "  ".join(f"{r:+.1f}" for r in residu))
    print(f"   som van de kwadraten: {np.sum(residu**2):.1f} cm2")

a2, a1, a0 = np.polyfit(t, s, 2)
print(f"versnelling a = 2 * {a2:.2f} cm/s2 = {2 * a2 / 100:.3f} m/s2")
print(f"voorspelling bij t = 3,0 s: {np.polyval([a2, a1, a0], 3.0):.0f} cm")
```



De residuen van de rechte vormen een boog; die van de parabool liggen op nul.

Uitvoer (gecontroleerd door het programma te draaien):

```
graad 1: coefficienten 47.91  -15.96
   residuen (cm): +16.0  -0.1  -9.4  -12.8  -9.6  -0.2  +16.1
   som van de kwadraten: 857.7 cm2
graad 2: coefficienten 19.97  -0.02  0.01
   residuen (cm): -0.0  -0.1  +0.2  -0.0  -0.0  -0.2  +0.1
   som van de kwadraten: 0.1 cm2
versnelling a = 2 * 19.97 cm/s2 = 0.399 m/s2
voorspelling bij t = 3,0 s: 180 cm
```

Lees de uitvoer als een onderzoeker. De rechte heeft residuen tot 16 cm met een duidelijk patroon (plus, min, plus): verworpen. De parabool heeft residuen van hoogstens 0,2 cm, kleiner dan de afleesfout van de video. Haar coëfficiënten $b = -0{,}02$ en $c = 0{,}01$ zijn praktisch nul, zoals de fysica voorspelt voor een knikker die vanuit stilstand bij $s = 0$ vertrekt. De voorspelling bij 3,0 s (180 cm) is een extrapolatie *voorbij het einde van de goot*: onzin, hoe goed de fit ook is. (Het −0.0 in de uitvoer is een heel klein negatief getal dat na afronden nul wordt.)

UITLEG · 4.16

Het geldigheidsgebied van een model

16/18

Elk model is een vereenvoudiging, en elke vereenvoudiging heeft een grens. Het **geldigheidsgebied** is het gebied waarin het model getoetst is en goed genoeg werkt voor wat je ermee wil doen. Binnen dat gebied voorspellen heet **interpoleren**; daarbuiten **extrapoleren**, en dat doe je alleen met een reden.

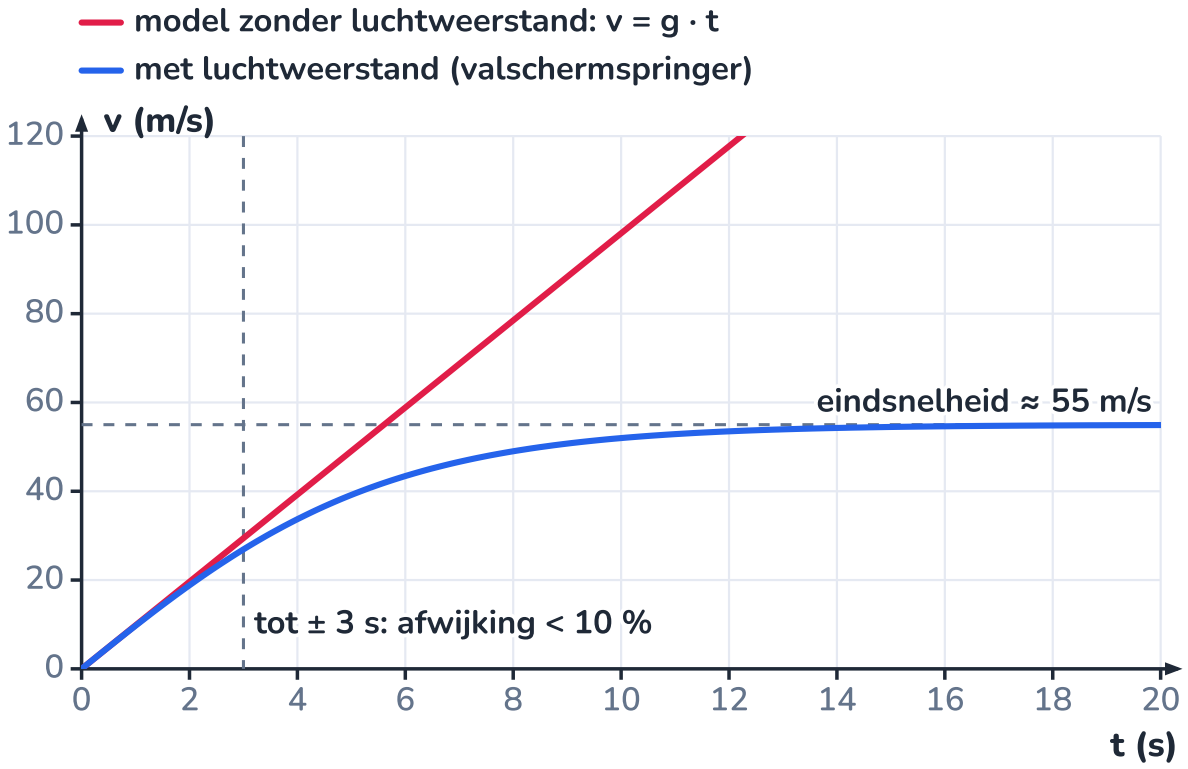
Een valschermspringer. Zonder luchtweerstand is $v = g \cdot t$, een rechte. De eerste seconden klopt dat goed, want bij lage snelheid is de luchtweerstand klein. Maar ze groeit met het kwadraat van de snelheid, tot ze even groot is als het gewicht: dan versnelt de springer niet meer. Buik naar beneden gebeurt dat rond 55 m/s, ongeveer 200 km/h. De rechte voorspelt na 60 s vrije val 589 m/s, sneller dan het geluid. In de grafiek zie je waar het model 'breekt': na 3 s wijkt het al bijna 10 % af, na 6 s 35 %.

Vier bronnen van grenzen, die je bij elk model nagaat:

- **verwaarloosde effecten** die groter worden: luchtweerstand, wrijving, warmteverlies;
- **materiaal- of systeemgrenzen**: een veer die blijvend vervormt (hoofdstuk F11), een goot die ophoudt, een batterij die leeg is;
- **veranderende omstandigheden**: een natte weg, een andere energieprijs, de tropopauze op 11 km;
- **het meetbereik**: wat je niet gemeten hebt, weet je niet.

Een formule geldt overal

Een formule in een boek is geen natuurwet zonder voorwaarden. $v = g\,t$ is 'vrije val zonder luchtweerstand', $d = v^2/(2a)$ is 'remmen met constante vertraging'. Wie de voorwaarden vergeet, rekent correct aan een verkeerde werkelijkheid.



De rechte v = g · t (rood) beschrijft alleen de eerste seconden; daarna nadert de echte snelheid (blauw) een eindsnelheid.

NASLAG · 4.17

Overzicht: functies en modellen

17/18

lineaire functie

$$f(x) = a\,x + b$$

symbool	betekenis	eenheid
a	richtingscoëfficiënt: verandering per eenheid x	eenheid y / eenheid
b	beginwaarde f(0)	eenheid y

rechte door twee punten

$$a = \frac{y_2 - y_1}{x_2 - x_1} \qquad y - y_1 = a\,(x - x_1)$$

kwadratische functie

$$f(x) = a\,x^2 + b\,x + c$$

symbool	betekenis	eenheid
a	opening (a > 0: dal, a < 0: berg) en steilheid	eenheid y / (eenheid x)²
c	beginwaarde f(0)	eenheid y

top en symmetrieas

$$x_{\text{top}} = -\frac{b}{2a} \qquad y_{\text{top}} = f(x_{\text{top}})$$

discriminant en nulpunten (abc-formule)

$$D = b^2 - 4ac \qquad x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

topvorm en productvorm

$$f(x) = a\,(x-p)^2 + q \qquad f(x) = a\,(x-x_1)(x$$

verticale worp zonder luchtweerstand

$$h(t) = h_0 + v_0\,t - \tfrac{1}{2}g\,t^2$$

symbool	betekenis	eenheid
h_0	beginhoogte	m
v_0	beginsnelheid (omhoog positief)	m/s
g	valversnelling, 9,81	m/s²

stopafstand

$$d = v\,t_r + \frac{v^2}{2\,a_{\text{rem}}}$$

symbool	betekenis	eenheid
v	snelheid	m/s
t_r	reactietijd	s
a_{rem}	remvertraging	m/s²

vraag in de context	wiskundig	hoe
wanneer is … nul? (leeg, op de grond, geen winst)	nulpunt	$f(x) = 0$ oplossen
wat is het meeste / minste? (hoogste punt, grootste oppervlakte)	top	$x = -b/(2a)$
wanneer zijn twee keuzes even duur? wanneer haalt A B in?	snijpunt	$f(x) = g(x)$ oplossen
wanneer is … groter dan een grens?	snijpunt met een horizontale lijn	$f(x) =$ grens oplossen, dan kijken welke kant
wat is de beginwaarde?	snijpunt met de verticale as	$f(0)$

data bij gelijke stappen in x	model	voorbeeld
eerste verschillen constant	lineair	kosten, ERB, kaars
tweede verschillen constant (gelijk aan $2a\,(\Delta x)^2$)	kwadratisch	val, remweg, oppervlakte
verhouding opeenvolgende waarden constant	exponentieel	bacteriën, afkoelen (hoofdstuk 5)

controlelijst bij elk model
invoer en uitvoer benoemd, met eenheid
domein: wiskundig én fysisch; punten of een doorlopende lijn?
parameters met eenheid en betekenis
oplossingen buiten het domein geschrapt, met uitleg
geldigheidsgebied: welke aannames, welk meetbereik?

HET GEHEEL · 4.18

Waar functies en modellen terugkomen

18/18

Een functie is de taal waarin je in de rest van StemExpert voorspelt. De rechte en de parabool zijn de eerste twee families; in hoofdstuk 5 komt de exponentiële functie erbij, in hoofdstuk 6 de goniometrische. In Expert leer je met afgeleiden de top van elke functie vinden, niet alleen van een parabool, en bouw je modellen uit veranderingen in plaats van uit formules.

F4 **Tabellen, grafieken en verbanden**

F14 **Beweging en snelheid**

I3 **Verbanden in data**

Tabel en grafiek, recht evenredig, interpoleren en extrapoleren: de basis van elke functie.

De (s,t)-grafiek van een eenparige beweging is een lineaire functie; de snelheid is haar helling.

De rechte uit metingen, residuen en lineariseren: hier gebruikt om een model te kiezen en te toetsen.

I5 Groei en verval: exponentiële modellen

Als de verhouding van opeenvolgende waarden constant is: groei en verval.

I7 Beweging beschrijven

ERB en EVRB: de lineaire en de kwadratische functie van de tijd, met de stopafstand als toepassing.

I9 Arbeid, energie en vermogen

Kinetische energie groeit met v^2: waarom de remweg kwadratisch is.

I28 Functies, lijsten en data in Python

Functies in Python met parameters en returnwaarden, en data inlezen om modellen te fitten.

E1 Afgeleiden: verandering meten

De helling van een kromme in elk punt; de top is waar die helling nul is.

E2 Optimaliseren

Het grootste kweekbed was een eerste optimalisatieprobleem; met afgeleiden los je er elk op.

E41 Van data naar model: regressie

Modellen met meer parameters, overfitting en validatie: waarom een zesdegraadsveelterm bedriegt.

Habitat

Voor de klimaat- en energiemodule maak je in opdracht 4.12 het kostenmodel van het hydroponiesysteem: zelf bouwen of kopen, met het kantelpunt en de kost per plantplaats. Het kweekbed van 2,88 m² uit het voorbeeld is een bouwsteen voor hoofdstuk 36 (hydroponie) en het integratieproject in hoofdstuk 40.

SAMENVATTING

In het kort

- Een functie koppelt aan elke invoer precies één uitvoer; je toont ze als tabel, grafiek of formule. In een grafiek: elke verticale lijn snijdt hoogstens één keer.
- Het domein is wat je mag invullen, het bereik wat eruit komt. In een model begrenst de fysica het domein vaak sterker dan de formule (geen negatieve tijd, geen halve plantplaats).
- De lineaire functie $f(x) = ax + b$ heeft een vaste start b en een vaste stap a (met eenheid); twee punten leggen ze vast.
- De kwadratische functie $f(x) = ax^2 + bx + c$ is een parabool met top bij $x = -b/(2a)$; de discriminant $D = b^2 - 4ac$ telt de nulpunten, de abc-formule geeft ze.
- Top, nulpunten en snijpunten krijgen betekenis in de context: hoogste punt, landing, kantelpunt, inhalen. Oplossingen buiten het domein schrap je, met uitleg.
- Een model kies je met de fysica, de verschillen (constante eerste verschillen: lineair; constante tweede verschillen: kwadratisch) en de residuen.
- Elk model heeft een geldigheidsgebied. Interpoleren binnen het getoetste gebied is veilig, extrapoleren alleen met een reden.

BEGRIPPEN

Wat je nu kent

functie

domein

Voorschrift dat aan elke toegelaten invoer x precies één uitvoer $f(x)$ koppelt.

De verzameling invoerwaarden waarvoor een functie (of een model) gebruikt mag worden; in een context vaak kleiner dan wat de formule toelaat.

bereik

De verzameling uitvoerwaarden die de functie op haar domein aanneemt.

lineaire functie

Functie $f(x) = ax + b$: gelijke stappen in x geven gelijke stappen in y; de grafiek is een rechte.

kwadratische functie

Functie $f(x) = ax^2 + bx + c$ met $a \neq 0$; de grafiek is een parabool.

top van een parabool

Het hoogste (bij $a < 0$) of laagste (bij $a > 0$) punt, bij $x = -b/(2a)$; de symmetrieas gaat erdoor.

nulpunt

Invoerwaarde waarvoor de functie nul is; in de grafiek een snijpunt met de horizontale as.

discriminant

Het getal $D = b^2 - 4ac$ dat zegt hoeveel nulpunten een kwadratische functie heeft: twee, één of geen.

snijpunt van twee functies

Punt waar twee functies dezelfde uitvoer geven; je vindt het door de voorschriften gelijk te stellen.

stopafstand

Reactieafstand plus remafstand: $d = v\,t_r + v^2/(2a_{\text{rem}})$, kwadratisch in de snelheid.

geldigheidsgebied

Het gebied waarin een model getoetst is en de werkelijkheid goed genoeg beschrijft.

tweede verschillen

De verschillen van de verschillen van opeenvolgende y-waarden bij gelijke stappen in x; constant bij een kwadratisch verband.

HANDBOEK · HOOFDSTUK 5

Groei en verval: exponentiële modellen

Vermenigvuldigen in plaats van optellen: van bacteriën en algen tot afkoelende thee en vervallende kernen

Sommige dingen groeien niet met een vaste hoeveelheid per stap, maar met een vaste **factor**: bacteriën, algen, het aantal transistoren op een chip. Andere dalen zo: de temperatuur van thee boven die van de kamer, de spanning over een condensator, het aantal radioactieve kernen. In dit hoofdstuk leer je rekenen met groeifactoren en procenten, verdubbelings- en halveringstijden afleiden met logaritmen, en een logaritmische schaal lezen. En je ontdekt waarom exponentiële groei onze intuïtie zo vaak te snel af is.

8× uitleg | 5× uitgewerkt voorbeeld | 1× proef | 1× naslag | 1× verhaal | 1× het geheel | 1× code | ± 9 lestijden

W Wiskundig modelleren | B Leven en systemen | E Energie

Zelfde voeding, zelfde warmte: in het doorzichtige reservoir verdubbelen de algen elke dag, in het lichtdichte niet.

Na dit hoofdstuk

- Je kan een groeifactor omzetten in een procentuele groei of afname en omgekeerd, ook naar een andere tijdseenheid.
- Je kan een exponentieel model N = N0 · g^t opstellen uit een beschrijving of uit twee metingen, en het herkennen aan constante verhoudingen.
- Je kan met logaritmen een exponent terugrekenen en de formule voor de verdubbelingstijd t2 = log 2 / log g afleiden en gebruiken.
- Je kan bacteriegroei, afkoeling, het ontladen van een condensator en radioactief verval met een exponentieel model beschrijven, met de juiste beginwaarde en eindwaarde.
- Je kan uitleggen waarom exponentiële groei lineaire groei altijd inhaalt en waarom ze in het begin misleidt.
- Je kan waarden aflezen op een logaritmische schaal en er een groeifactor en een verdubbelingstijd uit bepalen.
- Je kan het geldigheidsgebied van een exponentieel model benoemen: waar groei begrensd wordt en waar de aannames niet meer gelden.

UITLEG · 5.1

Wat je al weet, en wat nu komt

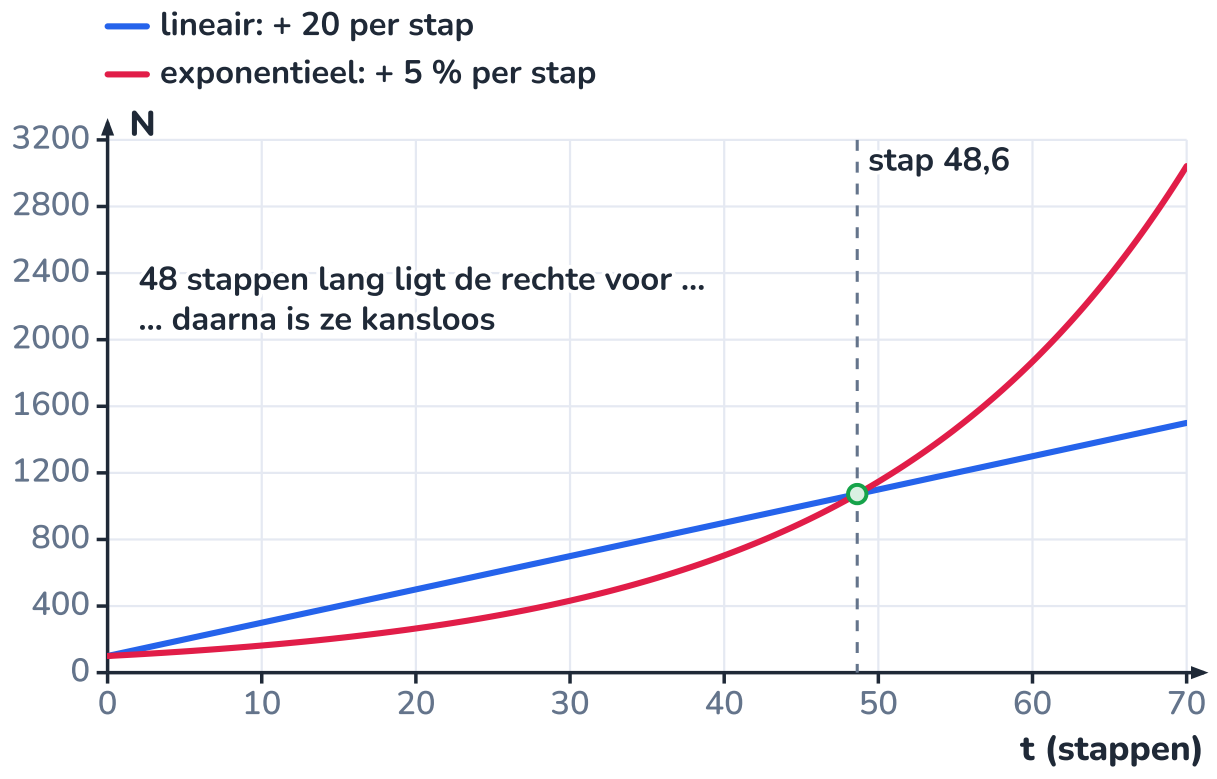
Je rekent al lang met **procenten**: 20 % korting, 6 % btw, 3 % meer leerlingen dan vorig jaar. In hoofdstuk 4 leerde je een model kiezen: constante eerste verschillen wijzen op een lineaire functie, constante tweede verschillen op een kwadratische. En er bleef een meetreeks over, de bacteriën, waarin niet de verschillen maar de **verhoudingen** constant waren: 100, 160, 256, 410, 655, telkens × 1,60.

Dat is het kenmerk van dit hoofdstuk. Een lineaire functie telt per stap **hetzelfde getal op**, een exponentiële functie **vermenigvuldigt met hetzelfde getal**. In de grafiek lijkt dat onschuldig: een groei van 5 % per stap blijft 48 stappen lang achter bij een vaste groei van 20 per stap. Daarna is het voorbij. Je leert in dit hoofdstuk

- rekenen met de **groeifactor** en met procenten, ook als de tijdseenheid verandert;
- de **verdubbelingstijd** en de **halveringstijd**, en hoe je ze afleidt met **logaritmen**;
- vier toepassingen: bacteriën, afkoelen, een condensator die ontlaadt, radioactief verval;
- een **logaritmische schaal** lezen, waarop een exponentiële functie een rechte wordt.

Machten opfrissen

$2^3 = 2 \cdot 2 \cdot 2 = 8$; $g^0 = 1$ voor elke g; $g^{-1} = 1/g$; $10^{-3} = 0{,}001$; $g^{1/2} = \sqrt{g}$; $(g^a)^b = g^{a \cdot b}$. Die laatste regel gebruik je om een groeifactor per dag om te rekenen naar een groeifactor per uur.



Dezelfde start: de rechte krijgt er vast 20 bij, de exponentiële kromme 5 % van wat er al is. Na stap 48,6 is de kromme voorgoed voor.

VERHAAL · 5.2

Een schaakbord vol rijst

Een oude legende, uit India of Perzië, vertelt over de uitvinder van het schaakspel. De koning is zo opgetogen over het spel dat de uitvinder zelf zijn beloning mag kiezen. Zijn vraag lijkt bescheiden: één rijstkorrel op het eerste veld van het schaakbord, twee op het tweede, vier op het derde, en zo telkens het dubbele, tot en met het 64ste veld. De koning lacht en laat een zak rijst brengen.

Reken mee, met 25 mg per korrel:

veld	korrels op dat veld	massa (bij 25 mg per korrel)
1	1	25 mg
10	$2^9 = 512$	13 g, een klein handje
20	$2^{19} = 524\,288$	13 kg, een zak
32	$2^{31} \approx 2{,}1 \cdot 10^9$	54 ton, twee vrachtwagens
40	$2^{39} \approx 5{,}5 \cdot 10^{11}$	14 000 ton, een vrachtschip
64	$2^{63} \approx 9{,}2 \cdot 10^{18}$	230 miljard ton
samen	$2^{64} - 1 \approx 1{,}8 \cdot 10^{19}$	**460 miljard ton**

De eerste helft van het bord, tot veld 32, vraagt samen ongeveer 107 ton: veel, maar een koning kan dat betalen. De tweede helft is een ander verhaal. Alles samen is het ongeveer 460 miljard ton rijst. De hele wereld oogst vandaag ongeveer 520 miljoen ton (gepelde) rijst per jaar: het schaakbord vraagt bijna 900 jaar wereldoogst.

Het verhaal is verzonnen, de rekening niet. Elke grootheid die met een vaste factor groeit, gedraagt zich zo: lang onschuldig, dan onbetaalbaar. Het omgekeerde geldt ook: een grootheid die per stap halveert, is na 64 stappen nog maar $1/2^{64} \approx 5 \cdot 10^{-20}$ van het begin.

Waarom dit verhaal ertoe doet

Op veld 64 liggen meer korrels dan op alle 63 vorige velden samen (2^{63} tegenover $2^{63} - 1$). Bij verdubbeling is de laatste stap altijd groter dan de hele geschiedenis ervoor. Dat is de reden waarom exponentiële problemen, van algen tot epidemieën, pas opvallen als het bijna te laat is.

Het eerste rijtje van het bord is een handvol korrels; de rest van het bord is een ander verhaal.

UITLEG · 5.3

Groeifactor en procentuele groei

3/18

Groeit een grootheid per tijdstap met p procent, dan wordt ze per stap vermenigvuldigd met de **groeifactor** $g = 1 + p/100$. Na t stappen:

EXPONENTIEEL MODEL

$$N(t) = N_0 \cdot g^t \qquad g = 1 + \frac{p}{100}$$

N_0 is de beginwaarde (bij $t = 0$), g de groeifactor per tijdstap. Groei: $g > 1$. Afname: $0 < g < 1$, met p negatief. De eenheid van t hoort bij g: 'per uur', 'per jaar'.

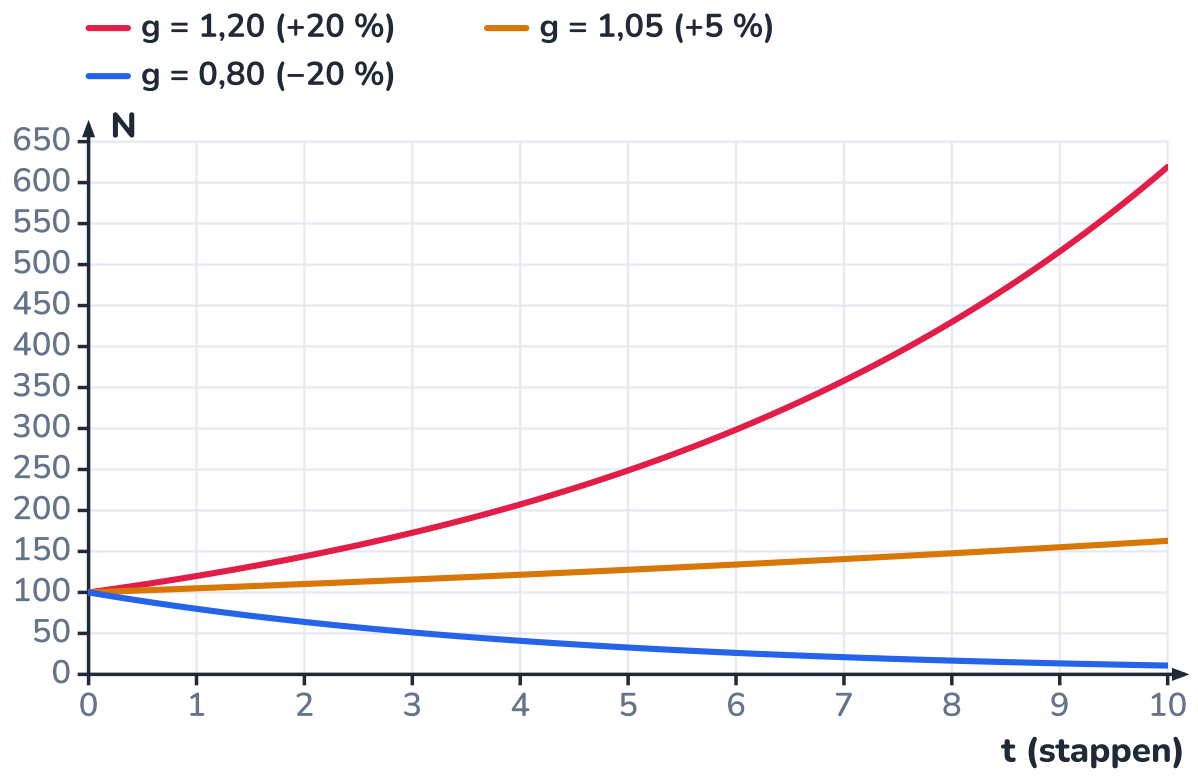
verandering per stap	groeifactor g
+3 %	1,03
+100 % (verdubbelen)	2
−20 %	0,80
−50 % (halveren)	0,5
+700 % (acht keer zoveel)	8

Andere tijdseenheid. Is de groeifactor per uur g_{u}, dan is hij per dag $g_{\text{d}} = g_{\text{u}}^{24}$. Omgekeerd: $g_{\text{u}} = g_{\text{d}}^{1/24}$. Groeit iets met 10 % per dag, dan is de groeifactor per week $1{,}1^7 = 1{,}95$: bijna het dubbele, dus +95 %, niet +70 %.

Procenten mag je optellen

+10 % per dag is niet +70 % per week, want elke dag rekent de 10 % op een groter getal. En eerst +20 % en dan −20 % brengt je niet terug bij het begin: $1{,}20 \cdot 0{,}80 = 0{,}96$, dus 4 % minder. Werk altijd met groeifactoren: die vermenigvuldig je.

Herkennen doe je aan de verhoudingen: bij gelijke stappen in t is $N(t+1)/N(t)$ constant (en gelijk aan g). In een grafiek: een kromme die steeds steiler wordt (groei) of steeds trager daalt en nooit nul wordt (afname).



Na 10 stappen: 619 bij +20 %, 163 bij +5 %, en 10,7 bij −20 %. De dalende kromme nadert nul, maar raakt ze nooit.

UITGEWERKT VOORBEELD · 5.4

Bacteriën in een voedingsbodem

4/18

Bij 37 °C, in een rijke voedingsbodem, verdubbelt de darmbacterie *E. coli* ongeveer elke 20 minuten. Een buisje bevat bij de start 1000 bacteriën per mL. Stel het model op met t in uren. Hoeveel bacteriën zijn er na 3 uur en na 8 uur? Wat is de procentuele groei per uur en per minuut?

GEGEVEN

- $N_0 = 1000$ per mL
- verdubbelingstijd $t_2 = 20\,\text{min}$

GEVRAAGD

- $N(t)$, $N(3\,\text{h})$, $N(8\,\text{h})$
- groei per uur en per minuut in procent

OPLOSSING

1. In één uur verdubbelt de populatie drie keer: $g_\text{u} = 2^3 = 8$ per uur. Model: $N(t) = 1000 \cdot 8^t$ (per mL, t in uren).
2. $N(3) = 1000 \cdot 8^3 = 512\,000$ per mL.
3. $N(8) = 1000 \cdot 8^8 = 1000 \cdot 2^{24} = 1{,}7 \cdot 10^{10}$ per mL.

4 Per uur: $g = 8$, dus $p = (8 - 1) \cdot 100\,\% = 700\,\%$. Per minuut: $g = 2^{1/20} = 1{,}0353$, dus 3,5 % per minuut. (Controle: $1{,}0353^{60} = 8{,}0$.)

ANTWOORD

$N(t) = 1000 \cdot 8^{\,t}$ per mL: 512 000 na 3 uur, $1{,}7 \cdot 10^{10}$ na 8 uur. De groei is 700 % per uur, of 3,5 % per minuut: hetzelfde model in twee tijdseenheden.

Klopt dit? De uitkomst na 8 uur klopt niet met de werkelijkheid: in zo'n buisje stopt de groei rond een miljard bacteriën per mL, omdat de voeding opraakt en afvalstoffen zich ophopen. Het exponentiële model geldt alleen in de groeifase, tussen een aanloopfase en een stationaire fase (hoofdstuk 35). En bij 4 °C, in de koelkast, is de verdubbelingstijd vele uren tot dagen: daarom bewaar je melk koel.

UITLEG · 5.5

Verdubbelingstijd en halveringstijd

5/18

Vaak is het handiger om een exponentieel model te beschrijven met de tijd die nodig is om te verdubbelen of te halveren, in plaats van met een groeifactor.

MET VERDUBBELINGSTIJD EN HALVERINGSTIJD

$$N(t) = N_0 \cdot 2^{\,t/t_2} \qquad N(t) = N_0 \cdot \left(\tfrac{1}{2}\right)^{t/t_{1/2}}$$

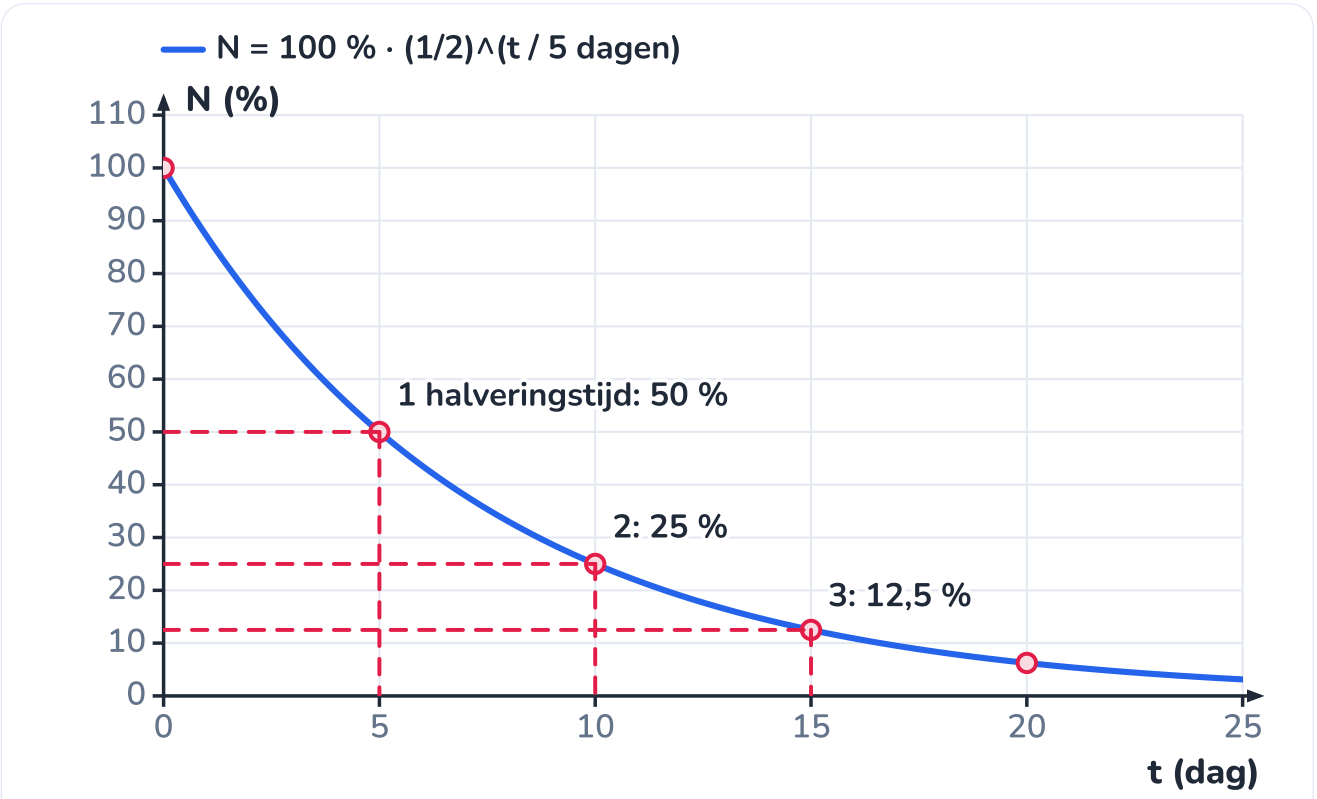
Na één verdubbelingstijd t_2 is N twee keer zo groot, na twee keer t_2 vier keer, na drie keer acht keer. Na één halveringstijd $t_{1/2}$ is de helft over, na twee een kwart, na tien ongeveer een duizendste $(1/1024)$.

Het mooie: de verdubbelingstijd hangt **niet** af van de beginwaarde. Van 1000 naar 2000 duurt even lang als van een miljoen naar twee miljoen. Dat is precies wat 'exponentieel' betekent: in gelijke tijden, gelijke factoren.

De vuistregel van 70. Bij een kleine procentuele groei p per tijdstap is de verdubbelingstijd ongeveer $70/p$ tijdstappen. Bij 2 % per jaar: 35 jaar; bij 7 % per jaar: 10 jaar. De regel is een benadering; bij 20 % geeft hij 3,5 in plaats van 3,8. Waar hij vandaan komt, zie je in het voorbeeld 'Hoe lang tot het verdubbelt?'.

Halveringstijd is geen 'halve levensduur'

Na twee halveringstijden is niet alles weg, maar nog een kwart. Een exponentieel dalende grootheid wordt nooit precies nul. In de praktijk zeg je: na ongeveer tien halveringstijden is er nog 0,1 % over, en dat is vaak verwaarloosbaar.



Elke 5 dagen halveert wat er over is: 100 %, 50 %, 25 %, 12,5 % … De stappen worden steeds kleiner.

UITLEG · 5.6

Logaritmen: de exponent terugvinden

6/18

Hoe lang duurt het tot een grootheid met 3 % per jaar verdubbelt? Je zoekt t in $1{,}03^t = 2$. De onbekende staat in de exponent. Om ze eruit te halen, heb je de **logaritme** nodig, de omgekeerde bewerking van een macht:

LOGARITME (GRONDTAL 10)

$$10^x = y \quad \Leftrightarrow \quad x = \log y$$

$\log 1000 = 3$ want $10^3 = 1000$; $\log 1 = 0$; $\log 0{,}01 = -2$; $\log 2 = 0{,}301$ want $10^{0{,}301} = 2$. Een logaritme bestaat alleen voor positieve getallen.

Twee rekenregels zijn genoeg voor dit hoofdstuk:

REKENREGELS

$$\log(a \cdot b) = \log a + \log b \qquad \log(a^n) = n \cdot \log a$$

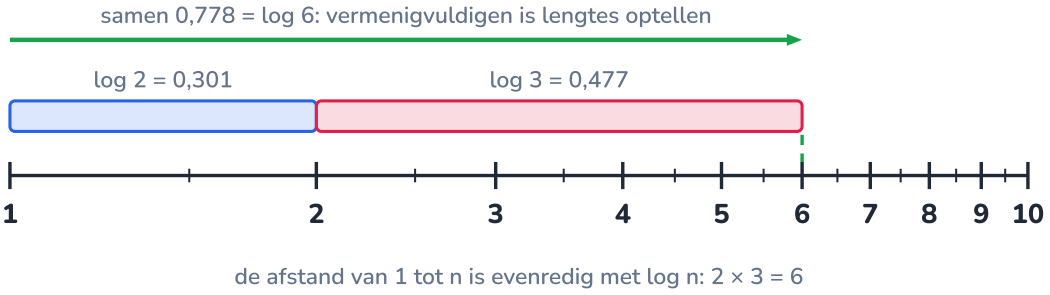
De eerste regel maakt van vermenigvuldigen optellen: dat is het principe van de **rekenlat**, tot in de jaren 1970 het rekentoestel van elke ingenieur (figuur). De tweede regel haalt de exponent naar beneden. Daarmee los je elke exponentiële vergelijking op:

$$g^t = k \;\Rightarrow\; \log\left(g^t\right) = \log k \;\Rightarrow\; t \cdot \log g = \log k \;\Rightarrow\; t = \frac{\log k}{\log g}$$

Voor $1{,}03^t = 2$: $t = \log 2 / \log 1{,}03 = 0{,}30103 / 0{,}012837 = 23{,}4$ jaar.

log of ln?

Je rekenmachine heeft ook een **ln**-toets: de natuurlijke logaritme, met als grondtal het getal e $\approx 2{,}718$ (dat verschijnt bij de condensator). Voor $t = \log k / \log g$ maakt het niet uit welke je gebruikt, zolang je in teller en noemer dezelfde neemt: $\ln 2 / \ln 1{,}03$ geeft ook 23,4.



Op een logaritmische schaal staat elk getal op een afstand log n van 1. Twee lengtes achter elkaar leggen is de getallen vermenigvuldigen.

UITGEWERKT VOORBEELD · 5.7

Hoe lang tot het verdubbelt? De formule afgeleid

7/18

Leid een formule af voor de verdubbelingstijd t_2 en de halveringstijd $t_{1/2}$ bij een groeifactor g. Pas ze toe op drie gevallen: (a) een waterverbruik dat groeit met 3 % per jaar; (b) de 'wet van Moore', een verdubbeling om de twee jaar: welke groeifactor per jaar is dat? (c) het radioactieve jodium-131, met een halveringstijd van 8,02 dagen: met hoeveel procent daalt het per dag?

GEGEVEN

- $N(t) = N_0 \cdot g^t$
- (a) $g = 1{,}03$ per jaar; (b) $t_2 = 2$ jaar; (c) $t_{1/2} = 8{,}02$ dagen

GEVRAAGD

- $t_2(g)$ en $t_{1/2}(g)$
- (a) t_2; (b) g per jaar; (c) p per dag

OPLOSSING

1 Verdubbeld betekent $N(t_2) = 2N_0$: $N_0 \cdot g^{t_2} = 2N_0$, dus $g^{t_2} = 2$.

2 Neem de logaritme van beide kanten: $\log(g^{t_2}) = \log 2$, dus $t_2 \cdot \log g = \log 2$ en $\boxed{t_2 = \dfrac{\log 2}{\log g}}$. Op dezelfde manier: $\boxed{t_{1/2} = \dfrac{\log \frac{1}{2}}{\log g} = -\dfrac{\log 2}{\log g}}$ (positief, want bij afname is $\log g < 0$).

3 (a) $t_2 = \dfrac{0{,}30103}{\log 1{,}03} = \dfrac{0{,}30103}{0{,}012837} = 23{,}4$ jaar. De vuistregel gaf $70/3 = 23{,}3$.

4 (b) Omgekeerd: $g^{t_2} = 2$ geeft $g = 2^{1/t_2} = 2^{1/2} = 1{,}414$ per jaar: +41 % per jaar.

5 (c) $g = \left(\frac{1}{2}\right)^{1/t_{1/2}} = 0{,}5^{1/8{,}02} = 0{,}9172$ per dag: een daling van 8,3 % per dag.

ANTWOORD

$t_2 = \log 2 / \log g$ **en** $t_{1/2} = -\log 2 / \log g$**. (a) 23,4 jaar; (b)** $g = 1{,}41$**, +41 % per jaar; (c) –8,3 % per dag.**

Klopt dit? Waar komt de vuistregel van 70 vandaan? Voor kleine p is $\ln(1 + p/100) \approx p/100$, en $\ln 2 = 0{,}693$. Dan is $t_2 = \ln 2 / \ln g \approx 0{,}693/(p/100) = 69{,}3/p$. Men rondt af naar 70 omdat dat makkelijker deelt, en omdat de benadering voor groeicijfers van enkele procenten dan net iets beter uitkomt.

Afkoelen: het verschil valt exponentieel

Een kop thee van 90 °C staat in een kamer van 20 °C. Ze koelt eerst snel af, dan steeds trager, en komt nooit onder 20 °C. Het is niet de temperatuur zelf die exponentieel daalt, maar het **verschil met de omgeving**. Isaac Newton beschreef dit verband in 1701: hoe groter het verschil, hoe sneller de warmte wegstroomt, dus hoe sneller het verschil kleiner wordt.

AFKOELINGSMODEL

$$T(t) = T_{\text{omg}} + (T_0 - T_{\text{omg}}) \cdot g^{\,t}$$

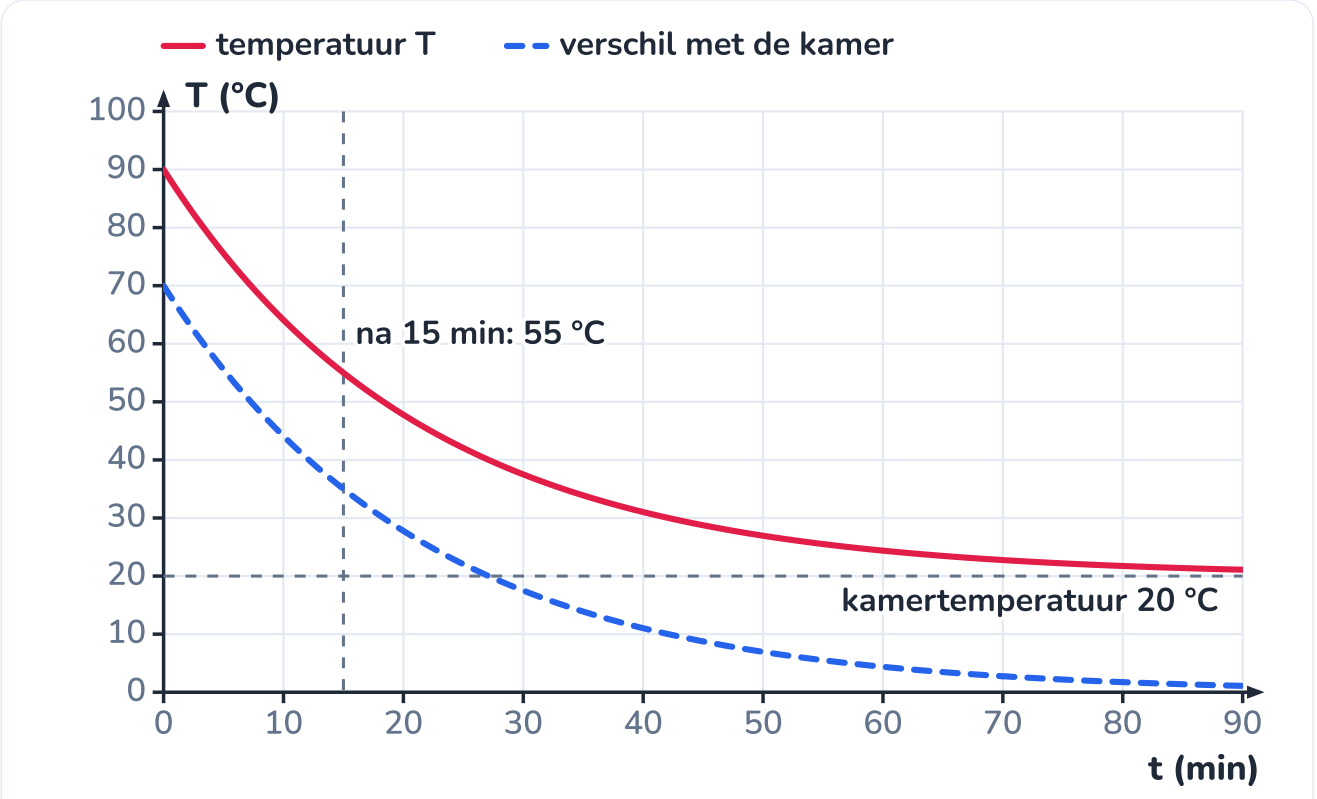
T_0 is de begintemperatuur, T_{omg} de omgevingstemperatuur (de eindwaarde), g de groeifactor van het verschil per tijdstap ($0 < g < 1$). Opwarmen naar een warmere omgeving gaat met dezelfde formule: dan is $T_0 - T_{\text{omg}}$ negatief.

In de grafiek halveert het verschil elke 15 minuten: van 70 °C naar 35 °C (de thee is dan 55 °C), naar 17,5 °C (37,5 °C), enzovoort. Elke halveringstijd komt de thee de helft van de resterende afstand dichter bij de kamertemperatuur.

Geldigheidsgebied. Het model veronderstelt een constante omgevingstemperatuur en een warmteverlies evenredig met het verschil. Dat klopt goed voor geleiding en voor stroming bij kleine verschillen. Bij een open beker koelt de thee in het begin sneller af dan het model zegt, door verdamping; en een grote watertank in een kas die 's nachts afkoelt, ziet de omgeving zelf ook kouder worden.

Vooruitblik

In Expert (hoofdstuk E4) schrijf je dit model als een **differentiaalvergelijking**: de snelheid van afkoelen is evenredig met het verschil, $\frac{\mathrm{d}T}{\mathrm{d}t} = -k\,(T - T_{\text{omg}})$. De exponentiële functie is daar de oplossing van. Het warmtemodel van het station in hoofdstuk E39 is hetzelfde idee, met een verwarming erbij.



De temperatuur (rood) nadert de kamertemperatuur; het verschil (blauw, stippel) halveert elke 15 minuten.

PROEF · 5.9

Hoe koelt een beker thee af?

9/18

ONDERZOEKSVRAAG

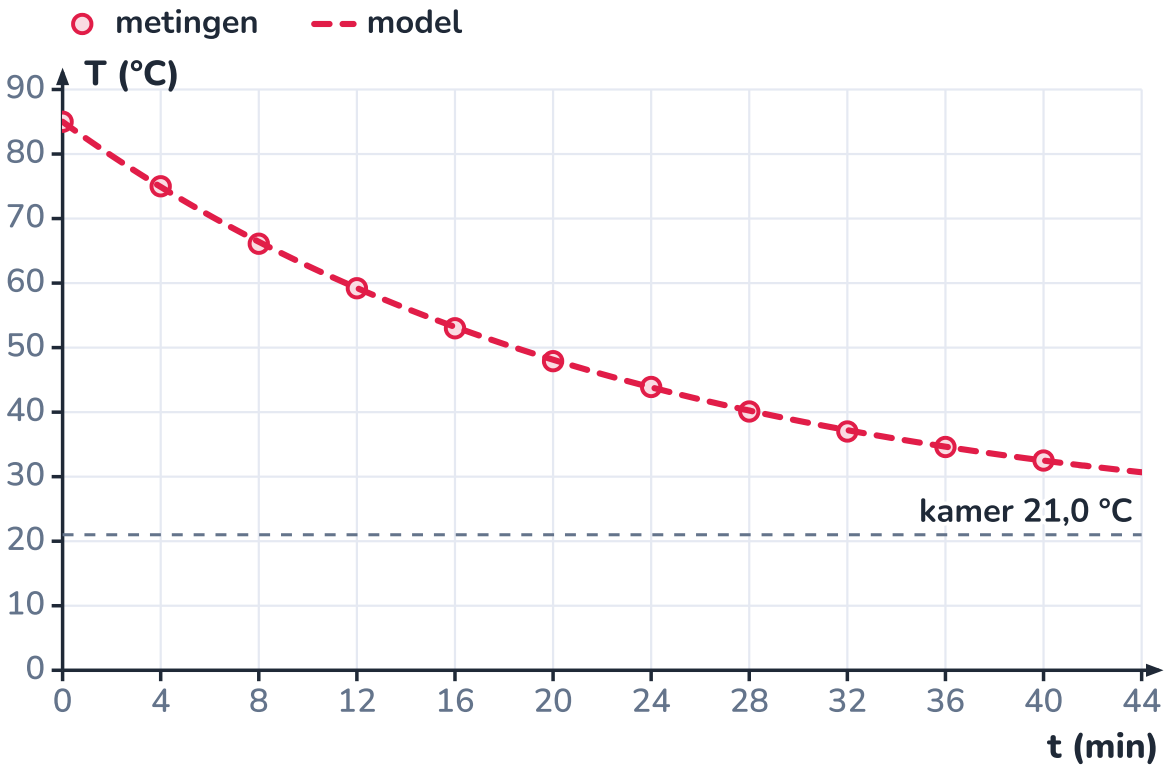
Daalt het temperatuurverschil tussen een beker heet water en de kamer exponentieel? Zo ja, wat is de halveringstijd van het verschil?

MATERIAAL

- beker (liefst dezelfde voor alle groepen), waterkoker
- digitale thermometer of temperatuursensor met datalogger
- thermometer voor de kamertemperatuur
- stopwatch
- rekenblad

Veiligheid

Heet water: zet de beker stevig neer, niet aan de rand van de tafel, en giet niet boven je handen.



De metingen (punten) volgen het exponentiële model naar de kamertemperatuur.

WERKWIJZE

1. Meet de kamertemperatuur, uit de buurt van ramen en verwarming.
2. Vul de beker met water van ongeveer 85 °C. Zet de sensor in het midden van het water, niet tegen de wand.
3. Roer zacht en noteer elke 4 minuten de temperatuur, 40 minuten lang. Laat de beker staan zonder deksel.
4. Bereken het verschil $T - T_{\text{omg}}$ en de verhouding van twee opeenvolgende verschillen.
5. Zet het verschil uit tegen de tijd, op een gewone en op een logaritmische as.

WAARNEMING

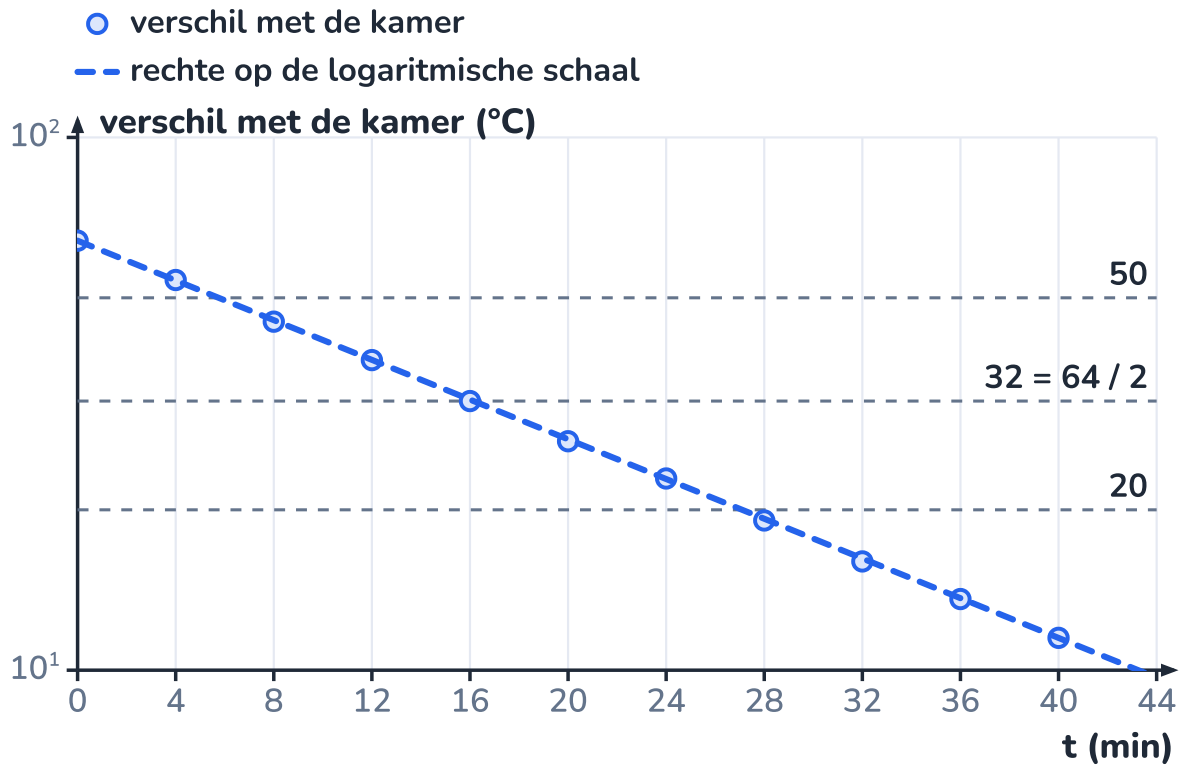
t (min)	0	4	8	12	16	20	24	28	32	36	40
T (°C)	85,0	75,0	66,1	59,2	53,0	47,9	4				
$T - T_{\text{omg}}$ (°C)	64,0	54,0	45,1	38,2	32,0	26,9	2				
verhouding per 4 min		0,844	0,835	0,847	0,838	0,841	0				

Kamertemperatuur 21,0 °C. De verhoudingen liggen allemaal tussen 0,834 en 0,851, zonder trend: gemiddeld 0,842 per 4 minuten. Op een logaritmische as liggen de verschillen op een rechte.

VERKLARING

ijdstappen: het verschil daalt exponentieel, met g 12 per 4 min. Per minuut is dat $0{,}842^{1/4} = 0{,}958$: het verschil wordt elke minuut 4,2 % kleiner. Halveringstijd van het verschil: $t_{1/2} = 4\,\text{min} \cdot \dfrac{\log 0{,}5}{\log 0{,}842} = 16\,\text{min}$. Het model: $T(t) = 21{,}0 + 64{,}0 \cdot 0{,}5^{\,t/16}$ (°C, t in min).

De halveringstijd hangt af van de beker, de hoeveelheid water en de luchtstroming: een dunne plastic beker of een ventilator maakt ze korter, een geïsoleerde beker veel langer. Precies dat getal gebruik je later om een warmtebuffer voor de klimaatmodule te ontwerpen (hoofdstukken 18 en 19).



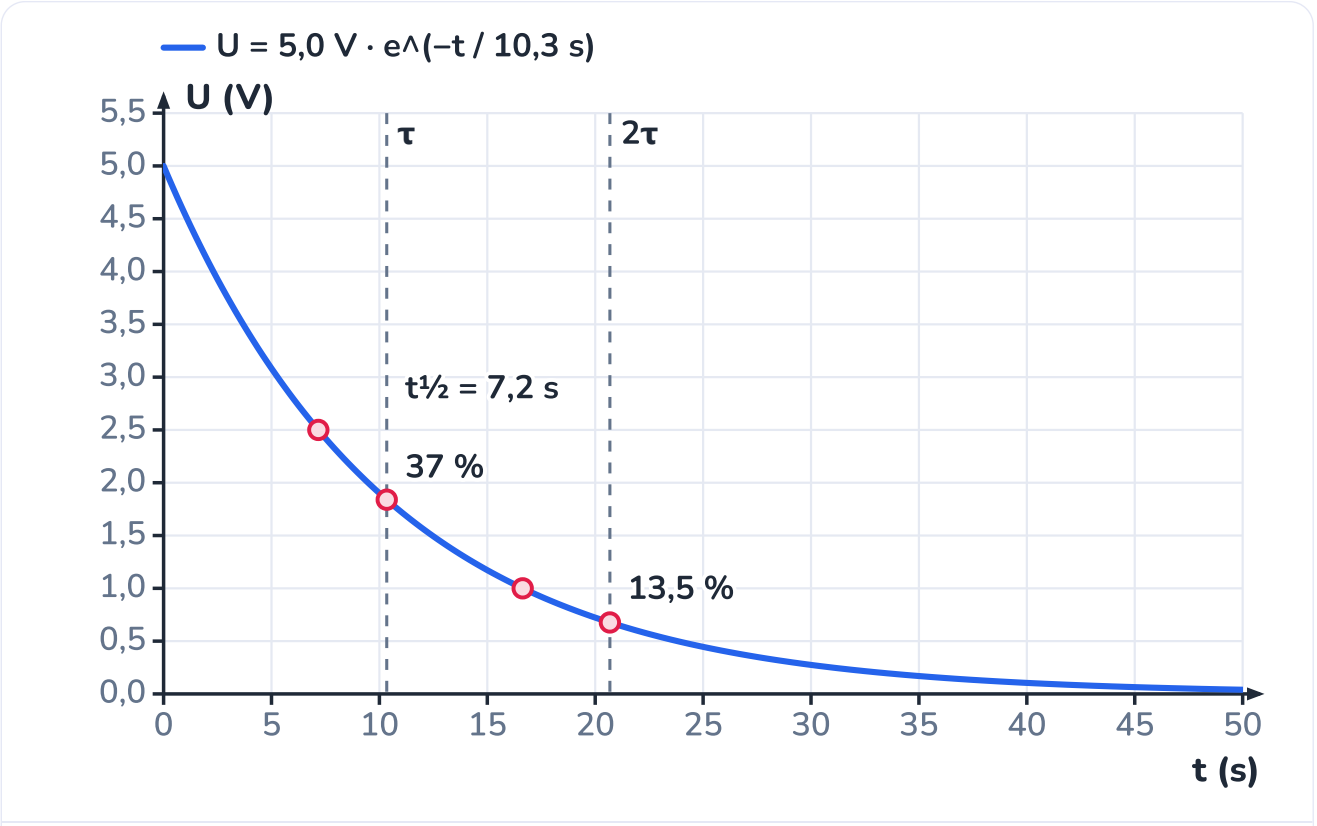
Op een logaritmische as wordt het verschil een rechte: het bewijs dat het exponentieel daalt. Het halveert van 64 naar 32 °C in 16 minuten.

UITGEWERKT VOORBEELD · 5.10

Een condensator ontlaadt

10/18

Een condensator van 470 µF is opgeladen tot 5,0 V en ontlaadt over een weerstand van 22 kΩ. De spanning volgt $U(t) = U_0 \cdot \mathrm{e}^{-t/\tau}$ met $\tau = R \cdot C$. Bereken de tijdconstante, de halveringstijd, de groeifactor per seconde, en de tijd tot de spanning onder 1,0 V zakt.



Na één tijdconstante (10,3 s) is nog 37 % over, na twee 13,5 %. De halveringstijd is 7,2 s.

GEGEVEN

- $C = 470\,\mu\text{F} = 470 \cdot 10^{-6}\,\text{F}$, $R = 22\,\text{k}\Omega$
- $U_0 = 5{,}0\,\text{V}$

GEVRAAGD

- τ, $t_{1/2}$, g per seconde
- t waarbij $U = 1{,}0\,\text{V}$

OPLOSSING

1. $\tau = R \cdot C = 22 \cdot 10^{3}\,\Omega \cdot 470 \cdot 10^{-6}\,\text{F} = 10{,}3\,\text{s}$ (ohm maal farad is seconde).
2. $\text{e}^{-t/\tau}$ is een gewone exponentiële functie: $\text{e}^{-t/\tau} = \left(\text{e}^{-1/\tau}\right)^{t}$, dus $g = \text{e}^{-1/10{,}34} = 0{,}908$ per seconde: de spanning daalt elke seconde 9,2 %.
3. Na één tijdconstante is $U = 5{,}0 \cdot \text{e}^{-1} = 1{,}84\,\text{V}$: nog 37 %. Na 2τ nog 13,5 %, na 5τ nog 0,7 %.
4. Halveringstijd: $\text{e}^{-t/\tau} = \tfrac{1}{2}$ geeft $t/\tau = \ln 2$, dus $t_{1/2} = \tau \ln 2 = 0{,}693 \cdot 10{,}34 = 7{,}2\,\text{s}$.
5. Onder 1,0 V: $5{,}0 \cdot \text{e}^{-t/\tau} = 1{,}0$, dus $\text{e}^{-t/\tau} = 0{,}20$ en $t = -\tau \ln 0{,}20 = \tau \ln 5 = 16{,}6\,\text{s}$.

ANTWOORD

$\tau = 10{,}3\,\text{s}$, $t_{1/2} = 7{,}2\,\text{s}$, $g = 0{,}908$ **per seconde; na 16,6 s is de spanning onder 1,0 V.**

Klopt dit? Het getal e $\approx 2{,}718$ is een grondtal zoals 2 of 10; natuurkundigen kiezen het omdat τ dan een eenvoudige betekenis krijgt: de tijd tot $1/\text{e}$ over is. Met log in plaats van ln vind je hetzelfde: $t = \tau \cdot \log 5 / \log \text{e} = 10{,}34 \cdot 0{,}699 / 0{,}434 = 16{,}6\,\text{s}$. Waarom condensatoren zo ontladen, leer je in hoofdstuk 24; in hoofdstuk E4 leid je de formule af.

Radioactief verval

Sommige atoomkernen zijn instabiel. Ze vallen vanzelf uiteen en zenden daarbij straling uit. Wanneer één bepaalde kern vervalt, is volledig toevallig: je kan het niet voorspellen. Maar elke kern heeft per tijdseenheid dezelfde kans om te vervallen. In een grote groep vervalt daardoor per tijdseenheid een vaste **fractie** van de kernen die er nog zijn: exponentieel verval.

RADIOACTIEF VERVAL

$$N(t) = N_0 \cdot \left(\tfrac{1}{2}\right)^{t/t_{1/2}} \qquad A(t) = A_0 \cdot \left(\tfrac{1}{2}\right)^{t/t_{1/2}}$$

N: het aantal kernen dat nog niet vervallen is. A: de **activiteit**, het aantal vervallen per seconde, in becquerel (1 Bq = 1 verval per seconde). Omdat A evenredig is met N, halveert ze in dezelfde tijd.

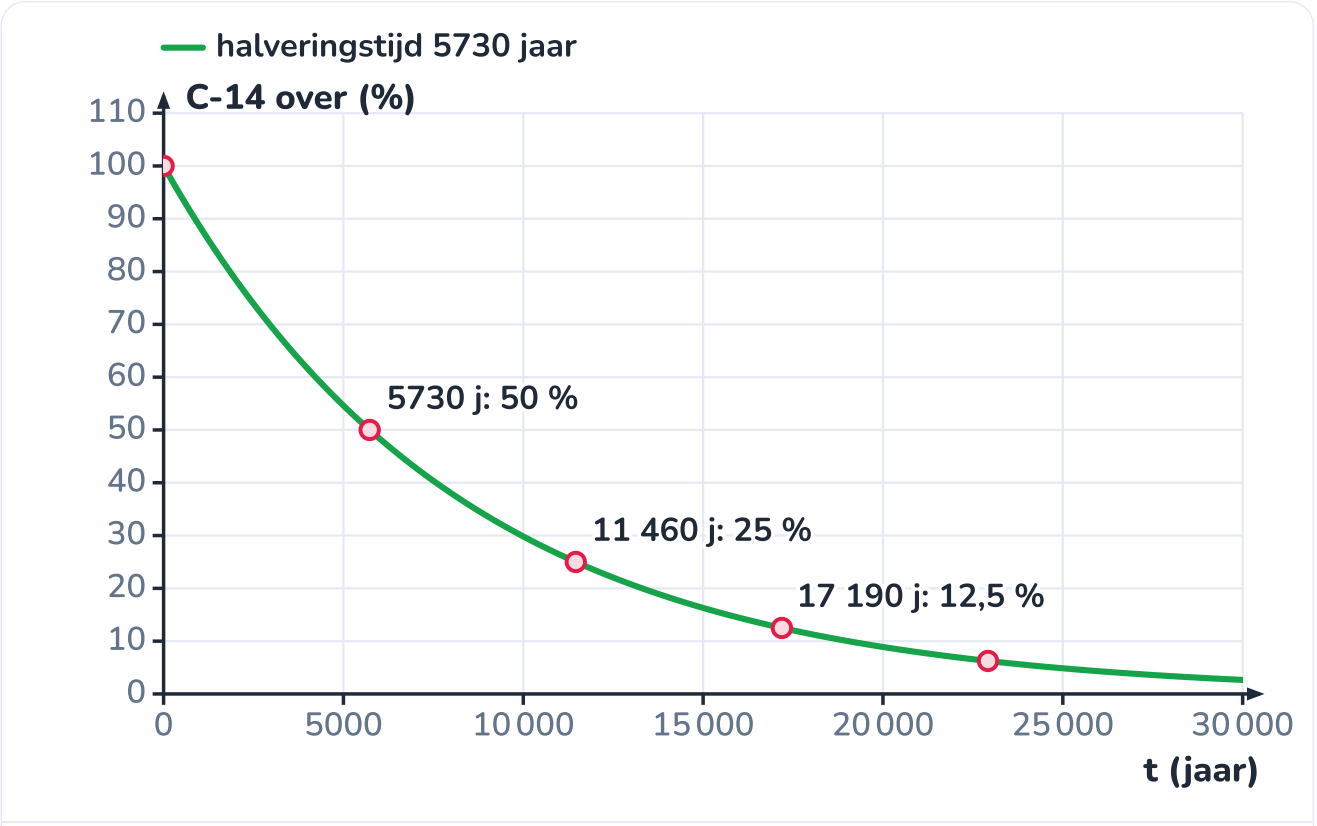
De halveringstijd is een eigenschap van de isotoop, en loopt van fracties van een seconde tot miljarden jaren:

isotoop	halveringstijd	waar je het tegenkomt
technetium-99m	6,0 uur	medische beeldvorming (scan van botten, hart)
radon-222	3,8 dagen	radioactief gas uit de bodem, onder meer in delen van de Ardennen
jodium-131	8,0 dagen	behandeling van de schildklier; vrijgekomen bij kernongevallen
cesium-137	30 jaar	splijtingsproduct uit kerncentrales; bron in meettoestellen
koolstof-14	5730 jaar	ouderdomsbepaling van hout, bot en textiel
kalium-40	1,25 miljard jaar	in bananen, in je eigen lichaam, in gesteenten
uranium-238	4,47 miljard jaar	ouderdom van gesteenten en van de aarde

Koolstof-14. Planten nemen koolstof op uit de lucht, met een vaste kleine fractie radioactief koolstof-14. Zolang een organisme leeft, blijft die fractie gelijk; na zijn dood vervalt de koolstof-14 zonder aanvulling. Meet je hoeveel er nog over is, dan ken je de ouderdom. De Amerikaanse chemicus Willard Libby ontwikkelde deze methode rond 1949 en kreeg er in 1960 de Nobelprijs voor.

Na twee halveringstijden is alles weg

Na één halveringstijd is de helft over, na twee een kwart, na drie een achtste. Een monster jodium-131 is na 8 dagen niet 'half op' in de zin van een half lege batterij: elke overblijvende kern heeft nog precies dezelfde kans om te vervallen als op dag één.



Elke 5730 jaar halveert de hoeveelheid koolstof-14. Na 50 000 jaar is nog 0,2 % over: de grens van de methode.

CODE · 5.12

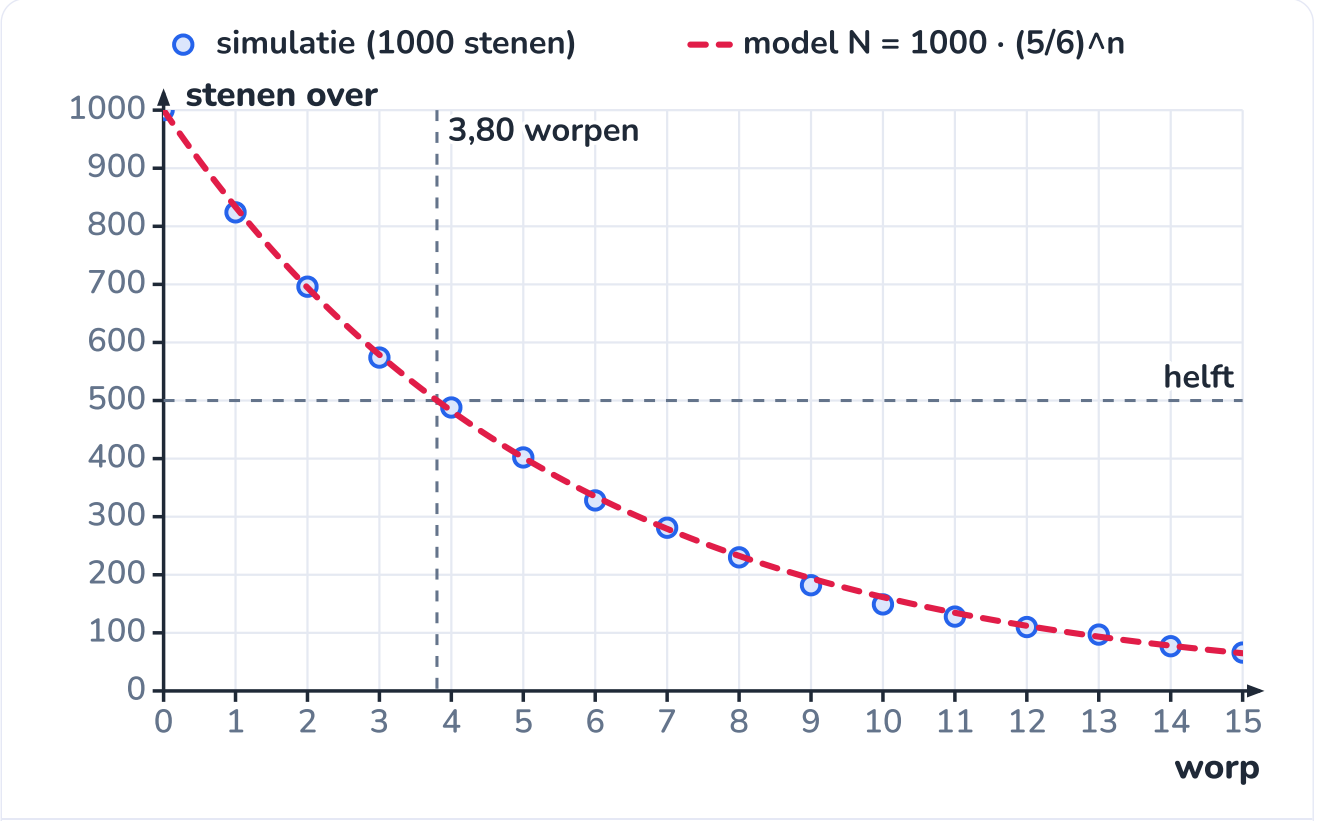
Verval met dobbelstenen, gesimuleerd

12/18

Een dobbelsteen is een goed model voor een radioactieve kern: bij elke worp is er een kans van 1/6 op een zes ('verval'), en de steen onthoudt niet hoe vaak hij al gegooid is. Gooi 1000 stenen, haal de zessen eruit, gooi de rest opnieuw. Het model voorspelt $N = 1000 \cdot (5/6)^n$ na n worpen, met een halveringstijd van $\log 0{,}5 / \log(5/6) = 3{,}80$ worpen. De computer gooit sneller dan jij.

Python **verval.py** 18 regels

```python
# verval.py - radioactief verval nabootsen met dobbelstenen
import math
import random

random.seed(159)              # vaste startwaarde: elke keer dezelfde 'toevallige' worpen
N0 = 1000                     # aantal dobbelstenen ('kernen') bij de start
p = 1 / 6                     # kans dat een steen een zes gooit ('vervalt') per worp

over = N0
print("worp  over  model")
for worp in range(1, 16):
    vervallen = sum(1 for _ in range(over) if random.randint(1, 6) == 6)
    over = over - vervallen
    model = N0 * (1 - p) ** worp
    print(f"{worp:4d} {over:5d} {model:6.1f}")

T_half = math.log(0.5) / math.log(1 - p)
print(f"halveringstijd volgens het model: {T_half:.2f} worpen")
```



Na 3,80 worpen is volgens het model de helft over; de simulatie zit dan tussen 574 en 488.

Uitvoer (gecontroleerd door het programma te draaien):

```
worp  over  model
   1   824  833.3
   2   696  694.4
   3   574  578.7
   4   488  482.3
   5   402  401.9
   6   328  334.9
   7   281  279.1
   8   230  232.6
   9   182  193.8
  10   149  161.5
  11   128  134.6
  12   110  112.2
  13    97   93.5
  14    77   77.9
  15    66   64.9
halveringstijd volgens het model: 3.80 worpen
```

De simulatie schommelt rond het model, soms erboven, soms eronder: toeval. Hoe kleiner de groep, hoe groter die schommelingen in verhouding (bij 1000 stenen enkele procenten, bij de laatste 66 stenen al meer). Dat is ook bij echte straling zo: een geigerteller tikt onregelmatig, maar gemiddeld over veel kernen geldt de halveringstijd heel precies. Met `random.seed(159)` krijg je elke keer dezelfde worpen; verander het getal en je krijgt een andere, even toevallige reeks. In opdracht 5.10 gooi je zelf.

Exponentieel tegenover lineair: waarom je intuïtie faalt

Onze intuïtie is lineair: wat gisteren een beetje groeide, groeit morgen een beetje meer. Bij exponentiële groei klopt dat niet, om drie redenen.

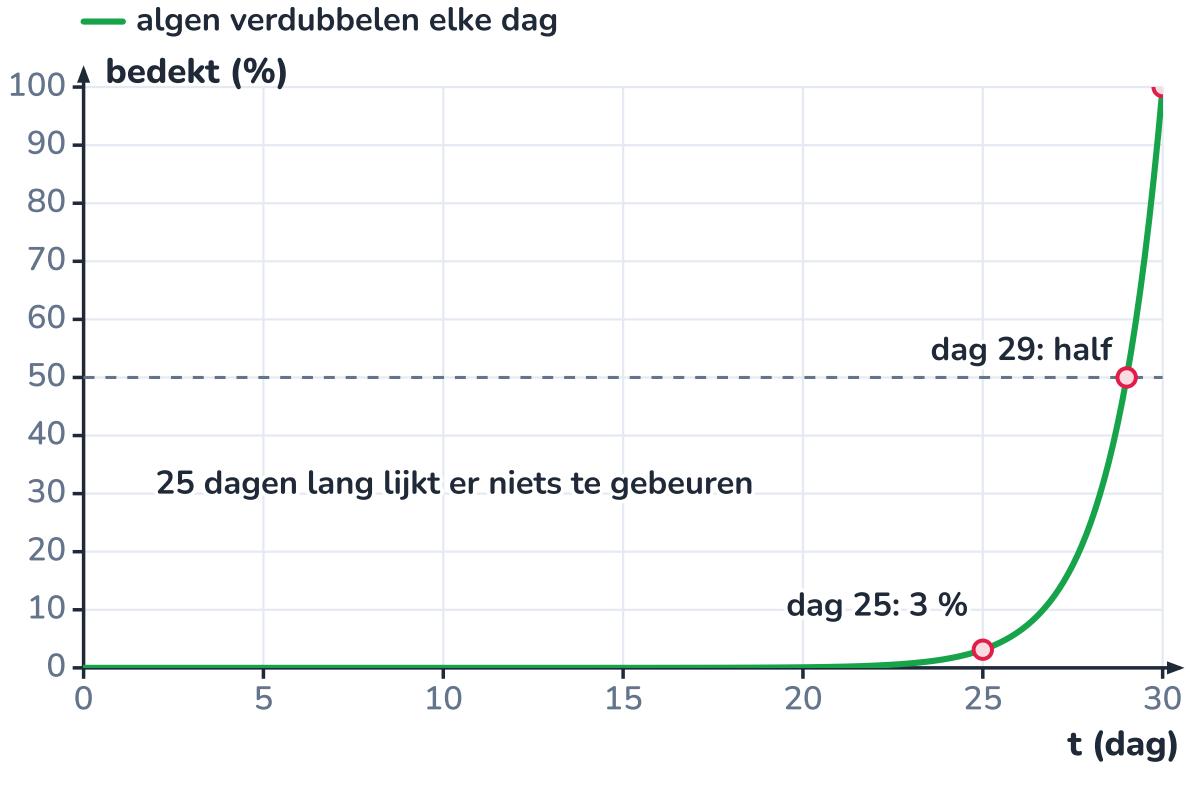
1. Lang is er niets te zien. In een vijver verdubbelt een laag algen elke dag. Op dag 30 is de vijver volledig bedekt. Wanneer was hij half bedekt? Op dag 29, één dag eerder. Op dag 25 was nog maar 3 % bedekt: wie dan naar de vijver keek, zag vooral water. Vier dagen later was er geen water meer te zien.

2. Kleine getallen worden snel astronomisch. Een blad papier van 0,1 mm dik, 42 keer dubbel gevouwen, zou $0{,}1\,\text{mm} \cdot 2^{42} = 440\,000\,\text{km}$ dik zijn: verder dan de maan (gemiddeld 384 400 km). In werkelijkheid lukt het vouwen maar enkele keren, maar de rekening klopt.

3. Wachten is duur. Verdubbelt iets om de 3 dagen, dan betekent een week later ingrijpen $2^{7/3} = 5$ keer zoveel werk. Bij een groeiend probleem, van algen in een reservoir tot een besmettelijke ziekte, is de goedkoopste dag om in te grijpen altijd vandaag.

En op lange termijn wint exponentiële groei **altijd** van lineaire groei, en zelfs van elke kwadratische of hogere macht: 5 % per stap haalt '+20 per stap' in na 48,6 stappen (eerste blad), en $1{,}01^t$ haalt t^2 ook in, alleen later.

> ⚠ **Geen groei is eeuwig exponentieel**
>
> Algen raken hun licht of hun voeding kwijt, bacteriën hun suiker, een epidemie de mensen die nog niet ziek of immuun zijn. Dan vlakt de groei af tot een grens: de **draagkracht**. Zo'n S-vormige kromme heet logistische groei (hoofdstuk E32). Het exponentiële model is dus het begin van het verhaal, maar vaak precies het deel waar je nog kan ingrijpen.



Bedekking in procent: 25 dagen lijkt er niets te gebeuren, de laatste vijf dagen gebeurt alles.

UITLEG · 5.14

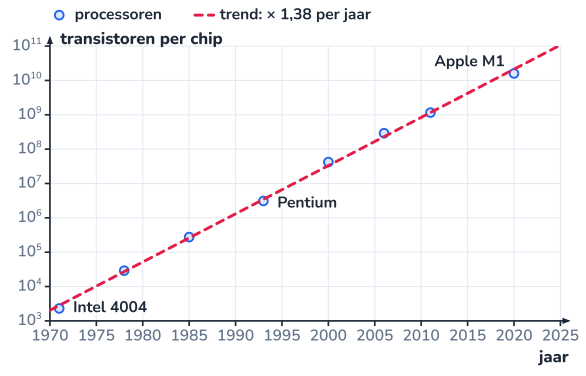
De logaritmische schaal lezen

Hoe teken je iets dat groeit van 2300 tot 16 miljard? Op een gewone as zie je alleen het laatste punt. Op een **logaritmische schaal** staan 1, 10, 100, 1000 … op gelijke afstanden: elke stap is een factor 10, een **decade**. Wat je uitzet, is eigenlijk $\log N$.

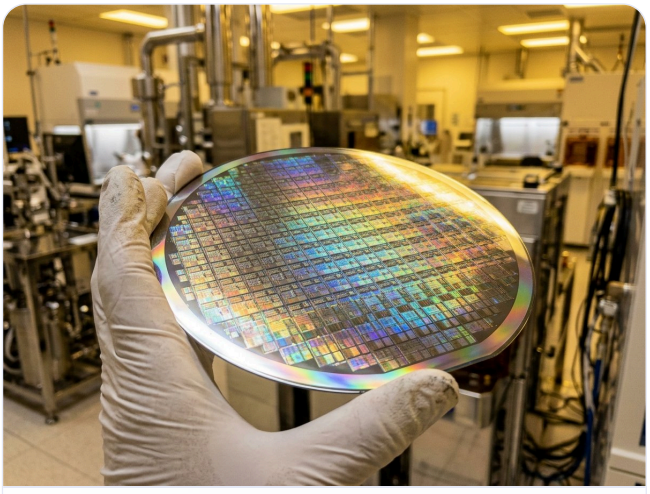
- **Gelijke afstanden zijn gelijke factoren.** Van 10 naar 100 is even ver als van 10^6 naar 10^7.
- **Tussenwaarden** staan niet in het midden: 2 staat op 30 % van een decade ($\log 2 = 0{,}30$), 3 op 48 %, 5 op 70 %. Halverwege tussen 10 en 100 staat 32 ($10^{1{,}5}$), niet 55.
- **Een exponentiële functie wordt een rechte**, want $\log N = \log N_0 + t \cdot \log g$: lineair in t, met helling $\log g$. Zo trek je een exponentieel verband recht, zoals je in hoofdstuk 3 krommen rechttrok.
- **De helling is de groeifactor.** Stijgt de rechte één decade in een tijd t_{10}, dan is $g^{t_{10}} = 10$ en de verdubbelingstijd $t_2 = t_{10} \cdot \log 2 = 0{,}30 \cdot t_{10}$.
- **Nul bestaat niet** op een logaritmische as: $\log 0$ is niet gedefinieerd. De as begint bij een macht van tien die je kiest.

De grafiek toont de **wet van Moore**. In 1965 voorspelde Gordon Moore, later medeoprichter van Intel, dat het aantal onderdelen op een chip elk jaar zou verdubbelen; in 1975 stelde hij dat bij tot elke twee jaar. Vijftig jaar lang bleven de punten verrassend dicht bij één rechte: een exponentiële groei met een factor van ongeveer 1,38 per jaar.

> **Andere logaritmische schalen**
>
> De **pH-schaal** (hoofdstuk 15): pH 5 is tien keer zo zuur als pH 6. De **decibel** voor geluid: +10 dB is tien keer zoveel geluidsvermogen. De **magnitude** van een aardbeving: één punt meer is tien keer zo grote uitslag op de seismometer. Telkens zijn gelijke stappen gelijke factoren.



Acht processoren van 1971 tot 2020, van 2300 tot 16 miljard transistoren. Op de logaritmische as liggen ze op een rechte.

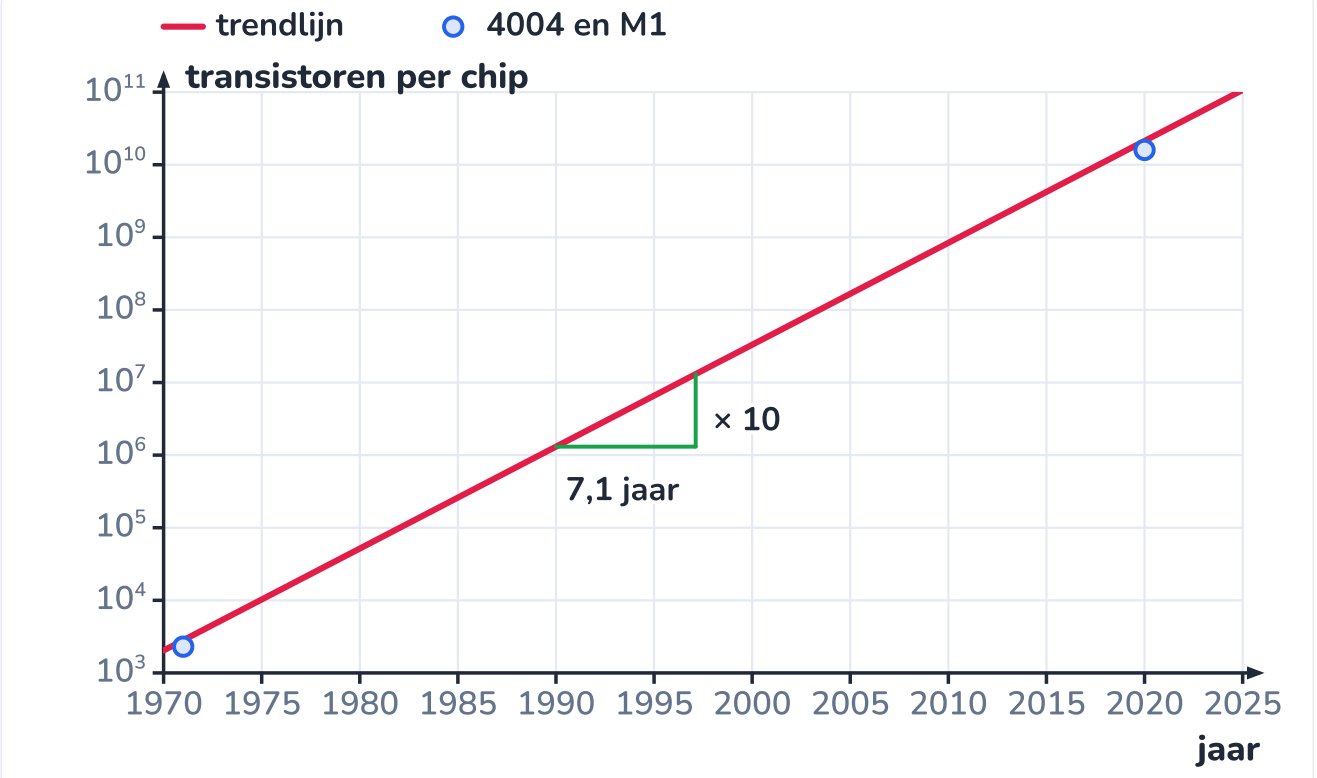


Een siliciumschijf met honderden chips: elk ervan bevat vandaag miljarden transistoren.

UITGEWERKT VOORBEELD · 5.15

De wet van Moore narekenen

De Intel 4004 (1971) had ongeveer 2300 transistoren, de Apple M1 (2020) 16 miljard. Bereken uit deze twee punten de groeifactor per jaar en de verdubbelingstijd. Hoe lang duurt een tienvoud? Wat voorspelt het model voor 2000, en hoe ver zit de Pentium 4 (42 miljoen) ernaast?



Op de trendlijn stijgt het aantal transistoren een decade (× 10) in 7,1 jaar.

GEGEVEN

- $N(1971) = 2{,}3 \cdot 10^{3}$
- $N(2020) = 1{,}6 \cdot 10^{10}$

GEVRAAGD

- g per jaar, t_2, t_{10}
- $N(2000)$ volgens het model

OPLOSSING

1 Groeifactor over 49 jaar: $\dfrac{1{,}6 \cdot 10^{10}}{2300} = 6{,}96 \cdot 10^{6}$. Dus $g^{49} = 6{,}96 \cdot 10^{6}$.

2 Met logaritmen: $49 \log g = \log(6{,}96 \cdot 10^{6}) = 6{,}842$, dus $\log g = 0{,}1396$ en $g = 10^{0{,}1396} = 1{,}379$ per jaar: +38 % per jaar.

3 Verdubbelingstijd: $t_2 = \dfrac{\log 2}{\log g} = \dfrac{0{,}3010}{0{,}1396} = 2{,}16$ jaar. Tienvoud: $t_{10} = \dfrac{\log 10}{\log g} = \dfrac{1}{0{,}1396} = 7{,}2$ jaar.

4 Model voor 2000: $N = 2300 \cdot 1{,}379^{29} = 2{,}6 \cdot 10^{7}$. De Pentium 4 had $4{,}2 \cdot 10^{7}$: 1,6 keer meer dan het model, op een logaritmische schaal maar een klein stukje boven de rechte ($\log 1{,}6 = 0{,}2$ decade).

ANTWOORD

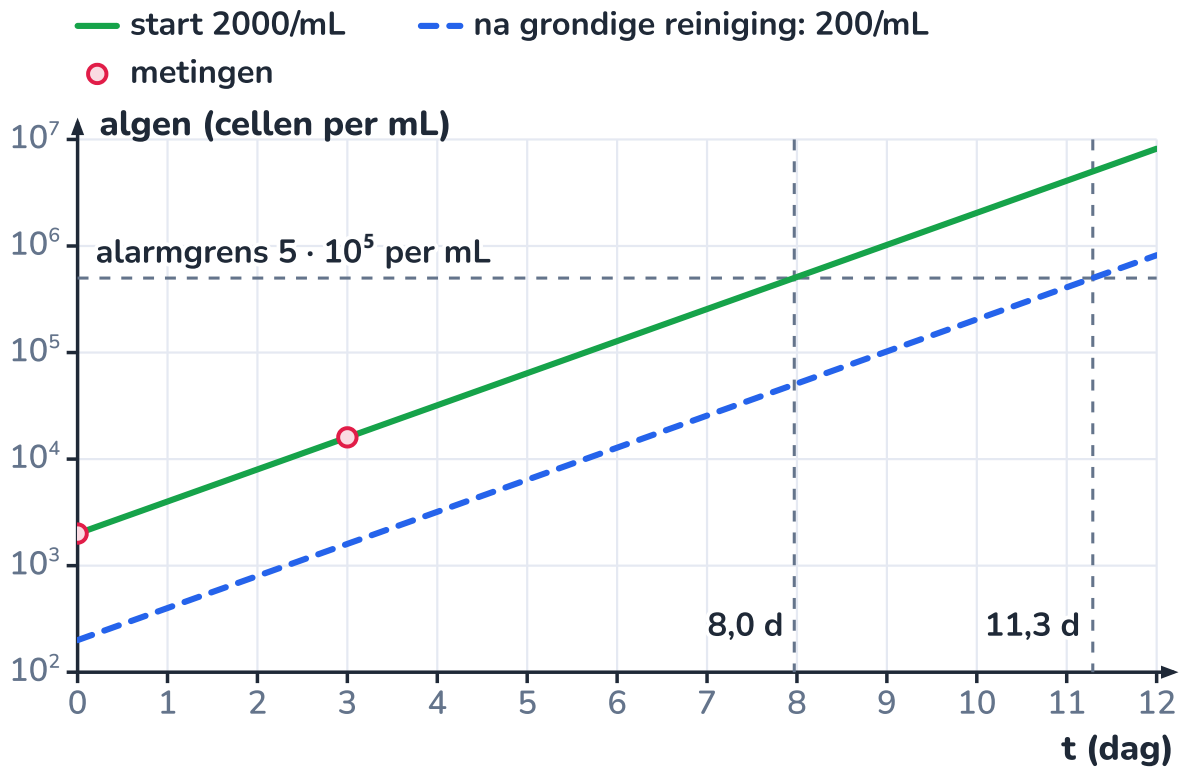
$g = 1{,}38$ per jaar, verdubbeling om de 2,2 jaar, tienvoud om de 7,2 jaar. Moore's bijgestelde voorspelling van 1975 (verdubbeling om de twee jaar) klopt dus opmerkelijk goed.

Klopt dit? Een kleinste-kwadratenrechte door log(N) van alle acht processoren (zoals in hoofdstuk 3, maar dan op log N) geeft g = 1,38 per jaar en 7,1 jaar per decade: bijna hetzelfde. Een verschil van een factor 1,6 lijkt groot, maar op deze schaal is het klein: de chips van elk jaar verschillen onderling meer dan dat. En de rechte kan niet eeuwig doorgaan: transistoren zijn nu nog enkele tientallen atomen breed.

UITGEWERKT VOORBEELD · 5.16

Habitat: wanneer wordt het reservoir groen?

Het voedingsreservoir van de klimaatmodule (60 L) is doorschijnend. Je meet de algen in het water met een sensor: $2{,}0 \cdot 10^3$ cellen per mL bij het vullen, $1{,}6 \cdot 10^4$ drie dagen later. Het team legt de alarmgrens op $5 \cdot 10^5$ cellen per mL, waar het water groenig begint te kleuren. Wanneer gaat het alarm af? Wat levert het op als je het reservoir bij elke vulling zo grondig reinigt dat je met tien keer minder algen start?



Een factor 10 minder bij de start schuift de rechte een decade omlaag: 3,3 dagen later alarm, niet meer.

GEGEVEN

- $N(0) = 2{,}0 \cdot 10^3$ /mL, $N(3\,\text{d}) = 1{,}6 \cdot 10^4$ /mL
- alarmgrens $5 \cdot 10^5$ /mL

GEVRAAGD

- g per dag en t_2
- tijd tot de alarmgrens, met en zonder grondige reiniging

OPLOSSING

1. In 3 dagen een factor $1{,}6 \cdot 10^4 / 2{,}0 \cdot 10^3 = 8$: $g^3 = 8$, dus $g = 2$ per dag. De algen verdubbelen elke dag: $t_2 = 1$ dag.
2. Alarm: $2{,}0 \cdot 10^3 \cdot 2^t = 5 \cdot 10^5$ geeft $2^t = 250$ en $t = \dfrac{\log 250}{\log 2} = \dfrac{2{,}398}{0{,}301} = 8{,}0$ dagen.
3. Met tien keer minder algen bij de start: $2^t = 2500$, $t = \log 2500 / \log 2 = 11{,}3$ dagen.
4. Tien keer minder algen levert dus maar $\log 10 / \log 2 = 3{,}3$ dagen uitstel op: er zijn 3,3 verdubbelingen nodig om een factor 10 in te halen.

ANTWOORD

De algen verdubbelen elke dag; het alarm gaat af na 8 dagen. Grondiger reinigen geeft 3,3 dagen uitstel, niet tien keer zoveel tijd. Wie de algen wil tegenhouden, moet de groeifactor verlagen, niet de beginwaarde: een lichtdicht reservoir neemt de algen hun licht af.

Klopt dit? Op de logaritmische grafiek zijn beide modellen evenwijdige rechten: dezelfde helling (dezelfde g), een decade uit elkaar. Het model geldt tot het licht of de voeding de groei begrenst; voor een alarmgrens ver onder die grens is het een goede, voorzichtige schatting. Opdracht 5.13 gaat verder met echte meetreeksen.

NASLAG · 5.17

Overzicht: exponentiële modellen

17/18

exponentieel model

$$N(t) = N_0 \cdot g^t \qquad g = 1 + \frac{p}{100}$$

symbool	betekenis	eenheid
N_0	beginwaarde	eenheid van N
g	groeifactor per tijdstap	–
p	procentuele verandering per tijdstap	%

andere tijdseenheid (n korte stappen in één lange)

$$g_{\text{lang}} = g_{\text{kort}}^{\,n} \qquad g_{\text{kort}} = g_{\text{lang}}^{\,1/n}$$

verdubbelings- en halveringstijd, vuistregel van 70

$$t_2 = \frac{\log 2}{\log g} \qquad t_{1/2} = -\frac{\log 2}{\log g} \qquad t_2 \approx \frac{70}{p}$$

met verdubbelings- of halveringstijd

$$N(t) = N_0 \cdot 2^{\,t/t_2} = N_0 \cdot \left(\tfrac{1}{2}\right)^{t/t_{1/2}}$$

logaritmen

$$10^x = y \Leftrightarrow x = \log y \qquad \log(ab) = \log a + \log b \qquad \log(a^n) = n \log a$$

een exponent terugrekenen

$$g^t = k \Rightarrow t = \frac{\log k}{\log g}$$

afkoelen of opwarmen naar de omgeving

$$T(t) = T_{\text{omg}} + (T_0 - T_{\text{omg}}) \cdot g^t$$

symbool	betekenis	eenheid
T_{omg}	omgevingstemperatuur (eindwaarde)	°C
T_0	begintemperatuur	°C

ontladen van een condensator

$$U(t) = U_0 \cdot \mathrm{e}^{-t/\tau} \qquad \tau = R\,C \qquad t_{1/2} = \tau \ln 2$$

symbool	betekenis	eenheid
τ	tijdconstante: na τ is nog 37 % over	s
R	weerstand	Ω
C	capaciteit	F

	lineair	exponentieel
per stap	zelfde getal erbij	zelfde factor maal
in een tabel constant	verschillen	verhoudingen
formule	$N = N_0 + a\,t$	$N = N_0 \cdot g^t$
rechte op	gewoon papier	papier met logaritmische verticale as
op lange termijn	blijft achter	wint altijd (bij $g > 1$)

isotoop	halveringstijd	waar je het tegenkomt
technetium-99m	6,0 uur	medische beeldvorming (scan van botten, hart)
radon-222	3,8 dagen	radioactief gas uit de bodem, onder meer in delen van de Ardennen
jodium-131	8,0 dagen	behandeling van de schildklier; vrijgekomen bij kernongevallen

isotoop	halveringstijd	waar je het tegenkomt
cesium-137	30 jaar	splijtingsproduct uit kerncentrales; bron in meettoestellen
koolstof-14	5730 jaar	ouderdomsbepaling van hout, bot en textiel
kalium-40	1,25 miljard jaar	in bananen, in je eigen lichaam, in gesteenten
uranium-238	4,47 miljard jaar	ouderdom van gesteenten en van de aarde

na … halveringstijden	1	2	3	5	7	10
nog over	50 %	25 %	12,5 %	3,1 %	0,8 %	0,1 %

HET GEHEEL · 5.18

Waar exponentiële modellen terugkomen

18/18

Exponentiële functies duiken op overal waar de verandering evenredig is met wat er al is: hoe meer bacteriën, hoe meer er bijkomen; hoe groter het temperatuurverschil, hoe sneller het kleiner wordt. Dat idee, 'de snelheid is evenredig met de grootte', wordt in Expert een differentiaalvergelijking, en verbindt biologie, energie en elektronica.

I3 Verbanden in data

Lineariseren: een exponentieel verband trek je recht met een logaritmische as.

I4 Functies als modellen

Modellen kiezen met verschillen en verhoudingen; domein en geldigheidsgebied.

I15 Zuren, basen en pH

De pH-schaal is logaritmisch: elke stap een factor 10 in zuurgraad.

I18 Warmte berekenen

Een watervat als warmtebuffer: hoe snel koelt het 's nachts af?

I24 Condensatoren en de RC-tijd

Laden en ontladen van een condensator, met de tijdconstante $\tau = R \cdot C$.

I35 Micro-organismen en biotechnologie

De groeicurve van een bacteriecultuur: aanloop, exponentiële groei, stationaire fase.

E4 Differentiaalvergelijkingen en simulatie

Afkoelingswet van Newton en RC-kring als $dy/dt = -k \cdot y$: de exponentiële functie als oplossing.

E30 Biotechnologie en genetische technieken

PCR verdubbelt DNA bij elke cyclus: 30 cycli geven een factor van een miljard.

E32 Ecologische modellen

Logistische groei: waar exponentiële groei tegen de draagkracht botst.

Habitat

Algen in het voedingsreservoir zijn het schoolvoorbeeld van exponentiële groei in de klimaatmodule: ze verdubbelen onder licht en met voeding ongeveer elke dag. In opdracht 5.13 analyseer je een meetreeks en kies je tussen reinigen, afdekken en lichtdicht maken. Voor Habitat leggen we vast: het reservoir van de klimaatmodule (60 L) is lichtdicht.

SAMENVATTING

In het kort

- Exponentieel: per tijdstap vermenigvuldigen met dezelfde groeifactor g: $N(t) = N_0 \cdot g^t$, met $g = 1 + p/100$. Herken het aan constante verhoudingen.
- Procenten tel je niet op, groeifactoren vermenigvuldig je: +10 % per dag is +95 % per week. Een andere tijdseenheid: $g_{\text{lang}} = g_{\text{kort}}^{n}$.

- De verdubbelingstijd $t_2 = \log 2 / \log g$ en de halveringstijd $t_{1/2} = -\log 2 / \log g$ hangen niet af van de beginwaarde. Vuistregel: $t_2 \approx 70/p$.
- Met logaritmen haal je een onbekende uit de exponent: $g^t = k$ geeft $t = \log k / \log g$.
- Bij afkoelen daalt het verschil met de omgeving exponentieel; een condensator ontlaadt met $e^{-t/RC}$; radioactieve kernen vervallen met een vaste halveringstijd.
- Exponentiële groei wint altijd van lineaire, maar lijkt lang onschuldig: de laatste verdubbeling is groter dan alles ervoor.
- Op een logaritmische schaal zijn gelijke afstanden gelijke factoren en wordt een exponentiële functie een rechte; uit de helling lees je de groeifactor.
- Geen groei is eeuwig exponentieel: voeding, licht of ruimte begrenzen ze. Het model geldt in de groeifase.

BEGRIPPEN

Wat je nu kent

exponentiële functie
Functie $N(t) = N_0 \cdot g^t$: per tijdstap wordt de grootheid met dezelfde factor g vermenigvuldigd.

groeifactor
De factor g waarmee een grootheid per tijdstap vermenigvuldigd wordt; $g > 1$ is groei, $0 < g < 1$ is afname.

procentuele groei
De groei per tijdstap in procent van de vorige waarde: $p = (g - 1) \cdot 100\,\%$.

verdubbelingstijd
De tijd t_2 waarin een exponentieel groeiende grootheid verdubbelt; $t_2 = \log 2 / \log g$.

halveringstijd
De tijd $t_{1/2}$ waarin een exponentieel afnemende grootheid halveert; $t_{1/2} = \log \frac{1}{2} / \log g$.

logaritme
De exponent: $\log y$ is het getal x waarvoor $10^x = y$. Met logaritmen reken je een exponent terug.

logaritmische schaal
As waarop gelijke afstanden gelijke factoren zijn (1, 10, 100 …); een exponentiële functie wordt er een rechte.

tijdconstante
Bij $e^{-t/\tau}$ de tijd τ waarna nog $1/e \approx 37\,\%$ over is; bij een RC-kring $\tau = R \cdot C$.

radioactief verval
Het willekeurig uiteenvallen van instabiele atoomkernen; per halveringstijd vervalt de helft van de overblijvende kernen.

becquerel
Eenheid van activiteit: één verval per seconde (Bq).

HANDBOEK · HOOFDSTUK 6

Goniometrie en vectoren

Hoeken en pijlen: van de helling van een oprit tot de hoek van de zon op het zonnepaneel

Hoe steil is 22 %? Met welke kracht glijdt een kist van een helling? Hoe hoog staat de zon in december, en onder welke hoek zet je een zonnepaneel? Allemaal vragen over **hoeken** en **richtingen**. In dit hoofdstuk leer je rekenen met sinus, cosinus en tangens in de rechthoekige driehoek, en werk je met **vectoren**: grootheden met een grootte én een richting, zoals kracht en snelheid. Je telt ze op, je ontbindt ze in stukken, en je kiest met echte zonnedata de beste hellingshoek voor het zonnepaneel van Habitat.

9× uitleg · 7× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

W Wiskundig modelleren · K Krachten, beweging en constructies · E Energie

In december staat de middagzon in België maar 15,6° boven de horizon: een steil paneel vangt haar het best.

Na dit hoofdstuk

- Je kan in een rechthoekige driehoek de zijden benoemen ten opzichte van een hoek en sinus, cosinus en tangens als verhoudingen schrijven.
- Je kan met sinus, cosinus en tangens een onbekende zijde of hoek berekenen, en de rekenmachine correct gebruiken (graden, inverse functies).
- Je kan een helling omrekenen van procent naar graden en omgekeerd, en uitleggen waarom 100 % geen verticale wand is.
- Je kan vectoren tekenen, optellen met de kop-staartmethode en de parallellogrammethode, en het verschil met gewone getallen uitleggen.
- Je kan een vector ontbinden in componenten en vectoren optellen met componenten, met grootte en richting van de resultante.
- Je kan de zwaartekracht op een helling ontbinden in een component langs en een component loodrecht op de helling.
- Je kan de zonnehoogte op het middaguur berekenen uit de breedtegraad en de declinatie, en er schaduwlengtes en de invalshoek op een zonnepaneel mee bepalen.
- Je kan een hellingshoek voor een zonnepaneel kiezen en verantwoorden met een afweging tussen jaaropbrengst en winteropbrengst.

UITLEG · 6.1

Wat je al weet, en wat nu komt

1/22

Uit de wiskundeles ken je de **stelling van Pythagoras**: in een rechthoekige driehoek is het kwadraat van de schuine zijde gelijk aan de som van de kwadraten van de twee rechthoekszijden. Je kent **gelijkvormige driehoeken**: dezelfde hoeken, dus dezelfde verhoudingen tussen de zijden. In hoofdstuk F11 tekende je een **kracht als pijl**, met een grootte, een richting en een aangrijpingspunt; in hoofdstuk F13 zag je dat een driehoek een stijve vorm is.

Pythagoras verbindt de drie *zijden*, maar zegt niets over de *hoeken*. Hoe steil is een oprit van 6,40 m lang die 40 cm stijgt? Onder welke hoek staat een ladder? Hoe hoog staat de zon als een stok van 1 m een schaduw van 1,23 m geeft? Daarvoor heb je de **goniometrie** nodig ('hoekmeting'): sinus, cosinus en tangens. In dit hoofdstuk

- reken je zijden en hoeken uit in een rechthoekige driehoek;
- zet je een helling om van procent naar graden en terug;
- werk je met **vectoren**: optellen, en ontbinden in componenten;
- pas je alles toe op de kracht op een helling en op de hoek van de zon.

Een helling is een rechthoekige driehoek. Pythagoras geeft de lengte, maar voor de steilheid heb je de hoek α nodig.

VERHAAL · 6.2

Eratosthenes meet de aarde met een schaduw

2/22

Rond 240 voor onze tijdrekening was **Eratosthenes** hoofd van de beroemde bibliotheek van Alexandrië, in Egypte. Volgens de overlevering hoorde hij dat de zon in Syene (het huidige Aswan), ver naar het zuiden, op de langste dag van het jaar om 12 uur recht boven stond: een verticale stok gaf er geen schaduw, en de zon scheen tot op de bodem van een diepe put. In Alexandrië gaf een verticale stok op datzelfde moment wel een schaduw.

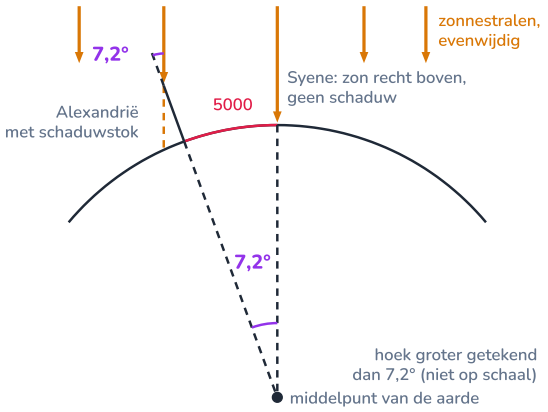
Eratosthenes redeneerde met twee aannames: de aarde is een bol, en de zon staat zo ver weg dat haar stralen evenwijdig invallen. Dan is de hoek tussen de stok in Alexandrië en de zonnestralen even groot als de hoek tussen de twee steden, gemeten vanuit het middelpunt van de aarde (evenwijdige lijnen,

gesneden door dezelfde rechte: overeenkomstige hoeken zijn gelijk). Die hoek mat hij: $\frac{1}{50}$ van een volle cirkel, 7,2°. De afstand tussen de twee steden schatte men op 5000 stadiën. Dan is de omtrek van de aarde $50 \cdot 5000 = 250\,000$ stadiën.

Hoe goed was dat? Niemand weet precies welke stadie Eratosthenes gebruikte; met de waarden die historici voorstellen (tussen ongeveer 157 en 185 m) kom je uit tussen 39 000 en 46 000 km. De echte omtrek over de polen is 40 008 km. Zijn methode was juist; de onzekerheid zat in de afstand en in het feit dat Syene niet precies ten zuiden van Alexandrië ligt.

Wat je uit dit verhaal meeneemt

Met een stok, een schaduw en een hoek mat iemand zonder ruimtevaart de omvang van de hele planeet. Hetzelfde idee gebruik je verderop: de lengte van een schaduw geeft de zonnehoogte, en de zonnehoogte geeft de beste hoek voor een zonnepaneel. In opdracht 6.15 doe je de meting van Eratosthenes over met twee scholen.



Evenwijdige zonnestralen: de hoek van de schaduw in Alexandrië is de hoek tussen de steden in het middelpunt van de aarde.

UITLEG · 6.3

Sinus, cosinus en tangens

3/22

Neem een rechthoekige driehoek en kies een van de scherpe hoeken, α. Ten opzichte van die hoek hebben de zijden een naam: de **schuine zijde** (tegenover de rechte hoek, de langste), de **overstaande rechthoekszijde** (tegenover α) en de **aanliggende rechthoekszijde** (die samen met de schuine zijde de hoek α vormt).

Alle rechthoekige driehoeken met dezelfde hoek α zijn **gelijkvormig**: groter of kleiner, maar met dezelfde verhoudingen tussen hun zijden. Die verhoudingen hangen dus alleen van α af, en krijgen een naam:

SINUS, COSINUS EN TANGENS

$$\sin\alpha = \frac{\text{overstaande}}{\text{schuine}} \qquad \cos\alpha = \frac{\text{aanliggende}}{\text{schuine}} \qquad \tan\alpha = \frac{\text{overstaande}}{\text{aanliggende}}$$

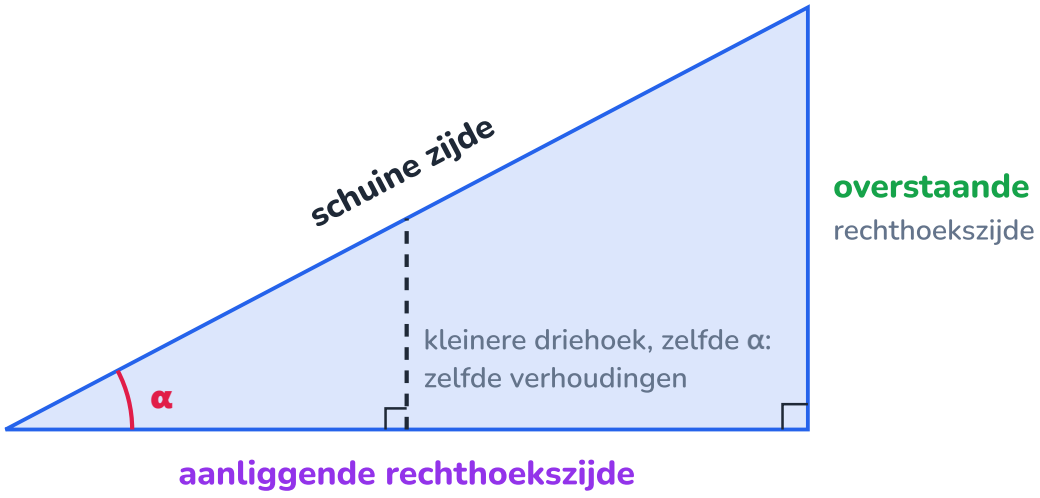
Ezelsbruggetje: **SOS CAS TOA** (Sinus = Overstaande / Schuine, …). Sinus en cosinus liggen tussen 0 en 1, want de schuine zijde is de langste. De tangens kan elk positief getal zijn.

Drie handige verbanden volgen er meteen uit:

- $\tan\alpha = \dfrac{\sin\alpha}{\cos\alpha}$ (deel teller en noemer door de schuine zijde);
- $\sin^2\alpha + \cos^2\alpha = 1$ (dat is Pythagoras, gedeeld door het kwadraat van de schuine zijde);

- de twee scherpe hoeken zijn samen 90°, en $\sin\alpha = \cos(90° - \alpha)$: wat voor de ene hoek overstaand is, is voor de andere aanliggend.

Een zijde berekenen. Ken je een hoek en een zijde, dan kies je de verhouding waarin de gekende en de gezochte zijde voorkomen. Een dak van 5,0 m lang staat onder 35°: het hoogteverschil is $5{,}0 \cdot \sin 35° = 2{,}87\,\text{m}$ en de horizontale overspanning $5{,}0 \cdot \cos 35° = 4{,}10\,\text{m}$.



De namen van de zijden hangen af van de hoek waarvan je vertrekt. De kleinere driehoek met dezelfde hoek heeft dezelfde verhoudingen.

UITLEG · 6.4

Een hoek terugrekenen, en exacte waarden

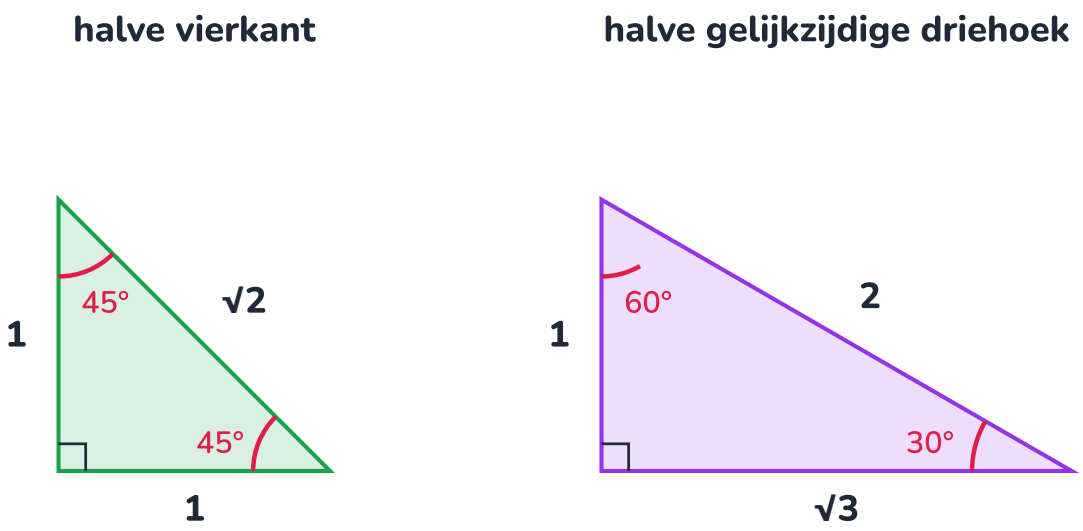
4/22

Ken je twee zijden, dan ken je een verhouding, en zoek je de hoek die erbij hoort. Daarvoor heeft je rekenmachine de **inverse functies** $\sin^{-1}$, $\cos^{-1}$ en $\tan^{-1}$ (ook boogsinus, boogcosinus en boogtangens genoemd; op een rekenblad en in Python: ASIN, ACOS, ATAN). Een oprit die 40 cm stijgt over 6,40 m horizontaal: $\tan\alpha = 0{,}40/6{,}40 = 0{,}0625$, dus $\alpha = \tan^{-1}(0{,}0625) = 3{,}58°$.

> ⚠ **Graden of radialen?**
> Een rekenmachine kan hoeken meten in graden (DEG) of in radialen (RAD), een andere eenheid waarbij een volle cirkel 2π is. Staat hij op RAD, dan is $\sin 30$ gelijk aan $-0{,}988$ in plaats van $0{,}5$. Controleer vóór een toets: $\sin 30 = 0{,}5$? Python en rekenbladen werken altijd in radialen: gebruik `math.radians()` of `RADIANS()` (opdracht 6.12).

Voor 30°, 45° en 60° bestaan **exacte waarden**, die je uit twee eenvoudige driehoeken haalt. Een vierkant met zijde 1, doormidden gesneden langs de diagonaal, geeft een driehoek met twee hoeken van 45° en een schuine zijde $\sqrt{2}$. Een gelijkzijdige driehoek met zijde 2, doormidden gesneden, geeft hoeken van 30° en 60°, rechthoekszijden 1 en $\sqrt{3}$.

α	0°	30°	45°	60°	90°
$\sin\alpha$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2} \approx 0{,}707$	$\frac{\sqrt{3}}{2} \approx 0{,}866$	1
$\cos\alpha$	1	$\frac{\sqrt{3}}{2} \approx 0{,}866$	$\frac{\sqrt{2}}{2} \approx 0{,}707$	$\frac{1}{2}$	0
$\tan\alpha$	0	$\frac{\sqrt{3}}{3} \approx 0{,}577$	1	$\sqrt{3} \approx 1{,}732$	bestaat niet



Links een half vierkant (45°), rechts een halve gelijkzijdige driehoek (30° en 60°). Lees de exacte waarden af als verhoudingen.

UITGEWERKT VOORBEELD · 6.5

Een ladder veilig plaatsen

5/22

Een ladder staat veilig als de voet op een kwart van de werkhoogte van de muur staat: 1 m uit de muur per 4 m hoogte (de '1-op-4-regel'). Onder welke hoek met de grond staat de ladder dan? Een ladder van 4,0 m wordt zo tegen een muur gezet. Hoe hoog reikt hij, en hoe ver staat de voet van de muur?

GEGEVEN

- verhouding hoogte : afstand = 4 : 1
- ladderlengte $s = 4{,}0\,\text{m}$

GEVRAAGD

- hoek α met de grond
- hoogte en afstand tot de muur

OPLOSSING

1. Muur, grond en ladder vormen een rechthoekige driehoek. Ten opzichte van de hoek α aan de grond is de hoogte de overstaande en de afstand de aanliggende rechthoekszijde.
2. $\tan\alpha = \dfrac{4}{1} = 4$, dus $\alpha = \tan^{-1} 4 = 76{,}0°$.
3. De hoek hangt niet af van de lengte van de ladder: elke ladder volgens de regel staat onder 76°.
4. Hoogte: $4{,}0 \cdot \sin 76{,}0° = 3{,}88\,\text{m}$. Afstand: $4{,}0 \cdot \cos 76{,}0° = 0{,}97\,\text{m}$.

ANTWOORD

De ladder staat onder ongeveer 76° met de grond; hij reikt tot 3,88 m hoogte, met de voet 0,97 m van de muur.

Klopt dit? Pythagoras: $3{,}88^2 + 0{,}97^2 = 15{,}05 + 0{,}94 = 15{,}99 \approx 4{,}0^2$. En de verhouding: $3{,}88/0{,}97 = 4{,}0$. Veiligheidsregels vermelden vaak 'ongeveer 75°': flauwer schuift de voet weg, steiler kiept de ladder achterover. Waarom precies, zie je met krachten en momenten in hoofdstuk 10.

UITLEG · 6.6

Hellingen in procent en in graden

6/22

Een helling van 10 % betekent: 10 m stijgen per 100 m **horizontaal**. Dat is de tangens van de hellingshoek, maal 100:

HELLING IN PROCENT EN IN GRADEN

$$p = 100\,\% \cdot \tan\alpha \qquad \alpha = \tan^{-1}\left(\frac{p}{100\,\%}\right)$$

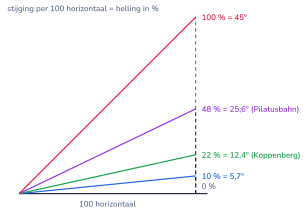
Bij kleine hellingen maakt het weinig uit of je per 100 m horizontaal of per 100 m langs de weg meet (bij 10 % geeft de tweede manier 9,95 % in plaats van 10 %); bij steile hellingen wel.

helling	hoek	voorbeeld
2 %	1,1°	spoorlijn, riolering (vaak nog minder)
5 %	2,9°	grens voor een lange oprit voor rolstoelen
10 %	5,7°	steile weg; zwaar voor een fietser
22 %	12,4°	steilste stuk van de Koppenberg (Ronde van Vlaanderen)
48 %	25,6°	Pilatusbahn (Zwitserland), de steilste tandradbaan ter wereld
100 %	45°	even hoog als lang: een trap is vaak nog steiler

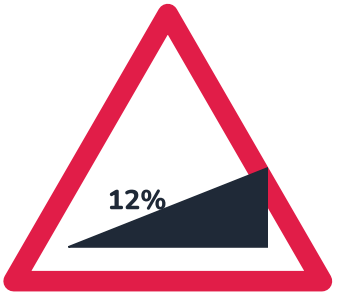
Gevaarsborden voor een steile helling of een gevaarlijke daling vermelden het hellingspercentage. Spoorwegen en rioleringen rekenen soms in promille (‰, per 1000): een rioolbuis van 5 ‰ daalt 5 mm per meter.

100 % is verticaal

100 % betekent 100 m stijgen per 100 m horizontaal: $\tan\alpha = 1$, dus 45°. Een verticale wand heeft geen horizontale afstand: zijn helling in procent is oneindig groot. En een helling van 20 % is niet 'twee keer zo steil als 10 %' in graden: 11,3° tegenover 5,7°, bijna dubbel, maar bij grote hellingen wijkt het verband duidelijk af van evenredig (100 % is 45°, 200 % is 63,4°).



Vier hellingen op schaal: 10 %, 22 %, 48 % en 100 %. De hoek groeit trager dan het percentage.



Een gevaarsbord voor een steile helling: het getal is de helling in procent.

Op zo'n kasseienhelling voelen 22 % en 12,4° heel anders dan ze op papier lijken.

UITGEWERKT VOORBEELD · 6.7

Habitat: een toegankelijke oprit naar de module

7/22

De vloer van de klimaatmodule ligt 40 cm boven het terrein. Iedereen moet binnen kunnen, ook met een rolstoel of een zware kar. In Vlaanderen legt de verordening toegankelijkheid de grootste helling vast

volgens het hoogteverschil (tabel). Hoe lang wordt de oprit, onder welke hoek, en is er een tussenbordes nodig?

te overbruggen hoogteverschil	helling hoogstens
tot 10 cm	10 %
10 tot 25 cm	8,3 %
25 tot 35 cm	7 %
35 tot 50 cm	6,25 %
meer dan 50 cm	5 %

Per 50 cm hoogteverschil, of na 10 m helling, komt er een vlak tussenbordes van minstens 1,50 m. (Vlaamse stedenbouwkundige verordening toegankelijkheid, artikel 19.)

Op schaal getekend is een helling van 6,25 % een bijna vlakke wig: 40 cm over 6,40 m.

GEGEVEN

- hoogteverschil $0{,}40\,\text{m}$
- de tabel met de toegelaten hellingen

GEVRAAGD

- toegelaten helling, horizontale lengte, lengte langs de helling, hoek

OPLOSSING

1. 40 cm valt in de klasse 35 tot 50 cm: hoogstens 6,25 %.
2. Horizontale lengte: $\tan\alpha = 0{,}0625 = \dfrac{0{,}40\,\text{m}}{x}$, dus $x = \dfrac{0{,}40}{0{,}0625} = 6{,}40\,\text{m}$.
3. Hoek: $\alpha = \tan^{-1}(0{,}0625) = 3{,}58°$.
4. Lengte langs de helling: $\dfrac{0{,}40}{\sin 3{,}58°} = 6{,}41\,\text{m}$ (of met Pythagoras: $\sqrt{6{,}40^2 + 0{,}40^2} = 6{,}41\,\text{m}$).
5. Tussenbordes: het hoogteverschil is kleiner dan 50 cm en de helling korter dan 10 m: niet nodig. Wel een vlak bordes van minstens 1,50 m bovenaan, voor de deur, om te draaien.

ANTWOORD

Een oprit van 6,40 m horizontaal (6,41 m langs de helling) onder 3,6°, zonder tussenbordes, met een vlak bordes aan de deur.

Klopt dit? 3,6° lijkt weinig, maar voor iemand die zichzelf in een rolstoel omhoog duwt, is 6 % al zwaar werk: daarom wordt de grens strenger naarmate het hoogteverschil groter is. Hoeveel kracht er nodig is om een kar van 80 kg over deze oprit te duwen, reken je uit in het voorbeeld over de batterijkar.

UITLEG · 6.8

Vectoren: grootte en richting

8/22

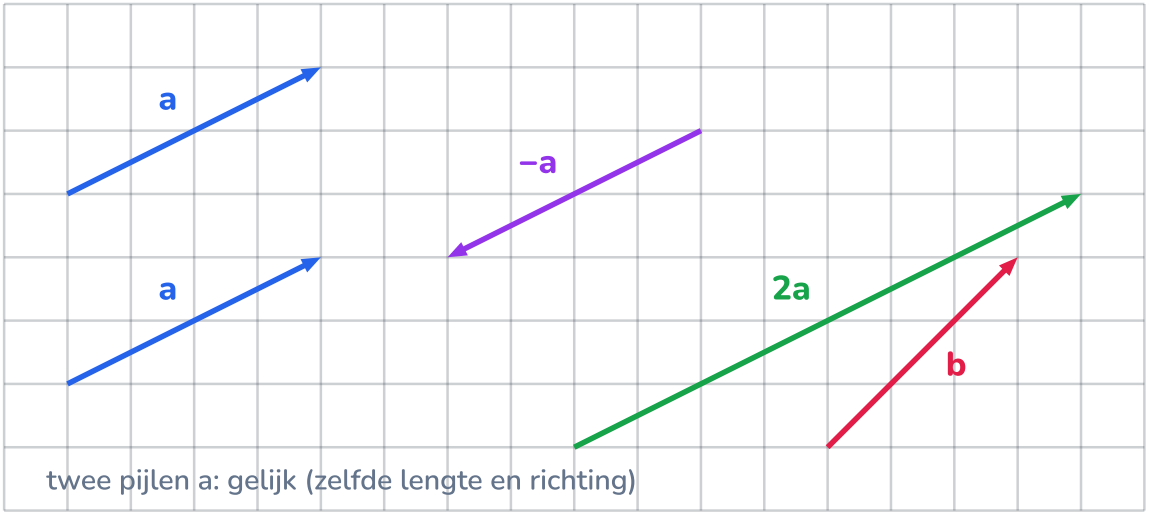
Een massa van 5 kg, een temperatuur van 20 °C, een energie van 300 J: die grootheden liggen volledig vast met een getal en een eenheid. Het zijn **scalairen**. Een kracht van 50 N is niet volledig beschreven: omhoog of opzij? Een snelheid van 3 m/s: naar het noorden of naar het oosten? Zulke grootheden hebben een **grootte** en een **richting**. Het zijn **vectoren**.

Je tekent een vector als een **pijl**: de lengte is de grootte (op een gekozen schaal), de pijl wijst de richting. In tekst schrijf je een vector met een pijltje, $\vec{F}$, en de grootte zonder pijltje, F (of $|\vec{F}|$). Voorbeelden: kracht, snelheid, verplaatsing, versnelling, het elektrisch veld.

- Twee vectoren zijn **gelijk** als ze dezelfde grootte en dezelfde richting hebben, waar ze ook getekend staan. (Bij krachten telt ook het aangrijpingspunt voor het draaien, hoofdstuk 10.)
- De **tegengestelde vector** $-\vec{a}$ heeft dezelfde grootte en de omgekeerde richting.
- Een vector maal een getal: $2\vec{a}$ is twee keer zo lang, zelfde richting; $-2\vec{a}$ twee keer zo lang, omgekeerde richting.

Vectoren tel je op als getallen

Twee krachten van 30 N en 40 N geven samen niet altijd 70 N. In dezelfde richting wel; tegengesteld 10 N; loodrecht op elkaar 50 N (Pythagoras). De som van twee vectoren hangt af van hun richtingen, en ligt altijd tussen het verschil en de som van hun groottes.



De twee pijlen a zijn dezelfde vector; –a wijst de andere kant op, 2a is dubbel zo lang, b heeft een andere richting.

UITLEG · 6.9

Vectoren optellen: kop-staart en parallellogram

9/22

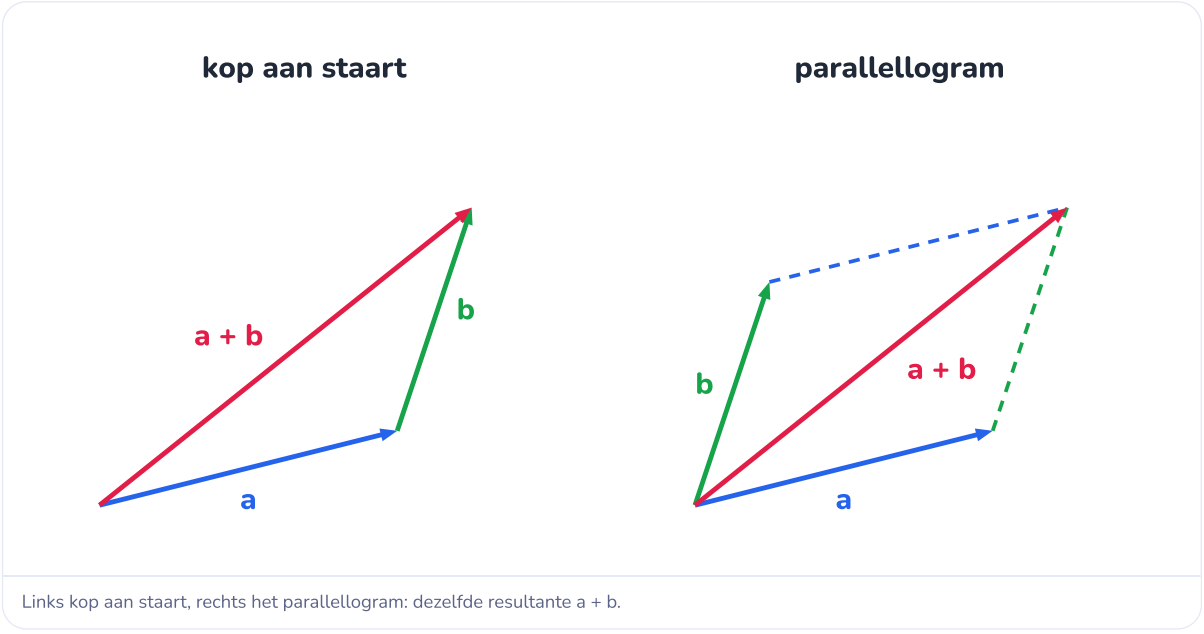
Loop 4 stappen naar rechts en 1 omhoog, en daarna 1 naar rechts en 3 omhoog. Waar kom je uit? De verplaatsingen tel je op door ze **achter elkaar** te zetten. Dat is de **kop-staartmethode**: zet de staart van $\vec{b}$ aan de kop van $\vec{a}$. De **resultante** $\vec{a}+\vec{b}$ loopt van de staart van $\vec{a}$ naar de kop van $\vec{b}$.

Bij krachten die in hetzelfde punt aangrijpen, is de **parallellogrammethode** natuurlijker: teken beide pijlen vanuit hetzelfde punt, vul aan tot een parallellogram, en de diagonaal vanuit dat punt is de som.

Het is dezelfde constructie: de overstaande zijden van een parallellogram zijn gelijke vectoren.

- De volgorde maakt niet uit: $\vec{a} + \vec{b} = \vec{b} + \vec{a}$ (de twee helften van het parallellogram).
- Meer dan twee vectoren: blijf kop aan staart zetten. Komt de laatste kop terug bij de eerste staart, dan is de som nul: de krachten zijn in evenwicht (hoofdstuk 8).
- Aftrekken is de tegengestelde optellen: $\vec{a} - \vec{b} = \vec{a} + (-\vec{b})$.

Op ruitjespapier met een goede schaal kan je de resultante **meten** (lengte met een lat, richting met een gradenboog). Nauwkeuriger is **rekenen**: staan de twee vectoren loodrecht op elkaar, dan geeft Pythagoras de grootte en de tangens de richting. Anders ontbind je ze eerst (twee bladen verder).



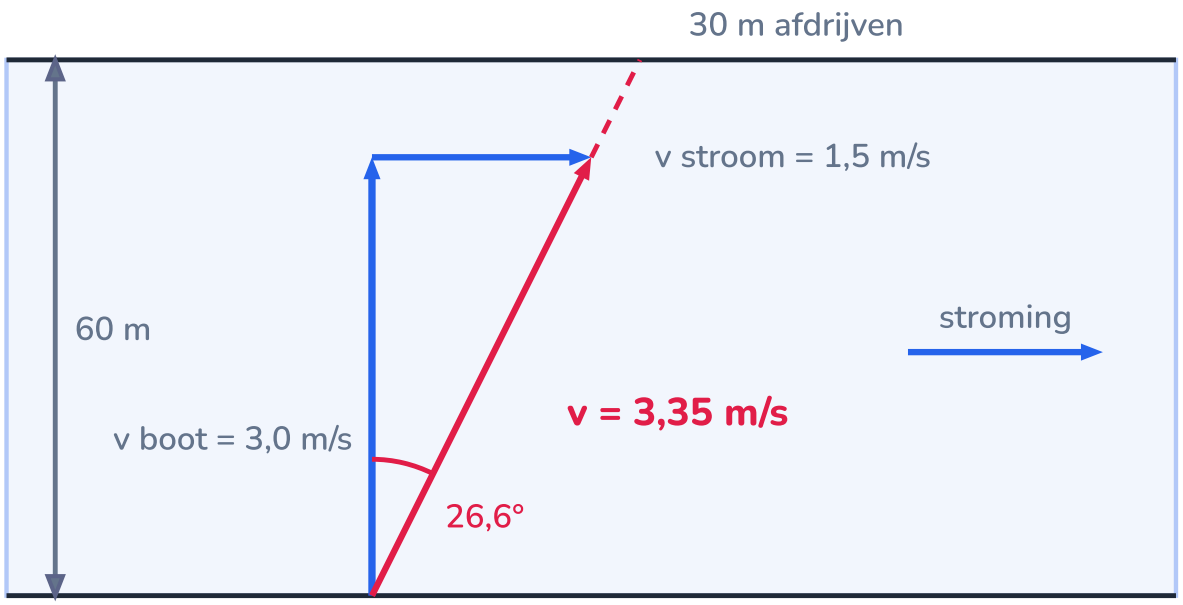
Links kop aan staart, rechts het parallellogram: dezelfde resultante a + b.

UITGEWERKT VOORBEELD · 6.10

Een boot steekt een rivier over

10/22

Een motorboot vaart ten opzichte van het water aan 3,0 m/s, met de neus loodrecht op de oever. De rivier is 60 m breed en stroomt aan 1,5 m/s. Wat is de snelheid van de boot ten opzichte van de oever, onder welke hoek beweegt hij, hoe lang duurt de oversteek, en hoe ver drijft hij af?



De snelheid van de boot en die van de stroming geven samen een schuine baan: 30 m afdrijven over 60 m breedte.

GEGEVEN

- $v_{\text{boot}} = 3{,}0\,\text{m/s}$ loodrecht op de oever
- $v_{\text{stroom}} = 1{,}5\,\text{m/s}$ langs de oever
- breedte 60 m

GEVRAAGD

- $\vec{v}$ ten opzichte van de oever (grootte, hoek)
- tijd en afdrijven

OPLOSSING

1. De snelheid ten opzichte van de oever is de som van de twee vectoren: $\vec{v} = \vec{v}_{\text{boot}} + \vec{v}_{\text{stroom}}$. Kop aan staart vormen ze een rechthoekige driehoek.
2. Grootte (Pythagoras): $v = \sqrt{3{,}0^2 + 1{,}5^2} = \sqrt{11{,}25} = 3{,}35\,\text{m/s}$.
3. Hoek met de loodlijn op de oever: $\tan\alpha = \dfrac{1{,}5}{3{,}0} = 0{,}5$, dus $\alpha = 26{,}6°$ stroomafwaarts.
4. Tijd: de stroming draagt niets bij in de richting dwars op de oever. De boot overbrugt de 60 m aan 3,0 m/s: $t = 60/3{,}0 = 20\,\text{s}$.
5. Afdrijven: in die 20 s neemt de stroming de boot mee over $1{,}5 \cdot 20 = 30\,\text{m}$.

ANTWOORD

De boot beweegt aan 3,35 m/s onder 26,6° met de loodlijn, is na 20 s aan de overkant, en komt 30 m stroomafwaarts aan land.

Klopt dit? De lengte van de schuine baan is $\sqrt{60^2 + 30^2} = 67{,}1\,\text{m}$, en $67{,}1/3{,}35 = 20\,\text{s}$: klopt. Wil de schipper recht naar de overkant, dan moet hij de neus schuin stroomopwaarts richten (opdracht 6.7).

UITLEG · 6.11

Ontbinden in componenten

Elke vector kan je schrijven als de som van twee loodrechte vectoren langs gekozen assen, meestal horizontaal en verticaal: zijn **componenten**. Maakt $\vec{F}$ een hoek α met de x-as, dan vormen $\vec{F}$ en zijn componenten een rechthoekige driehoek:

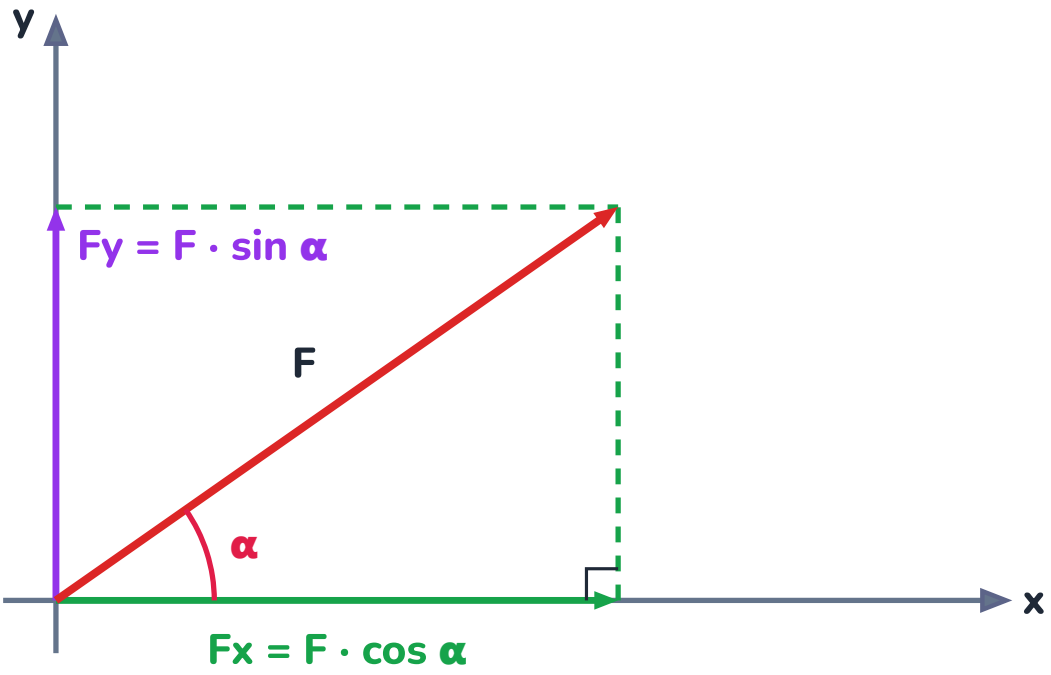
COMPONENTEN VAN EEN VECTOR

$$F_x = F\cos\alpha \qquad F_y = F\sin\alpha \qquad F = \sqrt{F_x^2 + F_y^2} \qquad \tan\alpha = \frac{F_y}{F_x}$$

Ontbinden (links) en terug samenstellen (rechts). Let bij $\tan^{-1}$ op het kwadrant: wijst de vector naar links ($F_x < 0$), tel dan 180° bij de hoek die de rekenmachine geeft, of redeneer met een schets.

Het grote voordeel: **vectoren optellen wordt getallen optellen**. De x-component van de som is de som van de x-componenten, en zo ook voor y. Daarna geeft Pythagoras de grootte van de resultante en de tangens haar richting. Zo tel je drie, vier of tien krachten op zonder te tekenen, en zo doen computers het ook.

Welke assen? Je mag ze vrij kiezen, zolang ze loodrecht staan. Kies ze slim: langs de beweging en loodrecht erop. Bij een voorwerp op een helling leg je de ene as langs de helling (volgende bladen).

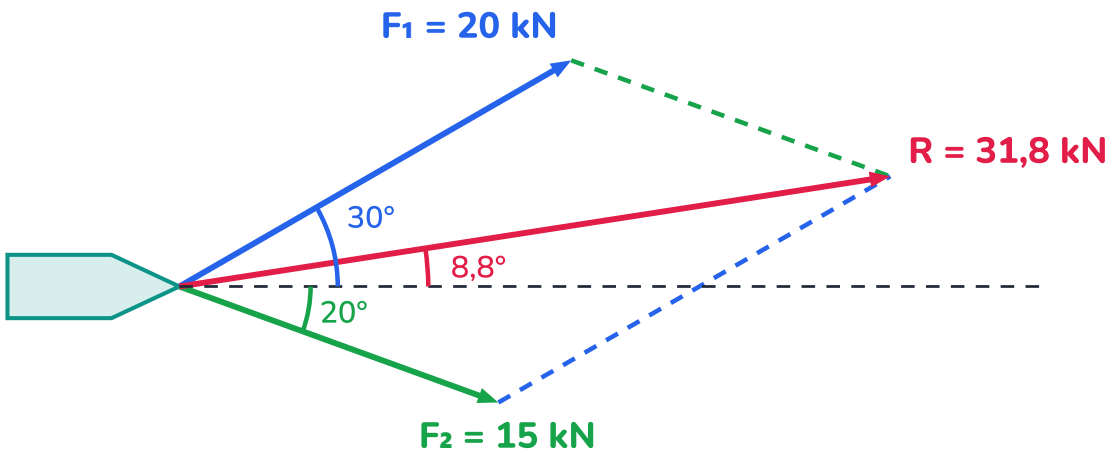


De kracht F is de diagonaal van de rechthoek met zijden Fx en Fy.

UITGEWERKT VOORBEELD · 6.12

Twee sleepboten trekken aan een schip

Twee sleepboten trekken een schip de haven in. Sleepboot 1 trekt met 20 kN onder 30° boven de vaarrichting, sleepboot 2 met 15 kN onder 20° eronder. Bereken de resultante kracht: hoe groot, en in welke richting?



schaal: 1 cm op de tekening = 10 kN (10 px per kN)

Het parallellogram van de twee krachten: de diagonaal is de resultante van 31,8 kN.

GEGEVEN

- $F_1 = 20\,\text{kN}$ onder $+30°$
- $F_2 = 15\,\text{kN}$ onder $-20°$ (ten opzichte van de vaarrichting, de x-as)

GEVRAAGD

- grootte en richting van $\vec{R} = \vec{F}_1 + \vec{F}_2$

OPLOSSING

1. Componenten van $\vec{F}_1$: $F_{1x} = 20\cos 30° = 17{,}32\,\text{kN}$; $F_{1y} = 20\sin 30° = 10{,}00\,\text{kN}$.
2. Componenten van $\vec{F}_2$: $F_{2x} = 15\cos(-20°) = 14{,}10\,\text{kN}$; $F_{2y} = 15\sin(-20°) = -5{,}13\,\text{kN}$.
3. Optellen per component: $R_x = 17{,}32 + 14{,}10 = 31{,}42\,\text{kN}$; $R_y = 10{,}00 - 5{,}13 = 4{,}87\,\text{kN}$.
4. Grootte: $R = \sqrt{31{,}42^2 + 4{,}87^2} = 31{,}8\,\text{kN}$. Richting: $\tan\alpha = 4{,}87/31{,}42$, dus $\alpha = 8{,}8°$ boven de vaarrichting.

ANTWOORD

De resultante is 31,8 kN onder 8,8° boven de vaarrichting: minder dan 20 + 15 = 35 kN, omdat de sleepboten deels tegen elkaar in trekken.

Klopt dit? Van de 35 kN gaat 31,4 kN in de vaarrichting; de dwarscomponenten ($+10{,}0$ en $-5{,}1\,\text{kN}$) heffen elkaar gedeeltelijk op. Wil de kapitein recht vooruit, dan moet sleepboot 2 de hoek vergroten of harder trekken tot $F_{2y} = -10{,}0\,\text{kN}$. Een tekening op schaal (10 px per kN) geeft dezelfde resultante, met een meetfout van een paar procent.

UITLEG · 6.13

De zwaartekracht op een helling

13/22

Op een blok op een helling werkt de zwaartekracht $F_z = m \cdot g$, recht naar beneden. Voor de beweging langs de helling is het handig die kracht te ontbinden langs twee assen: **langs** de helling en **loodrecht** erop.

ZWAARTEKRACHT OP EEN HELLING, ONTBONDEN

$$F_{\parallel} = m\,g\sin\alpha \qquad F_{\perp} = m\,g\cos\alpha$$

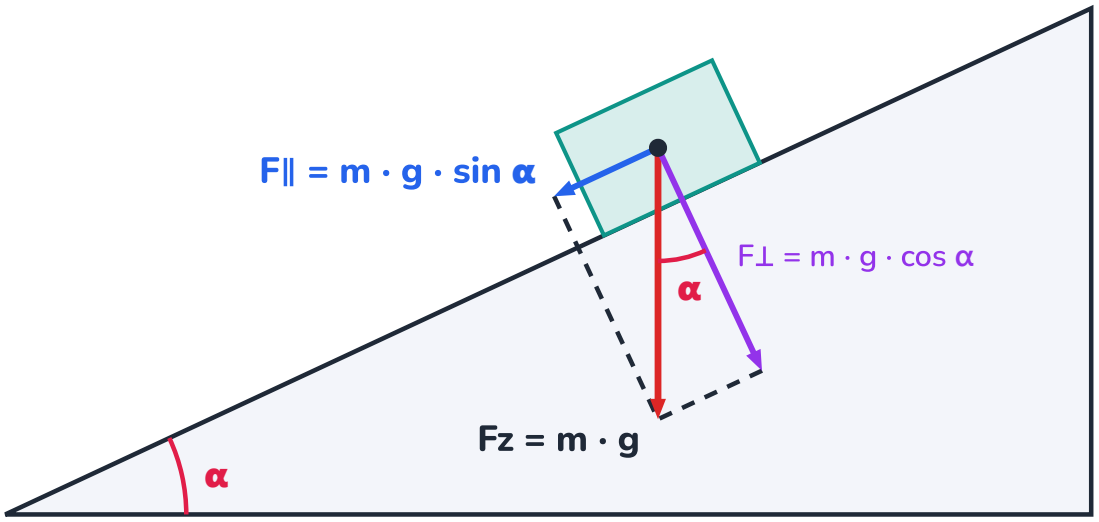
α is de hellingshoek. $F_{\parallel}$ trekt het blok langs de helling naar beneden, $F_{\perp}$ drukt het tegen de helling.

Waarom staat hier α? De zwaartekracht staat loodrecht op de grond, en $F_{\perp}$ staat loodrecht op de helling. Twee hoeken waarvan de benen twee aan twee loodrecht op elkaar staan, zijn gelijk: de hoek tussen $\vec{F}_z$ en $\vec{F}_{\perp}$ is dus de hellingshoek α. Tegenover die hoek ligt $F_{\parallel}$: $F_{\parallel} = F_z \sin\alpha$.

Controleer met de uitersten. Bij $\alpha = 0°$ (vlak) is $F_{\parallel} = 0$ en drukt het hele gewicht op de grond. Bij $\alpha = 90°$ (verticale wand) is $F_{\parallel} = m\,g$: vrije val langs de wand. Bij 45° zijn de twee even groot. Wie een formule met sin en cos niet vertrouwt, test haar zo op 0° en 90°.

Vooruitblik

$F_{\perp}$ wordt gecompenseerd door de **normaalkracht** van de helling op het blok, en $F_{\parallel}$ door **wrijving**, zolang het blok niet glijdt. Hoe groot de wrijving kan worden, en wat er gebeurt als $F_{\parallel}$ wint, is het onderwerp van hoofdstuk 8 (de wetten van Newton). Hoeveel arbeid je levert om iets langs een helling omhoog te duwen, zie je in hoofdstuk 9.



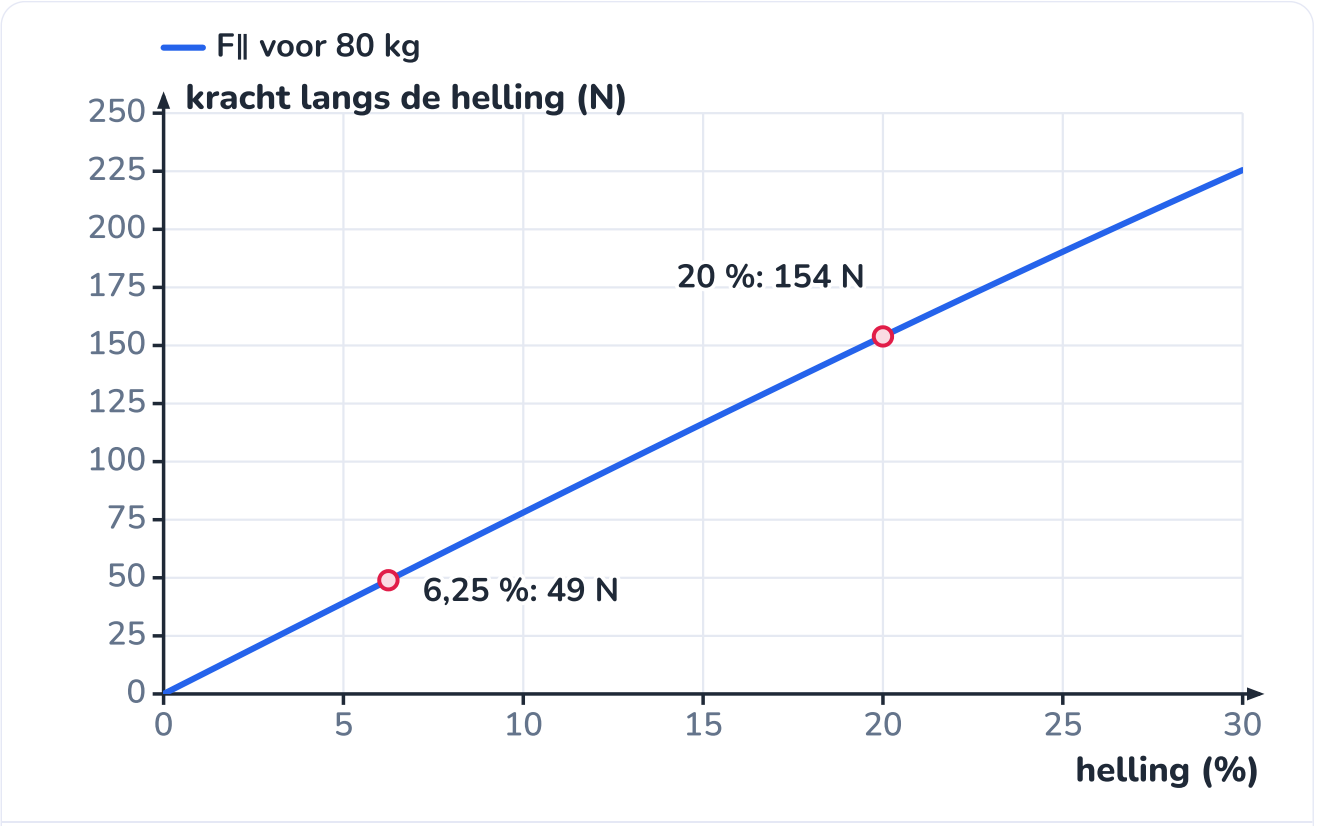
Fz ontbonden langs en loodrecht op de helling. De hoek tussen Fz en F⊥ is de hellingshoek α.

UITGEWERKT VOORBEELD · 6.14

Habitat: de batterijkar op de oprit

14/22

De batterijen van de klimaatmodule worden met een kar naar binnen gereden: kar en batterijen samen 80 kg. Met welke kracht (evenwijdig aan de helling) moet je minstens duwen om de kar met constante snelheid over de oprit van 6,25 % te krijgen, als je wrijving verwaarloost? Vergelijk met een oprit van 20 % en met recht omhoog tillen.



De kracht langs de helling groeit bijna evenredig met het hellingspercentage, zolang de helling klein is.

GEGEVEN

- $m = 80\,\text{kg}$, $g = 9{,}81\,\text{N/kg}$
- hellingen 6,25 % en 20 %

GEVRAAGD

- $F_{\parallel}$ bij beide hellingen
- vergelijking met tillen

OPLOSSING

1. Gewicht: $F_z = 80 \cdot 9{,}81 = 785\,\text{N}$.
2. 6,25 %: $\alpha = \tan^{-1}(0{,}0625) = 3{,}58°$. $F_{\parallel} = 785 \cdot \sin 3{,}58° = 49\,\text{N}$. $F_{\perp} = 785 \cdot \cos 3{,}58° = 783\,\text{N}$: bijna het hele gewicht drukt op de oprit.
3. 20 %: $\alpha = \tan^{-1}(0{,}20) = 11{,}3°$. $F_{\parallel} = 785 \cdot \sin 11{,}3° = 154\,\text{N}$.
4. Met constante snelheid en zonder wrijving moet je duwkracht de component $F_{\parallel}$ precies compenseren (hoofdstuk 8). Recht omhoog tillen vraagt 785 N.

ANTWOORD

Minstens 49 N op de oprit van 6,25 %, 154 N op een oprit van 20 %, tegenover 785 N om recht omhoog te tillen. Een flauwe helling deelt de kracht door 16.

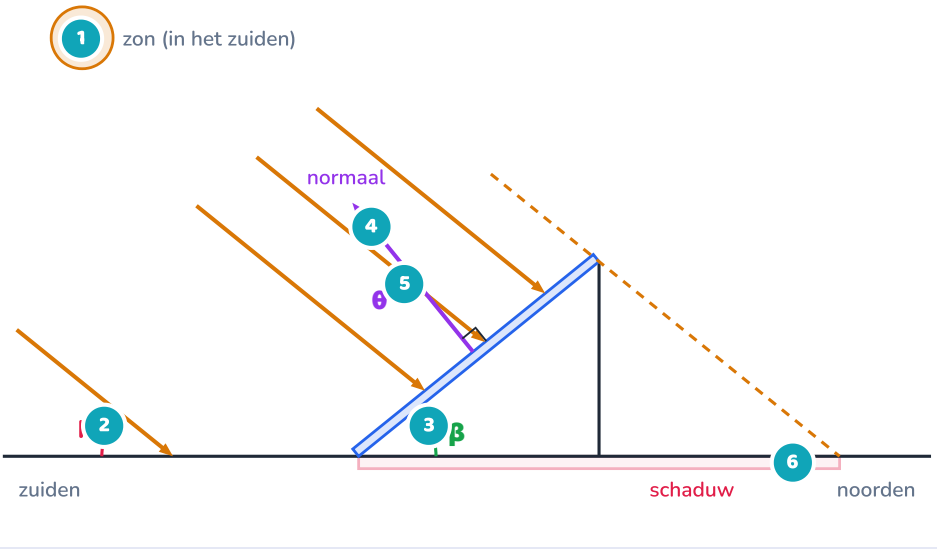
Klopt dit? Er is geen gratis lunch: over de flauwe helling duw je 6,41 m ver, bij tillen hef je 0,40 m. Kracht maal weg is in beide gevallen ongeveer 314 J (49 N × 6,41 m en 785 N × 0,40 m): de arbeid is dezelfde, zoals je in hoofdstuk 9 leert. In werkelijkheid komt er rolweerstand bij, typisch enkele procenten van het gewicht: reken op ongeveer 70 N op de oprit.

ONTLEED · 6.15

De zon en het zonnepaneel

15/22

Hoeveel zonlicht een paneel vangt, hangt af van de hoek waaronder de stralen invallen. De figuur toont de situatie op het middaguur rond 21 maart of 21 september in België: de zon staat 39° hoog in het zuiden, het paneel helt 39°. Wijs de nummers aan.



Zijaanzicht van zuid (links) naar noord (rechts), met de hoeken op schaal.

1 Evenwijdige stralen
De zon staat zo ver weg (150 miljoen km) dat haar stralen op een paneel, een stok of een gebouw evenwijdig invallen, zoals Eratosthenes aannam.

2 Zonnehoogte h
De hoek tussen de zonnestraal en de horizontale grond. Op het middaguur staat de zon in België in het zuiden, op zijn hoogst van die dag.

3 Hellingshoek β
De hoek tussen het paneel en de horizontale. Het paneel kijkt naar het zuiden; zijn lage rand ligt aan de zuidkant.

4 De normaal
De lijn loodrecht op het paneel. Ze maakt een hoek β met de verticale, dus een hoek 90° − β met de horizontale: hier 51°.

5 Invalshoek θ
De hoek tussen de zonnestraal en de normaal: $\theta = |(90^\circ - \beta) - h| = |51^\circ - 39^\circ| = 12^\circ$. Het paneel vangt een fractie $\cos\theta = \cos 12^\circ = 0{,}98$ van wat een paneel loodrecht op de stralen zou vangen. Bij $\theta = 0^\circ$ vangt het alles.

6 Schaduw
Achter het paneel ligt een schaduw. Een punt op hoogte H werpt een schaduw tot op $H / \tan h$ verder: hoe lager de zon, hoe langer de schaduw. Belangrijk als je rijen panelen achter elkaar zet.

UITLEG · 6.16

Hoe hoog staat de zon?

16/22

De aardas staat 23,44° schuin ten opzichte van de baan van de aarde rond de zon. Daardoor verschuift het punt waar de zon op het middaguur loodrecht boven staat in de loop van het jaar: rond 21 juni ligt het op 23,44° noorderbreedte (de Kreeftskeerkring), rond 21 december op 23,44° zuiderbreedte (de Steenbokskeerkring), en rond 21 maart en 23 september op de evenaar. Die breedtegraad heet de **declinatie** δ van de zon. Ze ligt altijd tussen −23,44° en +23,44°.

ZONNEHOOGTE OP HET MIDDAGUUR (NOORDELIJK HALFROND, BUITEN DE TROPEN)

$$h = 90^\circ - \varphi + \delta$$

φ is de breedtegraad van de plaats, δ de declinatie van de zon op die dag. Op de evenaar ($\varphi = 0$) staat de zon bij de equinox loodrecht ($h = 90^\circ$).

Voor België ($\varphi \approx 51^\circ$):

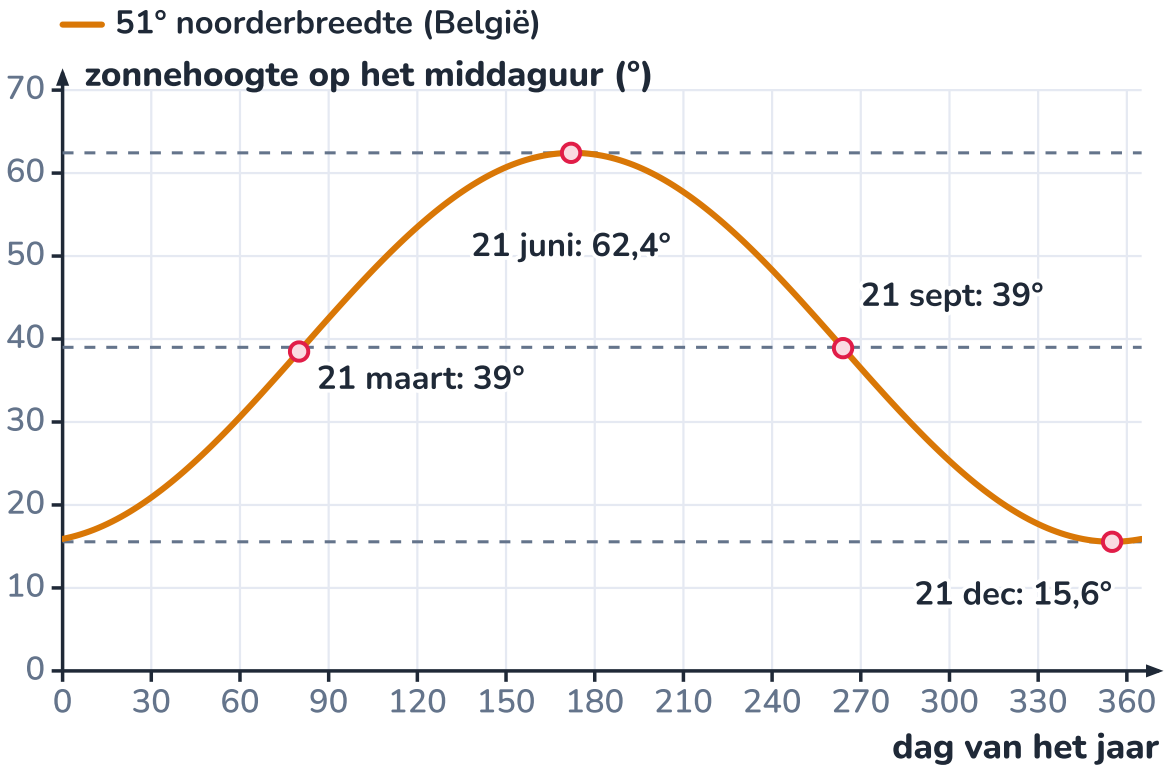
datum (ongeveer)	declinatie δ	zonnehoogte h	schaduw van een stok van 1 m
21 juni	+23,44°	62,4°	0,52 m
21 maart, 23 september	0°	39,0°	1,23 m
21 december	−23,44°	15,6°	3,59 m

De schaduw van een stok met hoogte H is $H / \tan h$. Omgekeerd meet je de zonnehoogte met een stok en een rolmeter: $h = \tan^{-1}(H/\text{schaduw})$ (volgende blad).

⚠ **Middaguur is niet 12 uur**

Het **middaguur** is het moment waarop de zon precies in het zuiden staat, op haar hoogst. In België valt dat niet om 12 uur op de klok: in de winter ongeveer tussen 12.25 en 13.00 uur, in de zomer (zomertijd) rond 13.30 à 13.50 uur. De verschuiving komt door de tijdzone, de zomertijd en de ellipsvormige baan van de aarde.

Wie de declinatie op elke dag wil, kan een benadering gebruiken: $\delta \approx -23{,}44^\circ \cdot \cos\left(\frac{360^\circ}{365} \cdot (N + 10)\right)$, met N het dagnummer (1 januari = 1). Ze zit hoogstens ongeveer 1,3° naast de echte waarde. Hoe je een cosinus van een hoek groter dan 90° uitrekent, leer je later; de rekenmachine en Python doen het al.



De middagzon in België schommelt tussen 15,6° en 62,4°: een verschil van 47°, twee keer de schuine stand van de aardas.

PROEF · 6.17

De zonnehoogte meten met een schaduwstok

17/22

ONDERZOEKSVRAAG

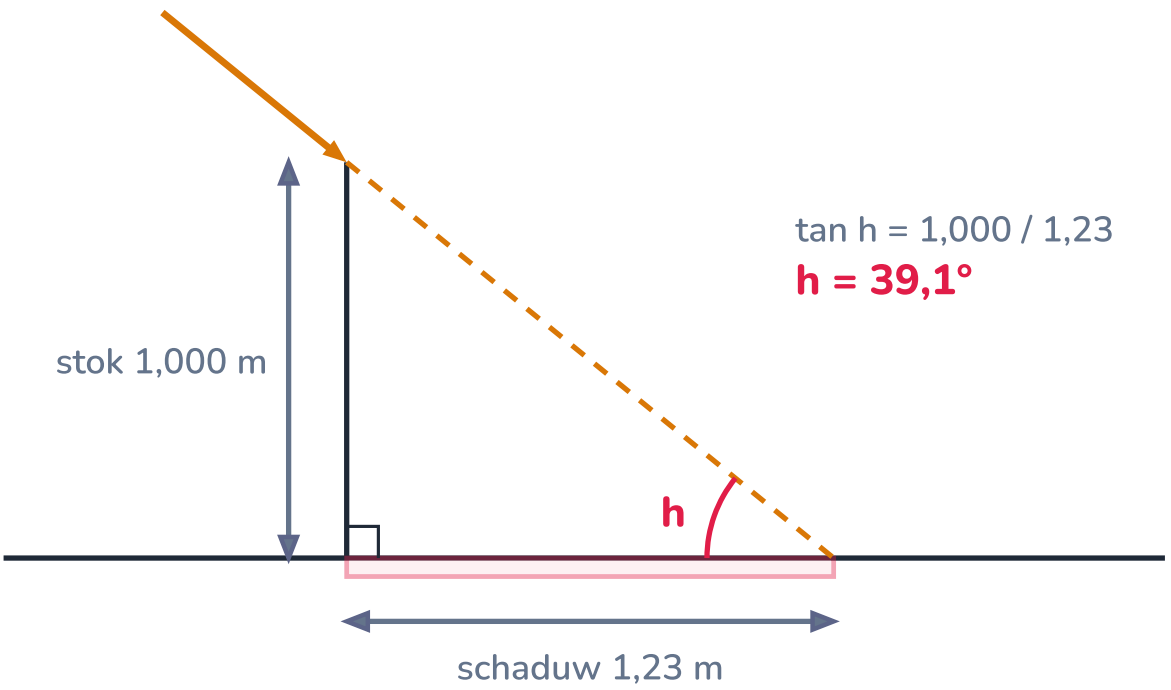
Hoe hoog staat de zon op het middaguur, en kan je uit je meting de breedtegraad van je school afleiden?

MATERIAAL

- rechte stok of bezemsteel, precies 1,000 m boven de grond
- schietlood of waterpas
- rolmeter
- krijt of piketten
- een vlakke, horizontale ondergrond en een zonnige dag
- klok

Veiligheid

Kijk nooit rechtstreeks in de zon, ook niet even: je meet de schaduw, niet de zon.



Stok, schaduw en zonnestraal vormen een rechthoekige driehoek: de tangens van h is stok gedeeld door schaduw.

WERKWIJZE

1. Zet de stok verticaal op een vlakke ondergrond (controleer met het schietlood of de waterpas).
2. Markeer vanaf ongeveer een half uur vóór het middaguur om de 5 minuten het uiteinde van de schaduw met krijt.
3. De kortste schaduw hoort bij het middaguur (dan wijst de schaduw ook precies naar het noorden). Meet haar lengte.
4. Bereken de zonnehoogte: $\tan h = 1{,}000\,\text{m} / \text{schaduw}$.
5. Zoek de declinatie van die dag op (of bereken ze met de benadering) en bereken de breedtegraad: $\varphi = 90^\circ - h + \delta$.

WAARNEMING

Meting op 21 maart (declinatie ongeveer 0°): de kortste schaduw is 1,23 m, om 12.50 uur. $h = \tan^{-1}(1{,}000/1{,}23) = 39{,}1°$.

VERKLARING

$\varphi = 90° - 39{,}1° + 0° = 50{,}9°$: de breedtegraad van Brussel (50,85°) tot op 0,1°. Een schaduwfout van 1 cm verandert h met ongeveer 0,2°: nauwkeurig meten loont.

Foutenbronnen: een stok die niet verticaal staat, een grond die niet horizontaal ligt, de vage rand van de schaduw (de zon is geen punt, maar een schijf van een halve graad), en het moment van meten. Rond het middaguur verandert de schaduwlengte heel traag, dus een paar minuten ernaast geeft weinig fout.

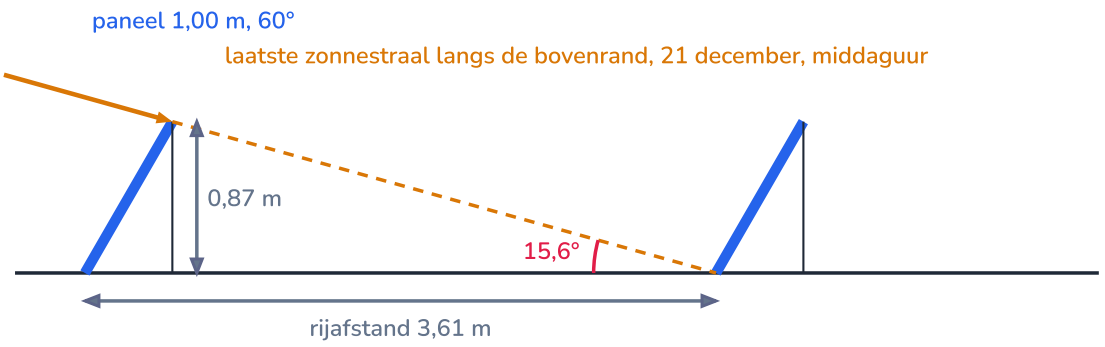
De krijtstreepjes volgen het uiteinde van de schaduw; de kortste hoort bij het middaguur.

UITGEWERKT VOORBEELD · 6.18

Hoe ver moeten twee rijen zonnepanelen uit elkaar?

18/22

Op een plat dak komen twee rijen zonnepanelen achter elkaar, op het zuiden, met een hellingshoek van 60°. Elk paneel is 1,00 m lang in de hellingsrichting. Hoe ver moeten de rijen uit elkaar staan, zodat de voorste rij de achterste op 21 december om 12 uur zonnetijd niet beschaduwt?



De stippellijn is de laatste zonnestraal die de voorste rij net raakt; daar mag de volgende rij beginnen.

GEGEVEN

- paneellengte $L = 1{,}00\,\text{m}$, $\beta = 60°$
- zonnehoogte op 21 december $h = 15{,}6°$

GEVRAAGD

- de rijafstand (van onderrand tot onderrand)

OPLOSSING

1. Hoogte van de bovenrand: $H = L \sin\beta = 1{,}00 \cdot \sin 60° = 0{,}87\,\text{m}$.
2. Horizontale diepte van het paneel: $L\cos\beta = 1{,}00 \cdot \cos 60° = 0{,}50\,\text{m}$.
3. Schaduw van de bovenrand, voorbij de bovenrand: $H / \tan h = 0{,}87 / \tan 15{,}6° = 3{,}11\,\text{m}$.
4. Rijafstand: $0{,}50 + 3{,}11 = 3{,}61\,\text{m}$.

ANTWOORD

De rijen moeten minstens 3,61 m uit elkaar staan (onderrand tot onderrand). Bij een hellingshoek van 39° zou 3,04 m volstaan, maar het paneel vangt dan minder winterzon.

Klopt dit? De rest van de dag staat de zon lager dan op het middaguur, dus 's ochtends en 's avonds is er toch wat schaduw. Ontwerpers nemen daarom vaak het middaguur van 21 december als ondergrens, of aanvaarden wat schaduw in december. De ruimte tussen de rijen is de prijs voor steile panelen.

UITGEWERKT VOORBEELD · 6.19

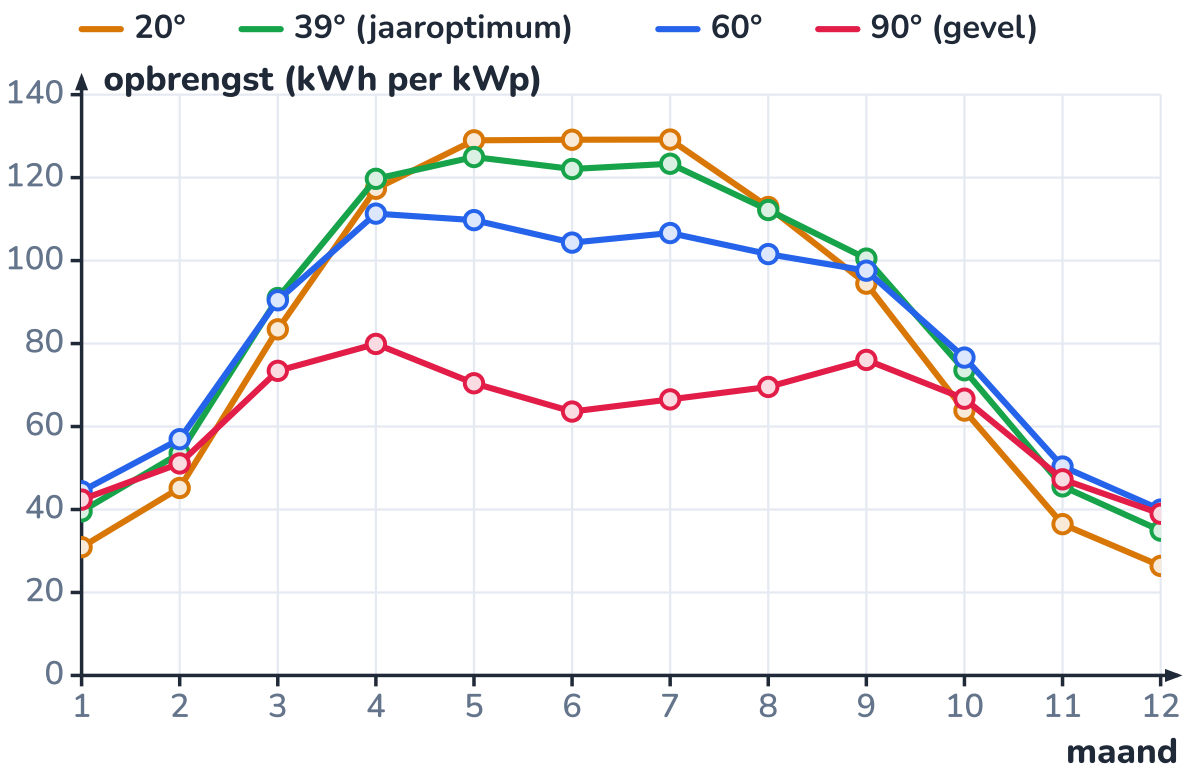
Habitat: de beste hellingshoek van het zonnepaneel

19/22

De klimaatmodule draait op eigen zonnestroom, met een batterij. Welke hellingshoek geef je het paneel (op het zuiden)? Werk in twee stappen: eerst met de zonnehoogte op het middaguur, daarna met echte jaardata uit PVGIS, het rekenprogramma van het Joint Research Centre van de Europese Commissie (Brussel, per kWp geïnstalleerd vermogen).

hellingshoek	jaar (kWh/kWp)	december	januari	juni
0°	869	14,3	18,2	127,8
20°	998	26,4	30,9	129,1
39°	1041	34,9	39,6	122,1
60°	990	40,0	44,4	104,3

hellingshoek	jaar (kWh/kWp)	december	januari	juni
75°	891	40,7	44,7	85,6



Maandopbrengst per kWp (PVGIS, Brussel): 60° levert minder in de zomer, maar meer van oktober tot februari dan 39°.

GEGEVEN

- $\varphi = 51°$; h tussen 15,6° en 62,4°
- maand- en jaaropbrengsten per hellingshoek (tabel)

GEVRAAGD

- de hellingshoek voor de grootste jaaropbrengst
- de hellingshoek voor de klimaatmodule, met verantwoording

OPLOSSING

1 Loodrecht op de middagzon staat een paneel als de normaal naar de zon wijst: $90° - \beta = h$, dus $\beta = 90° - h = \varphi - \delta$. In juni is dat $27{,}6°$, bij de equinoxen $51°$, in december $74{,}4°$.

2 Een vast paneel moet dus een compromis kiezen. Omdat de dagen in de zomer lang en zonnig zijn, weegt de zomer zwaarder door: PVGIS geeft de grootste jaaropbrengst bij $39°$ (1041 kWh per kWp), iets minder dan de breedtegraad. Een vuistregel voor België: breedtegraad min ongeveer 12°.

3 De klimaatmodule heeft een batterij die ook in de winter elke nacht moet volstaan. Vergelijk 60° met 39°: jaar $990/1041 = 0{,}95$ (−5 %), december $40{,}0/34{,}9 = 1{,}15$ (+15 %), januari $44{,}4/39{,}6 = 1{,}12$ (+12 %).

4 Steiler dan 60° levert in de winter nauwelijks nog iets op (75°: december 40,7, +2 %), maar kost veel in de zomer (jaar −10 %). Een steil paneel heeft nog een voordeel: sneeuw glijdt er sneller af.

ANTWOORD

Voor de grootste jaaropbrengst: 39°. Voor de klimaatmodule, waar de donkerste maanden de batterij bepalen: 60°. Je verliest 5 % op een heel jaar, maar wint 12 à 15 % in december en januari, precies wanneer de energie het schaarst is.

Klopt dit? Het middaguurmodel (stap 1) en de PVGIS-data (stap 2) wijzen dezelfde kant op: steiler voor de winter, vlakker voor de zomer. Het model houdt geen rekening met bewolking, diffuus licht en de lengte van de dag; daarom kies je de hoek met de echte data en gebruik je het model om te begrijpen waarom. Hoeveel wattpiek en welke batterij de module nodig heeft, reken je uit in hoofdstuk 20.

CODE · 6.20

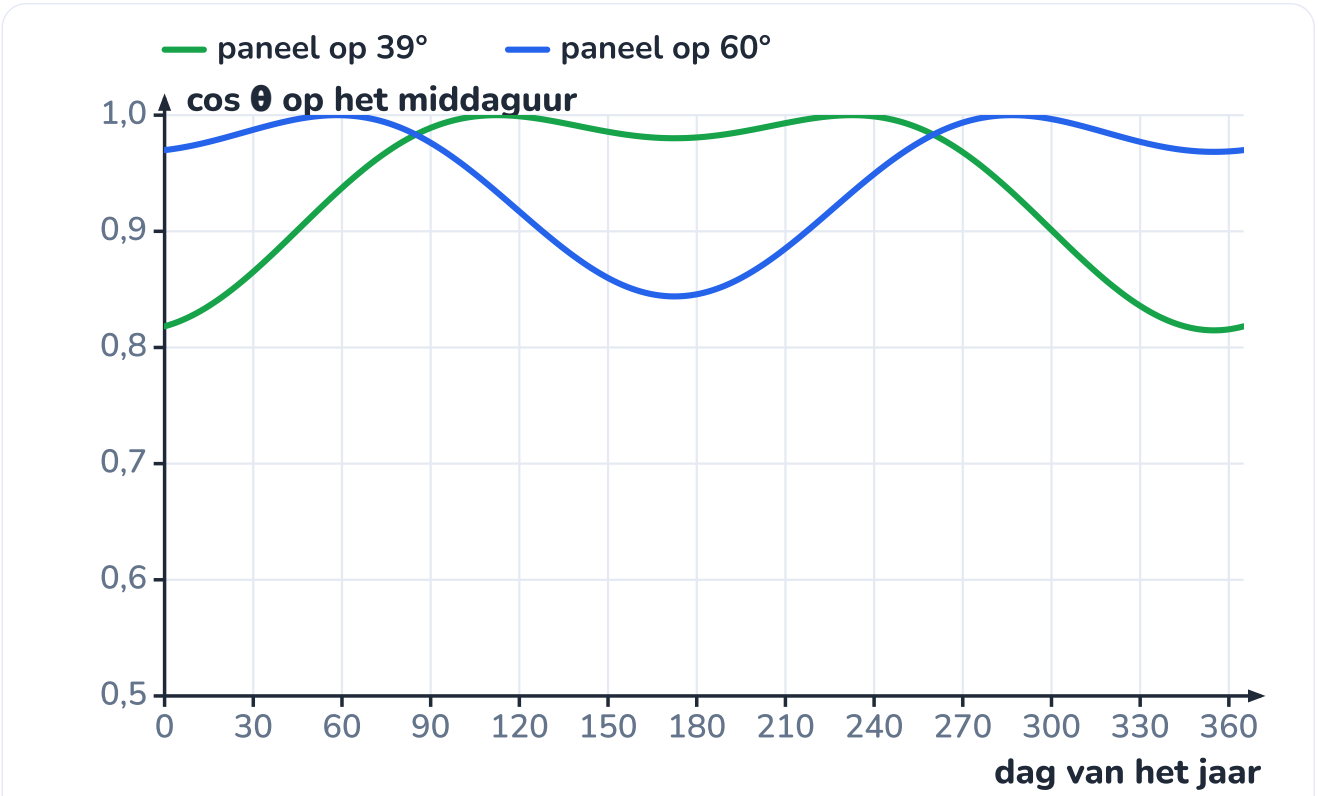
De zonnestand in Python

20/22

Het programma berekent voor vier dagen de declinatie (met de benadering van het vorige blad), de zonnehoogte op het middaguur, en voor een paneel op 39° en op 60° de factor $\cos\theta$: het deel van de rechtstreekse middagzon dat het paneel vangt. Let op `math.radians()`: Python rekent met radialen.

Python **zonnestand.py** 21 regels

```python
# zonnestand.py - zonnehoogte op het middaguur en de invalshoek op een schuin zonnepaneel
import math

BREEDTE = 51.0                    # breedtegraad van België (graden noorderbreedte)
DAGEN = {"21 maart": 80, "21 juni": 172, "21 september": 264, "21 december": 355}


def declinatie(dag):
    """Benadering van de declinatie van de zon (in graden) op dag 1 tot 365 van het jaar."""
    return -23.44 * math.cos(math.radians(360 / 365 * (dag + 10)))


for naam, dag in DAGEN.items():
    delta = declinatie(dag)
    hoogte = 90 - BREEDTE + delta                    # zonnehoogte op het middaguur (graden)
    regel = f"{naam:13s} declinatie {delta:6.1f}  zonnehoogte {hoogte:5.1f}"
    for kanteling in (39, 60):
        invalshoek = abs(90 - kanteling - hoogte)      # hoek tussen zonnestraal en normaal van het paneel
        factor = math.cos(math.radians(invalshoek))    # deel van de straling dat 'recht' invalt
        regel += f"  paneel {kanteling}: {factor:.2f}"
    print(regel)
```



De factor cos θ door het jaar, berekend met dezelfde formules: 60° wint van oktober tot februari, 39° de rest van het jaar.

Uitvoer (gecontroleerd door het programma te draaien):

```
21 maart      declinatie   -0.5  zonnehoogte  38.5  paneel 39: 0.98  paneel 60: 0.99
21 juni       declinatie   23.4  zonnehoogte  62.4  paneel 39: 0.98  paneel 60: 0.84
21 september  declinatie   -0.1  zonnehoogte  38.9  paneel 39: 0.98  paneel 60: 0.99
21 december   declinatie  -23.4  zonnehoogte  15.6  paneel 39: 0.81  paneel 60: 0.97
```

In december vangt het paneel op 60° 97 % van de middagzon, het paneel op 39° maar 81 %; in juni is het omgekeerd (84 % tegenover 98 %). Rond de equinoxen doen ze allebei bijna alles. De benadering geeft voor 21 maart een declinatie van −0,5° in plaats van ongeveer 0°: de fout van ongeveer een halve graad zie je in de zonnehoogte terug (38,5° in plaats van 39,0°).

NASLAG · 6.21

Overzicht: goniometrie en vectoren

21/22

in de rechthoekige driehoek

$\sin\alpha = \frac{o}{s}$ $\cos\alpha = \frac{a}{s}$ $\tan\alpha = \frac{o}{a}$

symbool	betekenis	eenheid
o	overstaande rechthoekszijde	m
a	aanliggende rechthoekszijde	m
s	schuine zijde	m

verbanden

$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}$ $\sin^2\alpha + \cos^2\alpha = 1$

helling in procent

$p = 100\,\% \cdot \tan\alpha$ $\sin\alpha = \cos(90° - \alpha)$

componenten van een vector

$F_x = F\cos\alpha$ $F_y = F\sin\alpha$ $F = \sqrt{F_x^2 + F_y^2}$

vectoren optellen met componenten

$R_x = F_{1x} + F_{2x}$ $R_y = F_{1y} + F_{2y}$ $\tan\alpha_R = \frac{R_y}{R_x}$

zwaartekracht op een helling

$F_{\parallel} = m\,g\sin\alpha$ $F_{\perp} = m\,g\cos\alpha$

zonnehoogte op het middaguur

$h = 90^\circ - \varphi + \delta \qquad -23{,}44^\circ \le \delta \le +23{,}44^\circ$

symbool	betekenis	eenheid
φ	breedtegraad	°
δ	declinatie van de zon	°

schaduw en invalshoek (zon in het zuiden)

$\text{schaduw} = \dfrac{H}{\tan h} \qquad \theta = |(90^\circ - \beta) - h| \qquad \text{fractie} = \cos\theta$

symbool	betekenis	eenheid
H	hoogte van het voorwerp	m
β	hellingshoek van het paneel	°
θ	invalshoek	°

α	0°	30°	45°	60°	90°
$\sin\alpha$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2} \approx 0{,}707$	$\frac{\sqrt{3}}{2} \approx 0{,}866$	1
$\cos\alpha$	1	$\frac{\sqrt{3}}{2} \approx 0{,}866$	$\frac{\sqrt{2}}{2} \approx 0{,}707$	$\frac{1}{2}$	0
$\tan\alpha$	0	$\frac{\sqrt{3}}{3} \approx 0{,}577$	1	$\sqrt{3} \approx 1{,}732$	bestaat niet

helling	hoek	voorbeeld
2 %	1,1°	spoorlijn, riolering (vaak nog minder)
5 %	2,9°	grens voor een lange oprit voor rolstoelen
10 %	5,7°	steile weg; zwaar voor een fietser
22 %	12,4°	steilste stuk van de Koppenberg (Ronde van Vlaanderen)
48 %	25,6°	Pilatusbahn (Zwitserland), de steilste tandradbaan ter wereld
100 %	45°	even hoog als lang: een trap is vaak nog steiler

controlelijst
rekenmachine op DEG; in Python en rekenbladen graden omzetten naar radialen
schets met de hoek en de namen van de zijden ten opzichte van díe hoek
test op 0° en 90°: geeft de formule het verwachte uiterste?
vectoren: grootte én richting in het antwoord, met eenheid
bij $\tan^{-1}$ het kwadrant controleren met een schets

HET GEHEEL · 6.22

Waar hoeken en vectoren terugkomen

22/22

Goniometrie en vectoren zijn het gereedschap van de hele mechanica: zodra krachten en snelheden niet op één lijn liggen, ontbind je ze. Ze zijn ook het gereedschap van de zonne-energie: de hoek van de zon bepaalt hoeveel energie een paneel oogst. En de sinus komt terug als functie van de tijd, in trillingen en wisselstroom.

F11 Krachten
Kracht als pijl met grootte, richting en aangrijpingspunt: de eerste vector.

F13 Constructies: sterk en stabiel
De driehoek als stijf element; in een vakwerk ontbind je de staafkrachten met sinus en cosinus.

I7 Beweging beschrijven
Verplaatsing, snelheid en versnelling zijn vectoren; vallen en rijden langs een lijn is het eendimensionale geval.

I8 De wetten van Newton

I9 Arbeid, energie en vermogen

I10 Moment en evenwicht

Het vrijlichaamsdiagram en het voorwerp op een helling, met normaalkracht en wrijving.

Arbeid is kracht maal weg langs de kracht: een flauwe helling vraagt minder kracht over een langere weg.

Evenwicht: de som van alle krachtvectoren is nul, en ook de som van de momenten.

I20 Zonne-energie dimensioneren

De invalshoek bepaalt de instraling op het paneel; daar kies je wattpiek en batterij.

E5 Vectoren en matrices

Vectoren in drie dimensies, het scalair product (arbeid) en het vectorproduct (moment).

E7 Sinusoïden, complexe getallen en fasoren

Sinus en cosinus als functies van een draaiende hoek, en fasoren om sinussen op te tellen als vectoren.

E8 Beweging in twee dimensies

De schuine worp: de beginsnelheid ontbonden in een horizontale en een verticale component.

Habitat

Voor Habitat leggen we vast: het zonnepaneel van de klimaatmodule staat op het zuiden onder een hellingshoek van 60°, gekozen voor de winteropbrengst (−5 % per jaar, +12 à 15 % in december en januari ten opzichte van het jaaroptimum van 39°). Twee rijen panelen van 1,00 m staan minstens 3,61 m uit elkaar. De oprit naar de module is 6,40 m lang (6,25 %). In opdracht 6.13 maak je de keuze zelf.

SAMENVATTING

In het kort

- In een rechthoekige driehoek: $\sin\alpha = o/s$, $\cos\alpha = a/s$, $\tan\alpha = o/a$. De verhoudingen hangen alleen van de hoek af (gelijkvormigheid).
- Een hoek terugrekenen doe je met $\sin^{-1}$, $\cos^{-1}$ of $\tan^{-1}$, met de rekenmachine op graden. Exacte waarden bij 30°, 45° en 60° komen uit een half vierkant en een halve gelijkzijdige driehoek.
- Een helling in procent is $100\,\% \cdot \tan\alpha$: 10 % is 5,7°, 100 % is 45°, en een verticale wand heeft geen eindig percentage.
- Vectoren hebben een grootte en een richting. Je telt ze op met kop-staart of met een parallellogram; de som hangt af van de richtingen.
- Ontbinden: $F_x = F\cos\alpha$, $F_y = F\sin\alpha$. Met componenten tel je vectoren op als getallen, en Pythagoras en de tangens geven grootte en richting van de resultante.
- Op een helling ontbind je de zwaartekracht in $F_{\parallel} = mg\sin\alpha$ langs en $F_{\perp} = mg\cos\alpha$ loodrecht op de helling.
- De middagzon staat op $h = 90^\circ - \varphi + \delta$; in België tussen 15,6° en 62,4°. Een schaduw is $H/\tan h$ lang.
- Een paneel vangt een fractie $\cos\theta$ van de rechtstreekse straling. De beste vaste hoek hangt af van wat je wil: 39° voor het jaar, 60° voor de winter.

BEGRIPPEN

Wat je nu kent

sinus

In een rechthoekige driehoek: overstaande rechthoekszijde gedeeld door de schuine zijde, $\sin\alpha = o/s$.

cosinus

In een rechthoekige driehoek: aanliggende rechthoekszijde gedeeld door de schuine zijde, $\cos\alpha = a/s$.

tangens

In een rechthoekige driehoek: overstaande gedeeld door aanliggende rechthoekszijde, $\tan\alpha = o/a$.

helling in procent

Hoogteverschil per 100 horizontale afstand: $p = 100 \cdot \tan\alpha$. 100 % is 45°.

vector

Grootheid met een grootte en een richting (kracht, snelheid, verplaatsing), getekend als pijl.

scalar

Grootheid die volledig bepaald is door een getal met eenheid (massa, temperatuur, energie).

resultante

De som van twee of meer vectoren: de ene vector die hetzelfde effect heeft.

componenten van een vector

De delen van een vector langs twee loodrechte richtingen: $F_x = F\cos\alpha$, $F_y = F\sin\alpha$.

zonnehoogte

De hoek tussen de richting van de zon en de horizon; op het middaguur $h = 90^\circ - \varphi + \delta$.

declinatie

De breedtegraad waar de zon op het middaguur loodrecht staat; ze schommelt tussen −23,44° en +23,44°.

invalshoek

De hoek θ tussen een zonnestraal en de normaal op een vlak; het vlak vangt een fractie $\cos\theta$ van de rechtstreekse straling.

I INTERMEDIATE · DEEL I3 · MECHANICA

HANDBOEK · HOOFDSTUK 7

Beweging beschrijven

Plaats, snelheid en versnelling: drie grafieken die samen één beweging vertellen

Een schuifpaneel dat zacht op gang komt, een steen die in een put valt, een auto die een noodstop maakt: om een beweging te voorspellen, moet je ze eerst nauwkeurig beschrijven. In dit hoofdstuk leer je werken met **plaats**, **snelheid** en **versnelling**, als formules en als grafieken. Je ontdekt dat de helling van de ene grafiek de volgende grafiek geeft, en dat de oppervlakte onder de (v,t)-grafiek de verplaatsing is. Daarmee beschrijf je de vrije val, reken je stopafstanden uit en ontwerp je de beweging van het automatische kasdak van Habitat.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 10 lestijden

K Krachten, beweging en constructies · W Wiskundig modelleren

Het schuifpaneel van het kasdak moet zacht vertrekken en zacht stoppen: hoe beschrijf je zo'n beweging precies?

Na dit hoofdstuk

- Je kan plaats, verplaatsing en afgelegde weg onderscheiden, met een gekozen positieve zin.
- Je kan gemiddelde en ogenblikkelijke snelheid berekenen en aflezen als helling van koorde en raaklijn in een (x,t)-grafiek.
- Je kan een eenparige rechtlijnige beweging (ERB) en een eenparig versnelde rechtlijnige beweging (EVRB) herkennen, beschrijven met formules en er problemen mee oplossen.
- Je kan de (x,t)-, (v,t)- en (a,t)-grafiek van eenzelfde beweging uit elkaar afleiden, met helling en oppervlakte.
- Je kan met de oppervlakte onder een (v,t)-grafiek een verplaatsing berekenen, ook bij een beweging in fasen.
- Je kan een vrije val berekenen met $g = 9{,}81\ \mathrm{m/s^2}$ en de grenzen van dat model benoemen.
- Je kan een stopafstand opsplitsen in reactieafstand en remafstand en uitleggen waarom snelheid zo zwaar doorweegt.
- Je kan een beweging meten met een sensor, er snelheid en versnelling uit berekenen en een beweging stap voor stap simuleren in Python.

UITLEG · 7.1

Wat je al weet, en wat nu komt

In hoofdstuk F14 berekende je een gemiddelde snelheid als afstand gedeeld door tijd, $v = s/t$, en zette je om tussen m/s en km/h (delen of vermenigvuldigen met 3,6). Je las een (s,t)-grafiek: een rechte betekent een constante snelheid, en hoe steiler de rechte, hoe sneller. In de figuur rijdt de fietser in 4 s 20 m verder: 5,0 m/s.

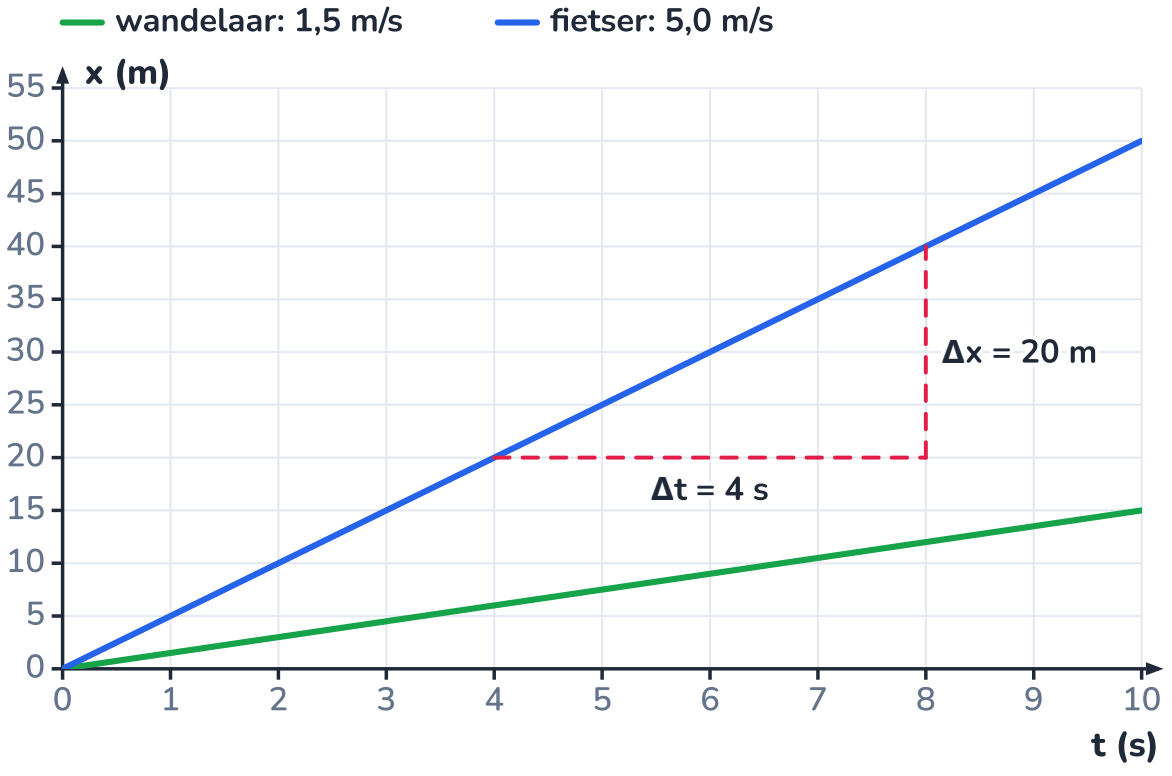
In hoofdstuk 4 zag je dat een vallende steen of een remmende auto geen rechte maar een **parabool** geeft, en rekende je met de formule voor de stopafstand zonder te weten waar ze vandaan komt. In hoofdstuk 6 leerde je dat snelheid een **vector** is: ze heeft een grootte en een richting.

Nu bouw je daarop verder. In dit hoofdstuk

- geef je een beweging een **as** met een positieve zin, zodat snelheid een teken krijgt;
- maak je onderscheid tussen de snelheid *gemiddeld* over een tijdsduur en *op één ogenblik*;
- voer je de **versnelling** in: hoe snel de snelheid verandert;
- leer je drie grafieken van één beweging samen lezen, met **helling** en **oppervlakte**;
- leid je de formules voor vallen en remmen zelf af.

Waarom één dimensie?

Alles in dit hoofdstuk gebeurt langs een rechte lijn: een gang, een rail, een verticale val, een rechte weg. Dan volstaat één getal voor de plaats en een teken voor de richting. Bewegingen in een vlak, zoals een schuine worp, komen in hoofdstuk E8: daar worden plaats, snelheid en versnelling vectoren met twee componenten.



Twee rechten: twee constante snelheden. De hellingsdriehoek op de fietser geeft 20 m / 4 s = 5,0 m/s.

UITLEG · 7.2

Plaats, verplaatsing en gemiddelde snelheid

Om een beweging langs een rechte te beschrijven, kies je eerst een **as**: een oorsprong (waar $x = 0$ is) en een **positieve zin**. De **plaats** x van het voorwerp is dan een getal met een eenheid, dat ook negatief mag zijn (links van de oorsprong).

- De **verplaatsing** tussen twee tijdstippen is het verschil van de plaatsen: $\Delta x = x_2 - x_1$. Ze is positief als het voorwerp netto in de positieve zin opgeschoven is, negatief als het netto teruggegaan is. Wat er onderweg gebeurde, telt niet.
- De **afgelegde weg** is de lengte van het hele traject, heen en terug samen. Die is nooit negatief, en nooit kleiner dan de grootte van de verplaatsing.

GEMIDDELDE SNELHEID

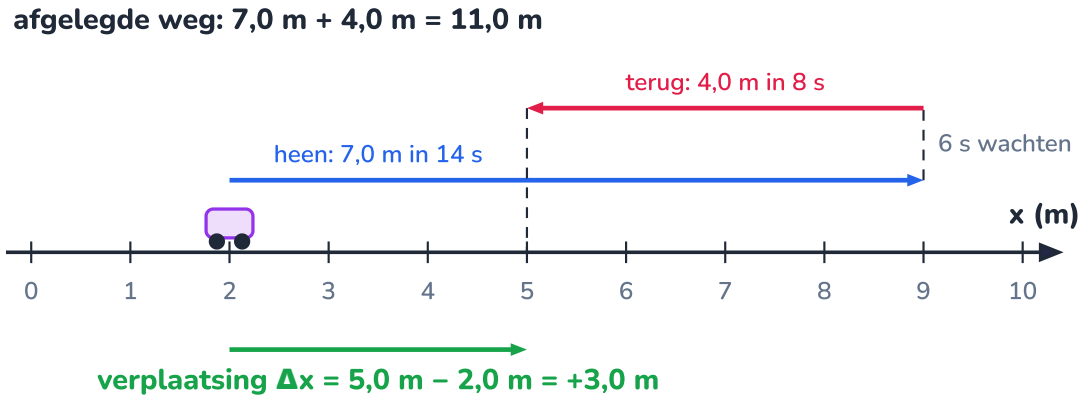
$$v_{\text{gem}} = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1}$$

Δx in m, Δt in s, v_{gem} in m/s. Het teken zegt in welke zin het voorwerp netto bewoog.

Wat je in hoofdstuk F14 berekende, afgelegde weg gedeeld door tijd, noemen we voortaan de **gemiddelde baansnelheid**. Beweegt een voorwerp in één zin zonder om te keren, dan zijn de twee gelijk. Keert het om, zoals het robotje in de figuur, dan niet.

Tijdstip en tijdsduur

$t = 14\,\text{s}$ is een **tijdstip** (een aflezing van de klok, net als x een aflezing op de as). $\Delta t = 14\,\text{s}$ is een **tijdsduur** (een verschil van twee aflezingen). Start je de klok bij het begin van de beweging, dan vallen ze samen, maar het blijven twee verschillende dingen.

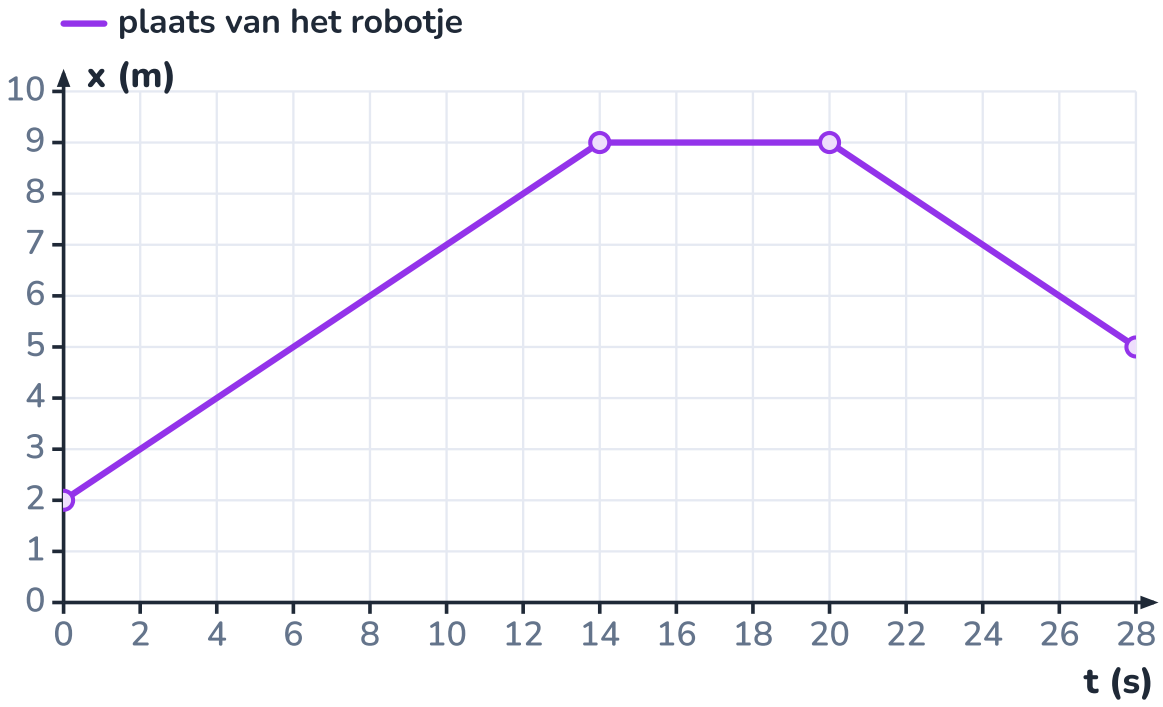


Het robotje rijdt van 2 m naar 9 m en terug naar 5 m: 11 m afgelegd, maar netto maar 3 m verplaatst.

UITGEWERKT VOORBEELD · 7.3

Het inspectierobotje in de gang

Een inspectierobotje (hoofdstuk 33) rijdt door de lange gang van het station, van de woonmodule naar de klimaatmodule. Op $t = 0$ staat het op $x = 2{,}0\,\text{m}$. Het rijdt in 14 s naar $x = 9{,}0\,\text{m}$, meet 6 s lang de temperatuur, en rijdt in 8 s terug naar $x = 5{,}0\,\text{m}$. Bereken de snelheid in elke fase, de verplaatsing, de afgelegde weg, de gemiddelde snelheid en de gemiddelde baansnelheid over de hele rit.



Stijgend, vlak, dalend: de helling van elk stuk is de snelheid in die fase.

GEGEVEN

- (t, x): $(0\,\text{s};\ 2{,}0\,\text{m})$, $(14\,\text{s};\ 9{,}0\,\text{m})$, $(20\,\text{s};\ 9{,}0\,\text{m})$, $(28\,\text{s};\ 5{,}0\,\text{m})$

GEVRAAGD

- v per fase, Δx, afgelegde weg, v_{gem} en gemiddelde baansnelheid

OPLOSSING

1. Heen: $v = \dfrac{9{,}0 - 2{,}0}{14 - 0} = \dfrac{7{,}0\,\text{m}}{14\,\text{s}} = +0{,}50\,\text{m/s}$.
2. Wachten: $v = \dfrac{9{,}0 - 9{,}0}{20 - 14} = 0\,\text{m/s}$.
3. Terug: $v = \dfrac{5{,}0 - 9{,}0}{28 - 20} = \dfrac{-4{,}0\,\text{m}}{8\,\text{s}} = -0{,}50\,\text{m/s}$: even snel, maar in de negatieve zin.
4. Verplaatsing: $\Delta x = 5{,}0 - 2{,}0 = +3{,}0\,\text{m}$. Afgelegde weg: $7{,}0 + 0 + 4{,}0 = 11{,}0\,\text{m}$.
5. Gemiddelde snelheid: $v_{\text{gem}} = \dfrac{3{,}0\,\text{m}}{28\,\text{s}} = 0{,}11\,\text{m/s}$. Gemiddelde baansnelheid: $\dfrac{11{,}0\,\text{m}}{28\,\text{s}} = 0{,}39\,\text{m/s}$.

ANTWOORD

Heen +0,50 m/s, wachten 0 m/s, terug −0,50 m/s. Netto 3,0 m verplaatst in 28 s ($v_{\text{gem}} = 0{,}11\,\text{m/s}$); 11,0 m afgelegd ($0{,}39\,\text{m/s}$ gemiddeld op de 'teller').

Klopt dit? In de (x,t)-grafiek zie je drie rechte stukken: stijgend (positieve helling), vlak (nul) en dalend (negatieve helling). De gemiddelde snelheid is de helling van de rechte van het eerste naar het laatste punt: van (0 s; 2 m) naar (28 s; 5 m). Voor de batterij van het robotje telt de afgelegde weg, voor de vraag 'waar is het nu?' de verplaatsing.

UITLEG · 7.4

Ogenblikkelijke snelheid: de raaklijn

4/20

Een auto die optrekt, rijdt elke seconde verder dan de vorige: zijn (x,t)-grafiek is een kromme. Welke snelheid heeft hij op $t = 6\,\text{s}$? De gemiddelde snelheid over de hele rit is de helling van de **koorde**, de rechte tussen begin- en eindpunt: in de figuur $48\,\text{m}/8\,\text{s} = 6{,}0\,\text{m/s}$. Maar op $t = 6\,\text{s}$ gaat de auto duidelijk sneller dan dat.

Neem de koorde over een steeds kleiner tijdsinterval rond $t = 6\,\text{s}$: van 6 tot 7 s, van 6 tot 6,1 s, van 6 tot 6,01 s. De koorde draait dan naar een grensstand: de **raaklijn**, die de kromme in dat ene punt raakt. De helling van de raaklijn is de **ogenblikkelijke snelheid**. Hier: $18\,\text{m}/2\,\text{s} = 9{,}0\,\text{m/s}$.

OGENBLIKKELIJKE SNELHEID

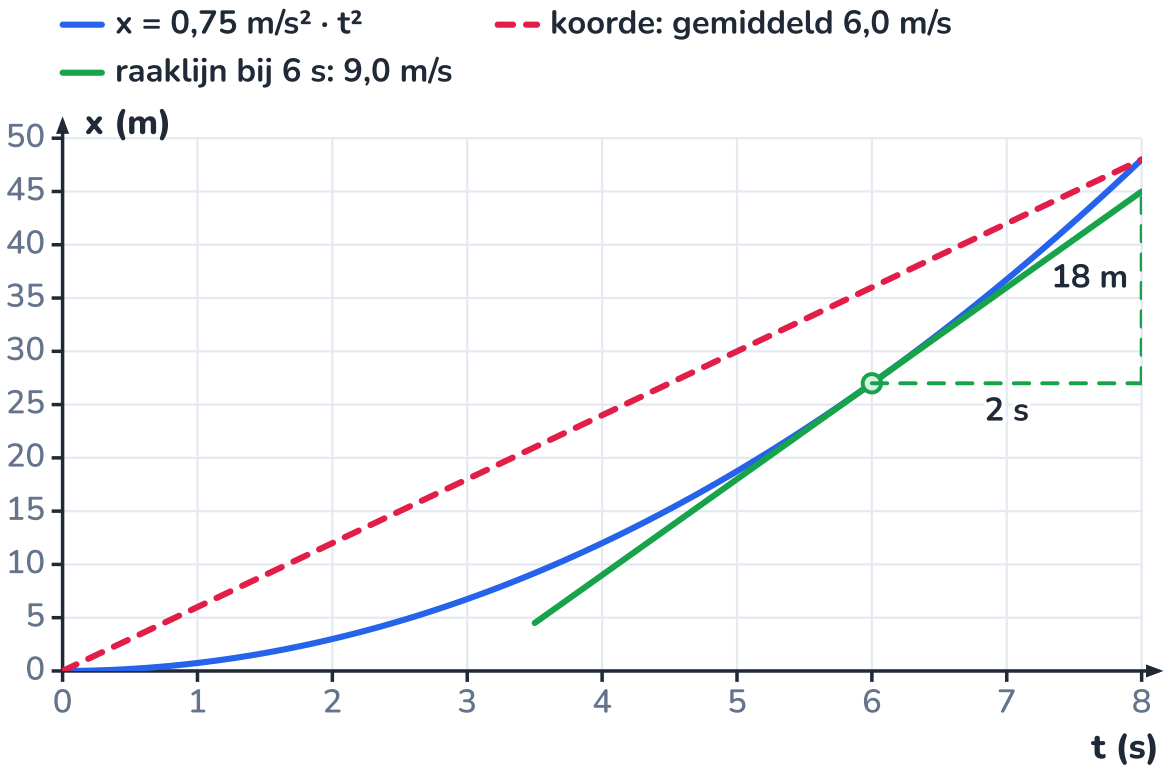
$$v \approx \frac{\Delta x}{\Delta t} \quad \text{voor een heel kleine } \Delta t$$

Exact is v de helling van de raaklijn aan de (x,t)-grafiek in dat punt.

Een snelheidsmeter, een flitspaal of een trajectcontrole doet in wezen hetzelfde, met een kleine of een grote Δt: een flitspaal meet de verplaatsing over een fractie van een seconde, een trajectcontrole over enkele minuten. Die laatste geeft dus een *gemiddelde* snelheid.

Vooruitblik

De grensstand van de koorde is precies wat de wiskunde een **afgeleide** noemt. In hoofdstuk E1 schrijf je $v = \frac{\text{d}x}{\text{d}t}$ en reken je hem exact uit. Hier lees je hem af met een raaklijn, of je benadert hem met een kleine Δt.



De koorde (gemiddeld 6,0 m/s over 8 s) en de raaklijn in t = 6 s (9,0 m/s op dat ogenblik).

UITLEG · 7.5

De eenparige rechtlijnige beweging (ERB)

5/20

Een beweging langs een rechte met een **constante snelheid** heet **eenparig rechtlijnig** (ERB). 'Eenparig' betekent: in gelijke tijden gelijke verplaatsingen. Dan is de ogenblikkelijke snelheid op elk moment gelijk aan de gemiddelde, en is de plaats een lineaire functie van de tijd (hoofdstuk 4):

ERB

$$x(t) = x_0 + v\,t \qquad v = \text{constant}$$

x_0 is de plaats op $t = 0$ (het snijpunt met de verticale as), v de snelheid (de helling). Een negatieve v geeft een dalende rechte.

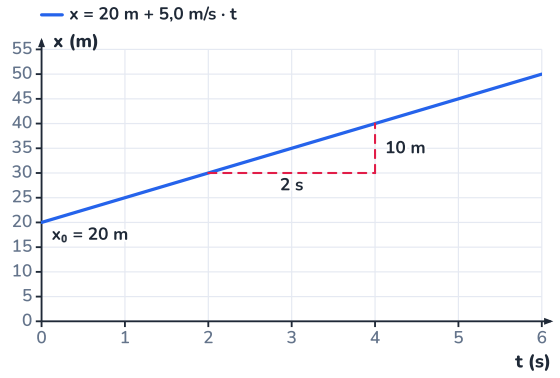
Dezelfde beweging kan je op drie manieren tekenen:

- **(x,t)**: een schuine rechte, met helling v;
- **(v,t)**: een horizontale lijn op hoogte v;
- **(a,t)**: een horizontale lijn op nul (de snelheid verandert niet).

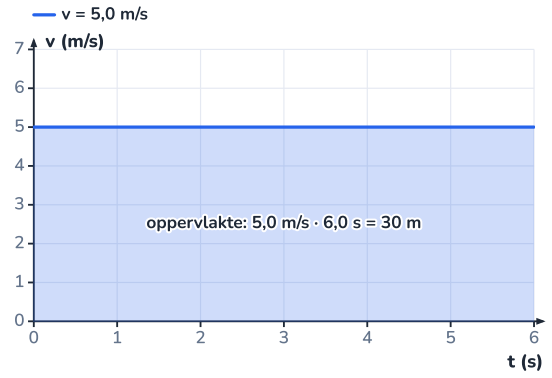
In de (v,t)-grafiek zit nog iets verborgen. Het gearceerde vlak onder de lijn is een rechthoek met hoogte v en breedte Δt. Zijn oppervlakte is $v \cdot \Delta t$: precies de **verplaatsing**. Let op de eenheden: m/s maal s geeft m. Dat idee, *oppervlakte onder de (v,t)-grafiek = verplaatsing*, blijft gelden als de snelheid niet constant is, en maakt het de rest van het hoofdstuk makkelijk.

⚠ Oppervlakte onder de as

Is de snelheid negatief, dan ligt de oppervlakte onder de t-as. Tel ze dan negatief: het voorwerp gaat terug. Zo gaf het robotje $+7{,}0\,\text{m}$ heen en $-4{,}0\,\text{m}$ terug, samen $+3{,}0\,\text{m}$ verplaatsing. Voor de afgelegde weg tel je alle oppervlaktes positief.



Snijpunt: de startplaats 20 m. Helling: 10 m per 2 s, de snelheid 5,0 m/s.



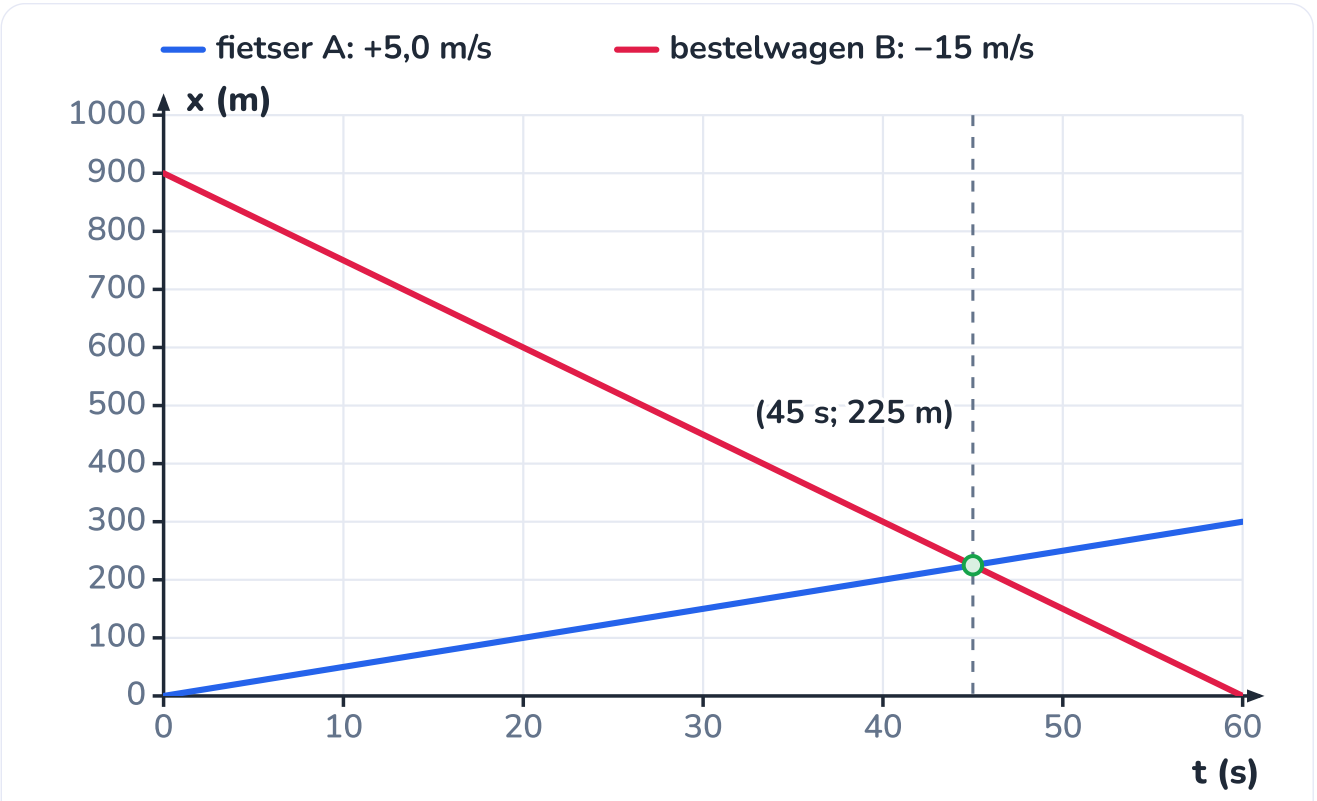
Dezelfde beweging als (v,t)-grafiek: de rechthoek van 5,0 m/s bij 6,0 s is de verplaatsing van 30 m.

UITGEWERKT VOORBEELD · 7.6

Twee tegenliggers

6/20

Op een rechte weg rijdt fietser A vanaf het dorpsplein ($x = 0$) naar het oosten aan 18 km/h. Op hetzelfde moment vertrekt 900 m oostelijker een bestelwagen B naar het westen aan 54 km/h. Wanneer en waar kruisen ze elkaar?



Twee rechten met een tegengestelde helling; hun snijpunt is de ontmoeting.

GEGEVEN

- A: $x_{0A} = 0$, $v_A = +18\,\text{km/h}$
- B: $x_{0B} = 900\,\text{m}$, $v_B = -54\,\text{km/h}$ (oost is positief)

GEVRAAGD

- tijdstip en plaats van de ontmoeting

OPLOSSING

1. Kies één as voor beide: oorsprong op het plein, oost positief. Zet de snelheden om naar m/s: $v_A = 18/3{,}6 = +5{,}0\,\text{m/s}$, $v_B = -54/3{,}6 = -15\,\text{m/s}$ (naar het westen, dus negatief).
2. Twee ERB's: $x_A(t) = 5{,}0\,t$ en $x_B(t) = 900 - 15\,t$ (x in m, t in s).
3. Ze kruisen elkaar als ze op dezelfde plaats zijn: $5{,}0\,t = 900 - 15\,t$, dus $20\,t = 900$ en $t = 45\,\text{s}$.
4. Plaats: $x_A(45) = 5{,}0 \cdot 45 = 225\,\text{m}$. Controle met B: $900 - 15 \cdot 45 = 225\,\text{m}$.

ANTWOORD

Na 45 s kruisen ze elkaar, 225 m ten oosten van het plein.

Klopt dit? Snelle controle: ze naderen elkaar met $5{,}0 + 15 = 20\,\text{m/s}$ en moeten samen 900 m overbruggen: $900/20 = 45\,\text{s}$. In de grafiek is het kruispunt het snijpunt van de twee rechten (hoofdstuk 4). Zonder gemeenschappelijke as en tekens voor de snelheid lukt geen van beide methoden.

UITLEG · 7.7

Versnelling: hoe snel de snelheid verandert

7/20

Als de snelheid verandert, versnelt het voorwerp. De **versnelling** a zegt met hoeveel de snelheid per seconde verandert:

VERSNELLING

$$a = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1}$$

Eenheid: (m/s) per s, geschreven als $\mathrm{m/s^2}$. $a = 2{,}0\,\mathrm{m/s^2}$ betekent: elke seconde komt er 2,0 m/s bij.

In een (v,t)-grafiek is de versnelling de **helling**, zoals de snelheid de helling is in een (x,t)-grafiek. Een horizontale lijn: geen versnelling. Een stijgende lijn: positieve versnelling. Een dalende lijn: negatieve versnelling. Is de versnelling tegengesteld aan de snelheid, dan wordt het voorwerp trager: we spreken dan van een **vertraging**.

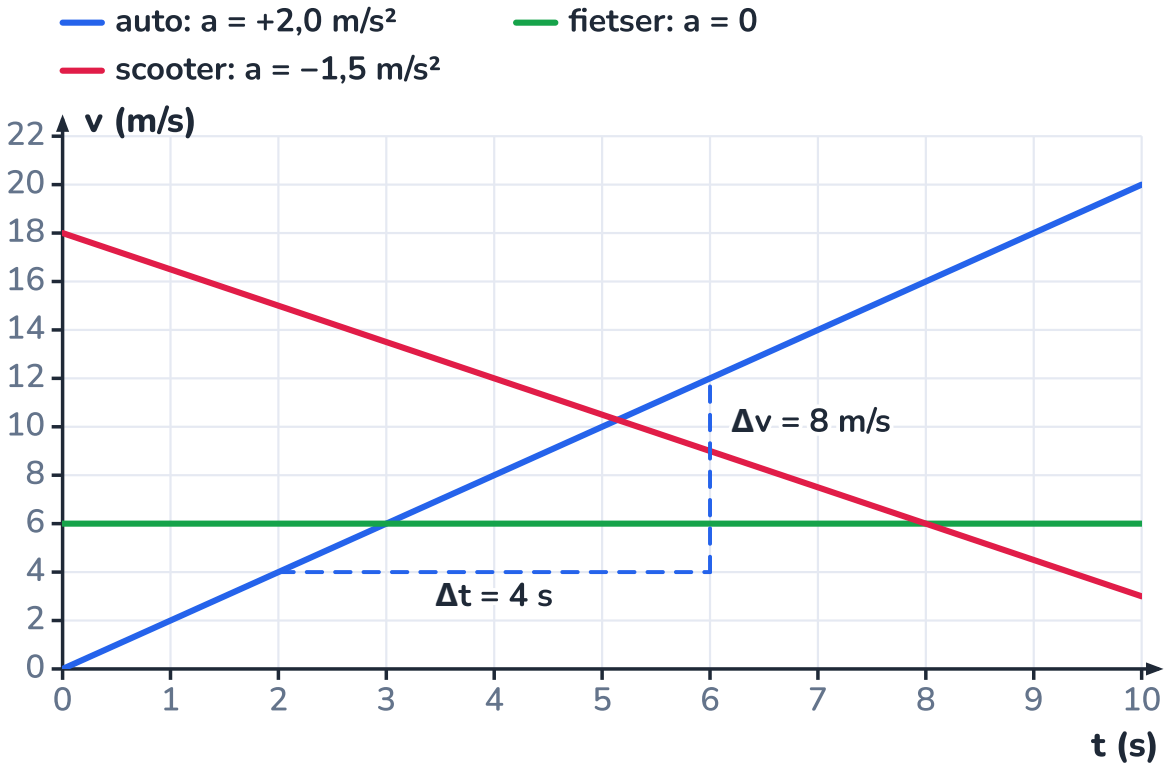
situatie	versnelling (ongeveer)
schuifpaneel van het kasdak (zacht aanlopen)	0,10 m/s²
lift die vertrekt	ongeveer 1 m/s²
gezinsauto van 0 naar 100 km/h in 10 s (gemiddeld)	2,8 m/s²
vrije val (zonder luchtweerstand)	9,81 m/s²
noodstop op droog asfalt	−7 à −8 m/s²

Negatieve versnelling betekent altijd vertragen

Het teken van a hangt af van de gekozen positieve zin, niet van 'sneller of trager'. Een steen die je loslaat, met 'omhoog' als positieve zin, heeft $a = -9{,}81\,\mathrm{m/s^2}$ en gaat toch steeds sneller: v wordt $-9{,}81$, $-19{,}6$, $-29{,}4\,\mathrm{m/s}$ … De regel is: hebben v en a hetzelfde teken, dan wordt het voorwerp sneller; een tegengesteld teken, dan trager.

Snelheid nul betekent versnelling nul

Een bal die je recht omhoog gooit, staat in het hoogste punt heel even stil: $v = 0$. Maar zijn snelheid blijft veranderen, van omhoog naar omlaag, met 9,81 m/s per seconde. Op de top is $a = -9{,}81\,\mathrm{m/s^2}$, niet nul. Anders zou de bal daar blijven hangen.



De auto wint 8 m/s in 4 s: 2,0 m/s². De fietser houdt zijn snelheid, de scooter verliest elke seconde 1,5 m/s.

UITLEG · 7.8

De eenparig versnelde rechtlijnige beweging (EVRB)

8/20

Bij een **constante versnelling** verandert de snelheid elke seconde met evenveel: de (v,t)-grafiek is een rechte. Dat is een **EVRB** (ook eenparig veranderlijke rechtlijnige beweging genoemd, want de versnelling mag negatief zijn). Met v_0 de snelheid op $t = 0$:

SNELHEID BIJ EEN EVRB

$$v(t) = v_0 + a\,t$$

Waar is het voorwerp? Gebruik de oppervlakte onder de (v,t)-grafiek. Tussen 0 en t is dat vlak een trapezium: een rechthoek met hoogte v_0 en een driehoek erbovenop met hoogte $a\,t$. De rechthoek levert $v_0\,t$, de driehoek $\frac{1}{2} \cdot t \cdot a\,t$. Samen:

PLAATS BIJ EEN EVRB

$$x(t) = x_0 + v_0\,t + \tfrac{1}{2}\,a\,t^2$$

Een kwadratische functie van t (hoofdstuk 4): de (x,t)-grafiek is een parabool. Met $v_0 = 2{,}0\,\text{m/s}$ en $a = 1{,}5\,\text{m/s}^2$ legt het voorwerp in 4 s $8 + 12 = 20\,\text{m}$ af.

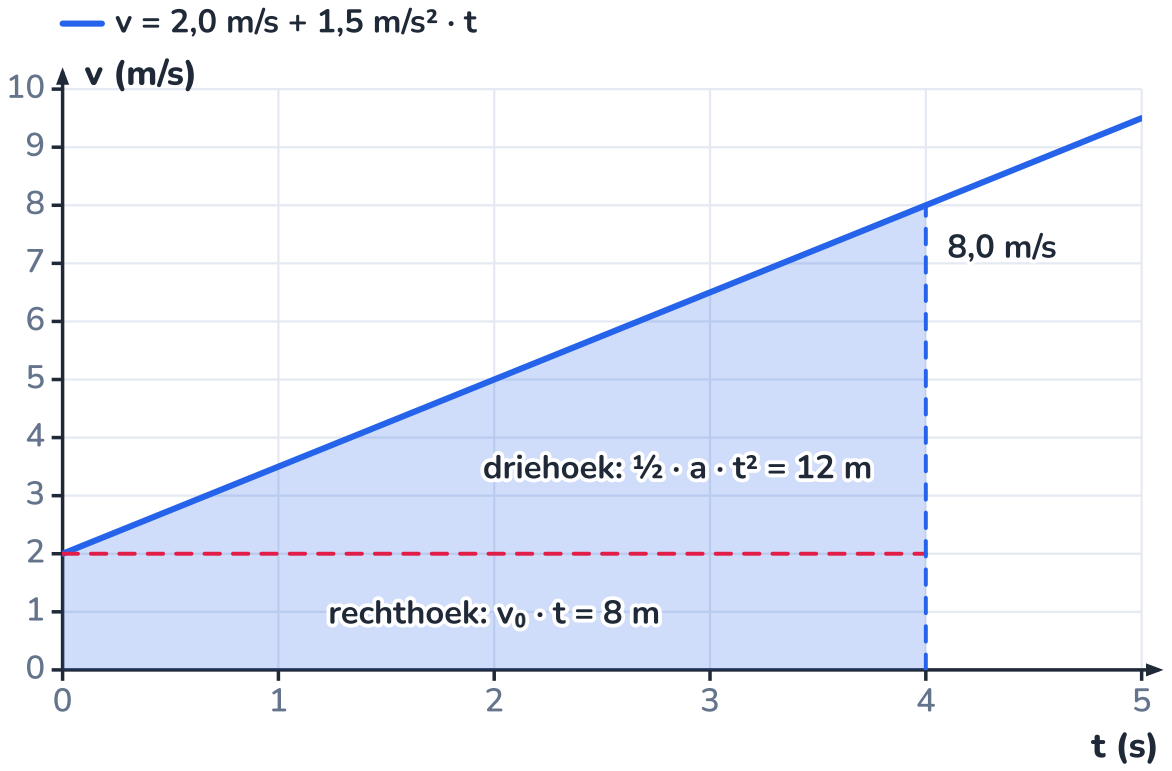
Zonder de tijd. Soms ken je de tijd niet, wel de afstand, zoals bij remmen. Het trapezium heeft oppervlakte $\Delta x = \dfrac{v_0 + v}{2} \cdot t$ (gemiddelde hoogte maal breedte), en uit $v = v_0 + a\,t$ volgt $t = \dfrac{v - v_0}{a}$. Vul in en vereenvoudig (merkwaardig product):

EVRB ZONDER TIJD, EN DE GEMIDDELDE SNELHEID

$$v^2 - v_0^2 = 2\,a\,\Delta x \qquad v_{\text{gem}} = \frac{v_0 + v}{2}$$

Controle met het voorbeeld: $8{,}0^2 - 2{,}0^2 = 60$ en $2 \cdot 1{,}5 \cdot 20 = 60$. Klopt.

De tweede formule geldt alleen bij een constante versnelling: dan is de gemiddelde snelheid het gemiddelde van begin- en eindsnelheid. Bij het robotje of de tram (werkboek) mag je ze niet gebruiken.



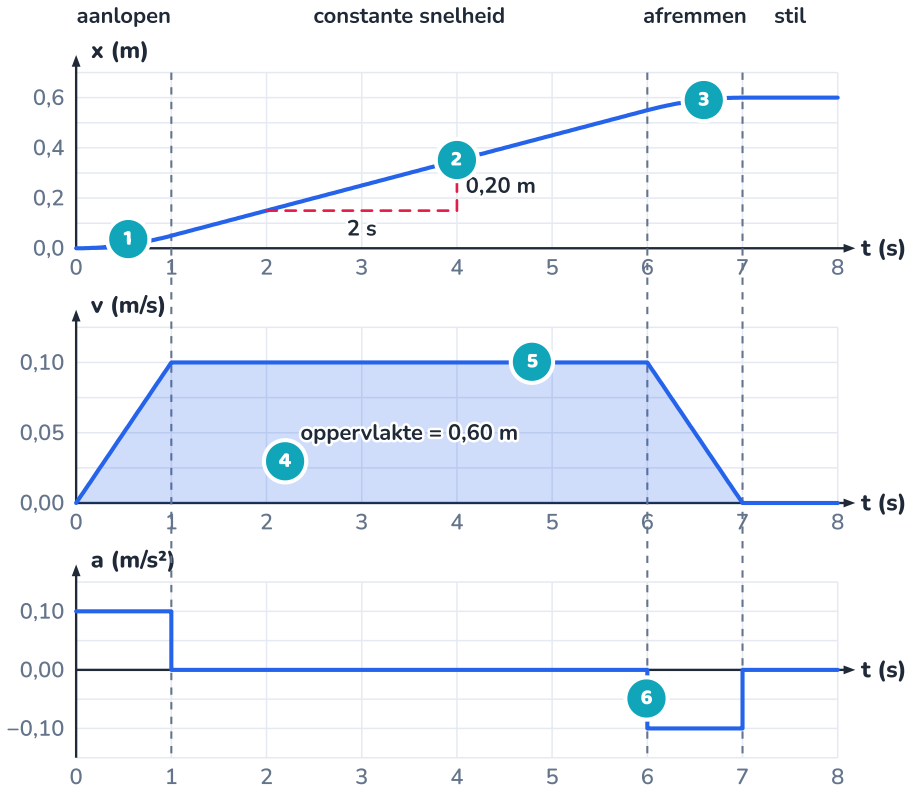
Rechthoek plus driehoek: 8 m + 12 m = 20 m verplaatsing in 4 s. Zo ontstaat de formule voor x(t).

ONTLEED · 7.9

Het kasdak in drie grafieken

9/20

Het kasdak van de klimaatmodule heeft een **schuifpaneel**: een polycarbonaatplaat in een aluminium kader die over twee rails langs de dakhelling schuift en een opening van 0,60 m vrijmaakt. De sturing laat de motor **zacht aanlopen en zacht stoppen**: 1,0 s versnellen tot 0,10 m/s, dan aan constante snelheid, en 1,0 s afremmen. Een schok aan het begin en het eind zou de riem, de rails en het paneel belasten (waarom, zie je in hoofdstuk 8). De drie grafieken tonen dezelfde beweging.



Plaats, snelheid en versnelling van het schuifpaneel op dezelfde tijdas.

1 **Aanlopen: een parabool**

$x = \frac{1}{2} \cdot 0{,}10 \cdot t^2$. De grafiek buigt naar boven (hol): elke tiende seconde gaat het paneel iets verder dan de vorige. Na 1,0 s heeft het maar 5 cm afgelegd.

2 **Helling = snelheid**

In de middenfase is de (x,t)-grafiek recht: 0,20 m in 2 s, dus 0,10 m/s. Dat is precies de hoogte van de (v,t)-grafiek eronder.

3 **Afremmen: de grafiek wordt vlak**

De helling neemt af tot nul: de (x,t)-grafiek buigt naar beneden (bol) en loopt horizontaal uit op 0,60 m. Een knik zou betekenen dat het paneel in één klap stilstaat.

4 **Oppervlakte = verplaatsing**

Het trapezium onder de (v,t)-grafiek: $\frac{7{,}0 + 5{,}0}{2} \cdot 0{,}10 = 0{,}60\,\text{m}$. Dat is de hoogte waarop de (x,t)-grafiek eindigt.

5 **Helling van (v,t) = versnelling**

Stijgend stuk: 0,10 m/s in 1,0 s, dus $a = 0{,}10\,\text{m/s}^2$. Vlak stuk: $a = 0$. Dalend stuk: $a = -0{,}10\,\text{m/s}^2$. De (a,t)-grafiek is die hellingen, als trappen.

6 **Een idealisering**

In dit model springt de versnelling in één klap van 0 naar −0,10 m/s². Bij een echte aandrijving gebeurt dat in een korte, maar niet nul tijd. Voor de verplaatsing maakt dat praktisch geen verschil; voor de schokken in de riem wel (hoofdstuk 8).

UITGEWERKT VOORBEELD · 7.10

Habitat: hoe lang duurt het openen?

10/20

De slag van het schuifpaneel is 0,60 m. De sturing versnelt en vertraagt met 0,10 m/s² en de topsnelheid is 0,10 m/s. Hoe lang moet de motor aan topsnelheid draaien, hoe lang duurt het openen, en wat is de gemiddelde snelheid?

Een getande riem trekt het paneel langs de rail; eindschakelaars stoppen de motor als het paneel helemaal open of dicht is.

GEGEVEN

- $\Delta x = 0{,}60\,\text{m}$
- $a = 0{,}10\,\text{m/s}^2$ (aanlopen), $-0{,}10\,\text{m/s}^2$ (afremmen)
- $v_{\text{max}} = 0{,}10\,\text{m/s}$

GEVRAAGD

- t_{constant}, totale tijd T, v_{gem}

OPLOSSING

1. Aanlopen (EVRB vanuit rust): $t_1 = \dfrac{v_{\text{max}}}{a} = \dfrac{0{,}10}{0{,}10} = 1{,}0\,\text{s}$; $\Delta x_1 = \tfrac{1}{2} \cdot 0{,}10 \cdot 1{,}0^2 = 0{,}050\,\text{m}$.
2. Afremmen is het spiegelbeeld: $t_3 = 1{,}0\,\text{s}$ en $\Delta x_3 = 0{,}050\,\text{m}$. (Controle met de formule zonder tijd: $0^2 - 0{,}10^2 = 2 \cdot (-0{,}10) \cdot \Delta x_3$ geeft $\Delta x_3 = 0{,}050\,\text{m}$.)
3. Middenfase (ERB): er blijft $0{,}60 - 0{,}050 - 0{,}050 = 0{,}50\,\text{m}$ over, aan 0,10 m/s: $t_2 = 0{,}50 / 0{,}10 = 5{,}0\,\text{s}$.
4. Totaal: $T = 1{,}0 + 5{,}0 + 1{,}0 = 7{,}0\,\text{s}$. Gemiddelde snelheid: $v_{\text{gem}} = 0{,}60 / 7{,}0 = 0{,}086\,\text{m/s}$.

ANTWOORD

De motor draait 5,0 s aan topsnelheid; openen duurt 7,0 s, met een gemiddelde snelheid van 0,086 m/s (8,6 cm/s).

Klopt dit? Oppervlakte van het trapezium onder de (v,t)-grafiek: $\dfrac{7{,}0 + 5{,}0}{2} \cdot 0{,}10 = 0{,}60\,\text{m}$. Klopt. Het zachte aanlopen kost weinig tijd: zonder aanloop (meteen 0,10 m/s) zou het openen 6,0 s duren, één seconde minder. Algemeen geldt $T = \dfrac{\Delta x}{v_{\text{max}}} + \dfrac{v_{\text{max}}}{a}$ (opdracht 7.13).

VERHAAL · 7.11

Twee loden kogels in Delft

11/20

Bijna tweeduizend jaar lang volgde men de Griekse filosoof Aristoteles: een zwaar voorwerp valt sneller dan een licht, en wel in verhouding tot zijn gewicht. Een kogel die tien keer zwaarder is, zou tien keer sneller vallen. Het klinkt logisch: een steen valt nu eenmaal sneller dan een veer.

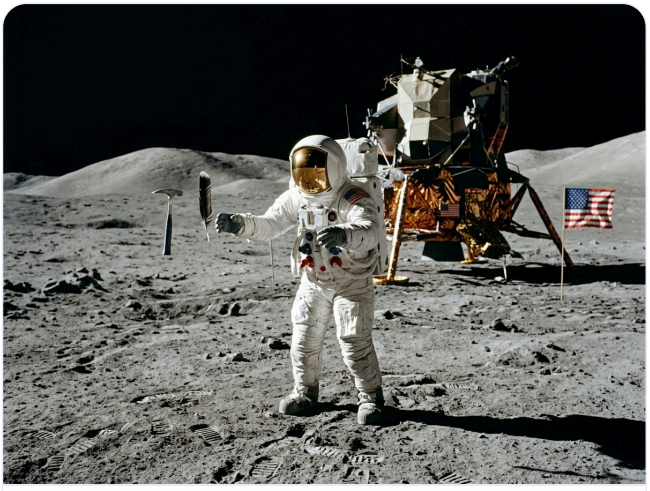
De Bruggeling **Simon Stevin**, ingenieur en wiskundige, testte het. In zijn boek over de weegkunde (*De Beghinselen der Weeghconst*, 1586) beschrijft hij hoe hij met zijn vriend Jan Cornets de Groot, burgemeester van Delft en vader van de rechtsgeleerde Hugo de Groot, twee loden kogels liet vallen, de ene tien keer zwaarder dan de andere, van ongeveer dertig voet hoog (een kleine 10 m) op een plank. Ze hoorden één klap, niet twee. Volgens Aristoteles had de lichte kogel pas lang na de zware moeten neerkomen. Het beroemde verhaal over Galilei die kogels van de scheve toren van Pisa liet vallen, kennen we alleen uit een levensbeschrijving door een leerling; historici twijfelen of die proef zo gebeurd is.

Waarom valt een veer dan toch trager? Door de **luchtweerstand**, die voor een lichte, grote veer veel meer uitmaakt dan voor een kleine, zware kogel. Zonder lucht vallen alle voorwerpen even snel. Dat werd op 2 augustus 1971 voor miljoenen tv-kijkers getoond: op de maan, waar geen lucht is, liet astronaut David Scott van de Apollo 15-missie een hamer en een valkenveer tegelijk los. Ze raakten tegelijk de grond, trager dan op aarde, want de valversnelling op de maan is maar 1,62 m/s².

Wat je uit dit verhaal meeneemt

Een goede proef kan een eeuwenoud 'gezond verstand' onderuithalen. En een model heeft voorwaarden: 'alle voorwerpen vallen even snel' geldt als de luchtweerstand te verwaarlozen is. Voor een loden kogel over 10 m is dat zo; voor een veer of een valschermspringer niet (hoofdstuk 4 en 8).

Stevin en De Groot hoorden de twee kogels tegelijk op de plank slaan: één klap, geen twee.



Zonder lucht vallen een hamer en een veer even snel: het klassieke experiment op de maan (1971).

UITLEG · 7.12

De vrije val

12/20

Een voorwerp dat alleen onder invloed van de zwaartekracht beweegt, maakt een **vrije val**. Zolang de luchtweerstand te verwaarlozen is, is dat een EVRB met een versnelling die voor alle voorwerpen dezelfde is: de **valversnelling** g.

VRIJE VAL VANUIT RUST (NAAR BENEDEN POSITIEF)

$$g = 9{,}81\,\mathrm{m/s^2} \qquad v = g\,t \qquad \Delta y = \tfrac{1}{2}\,g\,t^2 \qquad v = \sqrt{2\,g\,\Delta y}$$

Δy is de gevallen afstand. De laatste formule is de formule zonder tijd met $v_0 = 0$.

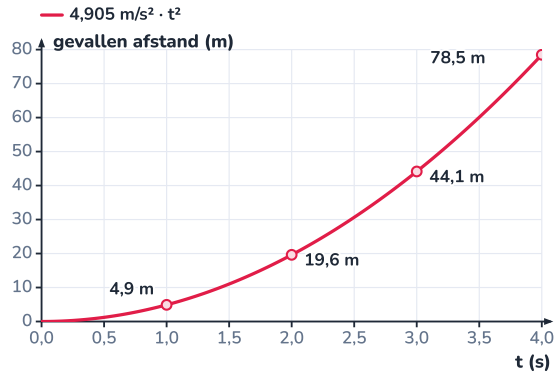
t (s)	0	1	2	3	4
snelheid $v = g\,t$ (m/s)	0	9,81	19,6	29,4	39,2
gevallen afstand $\frac{1}{2}g\,t^2$ (m)	0	4,9	19,6	44,1	78,5
afstand in die seconde (m)		4,9	14,7	24,5	34,3

In de laatste rij herken je de oneven getallen van Galilei uit hoofdstuk 4 (4,9 × 1, 3, 5, 7). De g in m/s² is hetzelfde getal als de 9,81 N/kg uit hoofdstuk F11: waarom dat zo is, zie je in hoofdstuk 8.

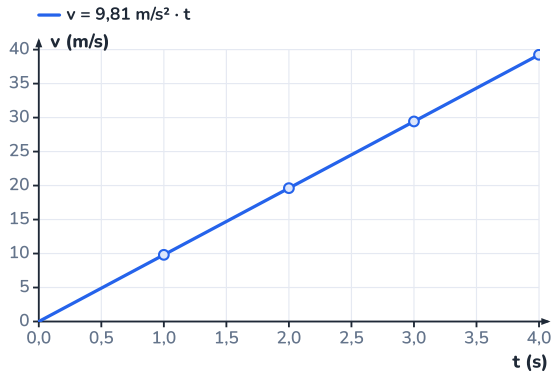
Omhoog gooien. Een bal die je omhoog gooit, maakt ook een vrije val, met een beginsnelheid. Neem omhoog als positieve zin: $v = v_0 - g\,t$ en $y = y_0 + v_0\,t - \frac{1}{2}g\,t^2$, de parabool van de worp uit hoofdstuk 4. De versnelling is tijdens de hele vlucht $-9{,}81\,\mathrm{m/s^2}$: bij het stijgen, op de top en bij het dalen.

⚠ Geldigheidsgebied

g is niet overal precies 9,81 m/s²: aan de evenaar ongeveer 9,78, aan de polen 9,83, in België 9,81. Belangrijker is de luchtweerstand: een steen die 1 s valt, wijkt nauwelijks af van het model; een valschermspringer haalt na ruim tien seconden een eindsnelheid van ongeveer 200 km/h (hoofdstuk 4) en versnelt dan niet meer. Waarom, leer je met de krachten in hoofdstuk 8.



De gevallen afstand groeit met het kwadraat van de tijd: na 2 s vier keer zo ver als na 1 s.



De snelheid groeit elke seconde met 9,81 m/s: een rechte door de oorsprong.

UITGEWERKT VOORBEELD · 7.13

Hoe diep is de put?

13/20

Je laat een steen in een droge put vallen en hoort hem 2,4 s later op de bodem slaan. Hoe diep is de put ongeveer? Met welke snelheid raakt de steen de bodem? Hoe groot is de fout omdat het geluid tijd nodig heeft om terug te komen?

GEGEVEN

- $t = 2{,}4\,\mathrm{s}$ (tot je het hoort)
- $g = 9{,}81\,\mathrm{m/s^2}$; geluidssnelheid $343\,\mathrm{m/s}$

GEVRAAGD

- diepte h
- snelheid v bij de bodem
- invloed van de looptijd van het geluid

OPLOSSING

1 Eerste benadering: verwaarloos de looptijd van het geluid en de luchtweerstand. Dan valt de steen 2,4 s: $h = \frac{1}{2} \cdot 9{,}81 \cdot 2{,}4^2 = 28\,\text{m}$.

2 Snelheid bij de bodem: $v = g\,t = 9{,}81 \cdot 2{,}4 = 23{,}5\,\text{m/s}$, ongeveer 85 km/h. (Of zonder tijd: $v = \sqrt{2 \cdot 9{,}81 \cdot 28{,}25} = 23{,}5\,\text{m/s}$.)

3 Het geluid legt 28 m terug af in $28/343 = 0{,}08\,\text{s}$. De steen viel dus maar ongeveer 2,32 s, en de put is minder diep: $\frac{1}{2} \cdot 9{,}81 \cdot 2{,}32^2 = 26\,\text{m}$.

ANTWOORD

De put is ongeveer 26 m diep (28 m als je het geluid verwaarloost), en de steen raakt de bodem met ongeveer 23 m/s.

Klopt dit? De correctie voor het geluid is ongeveer 7 %: kleiner dan de onzekerheid van een stopwatch die je met de hand indrukt (je reactietijd, ongeveer 0,2 s, geeft al bijna 5 m verschil). Een exacte oplossing, waarin valtijd en looptijd van het geluid samen 2,4 s moeten zijn, is opdracht 7.14.

UITLEG · 7.14

Reactietijd en remweg

14/20

Een bestuurder ziet een gevaar. Voor de remmen werken, verstrijkt de **reactietijd** t_r: zien, begrijpen, beslissen, de voet verplaatsen. In het verkeer rekent men met ongeveer 1 s (voor een alerte bestuurder in een eenvoudige situatie wat minder, moe of afgeleid veel meer). Tijdens die reactietijd rijdt de auto gewoon door aan zijn snelheid: een ERB. Daarna remt hij tot stilstand, bij benadering met een constante vertraging a: een EVRB.

STOPAFSTAND

$$d_{\text{stop}} = \underbrace{v\,t_r}_{\text{reactieafstand}} + \underbrace{\frac{v^2}{2\,a}}_{\text{remafstand}}$$

v in m/s, t_r in s, a de grootte van de remvertraging in m/s². De remafstand volgt uit de formule zonder tijd: $0^2 - v^2 = 2 \cdot (-a) \cdot \Delta x$.

In de (v,t)-grafiek zie je de twee stukken als twee oppervlaktes: een rechthoek (reactie) en een driehoek (remmen). De remvertraging hangt vooral af van het wegdek en de banden:

wegdek (goede banden)	remvertraging (orde van grootte)
droog asfalt	7 à 8 m/s²
nat asfalt	ongeveer 5 m/s²
sneeuw	ongeveer 2 m/s²
ijzel	1 m/s² of minder

De reactieafstand groeit evenredig met de snelheid, de remafstand met het **kwadraat**: twee keer zo snel is vier keer zo ver remmen. Daarom wordt een stopafstand bij hoge snelheid zo snel groot:

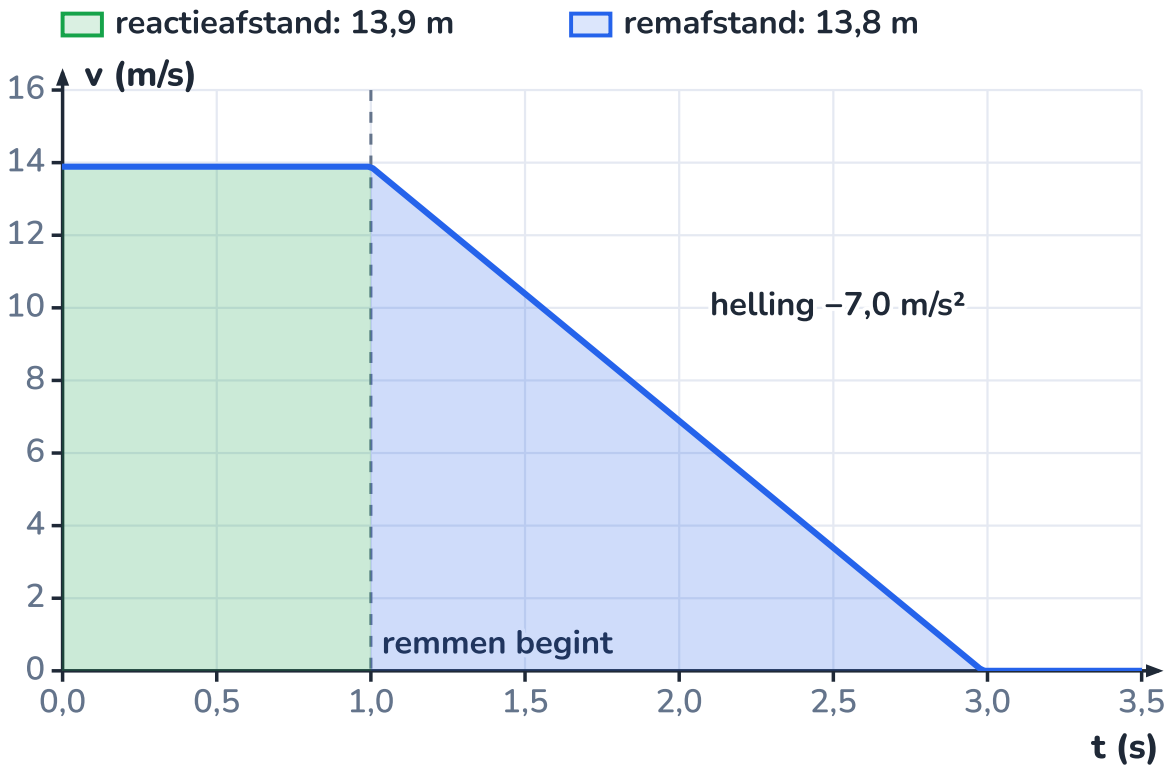
snelheid	reactieafstand	remafstand	stopafstand
30 km/h	8,3 m	5,0 m	13,3 m
50 km/h	13,9 m	13,8 m	27,7 m
70 km/h	19,4 m	27,0 m	46,5 m
90 km/h	25,0 m	44,6 m	69,6 m

snelheid	reactieafstand	remafstand	stopafstand
120 km/h	33,3 m	79,4 m	112,7 m

Reactietijd 1,0 s, remvertraging 7,0 m/s² (droog asfalt).

Afleiding kost meters

Wie een bericht leest op een smartphone, kijkt al snel twee seconden niet naar de weg. Bij 50 km/h is dat 28 m 'blind' rijden: meer dan de hele stopafstand van een alerte bestuurder. Alcohol, vermoeidheid en afleiding verlengen de reactietijd; nat of glad wegdek verkleint de remvertraging. Beide tellen op.



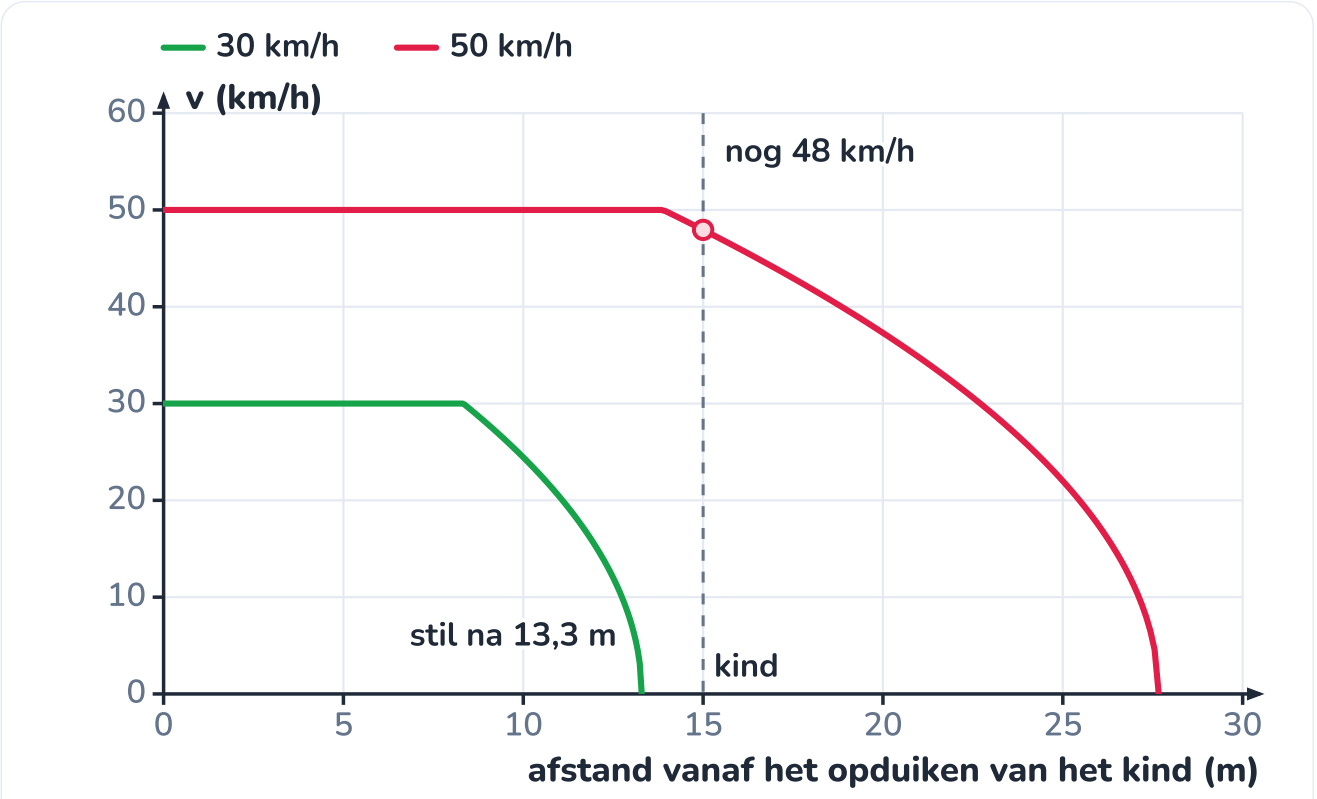
Eerst 1,0 s doorrijden (rechthoek), dan remmen tot stilstand (driehoek): samen 27,7 m.

UITGEWERKT VOORBEELD · 7.15

Vijftien meter voor een kind

15/20

In een schoolstraat stapt een kind plots 15 m voor een auto de weg op. Reken met een reactietijd van 1,0 s en een remvertraging van 7,0 m/s² (droog). Komt de auto op tijd tot stilstand als hij 30 km/h rijdt? En als hij 50 km/h rijdt: met welke snelheid raakt hij het kind dan nog?



Aan 30 km/h zakt de snelheid tot nul voor de 15 m; aan 50 km/h begint het remmen pas na 13,9 m.

GEGEVEN

- $d = 15\,\text{m}$, $t_r = 1{,}0\,\text{s}$, $a = 7{,}0\,\text{m/s}^2$
- $v = 30\,\text{km/h}$ of $50\,\text{km/h}$

GEVRAAGD

- stopafstand bij 30 km/h
- snelheid op 15 m bij 50 km/h

OPLOSSING

1. 30 km/h $= 8{,}33\,\text{m/s}$. Reactieafstand $8{,}33 \cdot 1{,}0 = 8{,}3\,\text{m}$; remafstand $\frac{8{,}33^2}{2 \cdot 7{,}0} = 5{,}0\,\text{m}$. Stopafstand 13,3 m: de auto staat 1,7 m voor het kind stil.
2. 50 km/h $= 13{,}9\,\text{m/s}$. Reactieafstand $13{,}9\,\text{m}$: na 1 s is de auto al bijna bij het kind, en hij heeft nog niet geremd.
3. Hij remt nog over $15 - 13{,}9 = 1{,}1\,\text{m}$. Formule zonder tijd: $v^2 = 13{,}9^2 - 2 \cdot 7{,}0 \cdot 1{,}1 = 177\,\text{m}^2/\text{s}^2$, dus $v = 13{,}3\,\text{m/s} = 48\,\text{km/h}$.

ANTWOORD

Aan 30 km/h staat de auto na 13,3 m stil, net voor het kind. Aan 50 km/h raakt hij het kind met ongeveer 48 km/h: bijna zonder snelheid te verliezen.

Klopt dit? Het verschil tussen 30 en 50 km/h lijkt klein op de teller, maar het is het verschil tussen stilstaan en een aanrijding aan bijna 50 km/h. Dat is de reden voor zones 30 rond scholen. Hoeveel energie er bij zo'n aanrijding vrijkomt, reken je uit in hoofdstuk 9.

PROEF · 7.16

Een karretje meten met een afstandssensor

16/20

ONDERZOEKSVRAAG

Een karretje rolt vanuit rust van een licht hellende plank. Is dat een EVRB, en hoe groot is de versnelling?

MATERIAAL

plank van 1,5 m, aan één kant 8 cm opgehoogd

karretje met lichte, vlot draaiende wielen

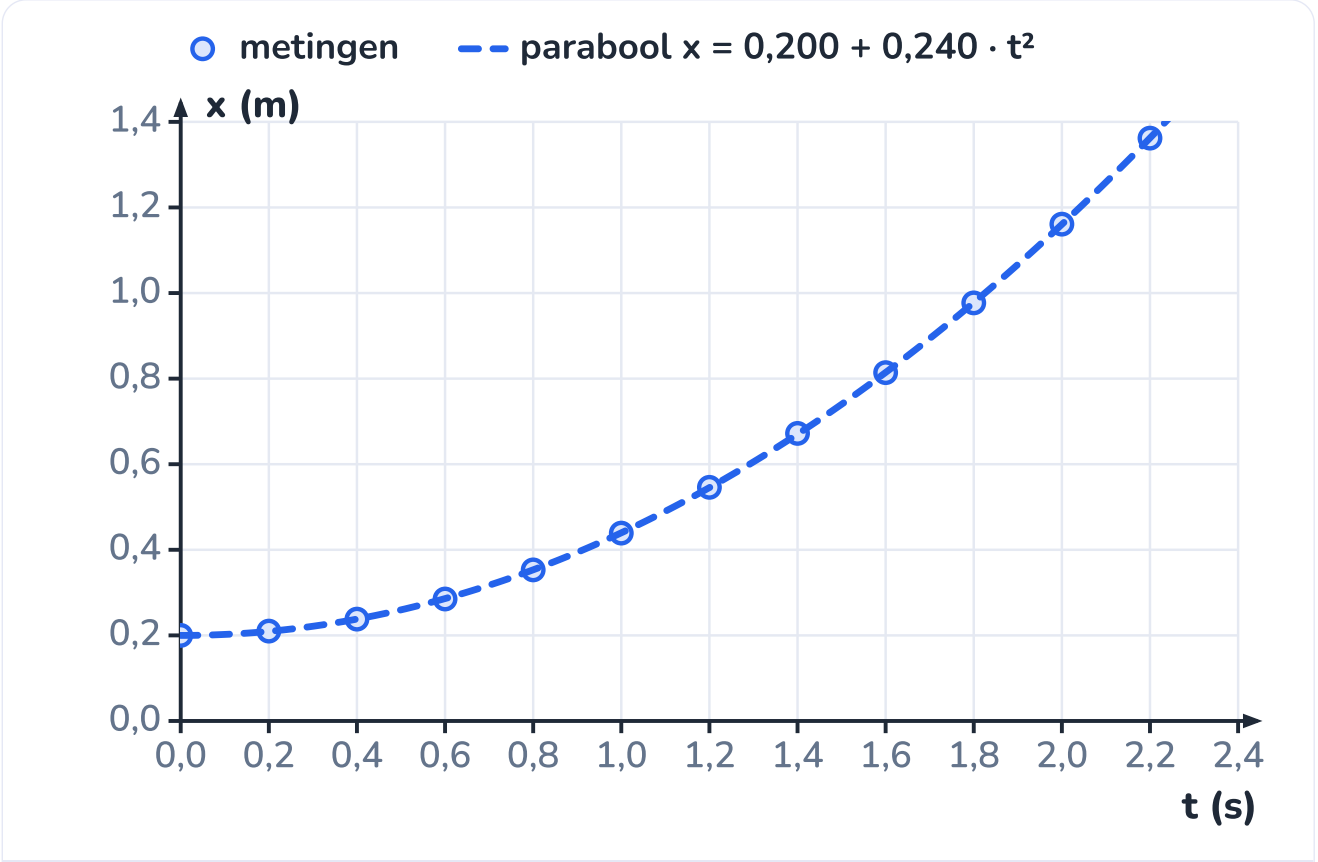
ultrasone afstandssensor HC-SR04 met Arduino (code op het volgende blad) en een computer

rekenblad of Python

als alternatief: smartphone op statief en een meetlint in beeld (videoanalyse)

Veiligheid

Vang het karretje op aan het eind van de plank. Werk op een tafel zonder snoeren in de baan.



De plaats groeit kwadratisch met de tijd: een parabool door de meetpunten.

WERKWIJZE

1. Zet de sensor bovenaan de plank, gericht langs de plank. Zet het karretje 20 cm voor de sensor en houd het vast.
2. Start de meting en laat het karretje los zonder duwen. De sensor meet elke 0,1 s de afstand.
3. Neem uit de meting één punt om de 0,2 s, vanaf het loslaten. Zet x tegen t in een grafiek.
4. Bereken per interval de gemiddelde snelheid $\Delta x / \Delta t$ en zet ze uit in het midden van het interval.
5. Is de (v,t)-grafiek een rechte? Bepaal de helling: de versnelling.

WAARNEMING

t (s)	0,0	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6	1,8	2,0	2,2
x (m)	0,200	0,210	0,238	0,285	0,353	0,439	0,546					

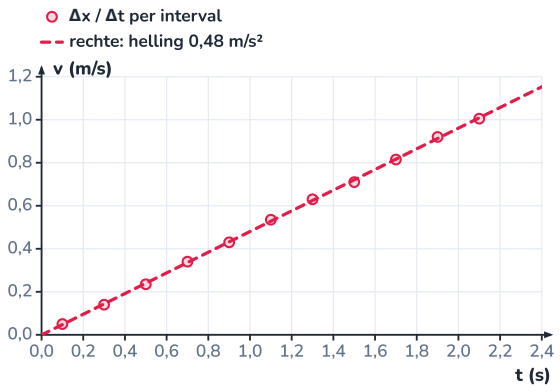
De (x,t)-punten liggen op een parabool. De snelheden per interval (0,050; 0,140; 0,235; ... 1,005 m/s) liggen op een rechte met een helling van 0,48 m/s². De rechte gaat (bijna) door de oorsprong: het karretje vertrok vanuit rust.

VERKLARING

[illegible] een constante versnelling: het is een LVRB met a [illegible] $0{,}48\,\mathrm{m/s^2}$. Controle: de parabool door de (x,t)-punten is $x = 0{,}200 + 0{,}240\,t^2$, en $\frac{1}{2}a = 0{,}240$ geeft ook $a = 0{,}48\,\mathrm{m/s^2}$.

Waarom zet je de snelheid in het **midden** van het interval? Bij een constante versnelling is de gemiddelde snelheid over een interval gelijk aan de ogenblikkelijke snelheid in het midden ervan ($v_{\text{gem}} = \frac{v_0+v}{2}$). Zou je ze aan het begin van het interval zetten, dan schuift de hele rechte 0,1 s op.

Zonder wrijving voorspelt hoofdstuk 6 een versnelling $g \sin\alpha = 9{,}81 \cdot 0{,}08/1{,}5 = 0{,}52\,\mathrm{m/s^2}$. De gemeten waarde is 8 % kleiner: rolweerstand, en de wielen moeten ook gaan draaien. Merk ook op dat de snelheden meer schommelen dan de plaatsen: bij een verschil tellen de absolute fouten op (hoofdstuk 1), dus twee keer 1,5 mm is 3 mm, en gedeeld door 0,2 s wordt dat tot 0,015 m/s op de snelheid. Afleiden versterkt ruis.



De snelheid per interval ligt op een rechte: de versnelling is constant, 0,48 m/s².

De sensor stuurt een ultrasone puls naar het karretje en meet hoe lang de echo onderweg is.

CODE · 7.17

Arduino: elke 0,1 s de plaats meten

17/20

De HC-SR04 zendt een korte ultrasone puls uit en meet hoe lang het duurt tot de echo terugkomt. Het geluid legt de afstand heen en terug af, dus $x = 343\,\mathrm{m/s} \cdot t_{\text{echo}}/2$. De sketch meet op vaste tijdstippen met `millis()` in plaats van `delay()`, zodat de tijdstap niet langer wordt door de meting zelf (hoofdstuk 29). De uitvoer is een tabel met een puntkomma tussen de kolommen, die je in een rekenblad of in Python inleest.

C++ (Arduino) afstand.ino 34 regels

```
// afstand.ino - de plaats van een karretje meten met een HC-SR04, elke 0,1 s
const int TRIG = 9;                    // trigger-pin van de sensor
const int ECHO = 10;                   // echo-pin van de sensor
const unsigned long STAP_MS = 100;     // meetinterval in milliseconden
unsigned long start;                   // tijdstip van de eerste meting
unsigned long n = 0;                   // nummer van de meting

float meetAfstand() {                  // afstand in meter
```

```
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);           // een puls van 10 microseconden start een meting
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  unsigned long duur = pulseIn(ECHO, HIGH, 30000UL);   // looptijd heen en terug (microseconden)
  return duur * 343.0 / 2.0 / 1000000.0;             // geluid: 343 m/s bij 20 graden
}

void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(115200);
  Serial.println("t (s);x (m)");
  start = millis();
}

void loop() {
  if (millis() - start >= n * STAP_MS) {   // tijd voor de volgende meting?
    float x = meetAfstand();
    Serial.print(n * STAP_MS / 1000.0, 1);  // tijd in seconden, 1 decimaal
    Serial.print(';');
    Serial.println(x, 3);                   // plaats in meter, 3 decimalen
    n++;
  }
}
```

Aansluiten: VCC op 5 V, GND op GND, TRIG op pin 9, ECHO op pin 10. De sensor meet van ongeveer 2 cm tot een paar meter, met een onzekerheid van enkele millimeter. Krijgt de sensor geen echo binnen 30 ms, dan geeft `pulseIn` 0 terug en print de sketch $x = 0$: gooi die regels weg. De geluidssnelheid hangt af van de temperatuur (331 m/s bij 0 °C, 343 m/s bij 20 °C): in een koude kas meet de sensor dus een paar procent te veel.

CODE · 7.18

Python: het kasdak stap voor stap

18/20

Een computer kent geen oppervlaktes of hellingen, maar kan wel heel veel kleine stapjes zetten. Het programma verdeelt de beweging van het schuifpaneel in stapjes van 0,01 s. In elk stapje verandert de snelheid met $a \cdot \Delta t$ (de helling van de (v,t)-grafiek) en de plaats met de oppervlakte van een smal trapezium onder de (v,t)-grafiek. Zo bouwt het de (v,t)- en de (x,t)-grafiek op uit de (a,t)-grafiek.

Python **kasdak.py** 26 regels

```python
# kasdak.py - de beweging van het schuifpaneel stap voor stap berekenen
DT = 0.01          # tijdstap (s)
A = 0.10           # versnelling bij aanlopen en afremmen (m/s2)


def versnelling(t):
    """Het bewegingsprofiel: 1 s aanlopen, 5 s constante snelheid, 1 s afremmen."""
    if t < 1.0:
        return A
    elif t < 6.0:
        return 0.0
    elif t < 7.0:
        return -A
    return 0.0


v, x = 0.0, 0.0
print("  t (s)   a (m/s2)   v (m/s)   x (m)")
for stap in range(800):                  # 800 stappen van 0,01 s = 8 s
    t = stap * DT
    a = versnelling(t + DT / 2)          # versnelling in het midden van de stap
```

```
    v_nieuw = max(v + a * DT, 0.0)        # a is de helling van (v,t); het paneel keert niet om
    x = x + (v + v_nieuw) / 2 * DT        # oppervlakte van een smal trapezium onder v
    v = v_nieuw
    if (stap + 1) % 50 == 0:              # om de 0,5 s een regel
        print(f"{t + DT:6.1f}   {a:8.2f}   {v:7.3f}   {x:6.3f}")
```

Uitvoer (gecontroleerd door het programma te draaien; de regels tussen 2 en 5,5 s zijn weggelaten):

```
t (s)   a (m/s2)   v (m/s)   x (m)
 0.5       0.10     0.050    0.013
 1.0       0.10     0.100    0.050
 1.5       0.00     0.100    0.100
 2.0       0.00     0.100    0.150
 ...
 5.5       0.00     0.100    0.500
 6.0       0.00     0.100    0.550
 6.5      -0.10     0.050    0.588
 7.0      -0.10     0.000    0.600
 7.5       0.00     0.000    0.600
 8.0       0.00     0.000    0.600
```

Na 7,0 s staat het paneel stil op 0,600 m, zoals het voorbeeld voorspelde. Het voordeel van deze aanpak: verander de functie `versnelling` (een ander profiel, een haperende motor, een versnelling die van de kracht afhangt) en het programma rekent de beweging opnieuw uit, ook als er geen formule meer voor bestaat. In hoofdstuk 8 laat je de versnelling volgen uit de krachten; in hoofdstuk E4 leer je hoe groot de fout van zo'n stapmethode is.

NASLAG · 7.19

Overzicht: beweging langs een rechte

19/20

definities

$\Delta x = x_2 - x_1 \qquad v_{\text{gem}} = \dfrac{\Delta x}{\Delta t} \qquad a = \dfrac{\Delta v}{\Delta t}$

symbool	betekenis	eenheid
x	plaats	m
v	snelheid	m/s
a	versnelling	m/s²
t	tijd	s

ERB (v constant, a = 0)

$$x = x_0 + v\,t$$

EVRB (a constant)

$v = v_0 + a\,t \qquad x = x_0 + v_0\,t + \frac{1}{2}\,a\,t^2$

EVRB zonder tijd

$v^2 - v_0^2 = 2\,a\,\Delta x \qquad v_{\text{gem}} = \dfrac{v_0 + v}{2}$

vrije val vanuit rust

$v = g\,t \qquad \Delta y = \frac{1}{2}\,g\,t^2 \qquad v = \sqrt{2\,g\,\Delta y}$

symbool	betekenis	eenheid
g	valversnelling, 9,81 in België	m/s²

stopafstand

$$d_{\text{stop}} = v\,t_r + \frac{v^2}{2\,a}$$

symbool	betekenis	eenheid
t_r	reactietijd, ongeveer 1 s	s
a	remvertraging, droog 7 à 8	m/s²

grafiek	helling is	oppervlakte eronder is
(x,t)	de snelheid v	(geen betekenis)
(v,t)	de versnelling a	de verplaatsing Δx
(a,t)	(niet nodig in dit hoofdstuk)	de snelheidsverandering Δv

beweging	(x,t)	(v,t)	(a,t)
stilstand	horizontaal	op nul	op nul
ERB	schuine rechte	horizontaal	op nul
EVRB, sneller	parabool, hol	rechte weg van nul	horizontaal, zelfde teken als v
EVRB, trager	parabool, bol	rechte naar nul	horizontaal, tegengesteld aan v

snelheid	reactieafstand	remafstand	stopafstand
30 km/h	8,3 m	5,0 m	13,3 m
50 km/h	13,9 m	13,8 m	27,7 m
70 km/h	19,4 m	27,0 m	46,5 m
90 km/h	25,0 m	44,6 m	69,6 m
120 km/h	33,3 m	79,4 m	112,7 m

Reactietijd 1,0 s, remvertraging 7,0 m/s² (droog asfalt).

controlelijst
as gekozen: oorsprong en positieve zin; tekens van v en a daarmee in overeenstemming
km/h omgezet naar m/s (delen door 3,6)
formules van de EVRB alleen gebruikt bij een constante versnelling; anders in fasen werken
antwoord gecontroleerd met een tweede methode (oppervlakte, formule zonder tijd)

HET GEHEEL · 7.20

Waar beweging terugkomt

20/20

Dit hoofdstuk beschrijft *hoe* iets beweegt, zonder te vragen *waarom*. Het waarom zijn de krachten: in het volgende hoofdstuk koppel je de versnelling aan de resulterende kracht. Samen vormen ze de basis van de mechanica. Wiskundig zijn helling en oppervlakte het begin van de afgeleide en de integraal, de twee pijlers van Expert.

F14 Beweging en snelheid
Gemiddelde snelheid, km/h en m/s, de (s,t)-grafiek en de eenparige beweging.

I4 Functies als modellen
De lineaire en de kwadratische functie: nu met een fysische betekenis voor elke coëfficiënt.

I6 Goniometrie en vectoren
Snelheid en versnelling zijn vectoren; op een helling is de versnelling g sin α.

I8 De wetten van Newton
Waarom een voorwerp versnelt: de resulterende kracht, F = m · a.

I9 Arbeid, energie en vermogen
Waarom de remweg met v² groeit: de bewegingsenergie ½ m v².

I26 Motoren en actuatoren
De motor van het kasdak kiezen en zacht laten aanlopen met PWM.

I33 Een mobiele robot
Een robot die zijn plaats bijhoudt uit snelheid en tijd, en afstand meet met ultrasoon geluid.

E1 Afgeleiden: verandering meten
De raaklijn als afgeleide: v = dx/dt en a = dv/dt, exact uitgerekend.

E3 Integralen: ophopen
De oppervlakte onder de (v,t)-grafiek als integraal.

E8 Beweging in twee dimensies
Plaats, snelheid en versnelling als vectoren: de schuine worp en de

cirkelbeweging.

Habitat

Voor Habitat leggen we vast: het kasdak van de klimaatmodule heeft een schuifpaneel met een slag van 0,60 m. De sturing versnelt en vertraagt met 0,10 m/s² tot een topsnelheid van 0,10 m/s: openen en sluiten duurt 7,0 s. In opdracht 7.13 ontwerp je een snellere sluiting voor als het begint te regenen; in hoofdstuk 8 bereken je de kracht die de dakmotor daarvoor moet leveren.

SAMENVATTING

In het kort

- Kies eerst een as: oorsprong en positieve zin. Verplaatsing $\Delta x = x_2 - x_1$ heeft een teken; afgelegde weg is altijd positief.
- Gemiddelde snelheid is $\Delta x / \Delta t$, de helling van de koorde in de (x,t)-grafiek; de ogenblikkelijke snelheid is de helling van de raaklijn.
- ERB: constante snelheid, $x = x_0 + v\,t$. EVRB: constante versnelling, $v = v_0 + a\,t$ en $x = x_0 + v_0 t + \frac{1}{2}at^2$, en zonder tijd $v^2 - v_0^2 = 2a\Delta x$.
- Versnelling $a = \Delta v / \Delta t$ in m/s² is de helling van de (v,t)-grafiek. Hetzelfde teken als v: sneller; tegengesteld: trager.
- De oppervlakte onder de (v,t)-grafiek is de verplaatsing, ook als de beweging uit fasen bestaat.
- Vrije val zonder luchtweerstand: een EVRB met $g = 9{,}81\ \mathrm{m/s^2}$ voor alle voorwerpen, $\Delta y = \frac{1}{2}gt^2$ en $v = \sqrt{2g\Delta y}$.
- Stopafstand = reactieafstand $v\,t_r$ + remafstand $v^2/(2a)$. Twee keer zo snel is vier keer zo ver remmen.
- Een computer rekent een beweging uit in kleine stapjes: v verandert met $a\,\Delta t$, x met de oppervlakte onder v.

BEGRIPPEN

Wat je nu kent

plaats
De coördinaat x van een voorwerp op een gekozen as met een oorsprong en een positieve zin.

verplaatsing
De verandering van plaats $\Delta x = x_2 - x_1$; kan positief, negatief of nul zijn.

afgelegde weg
De totale lengte van het gevolgde traject, altijd positief; bij heen en terug groter dan de verplaatsing.

gemiddelde snelheid
Verplaatsing gedeeld door tijdsduur, $v_{\text{gem}} = \Delta x / \Delta t$; de helling van de koorde in de (x,t)-grafiek.

ogenblikkelijke snelheid
De snelheid op één tijdstip: de helling van de raaklijn aan de (x,t)-grafiek.

versnelling
Verandering van snelheid per tijdseenheid, $a = \Delta v / \Delta t$, in m/s²; de helling van de (v,t)-grafiek.

ERB
Eenparig rechtlijnige beweging: langs een rechte met constante snelheid, $x = x_0 + v\,t$.

EVRB
Eenparig veranderlijke (versnelde) rechtlijnige beweging: constante versnelling, $v = v_0 + a\,t$ en $x = x_0 + v_0 t + \frac{1}{2}at^2$.

vrije val
Beweging onder invloed van alleen de zwaartekracht; zonder luchtweerstand een EVRB met $a = g = 9{,}81\ \mathrm{m/s^2}$.

reactieafstand
De afstand die je aflegt tussen het zien van een gevaar en het begin van het remmen: $v \cdot t_r$.

remafstand

De afstand die je nodig hebt om tot stilstand te remmen: $v^2/(2a)$ bij constante remvertraging a.

HANDBOEK · HOOFDSTUK 8

De wetten van Newton

Drie wetten die verklaren waarom iets versnelt, remt, of gewoon blijft doen wat het deed

Waarom glijdt een curlingsteen zo ver, waarom voel je je zwaarder in een lift die vertrekt, en hoeveel kracht moet een motor leveren om een dakpaneel een helling op te trekken? In hoofdstuk 7 beschreef je bewegingen; nu verklaar je ze. De drie wetten van Newton verbinden kracht, massa en versnelling. Met een **vrijlichaamsdiagram** breng je alle krachten op een voorwerp in kaart, met de **wrijvingscoëfficiënt** reken je wrijving uit, en op een helling combineer je alles met het ontbinden uit hoofdstuk 6. Zo bepaal je de kracht die de dakmotor van Habitat moet leveren.

8× uitleg · 6× uitgewerkt voorbeeld · 2× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

K Krachten, beweging en constructies · W Wiskundig modelleren

Een curlingsteen glijdt tientallen meters ver: bijna zonder wrijving blijft hij doen wat hij deed.

Na dit hoofdstuk

- Je kan de resulterende kracht bepalen van krachten langs één lijn en van loodrechte krachten.
- Je kan de eerste wet (traagheid) toepassen en uitleggen waarom er geen kracht nodig is om te blijven bewegen.
- Je kan met $F_{\text{res}} = m \cdot a$ een kracht, een massa of een versnelling berekenen, in een of twee richtingen.
- Je kan actie-reactieparen herkennen en uitleggen waarom ze elkaar niet opheffen.
- Je kan een vrijlichaamsdiagram tekenen met zwaartekracht, normaalkracht, spankracht, wrijving en duwkracht.
- Je kan statische en dynamische wrijving berekenen met een wrijvingscoëfficiënt, en een coëfficiënt meten.
- Je kan de krachten op een voorwerp op een helling ontbinden en de versnelling of de benodigde kracht berekenen.
- Je kan een beweging met een veranderlijke kracht stap voor stap simuleren met de tweede wet.

UITLEG · 8.1

Wat je al weet, en wat nu komt

1/21

In hoofdstuk F11 leerde je dat een kracht een voorwerp kan vervormen, versnellen, afremmen of doen draaien, dat je een kracht tekent als een pijl, en dat je ze meet in newton met een krachtmeter. Je berekende de zwaartekracht $F_z = m \cdot g$. In hoofdstuk 6 werden krachten vectoren: je telde ze op en ontbond ze, ook op een helling. In hoofdstuk 7 beschreef je bewegingen met snelheid en versnelling.

begrip	symbool en eenheid	waar
kracht: grootte, richting, aangrijpingspunt	$\vec{F}$ in newton (N)	F11
zwaartekracht	$F_z = m \cdot g$, $g = 9{,}81\,\mathrm{N/kg}$	F11
veerkracht en de krachtmeter (dynamometer)	wet van Hooke	F11
krachten optellen en ontbinden	$F_x = F\cos\alpha$, $F_y = F\sin\alpha$	hoofdstuk 6
zwaartekracht op een helling	$F_{\parallel} = mg\sin\alpha$, $F_{\perp} = mg\cos\alpha$	hoofdstuk 6
versnelling	$a = \Delta v/\Delta t$ in m/s²	hoofdstuk 7

Wat ontbreekt, is het verband: **hoe** bepalen de krachten op een voorwerp zijn beweging? Dat verband zijn de drie wetten van Newton. In dit hoofdstuk

- tel je alle krachten op een voorwerp op tot één **resulterende kracht**;
- leer je de drie wetten en de misvattingen die er hardnekkig rond hangen;
- teken je **vrijlichaamsdiagrammen**, het belangrijkste gereedschap van de mechanica;
- reken je met **wrijving** en met voorwerpen op een **helling**.

UITLEG · 8.2

De resulterende kracht

2/21

Op een voorwerp werken bijna altijd meerdere krachten tegelijk. Voor de beweging telt alleen hun **vectorsom**: de **resulterende kracht** $\vec{F}_{\text{res}}$ (ook resultante genoemd).

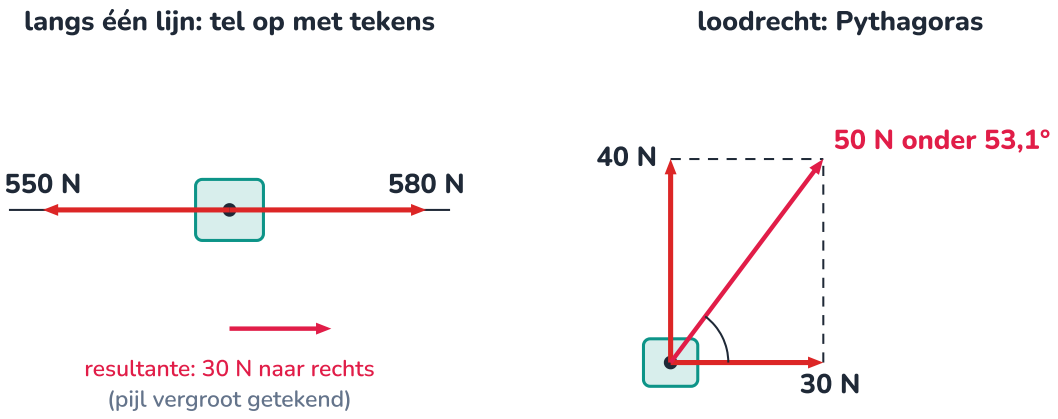
- **Langs één lijn**: kies een positieve zin, geef elke kracht een teken en tel op. Bij touwtrekken met 580 N naar rechts en 550 N naar links is de resultante $+580 - 550 = +30\,\mathrm{N}$: 30 N naar rechts.
- **Loodrecht op elkaar**: Pythagoras voor de grootte, de tangens voor de richting. 30 N en 40 N loodrecht geven $\sqrt{30^2 + 40^2} = 50\,\mathrm{N}$ onder $\tan^{-1}(40/30) = 53{,}1°$.
- **In het algemeen**: ontbind elke kracht langs twee loodrechte assen en tel per as op, zoals bij de sleepboten in hoofdstuk 6: $F_{\text{res},x} = \sum F_x$ en $F_{\text{res},y} = \sum F_y$.

RESULTERENDE KRACHT

$$\vec{F}_{\text{res}} = \vec{F}_1 + \vec{F}_2 + \ldots \qquad F_{\text{res},x} = \sum F_x \qquad F_{\text{res},y} = \sum F_y$$

Is de resulterende kracht nul, dan zijn de krachten **in evenwicht**. Dat betekent niet dat er geen krachten werken, alleen dat ze elkaar opheffen.

Belangrijk: je telt alleen krachten op die op **hetzelfde voorwerp** werken. Welke dat zijn, zie je het best in een vrijlichaamsdiagram (verderop).



Links heffen 580 N en 550 N elkaar bijna op; rechts vormen 30 N en 40 N de zijden van een rechthoek met diagonaal 50 N.

VERHAAL · 8.3

Waarom stopt een kar?

3/21

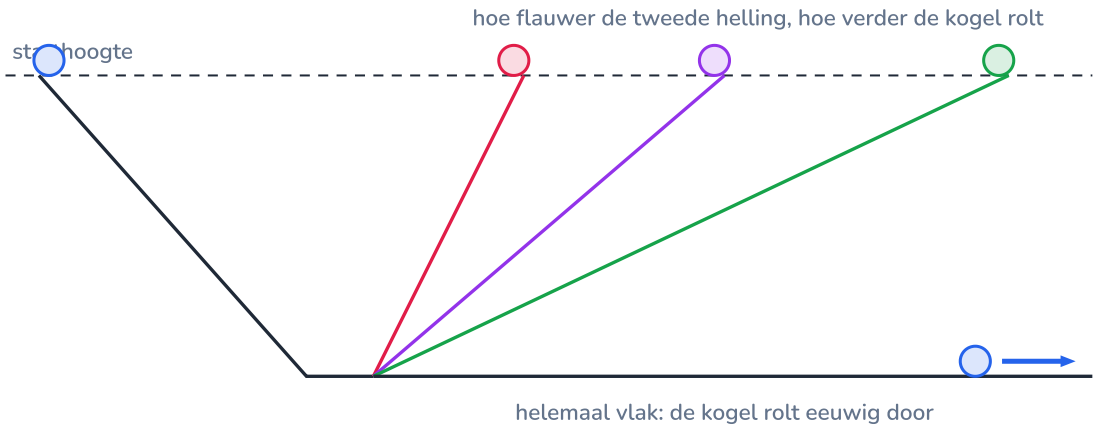
Duw een kar en laat los: ze rolt nog even en stopt. Voor de Griekse filosoof Aristoteles was dat de normale gang van zaken. Een beweging die niet 'natuurlijk' is (zoals vallen), heeft volgens hem een oorzaak nodig die blijft duwen of trekken. Houdt het duwen op, dan houdt de beweging op. Dat idee bleef bijna tweeduizend jaar overeind, omdat het klopt met wat je dagelijks ziet.

Galilei keek anders. Laat een kogel van een helling rollen en een tweede helling oplopen: hij komt (bijna) terug tot op zijn starthoogte, hoe flauw die tweede helling ook is. Hoe flauwer, hoe verder hij rolt om die hoogte te bereiken. Maak de tweede helling helemaal vlak, en de kogel bereikt die hoogte nooit: hij zou eeuwig doorrollen. Dat hij in werkelijkheid toch stopt, komt niet doordat de 'duw' opraakt, maar door wrijving en luchtweerstand: krachten die tegenwerken.

Isaac Newton maakte er in 1687, in zijn boek *Philosophiae Naturalis Principia Mathematica*, de eerste van drie wetten van. Met die drie wetten, en zijn wet van de zwaartekracht, kon hij de banen van de planeten, de getijden en de val van een appel met dezelfde wiskunde beschrijven. Meer dan drie eeuwen later rekenen ingenieurs er nog elke dag mee: de brug, de lift, de auto en het dakpaneel in dit hoofdstuk.

Wat je uit dit verhaal meeneemt

De vraag is niet 'waarom blijft iets bewegen?', maar 'waarom verandert zijn beweging?'. Een voorwerp dat zomaar verder beweegt, heeft geen verklaring nodig. Een voorwerp dat versnelt, remt of afbuigt, wel: daar zit een resulterende kracht achter.



De kogel rolt telkens tot op de starthoogte; op een vlakke baan zou hij, zonder wrijving, nooit stoppen.

UITLEG · 8.4

De eerste wet: traagheid

Elk voorwerp verzet zich tegen een verandering van zijn beweging. Die eigenschap heet **traagheid**. De eerste wet van Newton zegt het precies:

EERSTE WET VAN NEWTON (TRAAGHEIDSWET)

$$\vec{F}_{\text{res}} = \vec{0} \iff \vec{v} \text{ constant (in rust, of ERB)}$$

Een voorwerp waarop de resulterende kracht nul is, blijft in rust of beweegt eenparig rechtlijnig. En omgekeerd: beweegt een voorwerp met constante snelheid langs een rechte, dan is de resulterende kracht erop nul.

Voorbeelden die je nu kan verklaren:

- Bij een noodstop schiet je naar voren: de auto remt, jouw lichaam niet. De **gordel** levert de kracht die jou mee afremt.
- Een ruimtesonde zoals Voyager 1 vliegt al sinds 1980, na zijn laatste bezoek aan een planeet, zonder aandrijving verder: in de lege ruimte remt niets hem af.
- Het schuifpaneel van het kasdak beweegt in de middenfase aan constante snelheid (hoofdstuk 7). Dan is de kracht van de motor precies even groot als de zwaartekracht langs de helling plus de wrijving: samen nul.

Om te blijven bewegen, is een kracht nodig

Een fietser die stopt met trappen, vertraagt. Dat lijkt te tonen dat de trapkracht nodig is om te blijven rijden. Maar de fietser vertraagt door de luchtweerstand en de rolweerstand. Trapt hij aan constante snelheid, dan is zijn trapkracht precies zo groot als die weerstanden samen: de resulterende kracht is **nul**. Een kracht is nodig om een beweging te *veranderen*, niet om ze te *behouden*.

Traagheid is geen kracht

Je 'wordt niet naar voren geduwd' bij een noodstop: er is geen kracht die jou naar voren duwt. Jij gaat gewoon verder aan de snelheid die je had, terwijl de auto onder je afremt. Pas de gordel oefent een kracht op je uit.

De auto stopt, het lichaam wil verder: de gordel levert de kracht die de passagier afremt.

UITLEG · 8.5

De tweede wet: kracht, massa en versnelling

5/21

Is de resulterende kracht niet nul, dan versnelt het voorwerp, in de richting van die kracht. Hoe groot de versnelling is, hangt af van de kracht en van de massa:

TWEEDE WET VAN NEWTON

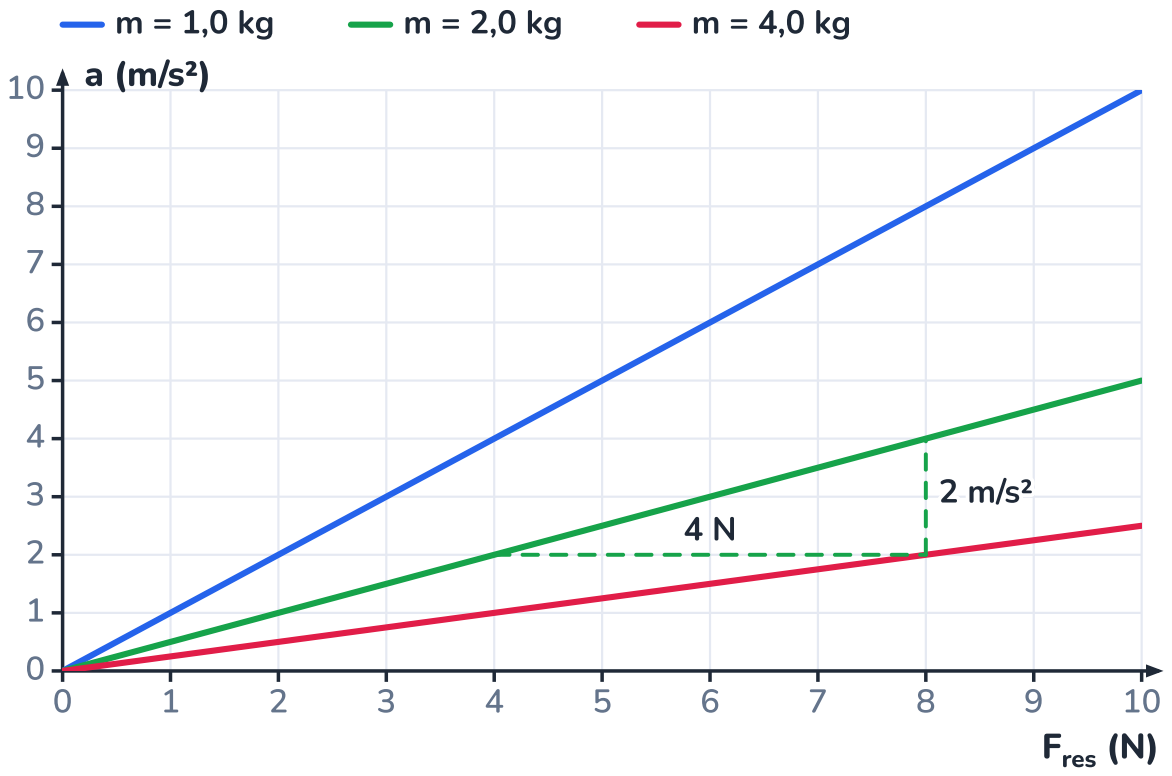
$$\vec{F}_{\text{res}} = m \cdot \vec{a} \qquad \Longleftrightarrow \qquad a = \frac{F_{\text{res}}}{m}$$

F_{res} in N, m in kg, a in m/s². Twee keer zoveel kracht: twee keer zoveel versnelling. Twee keer zoveel massa: half zoveel versnelling.

Die wet legt de eenheid van kracht vast: **één newton** is de kracht die een massa van 1 kg een versnelling van 1 m/s² geeft, $1\,\text{N} = 1\,\text{kg} \cdot \text{m/s}^2$. De massa is hier een maat voor de **traagheid**: hoe moeilijk een voorwerp van snelheid verandert. Een volle batterijkar van 80 kg krijg je moeilijker op gang dan een lege van 20 kg, ook op een perfect gladde vloer.

De zwaartekracht als speciaal geval. Bij een vrije val werkt alleen de zwaartekracht $F_z = m\,g$. De tweede wet geeft dan $a = \dfrac{m\,g}{m} = g$: de massa valt weg, en elk voorwerp valt met dezelfde versnelling. Dat is de verklaring voor de proef van Stevin (hoofdstuk 7), en de reden waarom g tegelijk 9,81 N/kg en 9,81 m/s² is.

In twee richtingen geldt de wet per as: $F_{\text{res},x} = m\,a_x$ en $F_{\text{res},y} = m\,a_y$. Beweegt een voorwerp alleen horizontaal, dan is $a_y = 0$ en moeten de verticale krachten elkaar opheffen.



Rechten door de oorsprong: a is recht evenredig met F. De helling is 1/m: hoe groter de massa, hoe vlakker de rechte.

UITGEWERKT VOORBEELD · 8.6

De batterijkar op een vlakke vloer

6/21

Je duwt de batterijkar van de klimaatmodule (kar en batterijen samen 80 kg) vanuit stilstand over een vlakke vloer met een horizontale kracht van 60 N. De rolweerstand is 2,5 % van de normaalkracht ($\mu_r = 0{,}025$). Hoe groot is de versnelling, en hoe snel rijdt de kar na 2,0 s? Daarna laat je los. Hoe ver rolt de kar nog?

GEGEVEN

- $m = 80\,\text{kg}$, $F_{\text{duw}} = 60\,\text{N}$, $\mu_r = 0{,}025$
- duwen gedurende 2,0 s vanuit rust

GEVRAAGD

- a en v na 2,0 s
- uitrolafstand na het loslaten

OPLOSSING

1. Verticaal: geen versnelling, dus $F_N = F_z = 80 \cdot 9{,}81 = 785\,\text{N}$. Rolweerstand: $F_r = 0{,}025 \cdot 785 = 19{,}6\,\text{N}$.
2. Horizontaal (bewegingszin positief): $F_{\text{res}} = 60 - 19{,}6 = 40{,}4\,\text{N}$, dus $a = \dfrac{40{,}4}{80} = 0{,}50\,\text{m/s}^2$.
3. Na 2,0 s (EVRB vanuit rust): $v = 0{,}505 \cdot 2{,}0 = 1{,}0\,\text{m/s}$.
4. Na het loslaten werkt horizontaal alleen de rolweerstand, tegen de beweging in: $a = -\dfrac{19{,}6}{80} = -0{,}245\,\text{m/s}^2$.
5. Uitrolafstand (EVRB zonder tijd): $0 - 1{,}01^2 = 2 \cdot (-0{,}245) \cdot \Delta x$, dus $\Delta x = 2{,}1\,\text{m}$, in $1{,}01/0{,}245 = 4{,}1\,\text{s}$.

ANTWOORD

De kar versnelt met 0,50 m/s² en rijdt na 2,0 s aan 1,0 m/s. Losgelaten rolt ze nog 2,1 m ver en staat na ongeveer 4 s stil.

Klopt dit? Dat de kar stopt, is geen teken dat 'de duwkracht op is': de rolweerstand remt ze af. Op een vloer zonder rolweerstand zou ze na het loslaten eeuwig aan 1,0 m/s verder rollen (eerste wet). En de grootte-orde: 20 N om een volle kar aan de gang te houden, dat voelt een mens als een lichte duw.

UITLEG · 8.7

De derde wet: actie en reactie

Krachten komen altijd in paren. Duw je tegen een muur, dan duwt de muur even hard terug op jou. Trekt de aarde aan een boek, dan trekt het boek even hard aan de aarde.

DERDE WET VAN NEWTON (ACTIE EN REACTIE)

$$\vec{F}_{A\to B} = -\vec{F}_{B\to A}$$

Oefent voorwerp A een kracht uit op voorwerp B, dan oefent B een even grote, tegengestelde kracht uit op A. De twee krachten zijn van dezelfde soort en werken op **verschillende** voorwerpen.

Zonder derde wet kan je niet eens stappen: je voet duwt de grond naar achter, en de grond duwt jou naar voor. Een roeier duwt water naar achter, het water duwt de boot naar voor. Een raket stoot gas naar achter uit, het gas duwt de raket vooruit, ook in de lege ruimte. En de riem die het dakpaneel omhoog trekt, trekt met evenveel kracht aan de motor: de motorsteun moet die kracht opvangen.

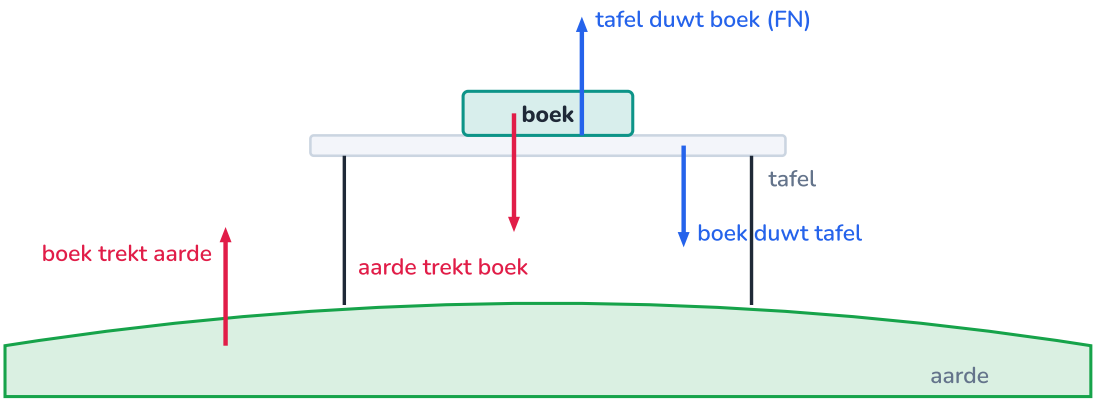
Actie en reactie heffen elkaar op

Als ze even groot en tegengesteld zijn, waarom beweegt er dan ooit iets? Omdat ze op **verschillende voorwerpen** werken. Je telt ze nooit samen op in één vrijlichaamsdiagram. Een paard dat een kar trekt: de kar trekt even hard terug aan het paard, maar voor de beweging van de kar telt alleen wat er op de kar werkt (de trekkracht en de wrijving).

Zwaartekracht en normaalkracht zijn geen paar

Op een boek op een tafel werken twee even grote, tegengestelde krachten: de zwaartekracht en de normaalkracht. Toch zijn ze **geen** actie-reactiepaar: ze werken allebei op het boek, ze zijn van een andere soort, en ze zijn alleen even groot omdat het boek niet versnelt (eerste wet). In een lift die optrekt, is de normaalkracht groter dan de zwaartekracht. De echte partners staan in de figuur.

rood: paar 1 (zwaartekracht) blauw: paar 2 (contactkracht)
de twee krachten van een paar werken op verschillende voorwerpen



Paar 1: de aarde trekt aan het boek en het boek aan de aarde. Paar 2: de tafel duwt op het boek en het boek op de tafel.

UITGEWERKT VOORBEELD · 8.8

Twee schaatsers duwen elkaar weg

8/21

Twee schaatsers staan stil op het ijs en duwen elkaar een halve seconde lang weg. Anna (60 kg) duwt met 120 N op Bram (40 kg). Welke kracht werkt er op Anna? Met welke snelheid glijden ze uit elkaar? De wrijving van de schaatsen verwaarloos je.

GEGEVEN

- $m_A = 60\,\text{kg}$, $m_B = 40\,\text{kg}$
- $F_{A\to B} = 120\,\text{N}$ gedurende $0{,}50\,\text{s}$

GEVRAAGD

- $F_{B\to A}$
- v_A en v_B na het duwen

OPLOSSING

1. Derde wet: Bram duwt even hard terug, $F_{B\to A} = 120\,\text{N}$, tegengesteld gericht. Wie 'eigenlijk' duwt, maakt niet uit.
2. Tweede wet voor elk apart (horizontaal is die kracht de enige): $a_A = \dfrac{120}{60} = 2{,}0\,\text{m/s}^2$ en $a_B = \dfrac{120}{40} = 3{,}0\,\text{m/s}^2$.
3. Na 0,50 s vanuit rust: $v_A = 2{,}0 \cdot 0{,}50 = 1{,}0\,\text{m/s}$ en $v_B = 3{,}0 \cdot 0{,}50 = 1{,}5\,\text{m/s}$, in tegengestelde zin.

ANTWOORD

Op Anna werkt ook 120 N, naar de andere kant. Anna glijdt weg met 1,0 m/s, Bram met 1,5 m/s in de tegengestelde zin.

Klopt dit? De krachten zijn gelijk, de versnellingen niet: de lichtste schaatser krijgt de grootste versnelling. Merk op dat $60 \cdot 1{,}0 = 40 \cdot 1{,}5 = 60\,\text{kg}\cdot\text{m/s}$: het product van massa en snelheid is voor beiden even groot. Dat is geen toeval; het heet *impuls* en komt terug in hoofdstuk E9.

UITLEG · 8.9

Het vrijlichaamsdiagram

9/21

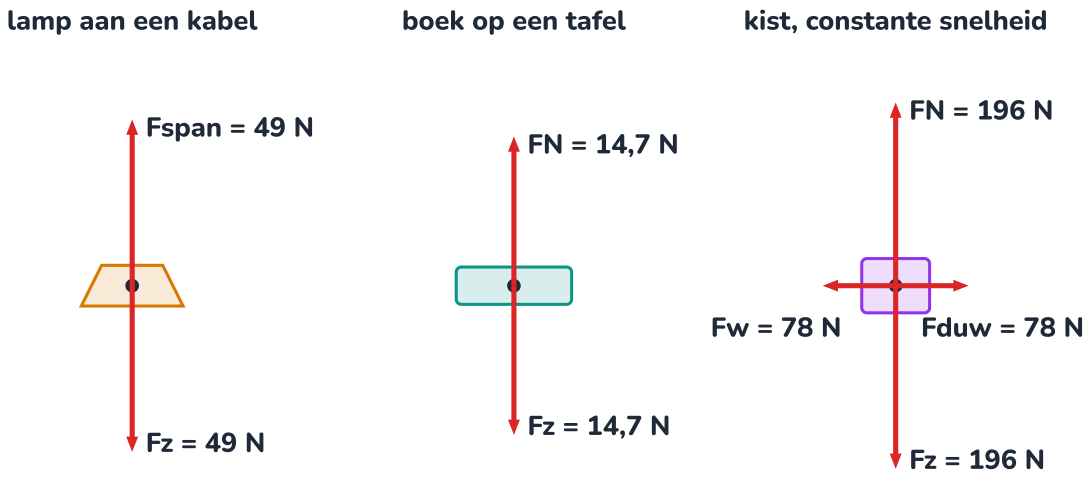
Een **vrijlichaamsdiagram** (VLD) is de tekening waarmee elke krachtenberekening begint. Je knipt één voorwerp los uit zijn omgeving en tekent alle krachten die de omgeving *erop* uitoefent. Werk altijd in dezelfde stappen:

1. **Kies het voorwerp.** Eén voorwerp (of een groep die samen beweegt). Teken het los, als een blokje of een punt.
2. **Zwaartekracht**: altijd, verticaal naar beneden, $F_z = m\,g$.
3. **Contactkrachten**: overal waar het voorwerp iets raakt. Een oppervlak geeft een **normaalkracht** F_N loodrecht op dat oppervlak, en vaak een **wrijvingskracht** F_w langs het oppervlak. Een touw, kabel of riem geeft een **spankracht** langs het touw, altijd trekkend. Een hand of een motor geeft een duw- of trekkracht.
4. **Kies assen**: liefst één as langs de (mogelijke) beweging.
5. **Controleer**: bij elke kracht moet je kunnen zeggen *welk ander voorwerp* ze uitoefent. Krachten die het voorwerp zelf uitoefent (de reactiekrachten) horen er niet in.

In dit hoofdstuk teken je alle krachten vanuit het zwaartepunt: het voorwerp is een punt. Waar een kracht precies aangrijpt, telt pas als het voorwerp kan kantelen of draaien (hoofdstuk 10).

De 'kracht van de beweging'

Een bal die omhoog vliegt, heeft na het loslaten geen 'werpkracht' meer die hem omhoog duwt: alleen de zwaartekracht (en wat luchtweerstand) werkt erop. Hij gaat omhoog door zijn traagheid, en de zwaartekracht remt hem af. Vraag bij elke pijl: wie oefent die kracht uit? Is het antwoord 'niemand', dan hoort de pijl er niet.



elke figuur op haar eigen schaal; alle krachten vanuit het zwaartepunt

Drie vrijlichaamsdiagrammen in evenwicht: in elke richting heffen de krachten elkaar op.

PROEF · 8.10

Een smartphone in de lift

10/21

ONDERZOEKSVRAAG

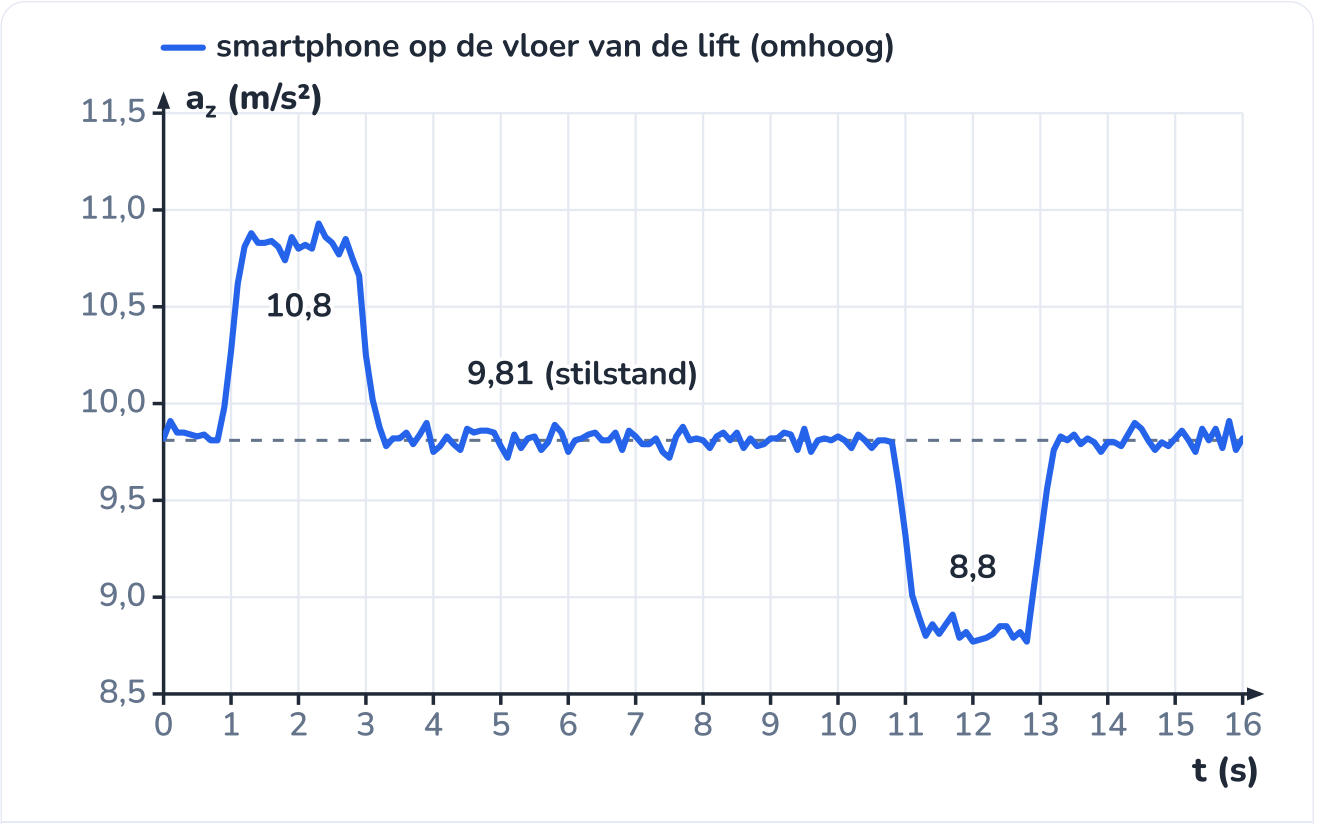
Wat meet de versnellingsmeter van een smartphone als de lift vertrekt, rijdt en stopt? En wat leer je daaruit over de normaalkracht?

MATERIAAL

- smartphone met een app die de versnellingsmeter uitleest (bijvoorbeeld phyphox, van de RWTH Aachen), stand 'versnelling met g'
- een lift van minstens vijf verdiepingen
- optioneel: een personenweegschaal

Veiligheid

Houd de liftdeuren vrij en laat andere gebruikers voorgaan. Meet niet in een drukke lift.



Voorbeeldmeting: 2 s lang 1,0 m/s² extra bij het vertrekken, 8 s constante snelheid, 2 s afremmen.

WERKWIJZE

1. Leg de smartphone plat op de vloer van de lift, met het scherm naar boven.
2. Start de meting, wacht enkele seconden, en rijd van het gelijkvloers naar een hoge verdieping.
3. Wacht tot de lift een paar seconden stilstaat en stop de meting. Bekijk de verticale component a_z.
4. Herhaal naar beneden. Sta je op een weegschaal, lees dan ook af wat die toont in elke fase.

WAARNEMING

In stilstand toont de telefoon 9,81 m/s² (niet nul!). Bij het vertrekken naar boven stijgt de waarde tot ongeveer 10,8 m/s², tijdens de rit is ze weer 9,81 m/s², en bij het afremmen zakt ze tot ongeveer 8,8 m/s². Een weegschaal toont bij het vertrekken meer dan je massa, bij het stoppen minder.

VERKLARING

Een versnellingsmeter meet eigenlijk de **contactkracht per kilogram**: de kracht waarmee de vloer op de telefoon duwt, gedeeld door de massa van de telefoon. Op de telefoon werken de zwaartekracht $m\,g$ en de normaalkracht F_N. De tweede wet (omhoog positief): $F_N - m\,g = m\,a$, dus

$$\frac{F_N}{m} = g + a$$

In stilstand of aan constante snelheid is $a = 0$ en meet je g. Tijdens het vertrekken meet je $9{,}81 + 1{,}0$: de lift versnelt met 1,0 m/s² omhoog. Tijdens het afremmen $9{,}81 - 1{,}0$. De oppervlakte onder a (de grafiek min 9,81) is de snelheidsverandering: $1{,}0 \cdot 2{,}0 = 2{,}0\,\mathrm{m/s}$, de rijsnelheid van de lift.

Een weegschaal meet ook een normaalkracht, en rekent ze om naar kilogram door te delen door 9,81. In een versnellende lift toont ze daardoor je **schijnbaar gewicht**, niet je massa.

UITGEWERKT VOORBEELD · 8.11

Wat toont de weegschaal in de lift?

11/21

Een leerling van 60 kg staat in de lift van de proef op een personenweegschaal. Wat toont de weegschaal tijdens het vertrekken (1,0 m/s² omhoog), tijdens de rit, en tijdens het afremmen? Wat zou ze tonen als de kabel breekt?

GEGEVEN

- $m = 60\,\text{kg}$
- $a = +1{,}0,\ 0$ en $-1{,}0\,\text{m/s}^2$ (omhoog positief)

GEVRAAGD

- F_N in elke fase en de aflezing in kg

OPLOSSING

1. VLD van de leerling: zwaartekracht $m\,g = 589\,\text{N}$ omlaag, normaalkracht F_N van de weegschaal omhoog. Tweede wet: $F_N - m\,g = m\,a$, dus $F_N = m\,(g + a)$.
2. Vertrekken: $F_N = 60 \cdot (9{,}81 + 1{,}0) = 649\,\text{N}$. De weegschaal deelt door 9,81: ze toont 66 kg.
3. Rit aan constante snelheid: $F_N = 589\,\text{N}$, de weegschaal toont 60 kg.
4. Afremmen: $F_N = 60 \cdot (9{,}81 - 1{,}0) = 529\,\text{N}$: de weegschaal toont 54 kg.
5. Vrije val ($a = -9{,}81\,\text{m/s}^2$): $F_N = 60 \cdot 0 = 0$. De weegschaal toont 0 kg: de leerling is *schijnbaar gewichtloos*. De zwaartekracht werkt nog volledig, maar niets hoeft hem nog te dragen.

ANTWOORD

66 kg bij het vertrekken, 60 kg tijdens de rit, 54 kg bij het afremmen, en 0 kg in vrije val.

Klopt dit? De massa van de leerling verandert niet; de kracht waarmee de vloer hem moet duwen wel. Zo voelen astronauten in een baan rond de aarde zich gewichtloos: ze vallen samen met hun ruimtestation voortdurend rond de aarde. Een liftkabel moet om dezelfde reden méér dan het gewicht kunnen dragen: een lift van 600 kg die met 1,0 m/s² vertrekt, hangt aan $600 \cdot 10{,}81 = 6486\,\text{N}$.

UITLEG · 8.12

Wrijving en de wrijvingscoëfficiënt

12/21

Duw zacht tegen een zware kist: ze beweegt niet. De vloer oefent een **statische wrijvingskracht** uit die precies even groot is als jouw duw, tegengesteld. Duw harder: de wrijving groeit mee. Maar niet onbeperkt: bij een bepaalde duwkracht schiet de kist los. Daarna glijdt ze, en werkt er een **dynamische (glijdende) wrijvingskracht**, die meestal iets kleiner is.

STATISCHE EN DYNAMISCHE WRIJVING

$$F_{w,s} \leq \mu_s \cdot F_N \qquad F_{w,k} = \mu_k \cdot F_N$$

μ_s en μ_k zijn de statische en de dynamische (kinetische) **wrijvingscoëfficiënt**, getallen zonder eenheid. F_N is de normaalkracht, niet altijd het gewicht. De wrijving werkt langs het oppervlak, tegen het (dreigende) glijden in.

Twee dingen die je misschien niet verwacht: de wrijving hangt bij benadering **niet** af van de grootte van het contactoppervlak (een baksteen op zijn smalle of zijn brede kant geeft dezelfde wrijving), en bij gewone snelheden nauwelijks van de snelheid. Wat wel telt: hoe hard de oppervlakken tegen elkaar gedrukt worden (F_N) en welke materialen het zijn (μ).

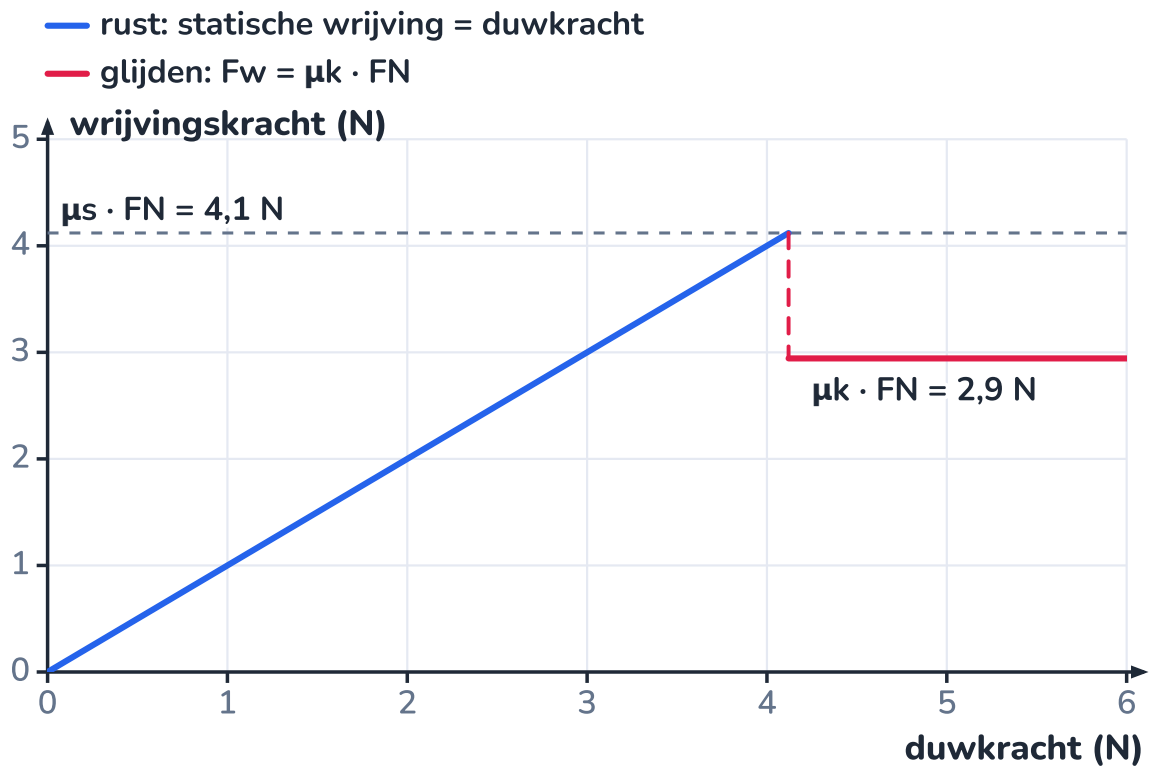
materialen	μ_s	μ_k
autoband op droog asfalt	0,8 à 1,0	ongeveer 0,7
autoband op nat asfalt	0,5 à 0,7	0,4 à 0,5
hout op hout	0,3 à 0,5	0,2 à 0,4

materialen	μ_s	μ_k
kunststof glijder (PE) op aluminium (rekenwaarde dakpaneel)	0,25	0,20
staal op ijs (schaats)	ongeveer 0,03	ongeveer 0,01
teflon op staal	ongeveer 0,05	ongeveer 0,05

Ordes van grootte: de echte waarde hangt af van de ruwheid, vuil, vocht en temperatuur. Wie een precieze waarde nodig heeft, meet ze (proef verderop). **Rolweerstand** werkt op dezelfde manier, $F_r = \mu_r \cdot F_N$, maar met veel kleinere coëfficiënten: fietsband op asfalt ongeveer 0,005, autoband ongeveer 0,01, de kar met rubberen wielen van de klimaatmodule 0,025.

Wrijving is niet altijd de vijand

Zonder wrijving kan je niet stappen, remmen of een bocht nemen, blijft een schroef niet vast en glijdt een riem over zijn poelie. Ingenieurs maken wrijving soms groot (remblokken, banden, antislip op een oprit) en soms klein (lagers, smering, teflon), afhankelijk van wat nodig is.



Tot 4,1 N past de statische wrijving zich aan de duw aan; eens het blok glijdt, is de wrijving constant 2,9 N.

PROEF · 8.13

Een wrijvingscoëfficiënt meten met een dynamometer

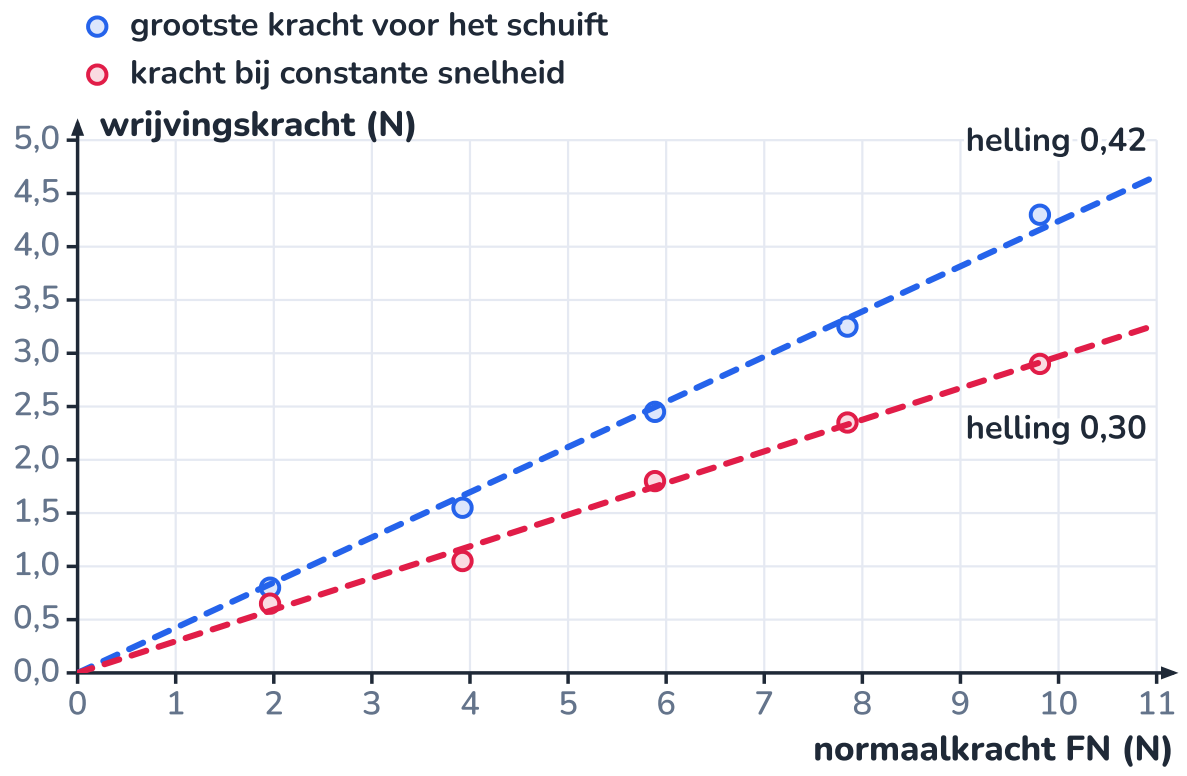
13/21

ONDERZOEKSVRAAG

Hoe hangt de wrijvingskracht op een houten blok af van de normaalkracht, en hoe groot zijn μ_s en μ_k voor hout op een tafelblad?

MATERIAAL

houten blok van 200 g met een haakje · vier massa's van 200 g · dynamometer van 5 N · een vlak, schoon tafelblad



Twee rechten door de oorsprong: de hellingen zijn μs = 0,42 en μk = 0,30.

WERKWIJZE

1. Haak de dynamometer aan het blok en trek horizontaal, eerst heel zacht, dan geleidelijk harder. Lees de grootste waarde af net voor het blok schuift: dat is de maximale statische wrijving.
2. Trek het blok daarna met een zo constant mogelijke, lage snelheid over de tafel en lees de kracht af: dat is de dynamische wrijving (eerste wet: de trekkracht is dan gelijk aan de wrijving).
3. Herhaal met 1, 2, 3 en 4 massa's op het blok. Herhaal elke meting drie keer en neem het gemiddelde.
4. Zet beide krachten uit tegen de normaalkracht $F_N = m\,g$ (op een vlakke tafel).
5. Leg het blok daarna op een smallere kant en meet opnieuw met dezelfde massa.

WAARNEMING

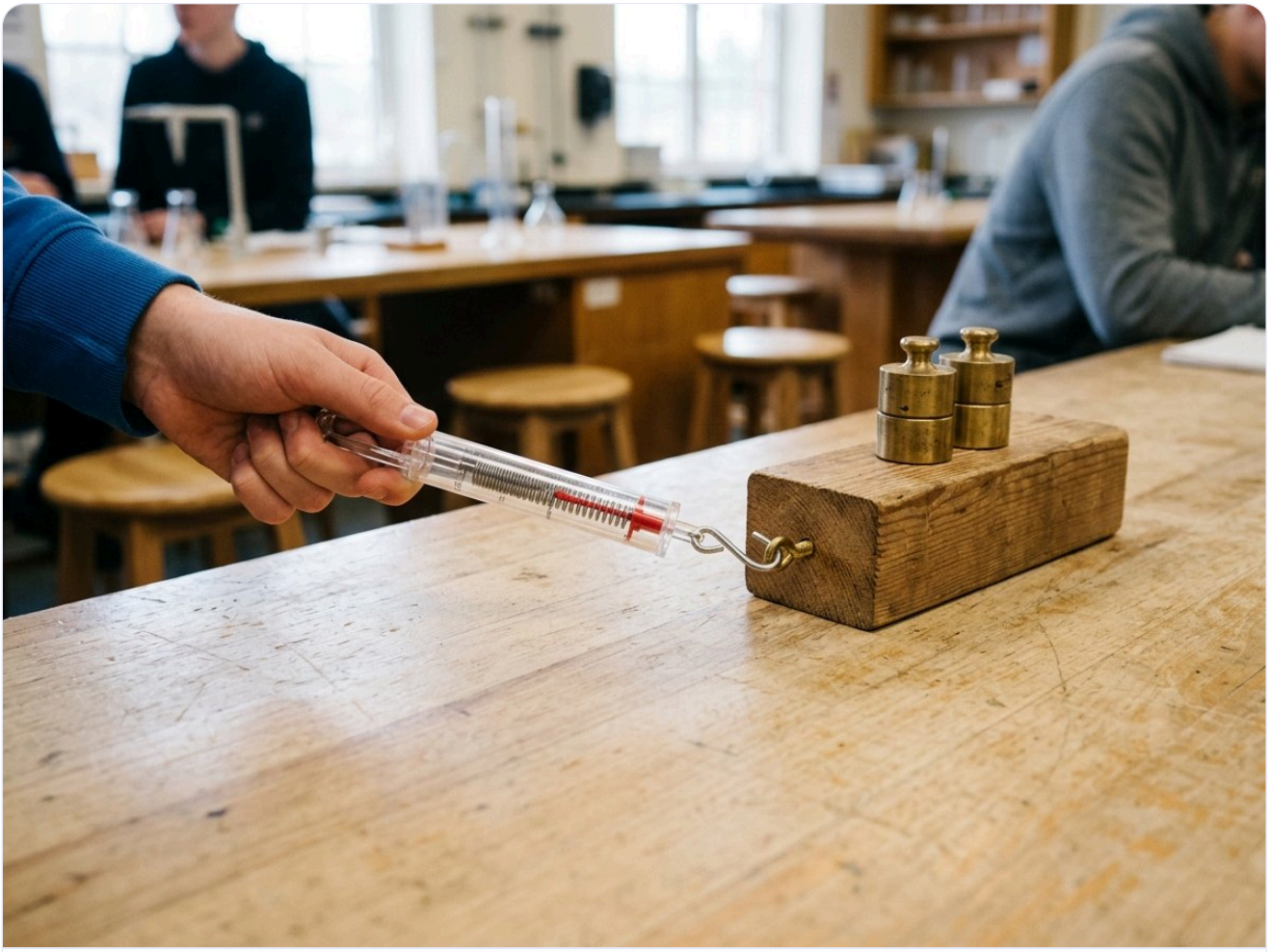
massa blok + gewichten (kg)	F_N (N)	grootste kracht voor het schuift (N)	kracht bij constante snelheid (N)
0,200	1,96	0,80	
0,400	3,92	1,55	
0,600	5,89	2,45	
0,800	7,85	3,25	
1,000	9,81	4,30	

Beide reeksen liggen op een rechte door de oorsprong. Op de smalle kant geeft het blok met 400 g 1,10 N bij constante snelheid, binnen de meetfout gelijk aan de 1,05 N op de brede kant.

VERKLARING

De wrijvingskracht is evenredig met de normaalkracht, en de evenredigheidsconstante is de wrijvingscoëfficiënt: de helling van de rechte. De best passende rechten door de oorsprong geven $\mu_s = 0{,}42$ en $\mu_k = 0{,}30$. De statische coëfficiënt is groter: een blok losrukken is moeilijker dan het in beweging houden. Het contactoppervlak maakt binnen de meetfout geen verschil.

Meetproblemen: bij het trekken aan 'constante snelheid' schokt het blok vaak een beetje (het hapert en schiet los), zodat de naald schommelt; neem dan het midden. Trek je schuin in plaats van horizontaal, dan verandert de normaalkracht (voorbeeld hierna).



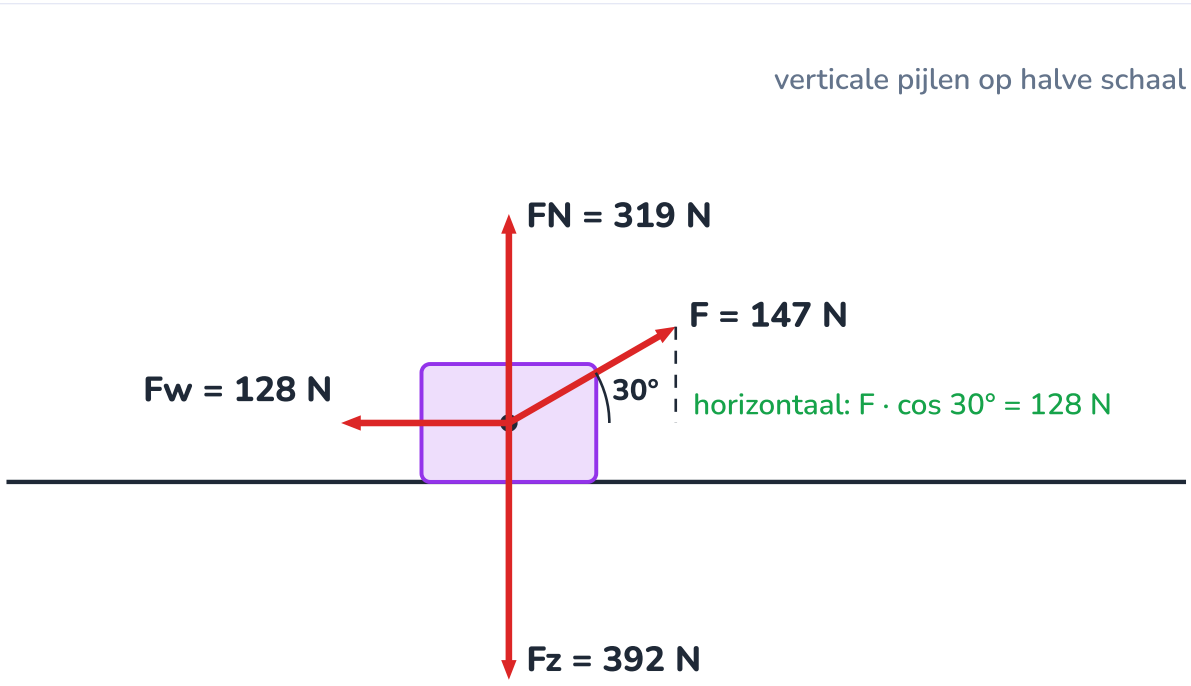
Trekken aan constante snelheid: dan is de kracht op de dynamometer gelijk aan de wrijving.

UITGEWERKT VOORBEELD · 8.14

Een kist slepen: horizontaal of schuin trekken?

14/21

Een kist van 40 kg staat op een betonvloer ($\mu_k = 0{,}40$). Met welke kracht moet je aan een touw trekken om de kist aan constante snelheid te verslepen: als het touw horizontaal loopt, en als het touw een hoek van 30° met de vloer maakt?



Het touw trekt schuin omhoog: minder normaalkracht, dus minder wrijving.

GEGEVEN

- $m = 40\,\text{kg}$, $\mu_k = 0{,}40$
- touw horizontaal of onder $\theta = 30^\circ$
- constante snelheid

GEVRAAGD

- trekkracht F in beide gevallen

OPLOSSING

1. Constante snelheid: de resulterende kracht is nul, in beide richtingen (eerste wet).
2. Horizontaal: $F_N = m\,g = 392\,\text{N}$ en $F = F_w = 0{,}40 \cdot 392 = 157\,\text{N}$.
3. Onder 30°: ontbind de trekkracht. Horizontaal $F\cos 30^\circ$, verticaal $F\sin 30^\circ$ omhoog. Die verticale component tilt de kist een beetje op, dus de vloer hoeft minder te duwen: $F_N = m\,g - F\sin 30^\circ$.
4. Horizontaal evenwicht: $F\cos 30^\circ = \mu_k\,(m\,g - F\sin 30^\circ)$. Breng alle termen met F samen: $F\,(\cos 30^\circ + \mu_k \sin 30^\circ) = \mu_k\,m\,g$, dus $F = \dfrac{0{,}40 \cdot 392}{0{,}866 + 0{,}20} = \dfrac{157}{1{,}066} = 147\,\text{N}$.
5. Controle: $F_N = 392 - 147 \cdot 0{,}5 = 319\,\text{N}$, $F_w = 0{,}40 \cdot 319 = 128\,\text{N}$ en $F\cos 30^\circ = 147 \cdot 0{,}866 = 128\,\text{N}$. Evenwicht.

ANTWOORD

Horizontaal trekken vraagt 157 N, schuin onder 30° maar 147 N: de verticale component van het touw maakt de normaalkracht, en dus de wrijving, kleiner.

Klopt dit? Een slee of een koffer op wieltjes trek je daarom schuin. Te schuin werkt weer averechts: dan gaat een groot deel van je kracht naar het optillen in plaats van naar het vooruittrekken (opdracht 8.10).

UITLEG · 8.15

Een voorwerp op een helling

15/21

In hoofdstuk 6 ontbond je de zwaartekracht op een helling in $F_{\parallel} = m\,g\sin\alpha$ langs en $F_{\perp} = m\,g\cos\alpha$ loodrecht op de helling. Nu komen de andere krachten erbij. Kies de assen langs en loodrecht op de helling:

- **Loodrecht**: het voorwerp gaat niet door de helling en komt er niet van los, dus $a_{\perp} = 0$ en $F_N = m\,g\cos\alpha$. Op een helling is de normaalkracht dus *kleiner* dan het gewicht.
- **Langs de helling**: $F_{\parallel}$, de wrijving (tegen het glijden in) en eventueel een duw- of trekkracht. De som is $m\,a$.

BLOK OP EEN HELLING, ZONDER ANDERE KRACHTEN

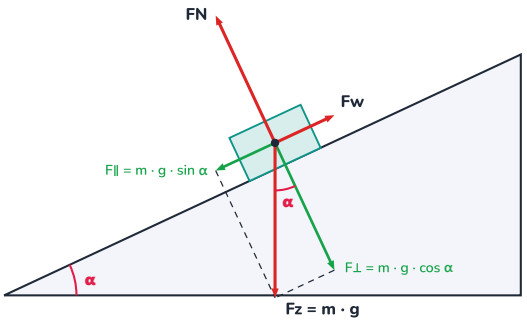
$$\text{blijft liggen als } \tan\alpha \le \mu_s \qquad a = g\,(\sin\alpha - \mu_k\cos\alpha) \text{ bij glijden}$$

De **grenshoek** volgt uit $m\,g\sin\alpha = \mu_s\,m\,g\cos\alpha$: de massa valt weg. Een lege en een volle kist glijden bij dezelfde hoek.

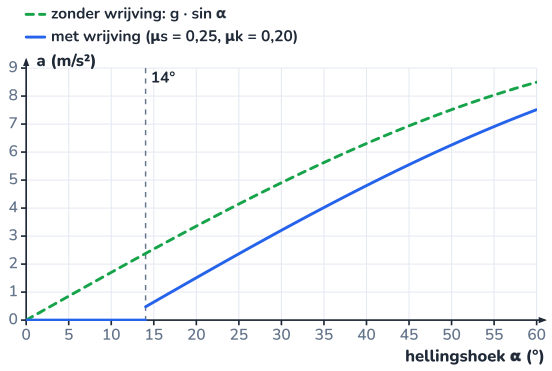
Wil je een voorwerp een helling **op** trekken, dan werken $F_{\parallel}$ en de wrijving allebei tegen: $F_{\text{trek}} = m\,g\sin\alpha + \mu_k\,m\,g\cos\alpha + m\,a$. Laat je het **afdalen** aan constante snelheid, dan helpt de wrijving je: $F_{\text{rem}} = m\,g\sin\alpha - \mu_k\,m\,g\cos\alpha$.

De normaalkracht is altijd gelijk aan het gewicht

Alleen op een vlakke ondergrond, zonder andere verticale krachten en zonder verticale versnelling. Op een helling is ze $m\,g\cos\alpha$; bij schuin trekken kleiner; in een lift die optrekt groter. Bereken F_N altijd met de tweede wet in de richting loodrecht op het oppervlak.



Blok in rust: FN heft F⊥ op, de statische wrijving Fw heft F∥ op.



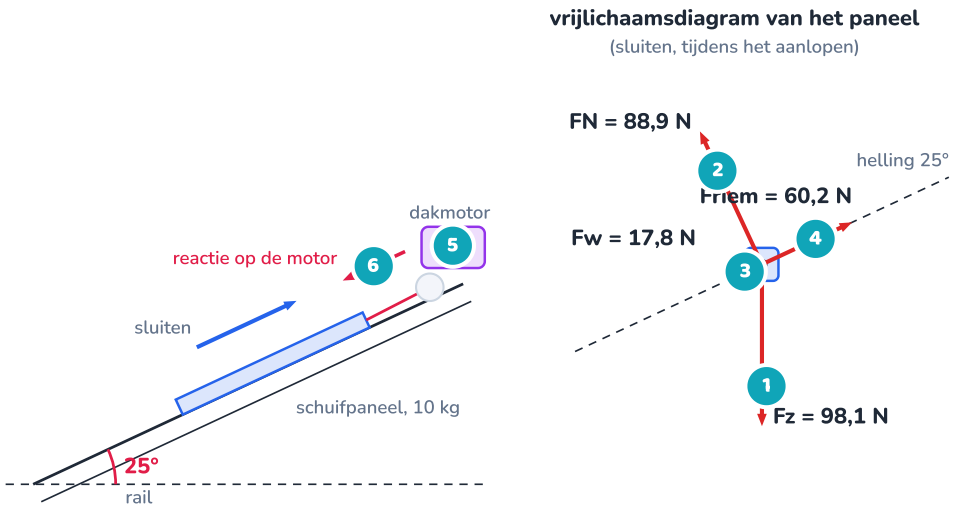
Onder 14° blijft het blok liggen (tan 14° = 0,25 = μs); daarboven glijdt het, altijd trager dan zonder wrijving.

ONTLEED · 8.16

Habitat: de krachten op het schuifpaneel

16/21

Het schuifpaneel van het kasdak (hoofdstuk 7) weegt 10 kg en glijdt met kunststof glijders over aluminium rails onder 25°. Bij het **openen** schuift het naar beneden, over het vaste deel van het dak, zodat bij de nok een opening ontstaat: daar ontsnapt de warme lucht het best. Bij het **sluiten** trekt de dakmotor het met een getande riem terug omhoog. Rekenwaarden voor de glijders: $\mu_s = 0{,}25$ en $\mu_k = 0{,}20$.



Links het dak in zijaanzicht, rechts het vrijlichaamsdiagram van het paneel bij het sluiten, op schaal (1,2 px per N).

1 Zwaartekracht
$F_z = 10 \cdot 9{,}81 = 98{,}1\ \text{N}$, verticaal. Langs de helling: $F_{\parallel} = 98{,}1 \sin 25° = 41{,}5\ \text{N}$ naar beneden; loodrecht: $98{,}1 \cos 25° = 88{,}9\ \text{N}$ tegen de rails.

2 Normaalkracht
De rails duwen loodrecht op het paneel: 88,9 N, want loodrecht op de rails versnelt het paneel niet. Kleiner dan het gewicht.

3 Wrijving
Tegen de beweging in: bij het sluiten naar beneden, bij het openen naar boven. $F_w = 0{,}20 \cdot 88{,}9 = 17{,}8\ \text{N}$.

4 Kracht van de riem
Een spankracht langs de riem, dus langs de helling omhoog. Hoe groot ze moet zijn, hangt af van de fase: zie het volgende voorbeeld.

5 Dakmotor met wormoverbrenging
Staat de motor stil, dan moet iets het paneel tegenhouden: tan 25° = 0,47 is groter dan μs = 0,25, dus los zou het paneel naar beneden glijden. Een wormoverbrenging is zelfremmend: de riem kan de motor niet terugdraaien.

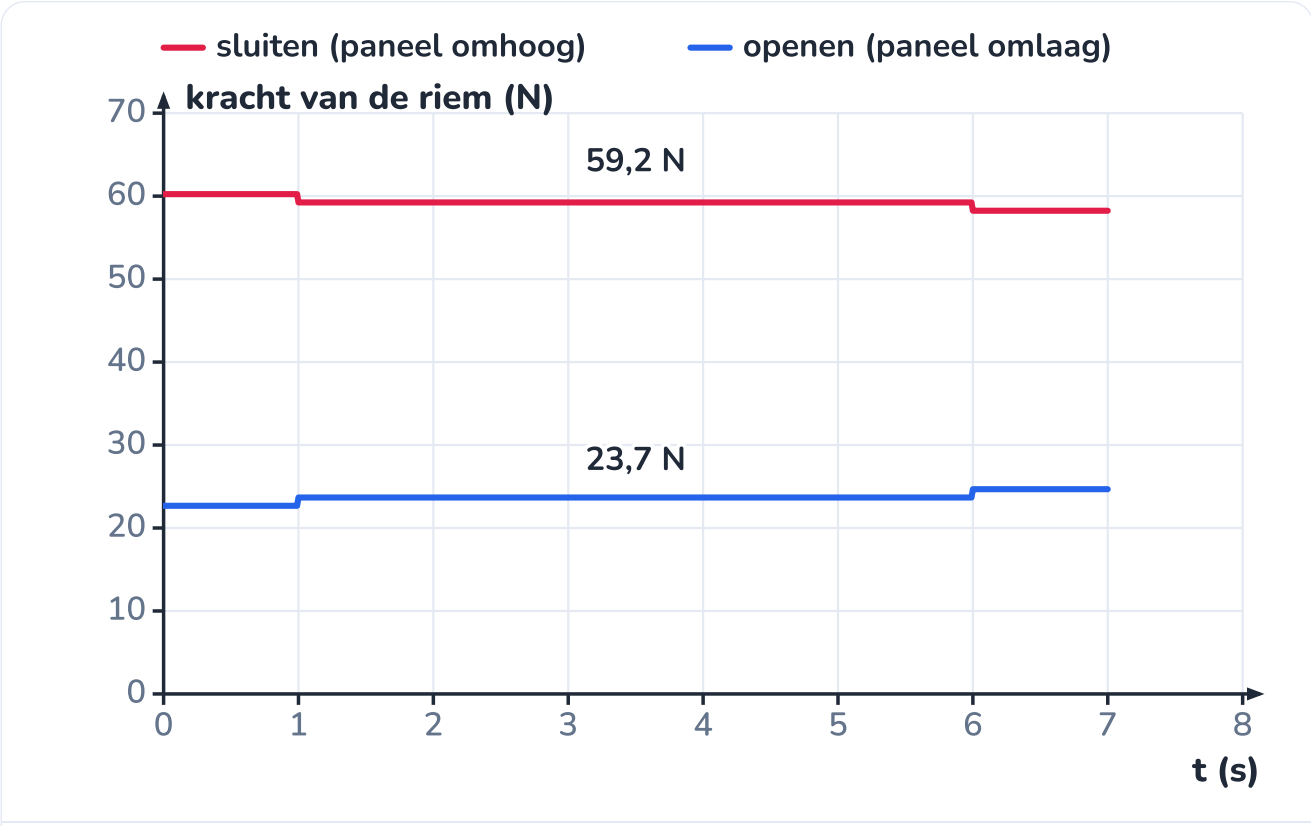
6 Reactie op de motor
Derde wet: trekt de riem met 60 N aan het paneel, dan trekt het paneel via de riem met 60 N aan de motor. De motorsteun en zijn bouten moeten die kracht opvangen, ook bij een schok.

UITGEWERKT VOORBEELD · 8.17

Habitat: hoeveel kracht moet de dakmotor leveren?

17/21

Bereken de kracht die de riem op het paneel moet uitoefenen in de zes fasen van hoofdstuk 7 (aanlopen, constante snelheid en afremmen, bij sluiten en bij openen). Het paneel versnelt en vertraagt met 0,10 m/s². Welke kracht is maatgevend voor de keuze van de motor?



De riemkracht in de drie fasen: bij het sluiten rond 59 N, bij het openen rond 24 N.

GEGEVEN

- $m = 10\,\text{kg}$, $\alpha = 25°$, $\mu_k = 0{,}20$
- $a = 0{,}10\,\text{m/s}^2$ bij aanlopen en afremmen

GEVRAAGD

- F_{riem} in de zes fasen
- de maatgevende kracht

OPLOSSING

1. Vaste krachten (zie het ontleedblad): $F_{\parallel} = 41{,}5\,\text{N}$, $F_N = 88{,}9\,\text{N}$, $F_w = 17{,}8\,\text{N}$.
2. **Sluiten** (omhoog positief; wrijving omlaag): $F_{\text{riem}} - F_{\parallel} - F_w = m\,a$, dus $F_{\text{riem}} = 41{,}5 + 17{,}8 + 10\,a$. Aanlopen ($a = +0{,}10$): 60,2 N. Constant: 59,2 N. Afremmen ($a = -0{,}10$): 58,2 N.
3. **Openen** (omlaag positief; wrijving omhoog; de riem houdt tegen, omhoog): $F_{\parallel} - F_w - F_{\text{riem}} = m\,a$, dus $F_{\text{riem}} = 41{,}5 - 17{,}8 - 10\,a$. Aanlopen: 22,7 N. Constant: 23,7 N. Afremmen: 24,7 N.
4. Maatgevend is het aanlopen bij het sluiten: 60,2 N. Het versnellen zelf ($m\,a = 1{,}0\,\text{N}$) weegt nauwelijks door; de helling en de wrijving bepalen alles.

ANTWOORD

De riem moet tot 60 N kunnen trekken (sluiten) en tot 25 N kunnen tegenhouden (openen). Met een veiligheidsfactor 2 voor vuil, ijs en slijtage kies je een aandrijving voor minstens 120 N.

Klopt dit? Het zachte aanlopen uit hoofdstuk 7 kost hier maar 1 N extra. Zonder aanlopen (van 0 naar 0,10 m/s in 0,01 s) zou $m\,a = 10 \cdot 10 = 100\,\text{N}$ extra zijn, een schok van meer dan het dubbele. Hoeveel vermogen de motor daarvoor nodig heeft, reken je uit in hoofdstuk 9; welke motor je kiest en hoe je hem aanstuurt, in hoofdstuk 26.

UITGEWERKT VOORBEELD · 8.18

Habitat: de batterijkar op de oprit

In hoofdstuk 6 vond je dat je minstens 49 N moet duwen om de batterijkar (80 kg) over de oprit van 6,25 % te krijgen, zonder wrijving. Reken nu met rolweerstand ($\mu_r = 0{,}025$). Hoeveel kracht vraagt het aan constante snelheid? En als je de kar vanuit stilstand op de oprit in 2,0 s op 1,0 m/s wil brengen? Wat gebeurt er als je halverwege loslaat?

Een oprit van 6,25 % lijkt vlak, maar de helling trekt de kar met 49 N terug naar beneden.

GEGEVEN

- $m = 80\,\text{kg}$, $\alpha = \tan^{-1}(0{,}0625) = 3{,}58°$
- $\mu_r = 0{,}025$

GEVRAAGD

- F bij constante snelheid en bij $a = 0{,}50\,\text{m/s}^2$
- gedrag na loslaten

OPLOSSING

1. $F_{\parallel} = 785 \sin 3{,}58° = 49{,}0\,\text{N}$; $F_N = 785 \cos 3{,}58° = 783\,\text{N}$; rolweerstand $F_r = 0{,}025 \cdot 783 = 19{,}6\,\text{N}$.
2. Constante snelheid omhoog: $F = 49{,}0 + 19{,}6 = 68{,}6\,\text{N}$ (hoofdstuk 6 schatte 'ongeveer 70 N').
3. Optrekken: $a = 1{,}0/2{,}0 = 0{,}50\,\text{m/s}^2$, dus $F = 68{,}6 + 80 \cdot 0{,}50 = 109\,\text{N}$.
4. Losgelaten: omhoog werkt nu niets meer. Rijdt de kar nog omhoog, dan remmen $F_{\parallel}$ en de rolweerstand allebei: $a = -\dfrac{49{,}0 + 19{,}6}{80} = -0{,}86\,\text{m/s}^2$, en na ruim een seconde staat ze stil.

5 Daarna keert de rolweerstand om (ze werkt altijd tegen de beweging), maar $F_{\parallel}$ wint ruim: de kar rolt terug, met $a = \frac{49{,}0 - 19{,}6}{80} = 0{,}37\,\mathrm{m/s^2}$ naar beneden.

ANTWOORD

69 N aan constante snelheid, 109 N om op 2,0 s op gang te komen. Losgelaten rolt de kar terug naar beneden, steeds sneller.

Klopt dit? Wie de kar de hele oprit laat terugrollen, heeft beneden een kar van 80 kg aan bijna 8 km/h (opdracht 8.14): een reden om een kar met een rem of een 'dodemansgreep' te kiezen, of de oprit met een vlak bordes te onderbreken.

CODE · 8.19

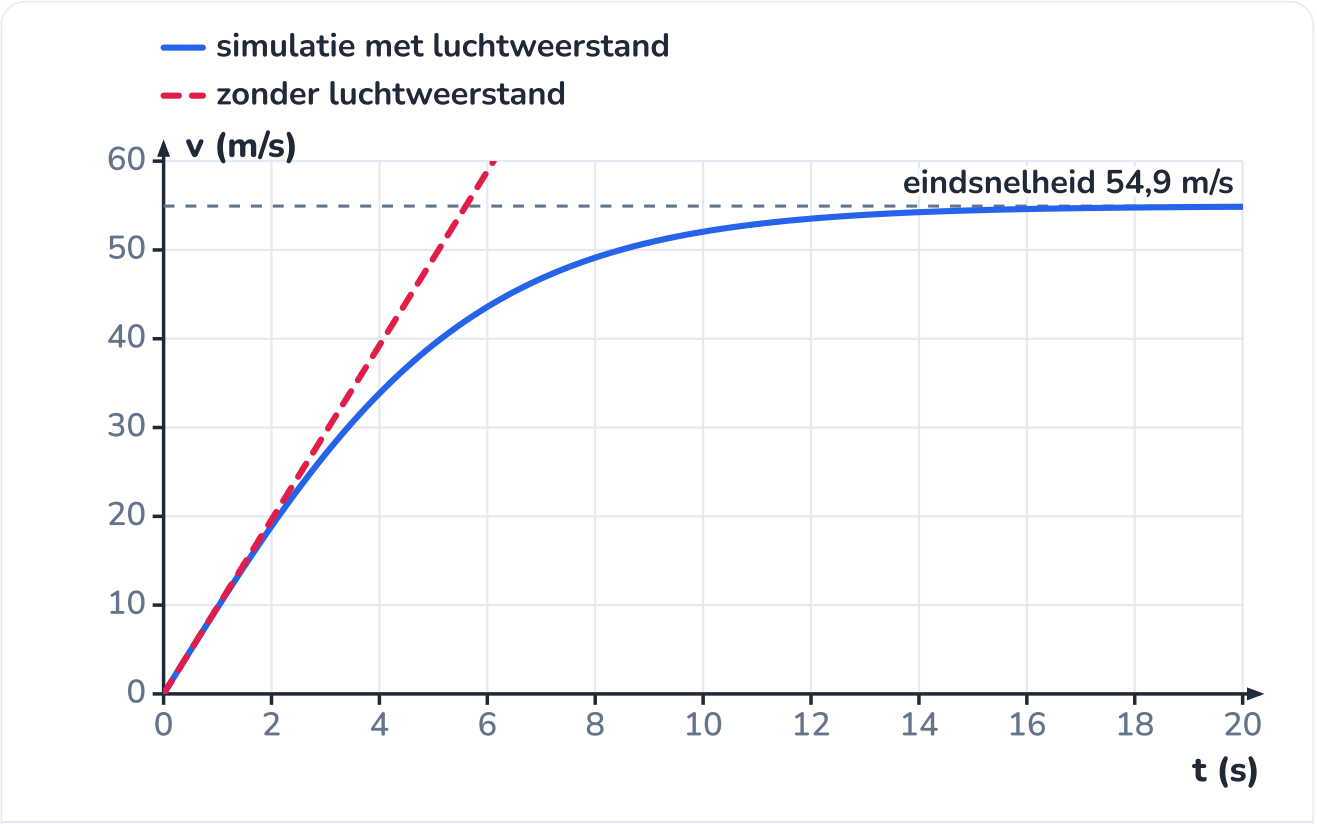
Python: vallen met luchtweerstand

19/21

Bij een valschermspringer is de kracht niet constant: de luchtweerstand groeit met het kwadraat van de snelheid, $F_{\text{lucht}} = K\,v^2$. Er is geen eenvoudige formule meer voor $v(t)$, maar met de stapmethode van hoofdstuk 7 is het eenvoudig: bereken in elk stapje de resulterende kracht, deel door de massa (tweede wet), en pas snelheid en plaats aan. Voor de springer (80 kg) is $K = 0{,}26\,\mathrm{kg/m}$.

Python **valspringer.py** 16 regels

```python
# valspringer.py - vrije val met luchtweerstand, stap voor stap met de tweede wet van Newton
G = 9.81          # valversnelling (m/s2)
M = 80.0          # massa van springer en uitrusting (kg)
K = 0.26          # luchtweerstandsconstante: F_lucht = K * v**2 (kg/m)
DT = 0.1          # tijdstap (s)

v, y = 0.0, 0.0    # snelheid en gevallen afstand, naar beneden positief
print(" t (s)   F_res (N)   a (m/s2)   v (m/s)   y (m)")
for stap in range(201):                    # 0 tot 20 s
    t = stap * DT
    F_res = M * G - K * v ** 2             # zwaartekracht omlaag, luchtweerstand omhoog
    a = F_res / M                          # tweede wet: a = F_res / m
    if stap % 20 == 0:                     # om de 2 s een regel
        print(f"{t:5.0f}   {F_res:9.1f}   {a:8.2f}   {v:7.1f}   {y:6.0f}")
    y = y + v * DT
    v = v + a * DT
```



Zonder luchtweerstand (stippellijn) blijft de snelheid lineair stijgen; met luchtweerstand nadert ze 54,9 m/s.

Uitvoer (gecontroleerd door het programma te draaien):

t (s)	F_res (N)	a (m/s2)	v (m/s)	y (m)
0	784.8	9.81	0.0	0
2	692.1	8.65	18.9	18
4	487.0	6.09	33.8	71
6	290.9	3.64	43.6	149
8	157.2	1.96	49.1	242
10	80.4	1.00	52.1	343
12	39.9	0.50	53.5	449
14	19.6	0.24	54.3	557
16	9.5	0.12	54.6	666
18	4.6	0.06	54.8	775
20	2.2	0.03	54.9	885

Bij de start is er geen luchtweerstand en is $a = g$. Hoe sneller de springer valt, hoe groter de luchtweerstand, hoe kleiner de resulterende kracht en dus de versnelling. De snelheid nadert de **eindsnelheid** waarbij $K\,v^2 = m\,g$: $v = \sqrt{m\,g/K} = \sqrt{80 \cdot 9{,}81 / 0{,}26} = 54{,}9\ \mathrm{m/s}$, bijna 200 km/h. Daar is de resulterende kracht nul, en valt de springer volgens de **eerste** wet aan constante snelheid verder. Het is dezelfde kromme als in hoofdstuk 4, nu verklaard.

NASLAG · 8.20

Overzicht: de wetten van Newton

20/21

eerste wet (traagheid)

$$\vec{F}_{\text{res}} = \vec{0} \iff \vec{v} \text{ constant}$$

tweede wet

$$\vec{F}_{\text{res}} = m \cdot \vec{a} \qquad F_{\text{res},x} = m\,a_x \qquad F_{\text{res},y} = m\,a_y$$

symbool	betekenis	eenheid
F	kracht	N = kg·m/s²
m	massa	kg
a	versnelling	m/s²

derde wet (actie en reactie, op verschillende voorwerpen)

$$\vec{F}_{A\to B} = -\,\vec{F}_{B\to A}$$

krachten

$$F_z = m\,g \qquad F_{w,s} \le \mu_s\,F_N \qquad F_{w,k} = \mu_k\,F_N \qquad F_N = F_r = m\,g\cos\alpha$$

symbool	betekenis	eenheid
μ_s, μ_k, μ_r	statische, dynamische wrijvings- en rolweerstandscoëfficiënt	geen

helling

$$\tan\alpha_{\text{grens}} = \mu_s \qquad a = g\,(\sin\alpha - \mu_k\cos\alpha)$$

normaalkracht in een lift (a omhoog positief)

$$F_N = m\,(g + a)$$

materialen	μ_s	μ_k
autoband op droog asfalt	0,8 à 1,0	ongeveer 0,7
autoband op nat asfalt	0,5 à 0,7	0,4 à 0,5
hout op hout	0,3 à 0,5	0,2 à 0,4
kunststof glijder (PE) op aluminium (rekenwaarde dakpaneel)	0,25	0,20
staal op ijs (schaats)	ongeveer 0,03	ongeveer 0,01
teflon op staal	ongeveer 0,05	ongeveer 0,05

Ordes van grootte: de echte waarde hangt af van de ruwheid, vuil, vocht en temperatuur. Wie een precieze waarde nodig heeft, meet ze (proef verderop). **Rolweerstand** werkt op dezelfde manier, $F_r = \mu_r \cdot F_N$, maar met veel kleinere coëfficiënten: fietsband op asfalt ongeveer 0,005, autoband ongeveer 0,01, de kar met rubberen wielen van de klimaatmodule 0,025.

stappenplan krachtenprobleem

1. Kies één voorwerp en teken het vrijlichaamsdiagram: zwaartekracht, en een kracht per contact.
2. Kies assen: één langs de beweging, één loodrecht erop. Ontbind schuine krachten.
3. Loodrecht op de beweging: meestal $a = 0$, daaruit F_N. Dan de wrijving met μF_N.
4. Langs de beweging: $\sum F = m\,a$. Los op naar wat gevraagd is.
5. Controleer eenheden, tekens en uitersten (α = 0° en α = 90°, μ = 0).

HET GEHEEL · 8.21

Waar Newton terugkomt

21/21

De drie wetten zijn het fundament van de mechanica: elke berekening van een brug, een lift, een voertuig of een robotarm begint met een vrijlichaamsdiagram en $\sum F = m\,a$. In de volgende hoofdstukken bekijk je dezelfde krachten door een andere bril: over een afstand (arbeid en energie), rond een draaipunt (moment), en verdeeld over een oppervlak (druk).

F11 Krachten

I6 Goniometrie en vectoren

I7 Beweging beschrijven

Kracht als pijl, zwaartekracht, massa en gewicht, de krachtmeter en wrijving in woorden.

Krachten optellen en ontbinden; de zwaartekracht op een helling.

De versnelling die de tweede wet voorspelt, en de bewegingsprofielen van het kasdak.

I9 Arbeid, energie en vermogen

Kracht maal weg: hoeveel energie en vermogen de dakmotor nodig heeft.

I10 Moment en evenwicht

Waar een kracht aangrijpt, telt als een voorwerp kan draaien: moment en evenwicht.

I11 Druk en hydraulica

Kracht verdeeld over een oppervlak: druk, en de opwaartse kracht in water.

I26 Motoren en actuatoren

De dakmotor kiezen op de kracht uit dit hoofdstuk, en hem aansturen met een H-brug.

E8 Beweging in twee dimensies

De tweede wet in twee dimensies: de schuine worp en de middelpuntzoekende kracht.

E9 Impuls en botsingen

Wat de schaatsers al lieten zien: kracht maal tijd is impuls, en impuls blijft behouden.

E12 Sterkteleer

Wat een kracht met een materiaal doet: spanning, rek en bezwijken.

Habitat

Voor Habitat leggen we vast: het schuifpaneel van het kasdak weegt 10 kg en glijdt op kunststof glijders over aluminium rails onder 25° (rekenwaarden μs = 0,25, μk = 0,20). Bij het openen schuift het naar beneden, bij het sluiten trekt de dakmotor het omhoog: de riem moet tot 60 N trekken, met een veiligheidsfactor 2 minstens 120 N, via een zelfremmende wormoverbrenging. De batterijkar rijdt op rubberen wielen met een rolweerstand van 2,5 % (69 N om ze over de oprit te duwen). In opdracht 8.13 kies je de dakmotor.

SAMENVATTING

In het kort

- De resulterende kracht is de vectorsom van alle krachten op één voorwerp; langs één lijn tel je met tekens, anders per as met componenten.
- Eerste wet: is de resulterende kracht nul, dan blijft de snelheid constant (rust of ERB). Een kracht is nodig om een beweging te veranderen, niet om ze te behouden.
- Tweede wet: $F_{\text{res}} = m \cdot a$. De massa is de traagheid; $1\,\text{N} = 1\,\text{kg} \cdot \text{m/s}^2$. Bij vrije val geeft ze $a = g$ voor elk voorwerp.
- Derde wet: krachten komen in paren, even groot en tegengesteld, op verschillende voorwerpen. Ze heffen elkaar dus nooit op.
- Een vrijlichaamsdiagram toont één voorwerp met alle krachten die de omgeving erop uitoefent: zwaartekracht plus een kracht per contact.
- Wrijving: statisch hoogstens $\mu_s F_N$, dynamisch $\mu_k F_N$. De normaalkracht is niet altijd het gewicht.
- Op een helling is $F_N = mg\cos\alpha$; een voorwerp blijft liggen zolang $\tan\alpha \le \mu_s$. Omhoog trekken kost $mg\sin\alpha + \mu_k mg\cos\alpha + ma$.
- Met de tweede wet en de stapmethode reken je ook bewegingen uit waarbij de kracht verandert, zoals vallen met luchtweerstand.

BEGRIPPEN

Wat je nu kent

resulterende kracht
De vectorsom van alle krachten op één voorwerp; ze bepaalt de versnelling.

eerste wet van Newton
Is de resulterende kracht nul, dan blijft een voorwerp in rust of beweegt het met constante snelheid langs een rechte (traagheid).

tweede wet van Newton
De resulterende kracht is massa maal versnelling: $\vec{F}_{\text{res}} = m \cdot \vec{a}$.

derde wet van Newton
Oefent A een kracht uit op B, dan oefent B een even grote, tegengestelde kracht uit op A (actie en reactie).

newton
De eenheid van kracht: $1\,\text{N} = 1\,\text{kg} \cdot \text{m/s}^2$, de kracht die 1 kg een versnelling van 1 m/s² geeft.

vrijlichaamsdiagram
Tekening van één voorwerp, los van zijn omgeving, met alle krachten die de omgeving erop uitoefent.

normaalkracht
De contactkracht van een oppervlak op een voorwerp, loodrecht op dat oppervlak.

statische wrijving
Wrijving tussen oppervlakken die niet over elkaar schuiven; ze past zich aan tot hoogstens $\mu_s F_N$.

dynamische wrijving
Wrijving tussen oppervlakken die over elkaar schuiven: $F_w = \mu_k F_N$.

wrijvingscoëfficiënt
Getal zonder eenheid dat zegt hoe sterk twee materialen wrijven: wrijvingskracht gedeeld door normaalkracht.

grenshoek
De hellingshoek waarbij een voorwerp net begint te glijden: $\tan\alpha = \mu_s$.

eindsnelheid
De snelheid waarbij de luchtweerstand even groot is als de zwaartekracht; de versnelling is dan nul.

HANDBOEK · HOOFDSTUK 9

Arbeid, energie en vermogen

Kracht maal weg: hoe energie van vorm verandert, hoeveel ervan verloren gaat, en hoe snel

Een kar een oprit op duwen, water naar boven pompen, een auto afremmen: telkens wordt er energie omgezet. In Fundamental leerde je de energievormen en het behoud van energie. Nu reken je ermee. **Arbeid** koppelt energie aan kracht en weg, **kinetische** en **potentiële energie** krijgen een formule, en zonder wrijving blijft hun som behouden. Met **vermogen** en **rendement** bereken je hoeveel energie een pomp werkelijk nodig heeft, en je ontdekt waarom een waterreservoir in Habitat een slechte batterij zou zijn.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

K Krachten, beweging en constructies · E Energie · W Wiskundig modelleren

Een batterij van water: 's nachts wordt het water omhoog gepompt, overdag stroomt het door turbines terug.

Na dit hoofdstuk

- Je kan arbeid berekenen als kracht maal verplaatsing, ook als de kracht een hoek maakt met de verplaatsing, en het teken van de arbeid verklaren.
- Je kan arbeid bepalen als oppervlakte onder een (F,s)-grafiek, ook bij een veranderlijke kracht.
- Je kan kinetische en potentiële energie berekenen en de formule voor de kinetische energie afleiden uit de tweede wet.
- Je kan met behoud van mechanische energie snelheden en hoogtes berekenen, zonder de tijd te kennen.
- Je kan het energieverlies door wrijving berekenen als arbeid van de wrijvingskracht en in een energieboekhouding verwerken.
- Je kan vermogen berekenen met $P = W/t$ en $P = F \cdot v$, en omrekenen tussen joule, wattuur en kilowattuur.
- Je kan het rendement van een machine berekenen en de verliezen voorstellen in een Sankey-diagram.
- Je kan het energieverbruik van de pomp van Habitat berekenen en vergelijken met de energie in een batterij.

UITLEG · 9.1

Wat je al weet, en wat nu komt

In hoofdstuk F15 leerde je dat energie niet verloren gaat maar van vorm verandert, dat je ze meet in joule, dat een toestel altijd een deel 'verliest' als warmte (rendement), en dat een Sankey-diagram die stromen zichtbaar maakt. In hoofdstuk F21 berekende je elektrische energie als vermogen maal tijd.

uit Fundamental (F15)	in dit hoofdstuk
bewegingsenergie	kinetische energie $E_k = \frac{1}{2}\,m\,v^2$
zwaarte-energie	potentiële energie $E_p = m\,g\,h$
energie gaat niet verloren, ze wordt omgezet	behoud van mechanische energie, en wat wrijving ermee doet
de joule	$1\,\mathrm{J} = 1\,\mathrm{N}\cdot\mathrm{m}$: arbeid als overdracht van energie
rendement in procent, Sankey-diagram	rendement van een pomp en een motor, berekend
vermogen van een toestel (F21)	$P = W/t$ en $P = F\cdot v$

In hoofdstuk 6 viel je iets op: om de batterijkar over de oprit te duwen, is maar 49 N nodig over 6,41 m; om ze recht omhoog te tillen 785 N over 0,40 m. Maar $49 \cdot 6{,}41 = 785 \cdot 0{,}40 = 314$. Toeval? Nee: dat product van kracht en weg is de **arbeid**, en het is de energie die in de kar gaat zitten, langs welke weg ook. In dit hoofdstuk

- reken je arbeid uit, ook als kracht en weg niet in dezelfde richting liggen;
- geef je bewegings- en zwaarte-energie een formule, en gebruik je het behoud ervan als rekenmethode;
- volg je wat wrijving met energie doet;
- reken je met vermogen en rendement aan de pomp en de dakmotor van Habitat.

UITLEG · 9.2

Arbeid: kracht maal weg

Een kracht die een voorwerp over een afstand verplaatst, draagt energie over. Die overgedragen energie heet **arbeid**. Werkt de kracht in de richting van de verplaatsing, dan is ze gewoon kracht maal afstand. Maakt de kracht een hoek α met de verplaatsing, dan telt alleen de component langs de verplaatsing, $F\cos\alpha$ (hoofdstuk 6):

ARBEID

$$W = F\cdot s\cdot\cos\alpha$$

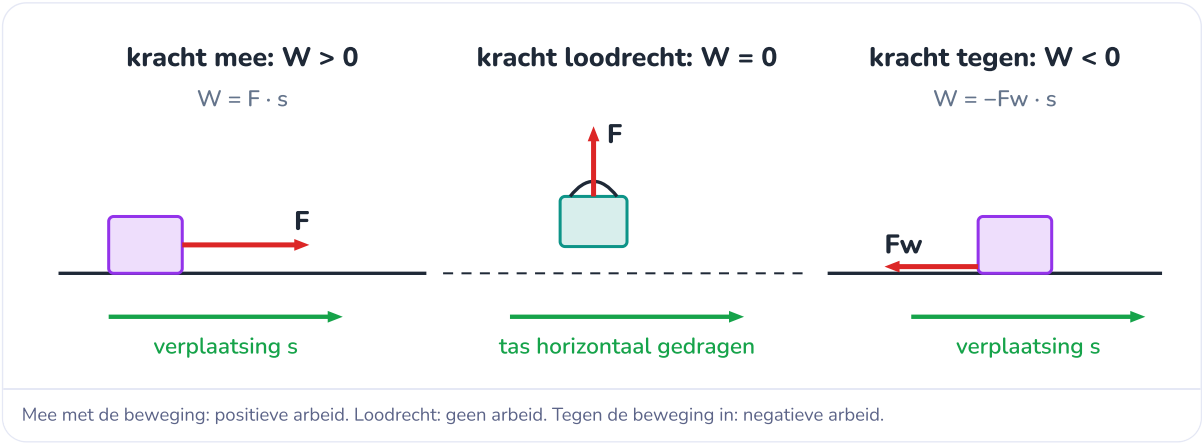
F in N, s (de verplaatsing) in m, W in joule: $1\,\mathrm{J} = 1\,\mathrm{N}\cdot\mathrm{m}$. Bij $\alpha = 0^\circ$ is $W = F\cdot s$.

Het teken van de arbeid heeft een betekenis:

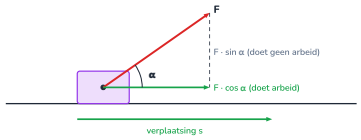
- $W > 0$ ($\alpha < 90^\circ$): de kracht **geeft** energie aan het voorwerp. Je duwt de kar vooruit.
- $W = 0$ ($\alpha = 90^\circ$): de kracht draagt geen energie over. Een boodschappentas horizontaal dragen: je tilt verticaal, de tas beweegt horizontaal. (Moe word je wel: je spieren verbruiken energie om te blijven aanspannen, maar die komt niet in de tas terecht.) Ook de normaalkracht op een horizontaal rijdende kar doet geen arbeid.
- $W < 0$ ($\alpha > 90^\circ$): de kracht **neemt** energie weg. Wrijving werkt tegen de beweging ($\alpha = 180^\circ$, $\cos 180^\circ = -1$): ze zet bewegingsenergie om in warmte.

Arbeid is inspanning

In de natuurkunde is arbeid alleen wat een kracht langs een verplaatsing overdraagt. Een zware kist een uur lang stilhouden kost veel inspanning, maar de arbeid op de kist is nul: ze verplaatst niet. Een plank die de kist draagt, doet precies hetzelfde zonder moe te worden.



Mee met de beweging: positieve arbeid. Loodrecht: geen arbeid. Tegen de beweging in: negatieve arbeid.



Een schuin touw: alleen F · cos α trekt mee in de bewegingsrichting; F · sin α tilt en doet geen arbeid.

UITGEWERKT VOORBEELD · 9.3

Habitat: de arbeid van elke kracht op de batterijkar

3/19

De batterijkar (80 kg) wordt aan constante snelheid over de hele oprit geduwd: 6,41 m langs de helling, 0,40 m hoger. Uit hoofdstuk 8: duwkracht 68,6 N langs de helling, $F_{\parallel} = 49{,}0\,\mathrm{N}$, rolweerstand 19,6 N, normaalkracht 783 N. Bereken de arbeid van elke kracht en tel ze op.

GEGEVEN

- $s = 6{,}41\,\mathrm{m}$ langs de helling omhoog
- $F_{\mathrm{duw}} = 68{,}6\,\mathrm{N}$, $F_{\parallel} = 49{,}0\,\mathrm{N}$, $F_r = 19{,}6\,\mathrm{N}$, $F_N = 783\,\mathrm{N}$

GEVRAAGD

- W per kracht, en de som

OPLOSSING

1. Duwkracht, in de zin van de verplaatsing ($\alpha = 0°$): $W = 68{,}6 \cdot 6{,}41 = +440\,\mathrm{J}$.
2. Zwaartekracht: haar component langs de helling wijst omlaag, tegen de verplaatsing: $W = -49{,}0 \cdot 6{,}41 = -314\,\mathrm{J}$. (Of met de volle kracht en de hoek tussen verticaal omlaag en de verplaatsing, $93{,}58°$: $785 \cdot 6{,}41 \cdot \cos 93{,}58° = -314\,\mathrm{J}$.)
3. Rolweerstand, tegen de beweging ($\alpha = 180°$): $W = -19{,}6 \cdot 6{,}41 = -126\,\mathrm{J}$.
4. Normaalkracht, loodrecht op de verplaatsing: $W = 0$.
5. Som: $440 - 314 - 126 + 0 = 0\,\mathrm{J}$.

ANTWOORD

Jij levert 440 J. Daarvan gaat 314 J naar hoogte (de zwaartekracht 'neemt' het op) en 126 J wordt warmte in de wielen en de vloer. De totale arbeid is nul.

Klopt dit? Dat de som nul is, is geen toeval: de kar gaat niet sneller of trager, dus haar bewegingsenergie verandert niet. De 314 J is precies m · g · h = 80 · 9,81 · 0,40: de zwaarte-energie die de kar bovenaan meer heeft (verderop). Zonder rolweerstand had je 314 J geleverd, zoals in hoofdstuk 6.

UITLEG · 9.4

Arbeid als oppervlakte

4/19

Teken de kracht in functie van de verplaatsing, een **(F,s)-grafiek**. Bij een constante kracht is de arbeid $F \cdot s$ de oppervlakte van een rechthoek, zoals de verplaatsing de oppervlakte was onder een (v,t)-grafiek (hoofdstuk 7). Dat blijft gelden als de kracht verandert: de arbeid is de **oppervlakte onder de (F,s)-grafiek**.

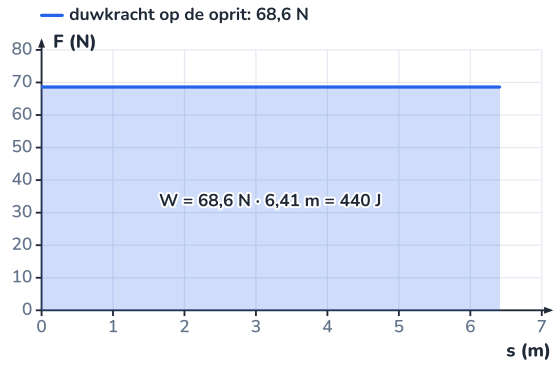
Een voorbeeld is een veer. Volgens de wet van Hooke (hoofdstuk F11) groeit de kracht evenredig met de uitrekking: $F = k \cdot s$, met k de veerconstante in N/m. Wie een veer uitrekt, begint met een kleine kracht en eindigt met een grote. De oppervlakte is een driehoek:

ARBEID OM EEN VEER UIT TE REKKEN

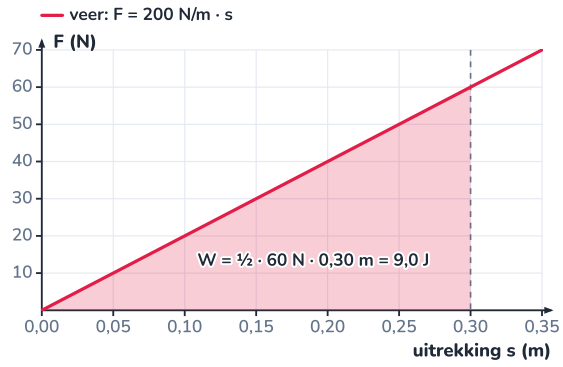
$$W = \tfrac{1}{2} F_{\text{eind}} \cdot s = \tfrac{1}{2} k\, s^2$$

Een veer met $k = 200\,\text{N/m}$ 30 cm uitrekken: $\tfrac{1}{2} \cdot 200 \cdot 0{,}30^2 = 9{,}0\,\text{J}$. Die energie zit daarna als **veerenergie** in de veer.

Moet je een oppervlakte onder een kromme bepalen, dan tel je hokjes of splits je ze in stroken: precies wat een computer doet (hoofdstuk 7, en hoofdstuk E3 met integralen).



Constante kracht: de arbeid is een rechthoek, 440 J voor de kar op de oprit.



Veer: de kracht groeit met de uitrekking, de arbeid is een driehoek van 9,0 J.

UITLEG · 9.5

Kinetische energie

5/19

Hoeveel energie zit er in een bewegend voorwerp? Reken uit hoeveel arbeid nodig is om het vanuit stilstand op snelheid v te brengen. Een constante resulterende kracht F geeft een versnelling $a = F/m$ (tweede wet). De afstand die daarvoor nodig is, volgt uit de formule zonder tijd (hoofdstuk 7): $v^2 = 2\,a\,s$, dus $s = \dfrac{v^2}{2a}$. De arbeid is dan

$$W = F \cdot s = m\,a \cdot \frac{v^2}{2\,a} = \tfrac{1}{2}\,m\,v^2$$

De versnelling valt weg: hard duwen over een korte afstand of zacht duwen over een lange, het resultaat is hetzelfde. Die energie noemen we de **kinetische energie** (bewegingsenergie):

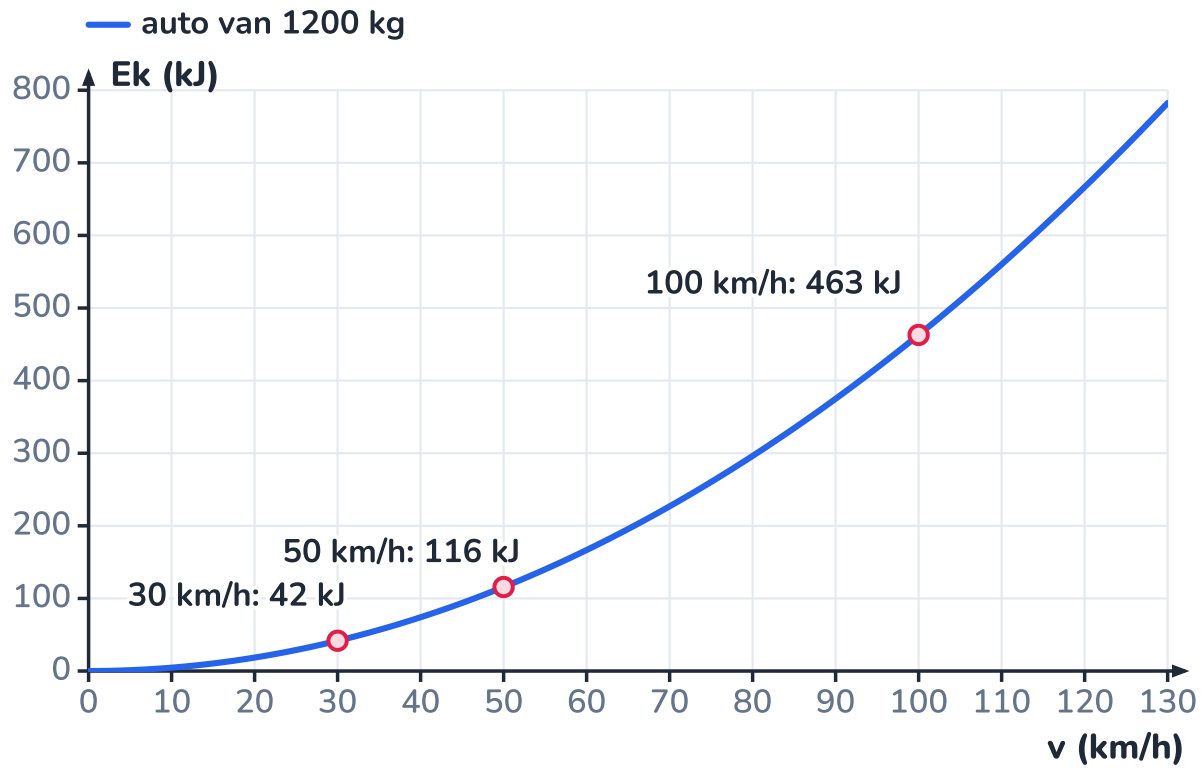
KINETISCHE ENERGIE EN DE ARBEID-ENERGIESTELLING

$$E_k = \tfrac{1}{2}\,m\,v^2 \qquad W_{\text{res}} = \Delta E_k$$

m in kg, v in m/s, E_k in J. De arbeid van de **resulterende** kracht is de verandering van de kinetische energie: positief als het voorwerp sneller gaat, negatief als het trager gaat.

voorwerp	massa	snelheid	E_k
schuifpaneel van het kasdak	10 kg	0,10 m/s	0,05 J
wandelaar	70 kg	1,4 m/s	69 J
fietser met fiets	90 kg	18 km/h	1,1 kJ
auto in de bebouwde kom	1200 kg	50 km/h	116 kJ
auto op de autosnelweg	1200 kg	120 km/h	667 kJ

Omdat v in het kwadraat staat, groeit de energie snel met de snelheid: twee keer zo snel is vier keer zoveel energie. Een auto aan 120 km/h heeft bijna zes keer de energie van dezelfde auto aan 50 km/h.



Een parabool: van 50 naar 100 km/h verviervoudigt de energie, van 116 naar 463 kJ.

UITGEWERKT VOORBEELD · 9.6

Waarom de remweg met het kwadraat groeit

6/19

In hoofdstuk 7 leidde je de remafstand $v^2/(2a)$ af met oppervlaktes. Toon nu met energie aan waarom een auto van 1200 kg die remt met een remkracht van 8400 N (7,0 m/s²) bij 100 km/h vier keer zo ver remt als bij 50 km/h. Hoe hoog zou je dezelfde auto moeten laten vallen om evenveel kinetische energie te krijgen als bij 50 km/h?

GEGEVEN

- $m = 1200\,\text{kg}$, $F_{\text{rem}} = 8400\,\text{N}$
- $v = 50\,\text{km/h} = 13{,}9\,\text{m/s}$ en $100\,\text{km/h} = 27{,}8\,\text{m/s}$

GEVRAAGD

- remafstand bij beide snelheden
- gelijkwaardige valhoogte

OPLOSSING

1. Tijdens het remmen is de remkracht de resulterende kracht, tegen de beweging in. Haar arbeid is $-F_{\text{rem}} \cdot d$, en die moet alle kinetische energie wegnemen: $F_{\text{rem}} \cdot d = \frac{1}{2}mv^2$.
2. 50 km/h: $E_k = \frac{1}{2} \cdot 1200 \cdot 13{,}9^2 = 116\,\text{kJ}$; $d = \dfrac{115\,700}{8400} = 13{,}8\,\text{m}$.
3. 100 km/h: $E_k = 463\,\text{kJ}$ (vier keer zoveel); $d = \dfrac{463\,000}{8400} = 55{,}1\,\text{m}$.
4. Valhoogte met dezelfde energie: $m\,g\,h = \frac{1}{2}mv^2$, dus $h = \dfrac{v^2}{2g} = \dfrac{13{,}9^2}{2 \cdot 9{,}81} = 9{,}8\,\text{m}$.

ANTWOORD

Remafstand 13,8 m bij 50 km/h en 55,1 m bij 100 km/h: de remkracht moet vier keer zoveel energie omzetten, dus over een vier keer zo lange afstand. Een botsing aan 50 km/h is als een val van bijna 10 m hoog, van de derde verdieping.

Klopt dit? Dezelfde 13,8 m als in hoofdstuk 7 ($13{,}9^2/14$): twee methoden, één antwoord. De remenergie (116 kJ) wordt warmte in de remschijven. Een elektrische auto kan een deel ervan terugwinnen door de motor als dynamo te gebruiken (opdracht 9.16).

UITLEG · 9.7

Potentiële energie: zwaarte-energie

7/19

Til een voorwerp met massa m traag een hoogte h op. Je kracht is even groot als de zwaartekracht, $m\,g$, en je arbeid is $m\,g \cdot h$. Die energie is niet weg: laat je het voorwerp los, dan krijg je ze terug als bewegingsenergie. Ze zit 'opgeslagen' in de hoogte, als **potentiële energie** of zwaarte-energie:

ZWAARTE-ENERGIE

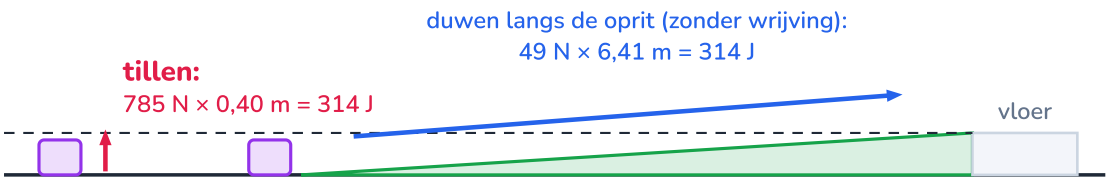
$$E_p = m\,g\,h \qquad W_{\text{zwaartekracht}} = -\Delta E_p$$

h is de hoogte boven een **referentieniveau** dat je zelf kiest (de vloer, de grond, de bodem van het reservoir). Alleen verschillen van E_p tellen, dus de keuze maakt voor de uitkomst niets uit.

Belangrijk: de arbeid van de zwaartekracht hangt alleen af van het **hoogteverschil**, niet van de weg. Recht omhoog tillen of over een lange, flauwe oprit duwen: de zwaartekracht neemt telkens $m\,g\,h$ op. Een oprit verkleint de kracht, niet de energie. Dat is de 'gulden regel' van alle eenvoudige machines: hefboom, katrol, hellend vlak. Wat je wint aan kracht, verlies je aan weg (hoofdstuk F12).

Alleen bij kleine hoogtes

$E_p = m\,g\,h$ gaat ervan uit dat g overal even groot is. Voor alles op aarde, tot vele kilometers hoog, klopt dat goed genoeg. Voor satellieten en raketten niet meer: daar neemt de zwaartekracht merkbaar af met de afstand tot het middelpunt van de aarde.



zelfde hoogteverschil, zelfde zwaarte-energie: Ep = m · g · h = 314 J

Tillen: grote kracht, korte weg. Duwen: kleine kracht, lange weg. Dezelfde 314 J (zonder wrijving).

UITLEG · 9.8

Behoud van mechanische energie

Laat een bal van 0,50 kg vallen van 10 m hoogte. Bovenaan heeft hij $E_p = 0{,}50 \cdot 9{,}81 \cdot 10 = 49{,}05\,\text{J}$ en geen E_k. Onderweg wint hij snelheid en verliest hij hoogte. Zolang alleen de zwaartekracht arbeid doet, is wat de ene vorm verliest precies wat de andere wint:

BEHOUD VAN MECHANISCHE ENERGIE (ZONDER WRIJVING)

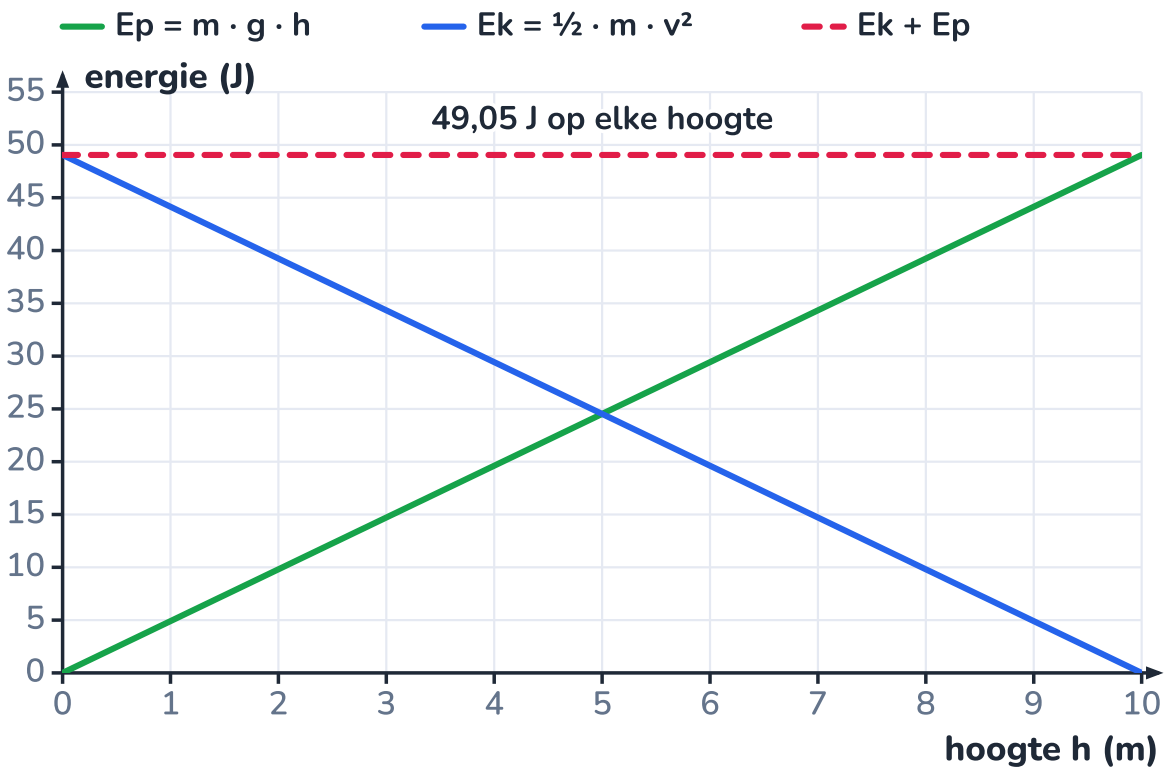
$$E_{k,1} + E_{p,1} = E_{k,2} + E_{p,2} \qquad \tfrac{1}{2} m v_1^2 + m g h_1 = \tfrac{1}{2} m v_2^2 + m g h_2$$

Geldt als alleen de zwaartekracht (of een veer) arbeid doet. Normaalkrachten mogen, want ze staan loodrecht op de beweging. Wrijving en luchtweerstand niet.

Die wet is een krachtige rekenmethode, want ze vraagt de tijd niet en ook de vorm van de baan niet. Deel door m en de massa valt weg. Voor een val vanuit rust over een hoogte h geeft ze meteen $v = \sqrt{2\,g\,h}$: dezelfde formule als in hoofdstuk 7, zonder één keer t te gebruiken. En ze geldt evengoed voor een glijbaan, een slinger of een achtbaan, waar de versnelling voortdurend verandert en de formules van de EVRB niet meer werken.

Waar blijft de energie bij wrijving?

Met wrijving daalt de mechanische energie. De energie is niet weg: ze wordt **warmte** (de deeltjes van de oppervlakken gaan sneller trillen, hoofdstuk F6). De totale energie blijft behouden: $E_k + E_p + Q$ is constant, met Q de wrijvingswarmte, gelijk aan de arbeid tegen de wrijving, $F_w \cdot s$.



Terwijl de bal valt (van 10 m naar 0 m), gaat Ep over in Ek. De som blijft 49,05 J.

UITGEWERKT VOORBEELD · 9.9

De bal uit hoofdstuk 4, nu met energie

9/19

In hoofdstuk 4 gooide je een bal recht omhoog: hij verliet je hand op 1,5 m hoogte met 10 m/s, en met een parabool vond je een top van 6,60 m. Bereken die top nu met behoud van energie. Met welke snelheid komt de bal op de grond, als je hem niet vangt? Luchtweerstand verwaarloos je.

GEGEVEN

- $h_1 = 1{,}5\,\text{m}$, $v_1 = 10\,\text{m/s}$
- top: $v = 0$; grond: $h = 0$

GEVRAAGD

- h_{top} en v_{grond}

OPLOSSING

1. Kies de grond als referentieniveau. Energie in de hand, per kg (deel door m): $\frac{1}{2}v_1^2 + g\,h_1 = 50 + 14{,}7 = 64{,}7\,\text{J/kg}$.
2. Top ($v = 0$): $g\,h_{\text{top}} = 64{,}7$, dus $h_{\text{top}} = 64{,}7/9{,}81 = 6{,}60\,\text{m}$.
3. Grond ($h = 0$): $\frac{1}{2}v^2 = 64{,}7$, dus $v = \sqrt{129{,}4} = 11{,}4\,\text{m/s}$.

ANTWOORD

De top ligt op 6,60 m, precies wat de parabool gaf. De bal raakt de grond met 11,4 m/s: sneller dan hij vertrok, want hij valt 1,5 m dieper dan je hand.

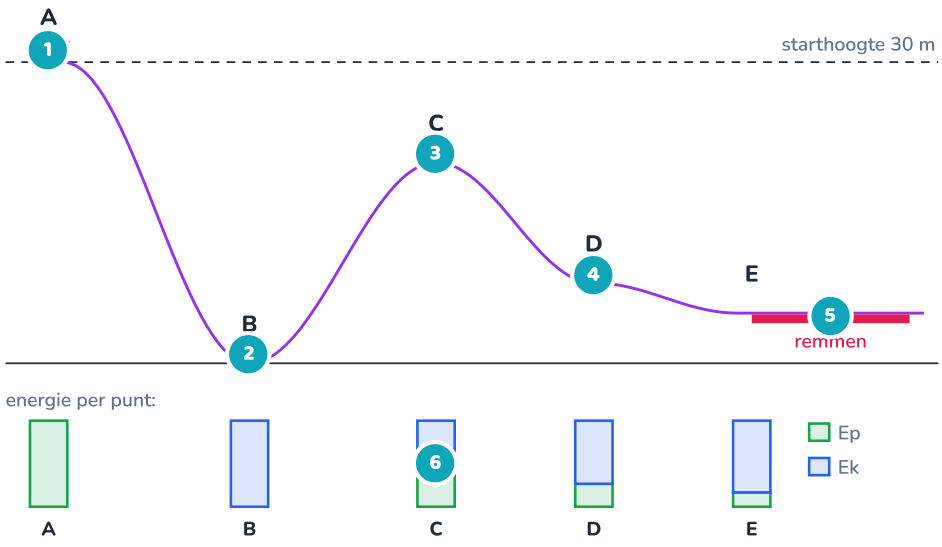
Klopt dit? De massa van de bal kwam er niet aan te pas, en de tijd ook niet. Met de parabool had je eerst het tijdstip van de top moeten zoeken. Met energie vind je de snelheid op elke hoogte in één stap; wil je weten wannéér, dan heb je de bewegingsformules nodig. De twee methoden vullen elkaar aan.

ONTLEED · 9.10

Energie in een achtbaan

10/19

Een achtbaan heeft geen motor in de karretjes. Een ketting trekt ze naar het hoogste punt, en daarna doet de zwaartekracht het werk. De figuur toont een baan zonder wrijving, met de energie in elk punt als staafje: groen de zwaarte-energie, blauw de bewegingsenergie. De hoogte van elk staafje is dezelfde.



Een achtbaan in zijaanzicht (hoogtes op schaal, 7 px per meter), met de energieverdeling in vijf punten.

1 **A: de top van de optakelheuvel**
30 m hoog, (bijna) stil. Alle energie is zwaarte-energie. Voor een karretje van 500 kg heeft de ketting $500 \cdot 9{,}81 \cdot 30 = 147\,\text{kJ}$ arbeid geleverd: meer komt er daarna nooit bij.

2 **B: het laagste punt**
Alle zwaarte-energie is bewegingsenergie geworden: $v = \sqrt{2 \cdot 9{,}81 \cdot 30} = 24{,}3\,\text{m/s}$, 87 km/h. Voor elk karretje, zwaar of licht.

3 **C: de tweede heuvel is lager**
Op 20 m is nog 10 m 'over': $v = \sqrt{2 \cdot 9{,}81 \cdot 10} = 14{,}0\,\text{m/s}$. Een heuvel van meer dan 30 m haalt het karretje nooit: er is geen energie voor.

4 **D: tweede dal**
Op 8 m: $v = \sqrt{2 \cdot 9{,}81 \cdot 22} = 20{,}8\,\text{m/s}$. De snelheid hangt alleen af van hoe diep het punt onder de start ligt, niet van de weg ernaartoe.

5 **E: de remmen**
Het station ligt op 5 m; het karretje komt aan met $\sqrt{2 \cdot 9{,}81 \cdot 25} = 22{,}1\,\text{m/s}$. Remmen (wrijving of wervelstroomremmen) zetten die bewegingsenergie om in warmte.

6 **In werkelijkheid**
Wrijving in de wielen en luchtweerstand nemen onderweg energie weg: elk staafje zou iets korter zijn dan het vorige, en de echte snelheden liggen enkele procenten lager. Daarom moet elke volgende heuvel duidelijk lager zijn dan de start.

UITLEG · 9.11

Wrijving, verlies en rendement

11/19

Geen enkele machine zet alle energie om in wat je wil. Een motor wordt warm, een pomp maakt het water wat warmer en draaikolken in de leidingen kosten energie. Het **rendement** η (de Griekse letter èta) zegt welk deel nuttig is:

RENDEMENT

$$\eta = \frac{E_{\text{nuttig}}}{E_{\text{toegevoerd}}} = \frac{P_{\text{nuttig}}}{P_{\text{toegevoerd}}}$$

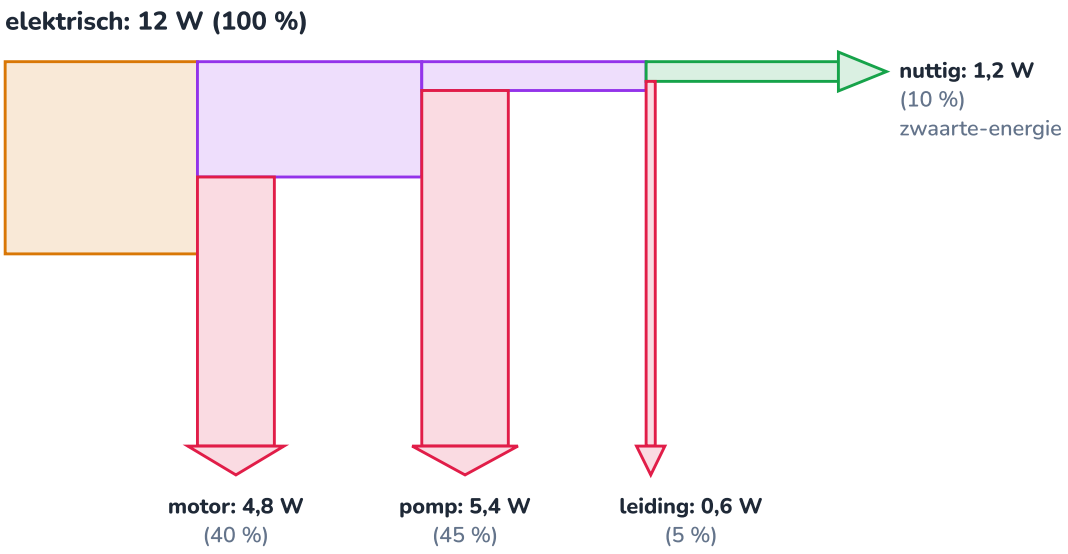
Een getal tussen 0 en 1, vaak in procent. Het verlies $E_{\text{toegevoerd}} - E_{\text{nuttig}}$ wordt uiteindelijk warmte.

Het **Sankey-diagram** (hoofdstuk F15) toont waar de energie heen gaat, met de breedte van elke pijl evenredig met het vermogen. Het diagram hiernaast hoort bij de pomp van de waterkringloop van Habitat: een 12 V-dompelpomp die 12 W opneemt en 5,0 liter per minuut 1,5 m hoog pompt. Het nuttige vermogen is de zwaarte-energie die het water per seconde krijgt: $\frac{m}{t}\,g\,h = \frac{5{,}0\,\text{kg}}{60\,\text{s}} \cdot 9{,}81 \cdot 1{,}5 = 1{,}2\,\text{W}$. Het rendement is dus maar 10 %.

Dat is typisch voor kleine pompen. Grote centrifugaalpompen halen 70 à 90 %, elektromotoren van enkele kilowatt 85 à 90 %. Hoe kleiner de machine, hoe groter het deel dat verloren gaat aan wrijving in lagers, lekstromen en draaikolken.

⚠ **Behoud van energie en rendement zijn geen tegenspraak**

De 10,8 W die 'verloren' gaat, is niet vernietigd. Ze zit als warmte in de motor, de pomp en het water. Voor de energiewet telt ze volledig mee; voor de gebruiker is ze waardeloos. Verlies betekent: omgezet in een vorm die je niet kan gebruiken.



Van 12 W elektrisch blijft 1,2 W over als zwaarte-energie van het water. De verdeling van de verliezen is een schatting.

UITLEG · 9.12

Vermogen: hoe snel energie omgezet wordt

12/19

Een lift en een trap brengen je naar dezelfde verdieping, met dezelfde zwaarte-energie. Het verschil is de **tijd**. Het **vermogen** is de arbeid (of de omgezette energie) per seconde:

VERMOGEN

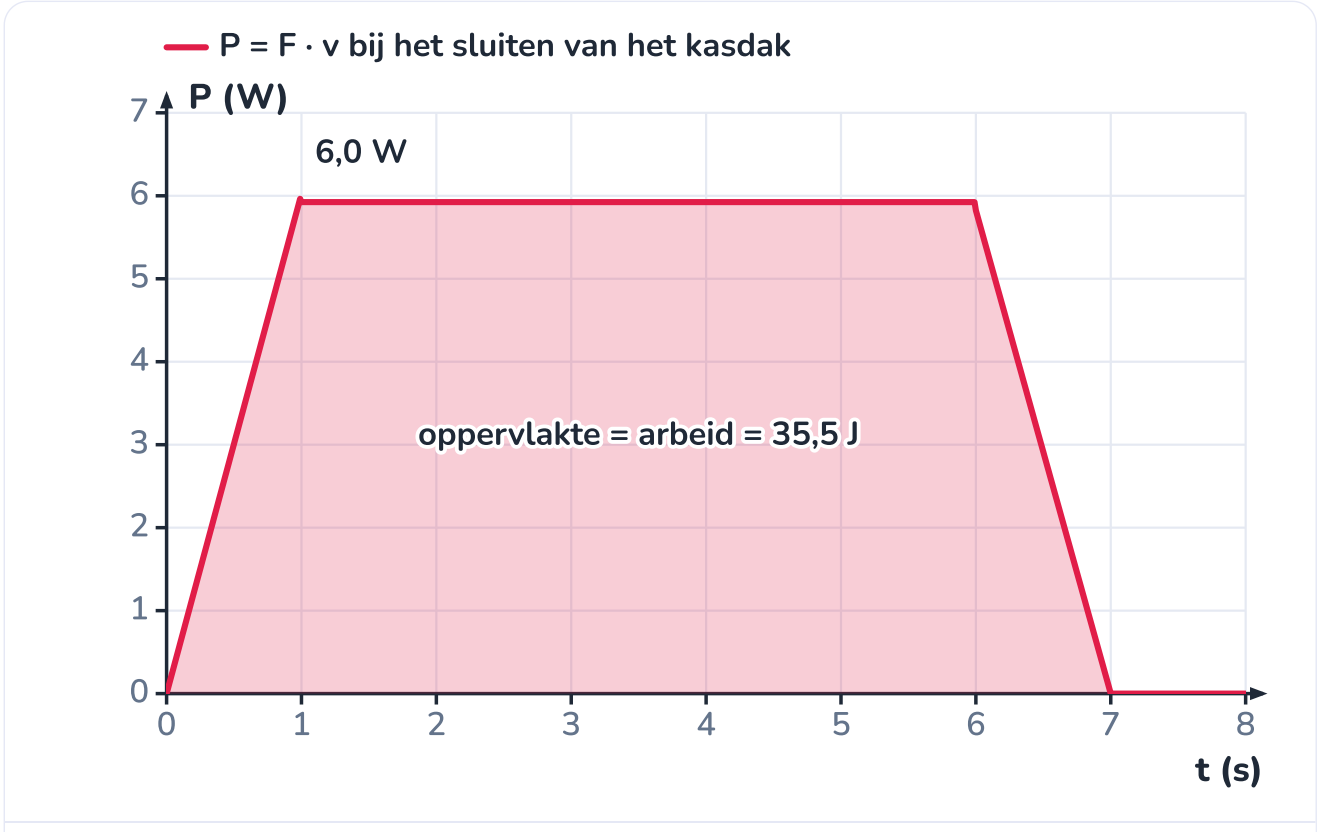
$$P = \frac{W}{t} = \frac{E}{t} \qquad P = F \cdot v$$

P in watt: $1\,\mathrm{W} = 1\,\mathrm{J/s}$. De tweede vorm volgt uit de eerste: $\frac{F \cdot s}{t} = F \cdot \frac{s}{t} = F \cdot v$, met de kracht in de richting van de snelheid.

Energie in kilowattuur. Een vermogen van 1 kW gedurende 1 uur levert $1000\,\mathrm{J/s} \cdot 3600\,\mathrm{s} = 3{,}6\,\mathrm{MJ}$: dat is **1 kWh**, de eenheid op je energiefactuur (hoofdstuk F21). Voor kleine toestellen reken je in wattuur: $1\,\mathrm{Wh} = 3600\,\mathrm{J}$.

toestel of situatie	vermogen (orde van grootte)
schuifpaneel van het kasdak (mechanisch)	6 W
pomp van de waterkringloop (elektrisch)	12 W
mens, lang volgehouden (recreatief fietsen)	100 à 200 W
mens, enkele seconden (trap oplopen)	500 W à 1 kW
waterkoker	2 kW
gezinsauto (motor, maximaal)	50 à 150 kW
pompcentrale van Coo (alle turbines)	ruim 1 GW

De dakmotor. Bij het sluiten trekt de riem met ongeveer 59 N aan 0,10 m/s: $P = F \cdot v = 5{,}9\,\mathrm{W}$ mechanisch. De grafiek toont $P = F \cdot v$ over de hele sluiting; de oppervlakte eronder is de arbeid, $59{,}2\,\mathrm{N} \cdot 0{,}60\,\mathrm{m} = 35{,}5\,\mathrm{J}$. Daarvan wordt 24,9 J zwaarte-energie van het paneel en 10,7 J warmte in de glijders.



Het vermogen stijgt tijdens het aanlopen tot 6,0 W en daalt tijdens het afremmen naar nul; de oppervlakte is 35,5 J.

PROEF · 9.13

Je eigen vermogen op de trap

13/19

ONDERZOEKSVRAAG

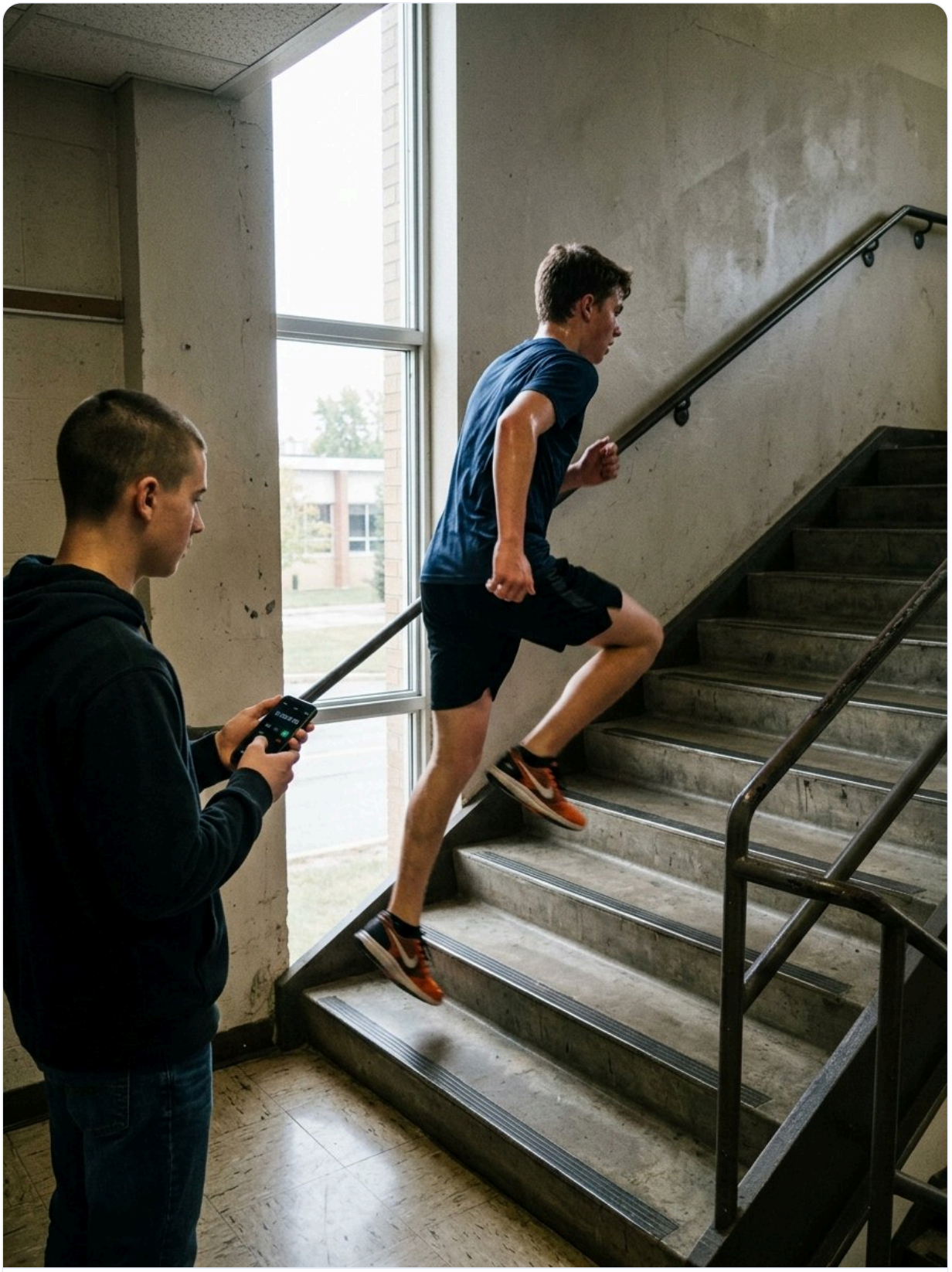
Hoeveel vermogen levert je lichaam als je een trap oploopt, en als je hem rustig op stapt?

MATERIAAL

- een trap met minstens 12 treden
- rolmeter
- stopwatch (of de stopwatch van een smartphone)
- personenweegschaal

Veiligheid

Loop niet sneller dan veilig kan, houd de leuning vast, en doe de proef niet als je een blessure of een hartprobleem hebt.



Zelfde trap, zelfde energie; wie sneller boven is, levert een groter vermogen.

WERKWIJZE

1. Meet je massa. Meet de hoogte van één trede en tel de treden; bereken de totale hoogte h.
2. Een partner meet de tijd van de eerste tot de laatste trede terwijl je rustig naar boven stapt.
3. Herhaal terwijl je zo snel als veilig kan naar boven loopt. Herhaal elke meting twee keer.

4 Bereken telkens de zwaarte-energie $m\,g\,h$ en het vermogen $P = m\,g\,h/t$.

WAARNEMING

Voorbeeld: 58 kg, 18 treden van 17,0 cm, dus $h = 3{,}06\,\text{m}$. Rustig stappen: 9,0 s. Lopen: 3,4 s. $E_p = 58 \cdot 9{,}81 \cdot 3{,}06 = 1741\,\text{J}$.

	tijd	vermogen
rustig stappen	9,0 s	$1741/9{,}0 = 193\,\text{W}$
lopen	3,4 s	$1741/3{,}4 = 512\,\text{W}$

VERKLARING

Dezelfde energie in minder tijd is een groter vermogen. Een halve kilowatt houdt een mens maar enkele seconden vol; lang volhouden lukt met ongeveer 100 à 200 W, een getrainde wielrenner haalt een uur lang ongeveer 300 à 400 W.

Wat je berekent, is alleen het *nuttige* vermogen: de zwaarte-energie per seconde. Je spieren hebben een rendement van ongeveer 20 à 25 %: om 512 W nuttig te leveren, verbruikt je lichaam er ruim 2 kW, en de rest wordt warmte. Daarom krijg je het warm op de trap. De beweging van armen en benen en de schokken bij elke stap tellen hier niet mee, dus je echte vermogen is nog iets groter.

UITGEWERKT VOORBEELD · 9.14

Habitat: water naar een hoger reservoir pompen

14/19

De pomp van de waterkringloop brengt de voedingsoplossing uit het reservoir van 60 L naar de bovenste kweekgoot, 1,5 m hoger, van waar ze door de goten terugstroomt. De pomp neemt 12 W op (12 V, 1,0 A) en verplaatst 5,0 L per minuut. Hoeveel zwaarte-energie krijgt het water als de pomp de hele 60 L één keer rondpompt? Hoe lang duurt dat, hoeveel elektrische energie kost het, en wat is het rendement?

De pomp duwt het water 1,5 m hoog; de zwaartekracht brengt het door de goten terug.

GEGEVEN

- $V = 60\,\text{L}$, dus $m = 60\,\text{kg}$ (de oplossing is bijna zuiver water, 1,00 kg per liter)
- $h = 1{,}5\,\text{m}$
- $P_{\text{el}} = 12\,\text{W}$, debiet 5,0 L/min

GEVRAAGD

- E_p, pomptijd, E_{el}, η

OPLOSSING

1. Zwaarte-energie: $E_p = m\,g\,h = 60 \cdot 9{,}81 \cdot 1{,}5 = 883\,\text{J}$.
2. Pomptijd: $60\,\text{L} / 5{,}0\,\text{L/min} = 12\,\text{min} = 720\,\text{s}$.
3. Elektrische energie: $E_{\text{el}} = P \cdot t = 12\,\text{W} \cdot 720\,\text{s} = 8640\,\text{J} = 2{,}4\,\text{Wh}$.
4. Rendement: $\eta = \dfrac{883}{8640} = 0{,}102$, ongeveer 10 %. Nuttig vermogen: $883/720 = 1{,}2\,\text{W}$.

ANTWOORD

Het water krijgt 883 J zwaarte-energie; de pomp doet er 12 minuten over en verbruikt 8,6 kJ (2,4 Wh). Het rendement is ongeveer 10 %.

Klopt dit? Die 883 J blijft niet: het water stroomt terug naar het reservoir en zijn zwaarte-energie wordt onderweg warmte (wrijving in de goten, klateren in het reservoir). In een kringloop moet de pomp dus voortdurend energie blijven leveren. Hoeveel dat per dag is, hangt af van hoe lang de pomp draait (opdracht 9.13). Voor de keuze van de pomp en de leidingen komt de wrijving in de leidingen erbij (hoofdstuk 11).

UITGEWERKT VOORBEELD · 9.15

Water als batterij?

15/19

Zou Habitat energie kunnen opslaan door water omhoog te pompen, en ze 's nachts terug te winnen met een turbine? Vergelijk met een batterij. Reken voorlopig met een batterij van 12 V en 100 Ah (hoofdstuk 16 en 20 kiezen de echte).

GEGEVEN

- batterij: $12\,\mathrm{V} \cdot 100\,\mathrm{Ah} = 1200\,\mathrm{Wh}$
- reservoir: 60 kg water, 1,5 m hoger: 883 J

GEVRAAGD

- hoeveel water op 1,5 m evenveel energie bevat als de batterij
- welk deel van de batterij één pompbeurt kost

OPLOSSING

1. Energie in de batterij: $1200\,\mathrm{Wh} \cdot 3600\,\mathrm{J/Wh} = 4{,}32\,\mathrm{MJ}$.
2. Massa water op 1,5 m met dezelfde energie: $m = \dfrac{E}{g\,h} = \dfrac{4{,}32 \cdot 10^6}{9{,}81 \cdot 1{,}5} = 294\,000\,\mathrm{kg}$, dus ongeveer $290\,\mathrm{m^3}$ water: het volume van een klein zwembad.
3. Anders bekeken: de batterij bevat evenveel energie als $\dfrac{4{,}32 \cdot 10^6}{883} \approx 4900$ volle reservoirs van 60 L op 1,5 m. Of: ze kan dat ene reservoir 7,3 km hoog tillen.
4. Eén keer 60 L rondpompen kost 2,4 Wh elektrisch: $2{,}4/1200 = 0{,}2\,\%$ van de batterij.

ANTWOORD

Water opslaan op 1,5 m hoogte is voor Habitat zinloos: voor de energie van één batterij (een lithiumbatterij van 1,2 kWh weegt ongeveer 10 à 15 kg) heb je bijna 300 ton water nodig. Omgekeerd kost het pompen weinig: één pompbeurt is 0,2 % van de batterij.

Klopt dit? Zwaarte-energie is 'dunne' energie: 1 kg op 1 m hoogte is maar 9,81 J. Wil je er echt energie mee opslaan, dan heb je enorme massa's én grote hoogteverschillen nodig. Dat is precies wat een pompcentrale doet (volgend blad).

VERHAAL · 9.16

Coo: een batterij van acht miljoen kubieke meter

16/19

In de Belgische Ardennen, bij Coo en Trois-Ponts, ligt de grootste energieopslag van België. Het is een **pompcentrale**, gebouwd tussen 1967 en 1979: twee kunstmatige meren op een heuvel, en een lager meer in de vallei van de Amblève, ongeveer 270 m lager. Samen kunnen de bovenbekkens ongeveer 8,5 miljoen kubieke meter water bevatten.

Als er veel goedkope stroom is (vroeger 's nachts, nu steeds vaker op zonnige, winderige dagen), pompen zes machines het water naar boven. Als er stroom tekort is, laten ze het terugstromen: dezelfde machines werken dan als turbines en generatoren. Samen leveren ze ruim 1 gigawatt, vijf à zes uur lang.

Reken na: $E_p = m\,g\,h = 8{,}5 \cdot 10^9\,\mathrm{kg} \cdot 9{,}81 \cdot 270\,\mathrm{m} = 2{,}3 \cdot 10^{13}\,\mathrm{J}$, ongeveer 6 GWh. Niet al dat water is bruikbaar en de omzetting kost energie; de centrale kan ongeveer 5 GWh leveren. Een volledige cyclus, oppompen en terug omzetten, heeft een rendement van ongeveer 75 %: voor elke 3 kWh die de centrale teruglevert, heeft ze er 4 verbruikt om te pompen.

Wat je uit dit verhaal meeneemt

Een pompcentrale werkt met precies dezelfde formule als het reservoir van Habitat. Het verschil zit in de getallen: 8,5 miljoen ton in plaats van 60 kg, 270 m in plaats van 1,5 m. Wie energie wil opslaan, moet eerst nagaan of de ordes van grootte kloppen. In 2021 werden de bovenbekkens uitgebreid met 600 000 m³: reken zelf na dat dat ruim 400 MWh extra is (opdracht 9.17).

In de machinezaal werkt dezelfde machine als pomp en als turbine, afhankelijk van de draairichting.

CODE · 9.17

Python: een energieboekhouding

17/19

Hoofdstuk 8 liet de batterijkar van de oprit rollen. Het programma rekent die beweging stap voor stap uit met de tweede wet, en houdt tegelijk drie energieposten bij: de kinetische energie, de zwaarte-energie boven de voet van de oprit, en de warmte die de rolweerstand ontwikkelt ($F_r \cdot \Delta s$ per stap). Als de natuurkunde klopt, moet de som constant blijven.

Python **energie.py** 30 regels

```python
# energie.py - de batterijkar rolt van de oprit: een energieboekhouding, stap voor stap
import math

G = 9.81
M = 80.0                          # kar met batterijen (kg)
ALFA = math.atan(0.0625)          # hellingshoek van een oprit van 6,25 % (radialen)
L = 6.41                          # lengte van de oprit (m)
H0 = 0.40                         # hoogte van de bovenkant (m)
MU_R = 0.025                      # rolweerstandscoefficient
DT = 0.001                        # tijdstap (s)

```

```
F_par = M * G * math.sin(ALFA)         # zwaartekracht langs de helling (N)
F_r = MU_R * M * G * math.cos(ALFA)    # rolweerstand (N)

s, v, warmte = 0.0, 0.0, 0.0           # weg langs de oprit (m), snelheid (m/s), wrijvingswarmte (J)
print(" t (s)   s (m)   Ek (J)   Ep (J)   warmte (J)   som (J)")
stap = 0
while s < L:
    if stap % 1000 == 0:                # elke seconde een regel
        Ek = 0.5 * M * v ** 2
        Ep = M * G * (H0 - s * math.sin(ALFA))
        print(f"{stap * DT:5.1f}   {s:5.2f}   {Ek:6.1f}   {Ep:6.1f}   {warmte:10.1f}   {Ek + Ep + warmte:6.1f}")
    a = (F_par - F_r) / M               # tweede wet langs de helling
    ds = v * DT + 0.5 * a * DT ** 2
    warmte = warmte + F_r * ds          # arbeid van de rolweerstand wordt warmte
    s = s + ds
    v = v + a * DT
    stap = stap + 1
Ek = 0.5 * M * v ** 2
print(f"beneden na {stap * DT:.2f} s: v = {v:.2f} m/s, Ek = {Ek:.1f} J, warmte = {warmte:.1f} J")
```

Uitvoer (gecontroleerd door het programma te draaien):

```
 t (s)   s (m)   Ek (J)   Ep (J)   warmte (J)   som (J)
  0.0    0.00      0.0    313.9          0.0    313.9
  1.0    0.18      5.4    304.9          3.6    313.9
  2.0    0.73     21.6    278.0         14.4    313.9
  3.0    1.65     48.5    233.0         32.4    313.9
  4.0    2.94     86.3    170.1         57.5    313.9
  5.0    4.59    134.8     89.2         89.9    313.9
beneden na 5.91 s: v = 2.17 m/s, Ek = 188.3 J, warmte = 125.6 J
```

De som blijft 313,9 J: de zwaarte-energie bovenaan ($80 \cdot 9{,}81 \cdot 0{,}40$). Beneden is 60 % bewegingsenergie geworden (188 J) en 40 % warmte (126 J). De kar komt aan met 2,17 m/s, zoals in opdracht 8.14. Zo'n controle, 'blijft de totale energie behouden?', gebruiken ingenieurs om een simulatie te testen: klopt de som niet, dan zit er een fout in het model of in de code.

NASLAG · 9.18

Overzicht: arbeid, energie en vermogen

18/19

arbeid

$$W = F \cdot s \cdot \cos\alpha$$

symbool	betekenis	eenheid
W	arbeid	J = N·m
F	kracht	N
s	verplaatsing	m
α	hoek tussen kracht en verplaatsing	°

veranderlijke kracht

$W =$ oppervlakte onder de (F, s)-grafiek

kinetische energie en arbeid-energiestelling

$$W_{\text{veer}} = \tfrac{1}{2} k\, s$$

$$E_k = \tfrac{1}{2} m\, v^2 \qquad W_{\text{res}} = \Delta E_k$$

zwaarte-energie

$$E_p = m\, g\, h \qquad W_{\text{zw}} = -\Delta E_p$$

symbool	betekenis	eenheid
h	hoogte boven het referentieniveau	m

behoud van energie (Q: wrijvingswarmte, $F_w \cdot s$)

$$\tfrac{1}{2} m\, v_1^2 + m\, g\, h_1 = \tfrac{1}{2} m\, v_2^2 + m\, g\, h_2\ (+\, Q)$$

vermogen en rendement

$$P = \frac{W}{t} \qquad P = F \cdot v \qquad \eta = \frac{P_{\text{nuttig}}}{P_{\text{toegevoerd}}}$$

symbool	betekenis	eenheid
P	vermogen	W = J/s
η	rendement	geen (of %)

eenheid	in joule
1 Wh (wattuur)	3600 J
1 kWh	3,6 MJ
1 kJ	1000 J
1 kcal (op voedingsverpakkingen)	4184 J

toestel of situatie	vermogen (orde van grootte)
schuifpaneel van het kasdak (mechanisch)	6 W
pomp van de waterkringloop (elektrisch)	12 W
mens, lang volgehouden (recreatief fietsen)	100 à 200 W
mens, enkele seconden (trap oplopen)	500 W à 1 kW
waterkoker	2 kW
gezinsauto (motor, maximaal)	50 à 150 kW
pompcentrale van Coo (alle turbines)	ruim 1 GW

Habitat: rekenwaarden uit dit hoofdstuk	
pomp waterkringloop	12 V, 1,0 A (12 W), 5,0 L/min bij 1,5 m opvoerhoogte, rendement ongeveer 10 %
60 L één keer rondpompen	883 J zwaarte-energie, 12 min, 2,4 Wh elektrisch
kasdak sluiten	35,5 J per sluiting, 6 W mechanisch
batterij (voorlopig)	1,2 kWh = 4,32 MJ

HET GEHEEL · 9.19

Waar energie terugkomt

19/19

Energie is de grootheid die alle leerlijnen verbindt: mechanisch, elektrisch, chemisch, warmte, licht. Dit hoofdstuk gaf haar de mechanische vorm, met kracht en weg. In de rest van Intermediate reken je ze om naar warmte, naar elektriciteit en naar de opbrengst van een zonnepaneel, en stel je voor de klimaatmodule een energiebudget op.

F15 Energie: vormen en omzettingen
Energievormen, behoud, rendement en het Sankey-diagram, nu met formules.

F21 Elektrisch vermogen en energie
Elektrisch vermogen P = U · I en energie in kWh: de pomp neemt 12 V · 1,0 A op.

I7 Beweging beschrijven
De formule zonder tijd, waaruit ½ m v² volgt; de remweg op twee manieren.

I8 De wetten van Newton
De krachten waarvan je hier de arbeid berekent, en de wrijving die energie in warmte omzet.

I11 Druk en hydraulica
Pompen en leidingen: opvoerhoogte, druk en de wrijving in de leidingen.

I16 Redox en batterijen
Wat er in een batterij gebeurt, en de capaciteit in Ah en Wh.

I18 Warmte berekenen
Waar de 'verloren' energie heen gaat: warmte en temperatuurstijging.

I20 Zonne-energie dimensioneren
Het energiebudget van de klimaatmodule, met de pomp als verbruiker.

E3 Integralen: ophopen
Arbeid bij een veranderlijke kracht als integraal van F over s.

E15 De hoofdwetten van de thermodynamica

Waarom je warmte nooit volledig terug in arbeid omzet: de tweede hoofdwet.

Habitat

Voor Habitat leggen we vast: de pomp van de waterkringloop is een 12 V-dompelpomp die 12 W opneemt en 5,0 L per minuut 1,5 m hoog pompt (rendement ongeveer 10 %). De 60 L van het voedingsreservoir één keer rondpompen kost 2,4 Wh. Het kasdak sluiten kost 35,5 J mechanische arbeid. Als voorlopige batterij rekenen we met 1,2 kWh. In opdracht 9.13 maak je het energiebudget van de waterkringloop.

SAMENVATTING

In het kort

- Arbeid is kracht maal verplaatsing langs de kracht: $W = F \cdot s \cdot \cos\alpha$, in joule. Mee met de beweging positief, loodrecht nul, tegen de beweging negatief.
- Bij een veranderlijke kracht is de arbeid de oppervlakte onder de (F,s)-grafiek; een veer uitrekken kost $\frac{1}{2}ks^2$.
- Kinetische energie $E_k = \frac{1}{2}mv^2$ volgt uit $F = ma$ en de formule zonder tijd. De arbeid van de resulterende kracht is ΔE_k.
- Zwaarte-energie $E_p = mgh$ hangt alleen af van het hoogteverschil, niet van de weg: een oprit verkleint de kracht, niet de energie.
- Zonder wrijving blijft $E_k + E_p$ constant. Zo reken je snelheden en hoogtes uit zonder de tijd te kennen.
- Wrijving zet mechanische energie om in warmte, $Q = F_w \cdot s$. De totale energie blijft behouden.
- Vermogen is energie per seconde: $P = W/t = F \cdot v$, in watt. $1\,\mathrm{kWh} = 3{,}6\,\mathrm{MJ}$.
- Rendement is nuttig gedeeld door toegevoerd; de pomp van Habitat haalt ongeveer 10 %. Zwaarte-energie is 'dun': 60 L op 1,5 m is maar 883 J.

BEGRIPPEN

Wat je nu kent

arbeid
Energie die een kracht overdraagt langs een verplaatsing: $W = F \cdot s \cdot \cos\alpha$, in joule.

joule
Eenheid van energie en arbeid: $1\,\mathrm{J} = 1\,\mathrm{N}\cdot\mathrm{m}$, de arbeid van 1 N over 1 m.

kinetische energie
Energie van een bewegend voorwerp: $E_k = \frac{1}{2}\,m\,v^2$.

potentiële energie
Hier: zwaarte-energie, $E_p = m\,g\,h$, met h de hoogte boven een gekozen referentieniveau.

mechanische energie
De som van kinetische en potentiële energie; zonder wrijving blijft ze constant.

arbeid-energiestelling
De arbeid van de resulterende kracht is gelijk aan de verandering van de kinetische energie.

vermogen
Energie (of arbeid) per tijdseenheid: $P = W/t$, in watt ($1\,\mathrm{W} = 1\,\mathrm{J/s}$); ook $P = F \cdot v$.

kilowattuur
Energie die een vermogen van 1 kW in 1 uur levert: $1\,\mathrm{kWh} = 3{,}6\,\mathrm{MJ}$.

rendement

opvoerhoogte

Het deel van de toegevoerde energie dat nuttig gebruikt wordt: $\eta = E_{\text{nuttig}} / E_{\text{toegevoerd}}$.

Het hoogteverschil waarover een pomp een vloeistof omhoog brengt.

I INTERMEDIATE · DEEL I3 · MECHANICA

HANDBOEK · HOOFDSTUK 10

Moment en evenwicht

Waarom een lange sleutel helpt, waar een balk op steunt, en wanneer iets omvalt

Een moer loskrijgen, een wip in evenwicht brengen, een vloerbalk die een vol reservoir draagt, een meetkas die de storm moet doorstaan: telkens gaat het niet alleen om hoe groot een kracht is, maar ook om **waar** ze aangrijpt. In hoofdstuk 8 was een voorwerp een punt. Nu kan het draaien. Het **moment** van een kracht zegt hoe sterk ze doet draaien. Met twee evenwichtsvoorwaarden bereken je de krachten op de steunpunten van een balk, met het **zwaartepunt** van samengestelde vormen vind je waar het gewicht aangrijpt, en met de **kantelveiligheid** beoordeel je of iets blijft staan.

8× uitleg · 7× uitgewerkt voorbeeld · 2× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

K Krachten, beweging en constructies · W Wiskundig modelleren · D Ontwerpen en maken

Een torenkraan is een reusachtige wip: het contragewicht houdt de last in evenwicht.

Na dit hoofdstuk

- Je kan het moment van een kracht om een punt berekenen met $M = F \cdot d$, met de loodrechte arm, ook bij een schuine kracht (arm of ontbinden).
- Je kan de hefboomwet uitleggen als momentenevenwicht en uitbreiden naar meer dan twee krachten.
- Je kan de twee evenwichtsvoorwaarden $\sum F = 0$ en $\sum M = 0$ opschrijven en uitleggen waarom je ze allebei nodig hebt (koppel).
- Je kan een vrijlichaamsdiagram van een balk tekenen, steunpunten vervangen door reactiekrachten en die reactiekrachten berekenen.
- Je kan een verdeelde belasting vervangen door haar resultante.
- Je kan het zwaartepunt van een samengestelde vorm of een samengesteld voorwerp berekenen, ook met een gat, en het zwaartepunt experimenteel bepalen.
- Je kan bepalen of een voorwerp kantelt, de kantelhoek berekenen en de kantelveiligheid van een ontwerp beoordelen.
- Je kan de steunpuntsreacties van een balk met Python berekenen voor veel lastgevallen tegelijk.

UITLEG · 10.1

Wat je al weet, en wat nu komt

1/22

In F11 leerde je dat een kracht een voorwerp kan doen **draaien**, en dat een kracht een aangrijpingspunt heeft. In F12 vond je de hefboomwet, en in F13 zag je dat iets omvalt als zijn zwaartepunt niet meer boven het steunvlak ligt. In hoofdstuk 8 tekende je vrijlichaamsdiagrammen en stelde je de som van de krachten gelijk aan nul voor een voorwerp in rust.

begrip	formule of regel	waar
kracht als pijl: grootte, richting, aangrijpingspunt	$\vec{F}$ in newton (N)	F11
hefboom, draaipunt, arm	hefboomwet $F_1 \cdot a_1 = F_2 \cdot a_2$	F12
zwaartepunt, steunvlak, kantelen	staat zolang het zwaartepunt boven het steunvlak blijft	F13
krachten ontbinden	$F_x = F\cos\alpha$, $F_y = F\sin\alpha$	hoofdstuk 6
evenwicht van krachten, vrijlichaamsdiagram	$\sum F_x = 0$, $\sum F_y = 0$	hoofdstuk 8

Maar in hoofdstuk 8 tekende je alle krachten vanuit één punt: het voorwerp kon alleen schuiven, niet draaien. Een ladder, een balk of een kraan kan dat wel. Daarom telt nu ook **waar** een kracht aangrijpt. In dit hoofdstuk

- reken je het **moment** van een kracht uit, ook als ze schuin staat;
- leer je de twee **evenwichtsvoorwaarden** en pas je ze toe op een wip, een ladder en een balk;
- bereken je de **steunpuntsreacties** van de vloerbalk onder het reservoir van Habitat;
- zoek je het **zwaartepunt** van samengestelde vormen, met rekenwerk en met een schietlood;
- beoordeel je of iets **kantelt**: een blok, een kast, de meetkas in de storm, een kraan.

VERHAAL · 10.2

Een toren die al achthonderd jaar valt

2/22

In 1173 begonnen bouwers in Pisa aan een klokkentoren naast de kathedraal. Na drie verdiepingen, rond 1178, helde hij al over: de toren staat op een ondiepe fundering van zowat 3 m in zachte klei en zand. De bouw lag daarna bijna honderd jaar stil. Toen de bouwers verder gingen, maakten ze de hogere verdiepingen aan de lage kant iets hoger om de scheefstand te verbergen: de toren is daardoor licht gebogen, als een banaan. Pas in de veertiende eeuw was hij af: ongeveer 56 m hoog en naar schatting 14 500 ton zwaar.

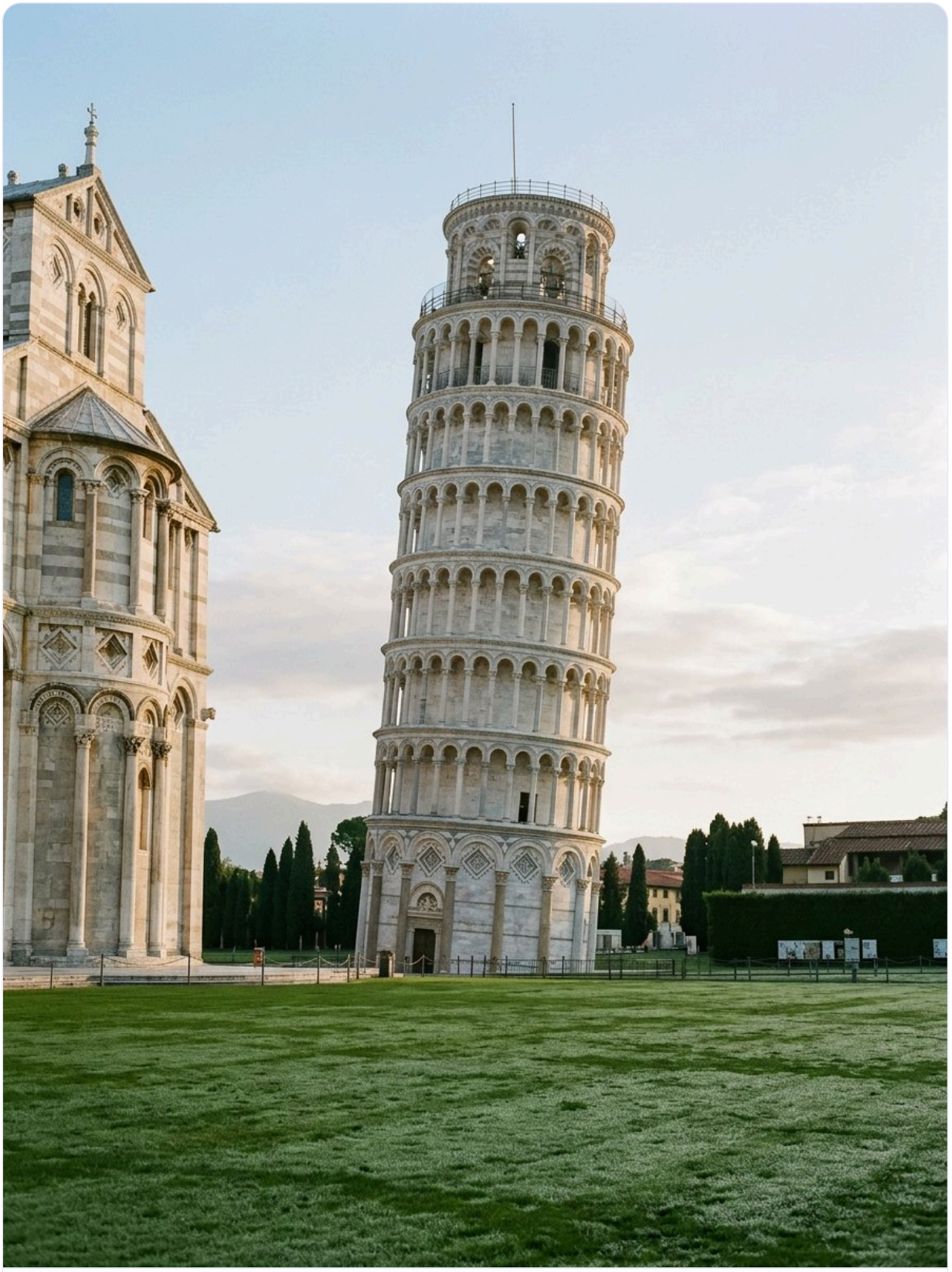
In 1990 helde de toren 5,5° en werd hij voor bezoekers gesloten. Waarom viel hij niet gewoon om? Zolang de verticale lijn door het **zwaartepunt** binnen de voet valt, duwt het gewicht de toren terug (F13). De voet is 15,5 m breed, en de lijn door het zwaartepunt viel nog ruim binnen de rand. Het gevaar zat in de grond: hoe schever de toren, hoe harder hij op de zuidkant drukt; die zachte grond geeft mee, en de toren helt nog meer. Een kringloop die vanzelf versnelt.

De redding was een les in momenten. Eerst legden ingenieurs honderden ton lood op de **noordkant**, de hoge kant: een gewicht met een arm tot het midden van de toren levert een **tegenmoment**. Daarna haalden ze met schuine boringen voorzichtig grond weg onder de noordkant, in totaal zowat 70 ton,

zodat die kant een beetje zakte. In 2001 ging de toren terug open. Hij helt nu ongeveer 4°, en de top staat zo'n 45 cm dichter bij het lood dan in 1990.

Wat je uit dit verhaal meeneemt

Of iets omvalt, hangt af van twee dingen: waar het zwaartepunt ligt ten opzichte van het steunvlak, en of de ondergrond de krachten kan dragen. Het eerste is dit hoofdstuk, het tweede komt terug in hoofdstuk 11 (druk op de grond).



De toren helt, maar de lijn door zijn zwaartepunt valt nog binnen zijn voet.

UITLEG · 10.3

Het moment van een kracht

3/22

Wie een moer probeert los te draaien, weet het uit ervaring: duw op het uiteinde van de sleutel, loodrecht, en het lukt het best. Hoe sterk een kracht een voorwerp rond een punt O doet draaien, heet het **moment** van die kracht ten opzichte van O:

MOMENT VAN EEN KRACHT

$$M = F \cdot d$$

F in N; d de **arm**: de loodrechte afstand van het draaipunt O tot de **werklijn** van de kracht (de lijn waarop de kracht ligt), in m. M in newtonmeter (N·m).

De figuur toont drie keer dezelfde kracht op dezelfde plaats. Loodrecht op de sleutel is de arm de hele lengte r van de sleutel. Schuin is de arm kleiner: $d = r \cdot \sin\theta$, met θ de hoek tussen kracht en sleutel. Langs de sleutel gaat de werklijn door O: de arm is nul en er gebeurt niets, hoe hard je ook trekt.

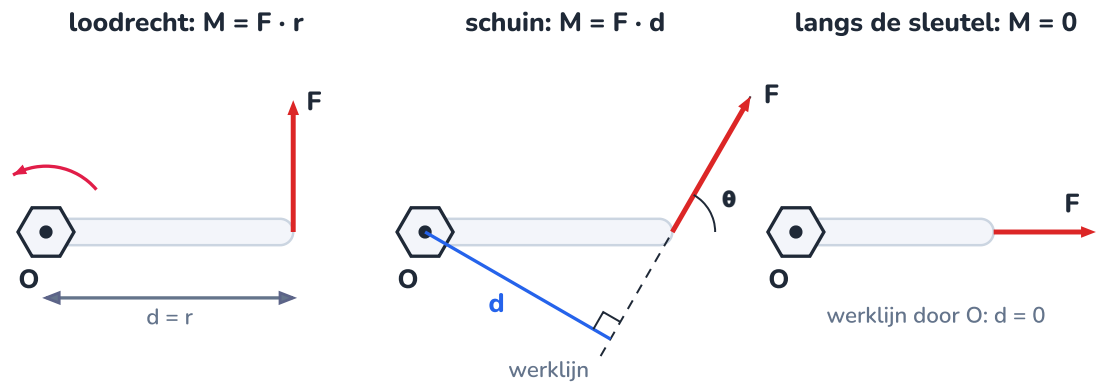
De draaizin telt. Een kracht die tegen de klok in doet draaien, geeft een positief moment; met de klok mee een negatief (dezelfde afspraak als voor hoeken in hoofdstuk 6). Werken er meerdere krachten, dan tel je hun momenten op met hun teken.

De arm is de afstand tot het aangrijpingspunt

Alleen als de kracht loodrecht staat op de lijn naar het draaipunt. In het algemeen is de arm de *loodrechte* afstand tot de werklijn. Een deur die je vlak naast de scharnieren openduwt, of schuin naar de scharnieren toe, gaat moeilijk open, ook al is je kracht groot.

Newtonmeter is geen joule

Ook arbeid (hoofdstuk 9) is kracht maal afstand, maar daar liggen kracht en verplaatsing *langs* elkaar. Bij een moment staan kracht en arm *loodrecht* op elkaar. Het zijn verschillende grootheden; een moment schrijf je daarom altijd in N·m, nooit in J. Het 'koppel' van een motor (hoofdstuk 26) is zo'n moment: een motor met een koppel van 2 N·m draait zijn as alsof je met 20 N aan een sleutel van 10 cm trekt.



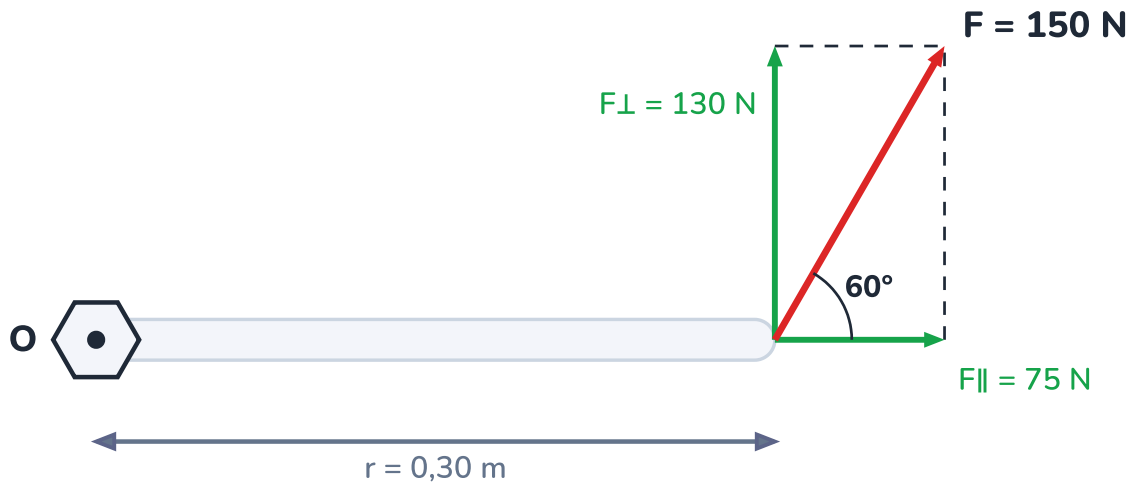
Dezelfde kracht, drie armen: de hele sleutel, een deel ervan (d = r sin θ), en nul als de werklijn door O gaat.

Een momentsleutel klikt zodra het ingestelde moment bereikt is: niet te los, niet te vast.

UITGEWERKT VOORBEELD · 10.4

Een moer aandraaien met een schuine kracht

Je trekt met 150 N aan een moersleutel, 0,30 m van het midden van de moer, maar niet loodrecht: je kracht maakt een hoek van 60° met de sleutel. Hoe groot is het moment? De fabrikant schrijft voor deze bout een aanhaalmoment van 45 N·m voor. Welke kracht moet je dan uitoefenen, onder 60° en loodrecht?



Alleen F⊥ doet draaien; F∥ trekt langs de sleutel en duwt tegen de moer.

GEGEVEN

- $F = 150\,\text{N}$, $r = 0{,}30\,\text{m}$, $\theta = 60°$
- gevraagd moment $45\,\text{N}\cdot\text{m}$

GEVRAAGD

- M
- F voor $45\,\text{N}\cdot\text{m}$ onder 60° en loodrecht

OPLOSSING

1. **Methode 1, met de arm.** De loodrechte afstand van O tot de werklijn is $d = r\sin\theta = 0{,}30\cdot\sin 60° = 0{,}260\,\text{m}$. Dus $M = 150\cdot 0{,}260 = 39{,}0\,\text{N}\cdot\text{m}$.
2. **Methode 2, door te ontbinden** (hoofdstuk 6). Loodrecht op de sleutel: $F_\perp = 150\sin 60° = 129{,}9\,\text{N}$, met arm 0,30 m: $129{,}9\cdot 0{,}30 = 39{,}0\,\text{N}\cdot\text{m}$. Langs de sleutel: $F_\parallel = 150\cos 60° = 75\,\text{N}$; die werklijn gaat door O en geeft geen moment.
3. Voor 45 N·m onder 60°: $F = \dfrac{M}{d} = \dfrac{45}{0{,}260} = 173\,\text{N}$.
4. Loodrecht: $F = \dfrac{45}{0{,}30} = 150\,\text{N}$.

ANTWOORD

Het moment is 39,0 N·m: te weinig. Onder 60° moet je met 173 N trekken; loodrecht volstaat 150 N.

Klopt dit? De twee methoden geven hetzelfde, en dat is geen toeval: $F\cdot(r\sin\theta) = (F\sin\theta)\cdot r$. Kies de methode die het eenvoudigst is voor de figuur. En de uitersten kloppen: bij 90° is $\sin\theta = 1$ (het grootste moment), bij 0° is het moment nul.

UITLEG · 10.5

Van hefboomwet naar momentenevenwicht

De hefboomwet uit F12, $F_1 \cdot a_1 = F_2 \cdot a_2$, is een uitspraak over momenten. Links van het draaipunt probeert F_1 de hefboom tegen de klok in te draaien, rechts probeert F_2 hem met de klok mee te draaien. Zijn die twee momenten even groot, dan draait de hefboom niet: $F_1 a_1 - F_2 a_2 = 0$.

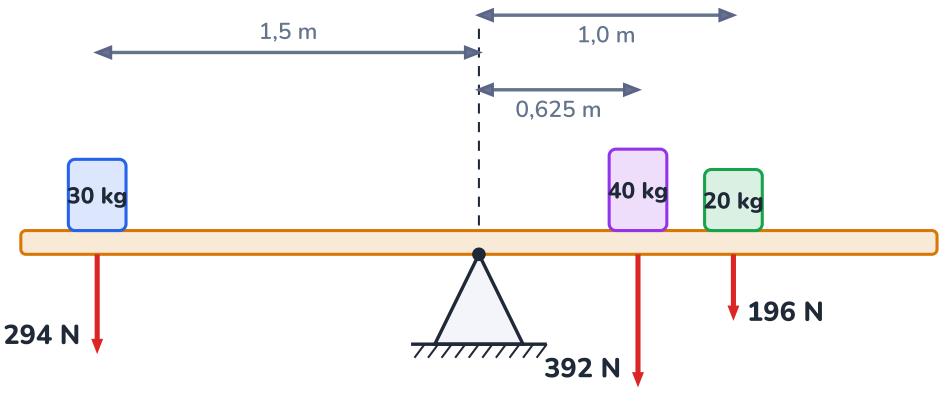
Zo geformuleerd werkt de wet voor elk aantal krachten: **een voorwerp draait niet als de som van alle momenten nul is**. Drie kinderen op een wip: Anna (30 kg) zit 1,5 m links van het draaipunt, Chloë (20 kg) zit 1,0 m rechts. Waar moet Bram (40 kg) gaan zitten? Tegen de klok in positief:

$$30\,g \cdot 1{,}5 - 20\,g \cdot 1{,}0 - 40\,g \cdot x = 0 \quad \Rightarrow \quad x = \frac{45 - 20}{40} = 0{,}625\,\text{m}$$

De valversnelling valt weg: je mag hier rekenen met kilogram in plaats van newton. Het draaipunt zelf duwt de plank omhoog met het totale gewicht, $90 \cdot 9{,}81 = 883\,\text{N}$: ook de som van de krachten moet nul zijn. Dat is de tweede voorwaarde voor evenwicht (volgend blad).

soort (F12)	wat zit in het midden?	voorbeelden
eerste soort	het draaipunt	wip, schaar, koevoet, tang
tweede soort	de last	kruiwagen, notenkraker, flesopener
derde soort	de kracht	pincet, je onderarm (biceps), bezem

In elke soort hefboom geldt hetzelfde momentenevenwicht. Bij de tweede en derde soort liggen kracht en last aan dezelfde kant van het draaipunt; ze draaien dan in tegengestelde zin omdat ze in tegengestelde richting werken.



Tegen de klok in: 294 N · 1,5 m = 441 N·m. Met de klok mee: 392 N · 0,625 m + 196 N · 1,0 m = 441 N·m.

UITLEG · 10.6

De twee evenwichtsvoorwaarden

In hoofdstuk 8 was een voorwerp in evenwicht als de resulterende kracht nul was. Dat is niet genoeg. Neem een plank en duw links met 60 N omhoog, rechts met 60 N omlaag. De som van de krachten is nul, de plank schuift niet weg, maar ze **draait**. Twee even grote, tegengestelde, evenwijdige krachten die niet op één lijn liggen, heten een **koppel**. Je draait zo aan een stuur of aan een waterkraan.

EVENWICHT VAN EEN VOORWERP IN EEN VLAK

$$\sum F_x = 0 \qquad \sum F_y = 0 \qquad \sum M = 0$$

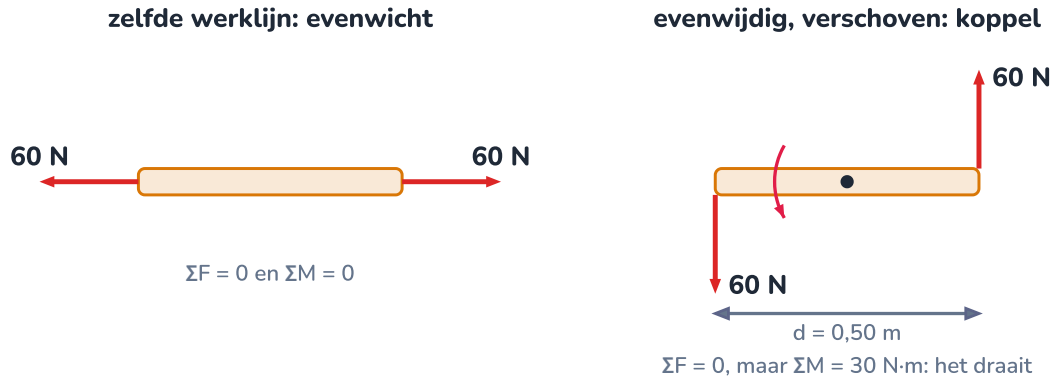
Een voorwerp is in evenwicht (het blijft in rust, schuift niet en draait niet) als de krachten elkaar opheffen in elke richting **en** de momenten elkaar opheffen.

Drie eigenschappen maken die voorwaarden tot een sterk gereedschap:

- **Het draaipunt mag je vrij kiezen.** Is een voorwerp in evenwicht, dan is de som van de momenten nul om élk punt, ook om een punt waar geen as zit. Kies het punt waar een onbekende kracht aangrijpt: haar arm is nul, en ze verdwijnt uit de vergelijking.
- **Drie vergelijkingen, drie onbekenden.** In een vlak heb je drie voorwaarden, dus kan je hoogstens drie onbekende krachten berekenen. Constructies waarvoor dat lukt, heten *statisch bepaald* (hoofdstuk E13).
- **Het moment van een koppel** is $F \cdot d$, met d de afstand tussen de twee werklijnen, en dat om elk punt. Een koppel kan je alleen opheffen met een ander koppel.

Stappenplan voor een evenwichtsprobleem

1. Teken het vrijlichaamsdiagram, nu met elke kracht op haar **echte** aangrijpingspunt. 2. Kies assen en een positieve draaizin. 3. Kies een momentpunt waar zoveel mogelijk onbekenden aangrijpen, en schrijf $\sum M = 0$. 4. Vul aan met $\sum F_x = 0$ en $\sum F_y = 0$. 5. Controleer met de momenten om een ander punt.



Links heffen de krachten elkaar volledig op. Rechts is de resulterende kracht nul, maar het koppel draait de plank.

PROEF · 10.7

Twee weegschalen onder een lat

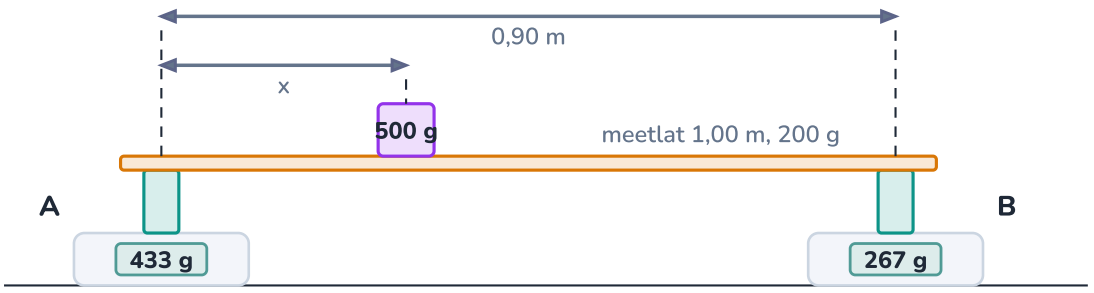
7/22

ONDERZOEKSVRAAG

Een lat ligt met haar uiteinden op twee weegschalen. Hoe verdeelt het gewicht van een massa zich over de twee steunpunten als je de massa langs de lat verschuift?

MATERIAAL

- meetlat van 1,00 m (200 g) of een lichte plank
- twee houten blokjes
- twee digitale keukenweegschalen
- massa van 500 g
- rolmeter



Met de massa op x = 30 cm draagt A 433 g en B 267 g.

WERKWIJZE

1. Zet de blokjes op de weegschalen, 0,90 m uit elkaar, en zet beide weegschalen op nul (tarra).
2. Leg de lat op de blokjes. Noteer wat elke weegschaal toont.
3. Zet de massa op $x = 0$ (recht boven blokje A) en noteer beide aflezingen.
4. Verschuif de massa telkens 15 cm naar B en noteer opnieuw, tot $x = 90\,\text{cm}$.
5. Zet beide aflezingen uit tegen x, en bereken telkens hun som.

WAARNEMING

x (cm)	0	15	30	45	60	75	90
weegschaal A (g)	601	517	432	351	266	184	99
weegschaal B (g)	99	184	268	349	434	516	600
som (g)	700	701	700	700	700	700	699

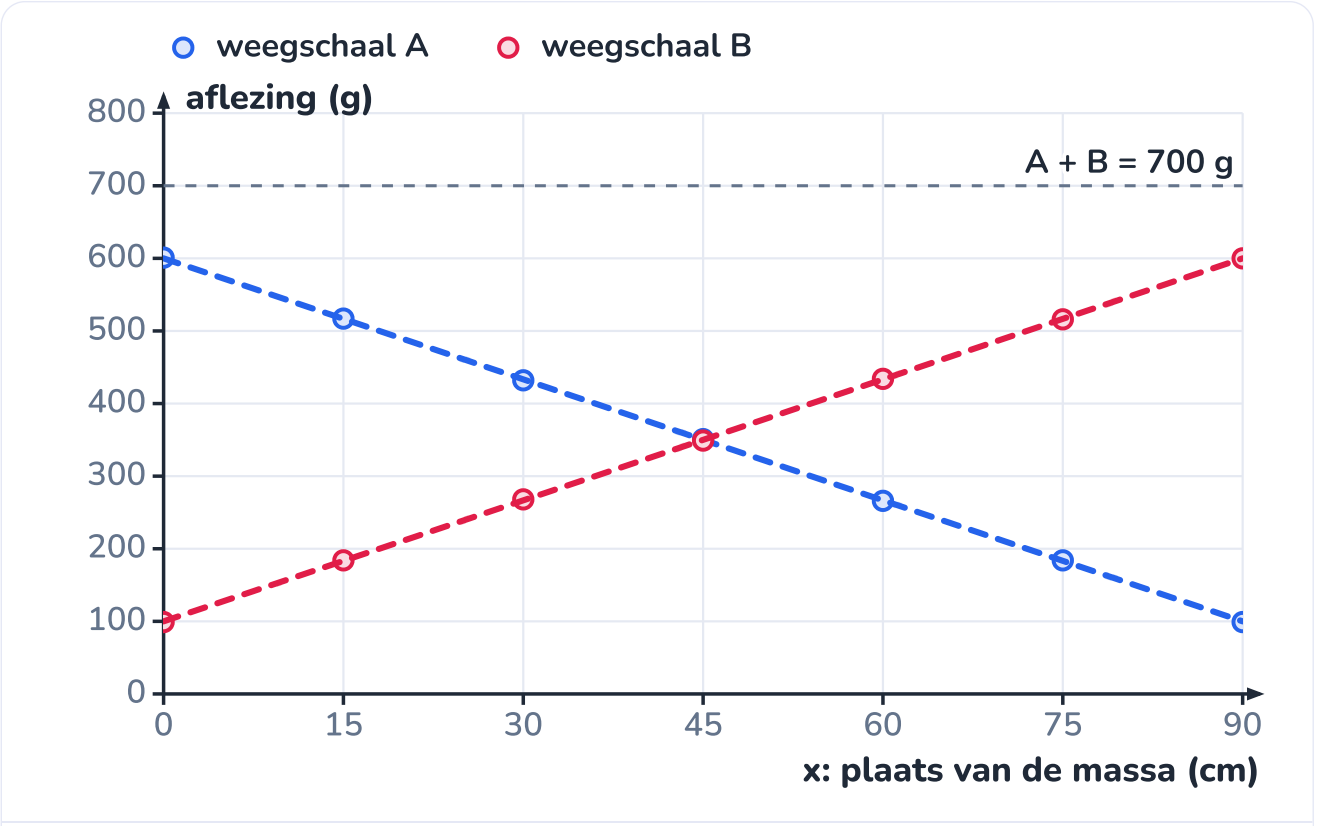
Zonder massa toont elke weegschaal 100 g. De aflezingen veranderen lineair met x, in tegengestelde zin; hun som blijft 700 g.

VERKLARING

Op de lat werken vier krachten: haar eigen gewicht (in het midden), het gewicht van de massa, en de twee steunkrachten R_A en R_B die de weegschalen tonen. Momenten om A (tegen de klok in positief, B rechts): $R_B \cdot 0{,}90 - m_{\text{lat}}\,g \cdot 0{,}45 - m\,g \cdot x = 0$, dus

$$R_B = \frac{0{,}200\,g \cdot 0{,}45 + 0{,}500\,g \cdot x}{0{,}90}$$

Een lineaire functie van x: in grammen $100 + 500 \cdot x/0{,}90$. De som van de krachten geeft $R_A = 700\,g - R_B$. Staat de massa boven A, dan draagt A ze helemaal; in het midden dragen A en B elk de helft. Een weegschaal toont massa, geen kracht; omdat g in elke term staat, mag je hier gewoon met gram rekenen. De afwijkingen van 1 à 2 g komen van de weegschalen (resolutie 1 g) en van het plaatsen van de massa.



Voorbeeldmetingen (punten) en het model uit het momentenevenwicht (stippellijnen).

UITLEG · 10.8

Steunpunten en steunpuntsreacties

8/22

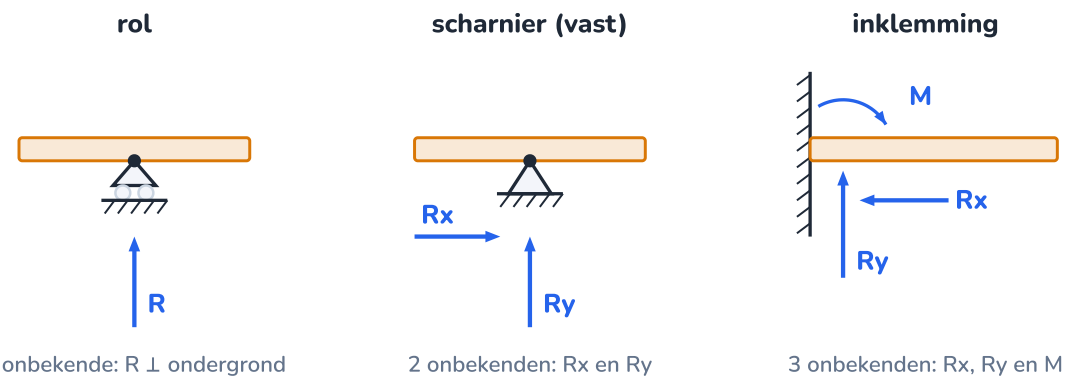
Een balk, een brug of een plank rust op **steunpunten** (opleggingen). In een vrijlichaamsdiagram laat je de steunpunten weg en teken je in de plaats de krachten die ze op de balk uitoefenen: de **steunpuntsreacties** of reactiekrachten. Welke reacties een steunpunt kan leveren, hangt af van hoe het de balk vasthoudt:

- Een **rol** (glijdende oplegging) laat de balk langs de ondergrond schuiven en draaien. Hij levert één kracht, loodrecht op de ondergrond.
- Een **scharnier** (vaste oplegging) houdt het punt vast, maar laat de balk draaien. Hij levert een kracht in elke richting: twee componenten R_x en R_y.
- Een **inklemming** (een balk die in een muur zit, een paal in de grond) houdt vast én belet het draaien: twee krachtcomponenten en een moment.

De klassieke balk heeft een scharnier aan één kant en een rol aan de andere. Dat zijn precies drie onbekenden, genoeg voor de drie evenwichtsvoorwaarden. De rol is ook praktisch: een balk zet uit als hij warm wordt, en de rol laat hem schuiven. Zo is het kasframe van de meetkas gemaakt: in het midden vast, met sleufgaten aan de uiteinden (F6). Bruggen hebben om dezelfde reden aan één kant rollagers.

Alleen verticale lasten? Dan is Rx nul

Werken alle lasten verticaal, dan zegt $\sum F_x = 0$ dat de horizontale reactie van het scharnier nul is. Er blijven twee onbekenden over, R_A en R_B: één momentenvergelijking en één krachtenvergelijking volstaan.

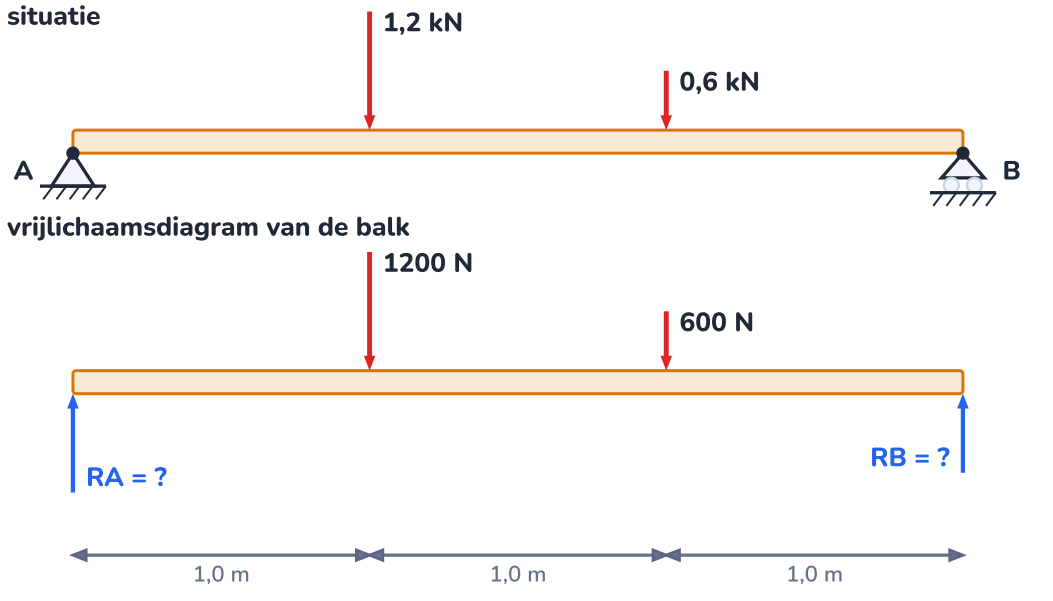


Symbool en reacties: een rol levert één kracht, een scharnier twee componenten, een inklemming ook nog een moment.

UITGEWERKT VOORBEELD · 10.9

Een balk op twee steunpunten

Een balk van 3,0 m ligt met een scharnier in A en een rol in B. Op 1,0 m van A hangt een last van 1,2 kN, op 2,0 m van A een last van 0,6 kN. Het eigen gewicht van de balk verwaarloos je. Bereken de steunpuntsreacties.



Boven de situatie, onder het vrijlichaamsdiagram: de steunpunten zijn vervangen door RA en RB (pijlen op schaal).

GEGEVEN

- $L = 3{,}0\,\text{m}$
- $F_1 = 1200\,\text{N}$ op $1{,}0\,\text{m}$, $F_2 = 600\,\text{N}$ op $2{,}0\,\text{m}$ van A

GEVRAAGD

- R_A en R_B

OPLOSSING

1. Vrijlichaamsdiagram: de twee lasten omlaag, R_A en R_B omhoog. Alle krachten zijn verticaal, dus de horizontale reactie in A is nul.

2. Momenten om A (dan valt R_A weg; tegen de klok in positief): $R_B \cdot 3{,}0 - 1200 \cdot 1{,}0 - 600 \cdot 2{,}0 = 0$, dus $R_B = \dfrac{1200 + 1200}{3{,}0} = 800\,\text{N}$.

3 Som van de verticale krachten: $R_A + R_B - 1200 - 600 = 0$, dus $R_A = 1800 - 800 = 1000\,\text{N}$.

4 Controle met momenten om B: $-R_A \cdot 3{,}0 + 1200 \cdot 2{,}0 + 600 \cdot 1{,}0 = -3000 + 2400 + 600 = 0$. Klopt.

ANTWOORD

$R_A = 1000\,\text{N}$ **en** $R_B = 800\,\text{N}$**. Samen dragen ze de 1800 N van de lasten.**

Klopt dit? De grootste last hangt dichter bij A, dus A moet het meest dragen: dat klopt. Voor één last F op afstand a van A en b van B vind je op dezelfde manier $R_A = F \cdot b/L$ en $R_B = F \cdot a/L$: het steunpunt het dichtst bij de last draagt het meest. Bij meerdere lasten tel je de bijdragen op.

UITLEG · 10.10

Een verdeelde belasting vervangen door haar resultante

10/22

Niet elke last grijpt aan in één punt. Het eigen gewicht van een balk, een rij boeken op een plank, sneeuw op een dak of het water in een reservoir zijn over een lengte **verdeeld**. Zo'n last geef je aan met q: de kracht per meter, in N/m.

RESULTANTE VAN EEN GELIJKMATIG VERDEELDE BELASTING

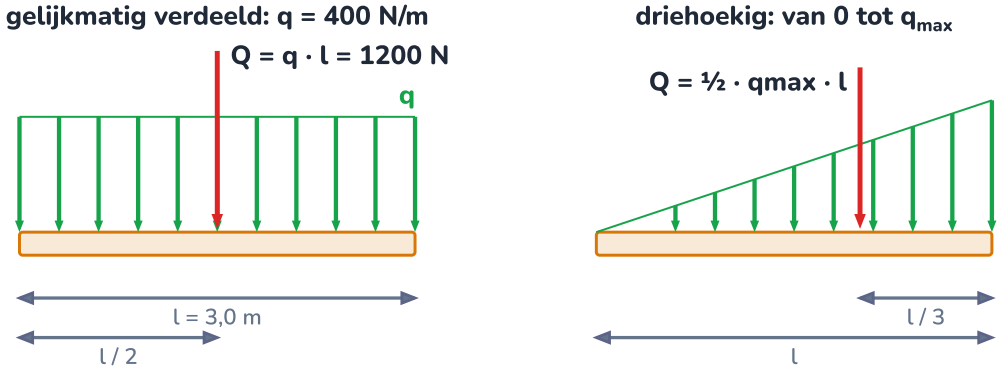
$$Q = q \cdot l \quad \text{in het midden van } l$$

Voor het evenwicht van de hele balk mag je een verdeelde last vervangen door één kracht Q die even groot is als de totale last, en die aangrijpt in het zwaartepunt van het 'lastvlak': bij een gelijkmatige last in het midden.

Waarom mag dat? Denk de last opgedeeld in veel kleine stukjes. Hun krachten tellen op tot $q \cdot l$, en hun momenten tellen op tot $q \cdot l$ maal de afstand tot het midden, net zoals bij het zwaartepunt verderop in dit hoofdstuk. Een **driehoekige** last, die gelijkmatig groeit van 0 tot $q_{\max}$, heeft als resultante de oppervlakte van de driehoek, $\frac{1}{2}\, q_{\max}\, l$, op een derde van de lengte aan de zware kant. Die vorm zie je terug bij de waterdruk op de wand van een reservoir (hoofdstuk 11).

Alleen voor het evenwicht van het geheel

De resultante vertelt je hoeveel elk steunpunt draagt. Ze zegt niets over wat er *in* de balk gebeurt: een balk met een gelijkmatige last buigt anders door dan een balk met dezelfde last in het midden. Hoe je de spanningen en de doorbuiging berekent, leer je in hoofdstuk E12.

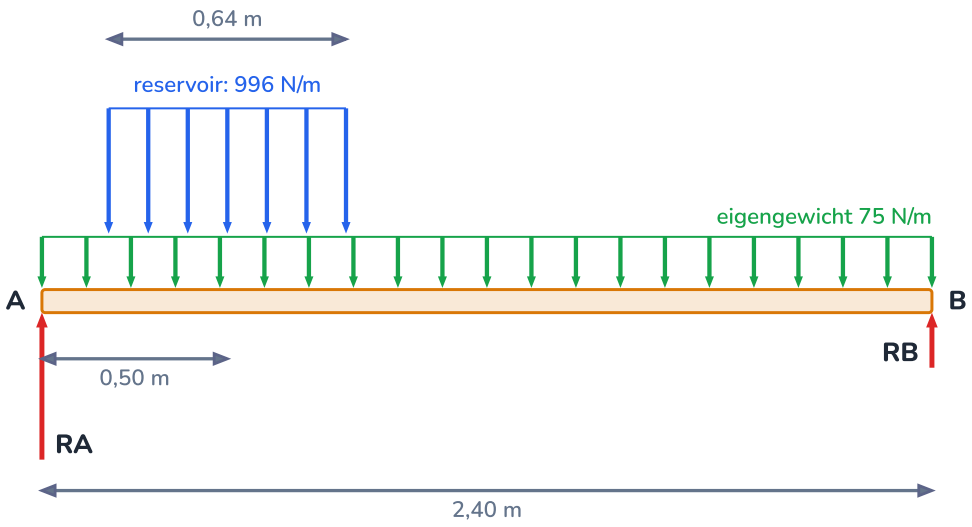


Links: de resultante ligt in het midden. Rechts: ze ligt op een derde van de lengte, aan de kant van de grootste last.

UITGEWERKT VOORBEELD · 10.11

Habitat: de draagbalk onder het voedingsreservoir

De vloer van de klimaatmodule ligt 40 cm boven het terrein (hoofdstuk 6) en rust op vloerbalken van vuren hout (45 × 145 mm), om de 40 cm, die 2,40 m overspannen tussen twee liggers A en B. Het voedingsreservoir (60 L oplossing, samen met de bak, het deksel en de pomp 65 kg) staat met zijn lange zijde (0,64 m) langs een vloerbalk, met zijn midden op 0,50 m van ligger A. Balk en vloerstrook wegen samen 75 N per meter. Welke krachten moeten de liggers opvangen?



Het vrijlichaamsdiagram: twee verdeelde lasten en twee reactiekrachten (hoogtes van de lastpijlen niet op schaal).

GEGEVEN

- $L = 2{,}40\,\text{m}$; eigengewicht $q = 75\,\text{N/m}$ over de hele balk
- reservoir: $65\,\text{kg}$ over $0{,}64\,\text{m}$, midden op $0{,}50\,\text{m}$ van A
- aan de veilige kant: één balk draagt het hele reservoir

GEVRAAGD

- R_A en R_B

OPLOSSING

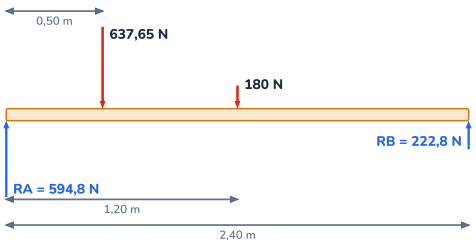
1. **Lasten.** Reservoir: $F = 65 \cdot 9{,}81 = 637{,}65\,\text{N}$, verdeeld over 0,64 m, dus $q_{\text{res}} = 637{,}65/0{,}64 = 996\,\text{N/m}$. Eigengewicht: $Q = 75 \cdot 2{,}40 = 180\,\text{N}$.
2. **Resultanten.** De reservoirlast grijpt aan in het midden van het reservoir, op 0,50 m van A. Het eigengewicht in het midden van de balk, op 1,20 m van A.
3. **Momenten om A:** $R_B \cdot 2{,}40 - 637{,}65 \cdot 0{,}50 - 180 \cdot 1{,}20 = 0$, dus $R_B = \dfrac{318{,}8 + 216}{2{,}40} = 222{,}8\,\text{N}$.
4. **Som van de krachten:** $R_A = 637{,}65 + 180 - 222{,}8 = 594{,}8\,\text{N}$.
5. **Controle** met momenten om B: $R_A = \dfrac{637{,}65 \cdot 1{,}90 + 180 \cdot 1{,}20}{2{,}40} = \dfrac{1211{,}5 + 216}{2{,}40} = 594{,}8\,\text{N}$.

ANTWOORD

Ligger A moet 595 N opvangen, ligger B 223 N. A draagt het meest, omdat het reservoir dicht bij A staat.

Klopt dit? Of de liggers en de metalen balkdragers dat aankunnen, lees je in hun technische fiche; in opdracht 10.14 zet je er nog iemand bij die het reservoir bijvult. Of de balk zelf sterk genoeg is en hoeveel hij doorbuigt, hangt af van wat er binnenin gebeurt: dat is

hoofdstuk E12. 'Aan de veilige kant' betekent: in werkelijkheid verdelen de vloerplaten het gewicht over twee of drie balken, dus elke balk krijgt minder dan we rekenden.



Vervangen door resultanten: 637,65 N op 0,50 m en 180 N op 1,20 m (pijlen op schaal).

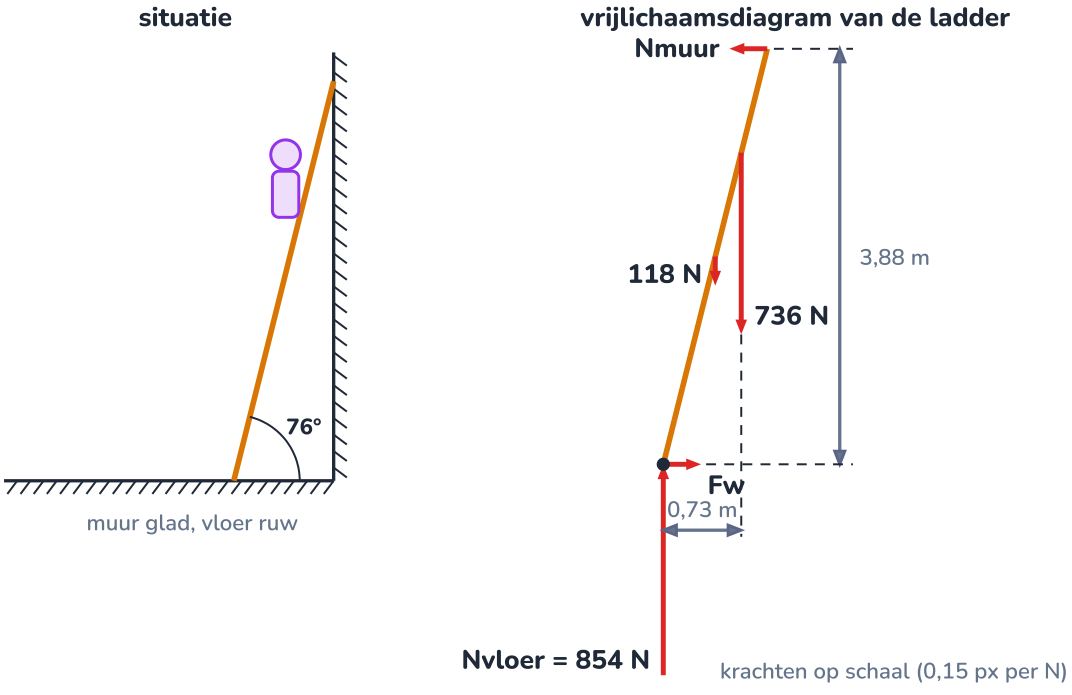
Onder de vloer: vloerbalken op liggers. Het volle reservoir drukt vooral op de balk eronder.

UITGEWERKT VOORBEELD · 10.12

Een ladder tegen een gladde muur

12/22

In hoofdstuk 6 zette je een ladder volgens de 1-op-4-regel: onder 76° met de grond. Waarom schuift een ladder weg als hij te schuin staat? Een ladder van 4,0 m en 12 kg staat tegen een gladde muur (geen wrijving). Een persoon van 75 kg staat op 3,0 m langs de ladder. Hoe groot moet de wrijvingskracht aan de voet zijn? De wrijvingscoëfficiënt tussen voet en vloer is $\mu_s = 0{,}40$. Staat de ladder veilig? En als hij onder 60° stond?



Links de situatie, rechts de ladder los van muur en vloer, met alle krachten op hun aangrijpingspunt.

GEGEVEN

- $L = 4{,}0\,\text{m}$, $\alpha = 76{,}0°$ ($\tan\alpha = 4$)

GEVRAAGD

- N_{muur}, N_{vloer}, F_w en of de ladder blijft staan

- ladder 12 kg (zwaartepunt in het midden), persoon 75 kg op 3,0 m
- muur glad; $\mu_s = 0{,}40$ aan de voet

OPLOSSING

1 **Vrijlichaamsdiagram.** Twee gewichten omlaag (118 N in het midden, 736 N op 3,0 m). Aan de top alleen een normaalkracht van de muur, horizontaal (de muur is glad). Aan de voet een normaalkracht omhoog en een wrijvingskracht naar de muur toe. Drie onbekenden.

2 **Verticaal:** $N_{\text{vloer}} = (12 + 75) \cdot 9{,}81 = 853{,}5\,\text{N}$.

3 **Momenten om de voet** (daar grijpen twee onbekenden aan). De arm van een verticale kracht is haar horizontale afstand tot de voet, $s\cos\alpha$; de arm van de horizontale muurkracht is de hoogte van de top, $L\sin\alpha = 3{,}88\,\text{m}$: $N_{\text{muur}} \cdot 3{,}88 = 117{,}7 \cdot 2{,}0\cos 76° + 735{,}8 \cdot 3{,}0\cos 76°$, dus $N_{\text{muur}} = \dfrac{57{,}1 + 535{,}4}{3{,}88} = 152{,}7\,\text{N}$.

4 **Horizontaal:** $F_w = N_{\text{muur}} = 152{,}7\,\text{N}$. Nodig is dus $\mu \geq \dfrac{152{,}7}{853{,}5} = 0{,}18$.

5 Beschikbaar is $0{,}40 \cdot 853{,}5 = 341\,\text{N}$: ruim genoeg. Onder 60° wordt de arm van de gewichten groter en die van de muurkracht kleiner: met de persoon helemaal bovenaan is dan $F_w = 459\,\text{N}$ nodig, meer dan 341 N. Onder 60° schuift de ladder weg zodra de persoon hoger komt dan 2,89 m.

ANTWOORD

Onder 76° is 153 N wrijving nodig, en de vloer kan er 341 N leveren: de ladder staat veilig, zelfs met de persoon bovenaan (199 N). Onder 60° glijdt de voet weg voor de persoon de top bereikt.

Klopt dit? Hoe hoger je klimt, hoe groter de arm van je gewicht en hoe meer wrijving nodig is: ladders schuiven daarom meestal weg als iemand bovenaan staat, niet bij de eerste sport. Een steile ladder verkleint die arm; te steil kiept hij achterover. Vandaar de 75 à 76° uit hoofdstuk 6.

UITLEG · 10.13

Het zwaartepunt

13/22

Elk stukje van een voorwerp heeft gewicht. Voor het evenwicht mag je al die kleine krachten vervangen door één kracht, het totale gewicht, die aangrijpt in het **zwaartepunt** Z. Hang je een voorwerp op aan een punt, dan draait het tot het zwaartepunt recht onder dat punt hangt; ondersteun je het in het zwaartepunt, dan blijft het in elke stand in evenwicht.

Voor symmetrische vormen ligt Z op de symmetrie-as: in het midden van een rechthoek, een cirkel of een balk. Een **samengestelde vorm** deel je op in stukken waarvan je het zwaartepunt kent. Het moment van het totale gewicht moet gelijk zijn aan de som van de momenten van de stukken. Voor platen van gelijke dikte is het gewicht evenredig met de oppervlakte, dus

ZWAARTEPUNT VAN EEN SAMENGESTELDE VLAKKE FIGUUR

$$x_Z = \frac{\sum A_i\, x_i}{\sum A_i} \qquad y_Z = \frac{\sum A_i\, y_i}{\sum A_i}$$

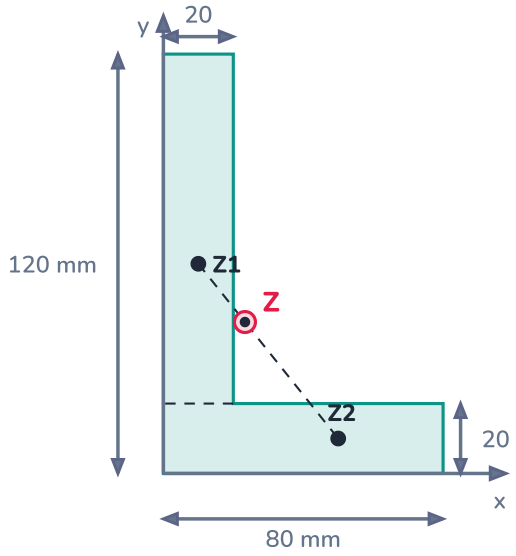
A_i de oppervlakte en $(x_i;\, y_i)$ het zwaartepunt van stuk i. Voor een samengesteld voorwerp vervang je A_i door de massa m_i. Een **gat** tel je mee met een negatieve oppervlakte.

vorm	oppervlakte	zwaartepunt
rechthoek $b \times h$	$b \cdot h$	in het midden: $b/2$ en $h/2$

vorm	oppervlakte	zwaartepunt
driehoek met basis b en hoogte h	$\frac{1}{2}\,b \cdot h$	op $h/3$ boven de basis (snijpunt van de zwaartelijnen)
cirkel met straal r	πr^2	in het middelpunt
halve cirkel met straal r	$\frac{1}{2}\,\pi r^2$	op $\frac{4r}{3\pi} \approx 0{,}42\,r$ van de rechte rand

Het zwaartepunt ligt altijd in het voorwerp

Bij een L-vorm, een ring, een boemerang of een hoefijzer ligt het zwaartepunt in de lege ruimte. Een hoogspringer die met de Fosbury flop over de lat gaat, buigt zijn lichaam er als een boog omheen: zijn zwaartepunt kan net onder de lat door, terwijl zijn lichaam erover gaat.



deel 1: 20 × 120 mm

A1 = 2400 mm², Z1 = (10; 60)

deel 2: 60 × 20 mm

A2 = 1200 mm², Z2 = (50; 10)

xZ = (2400 · 10 + 1200 · 50) / 3600 = 23,3 mm

yZ = (2400 · 60 + 1200 · 10) / 3600 = 43,3 mm

Z ligt op de lijn Z1–Z2, dichter bij het grootste deel, en buiten het materiaal.

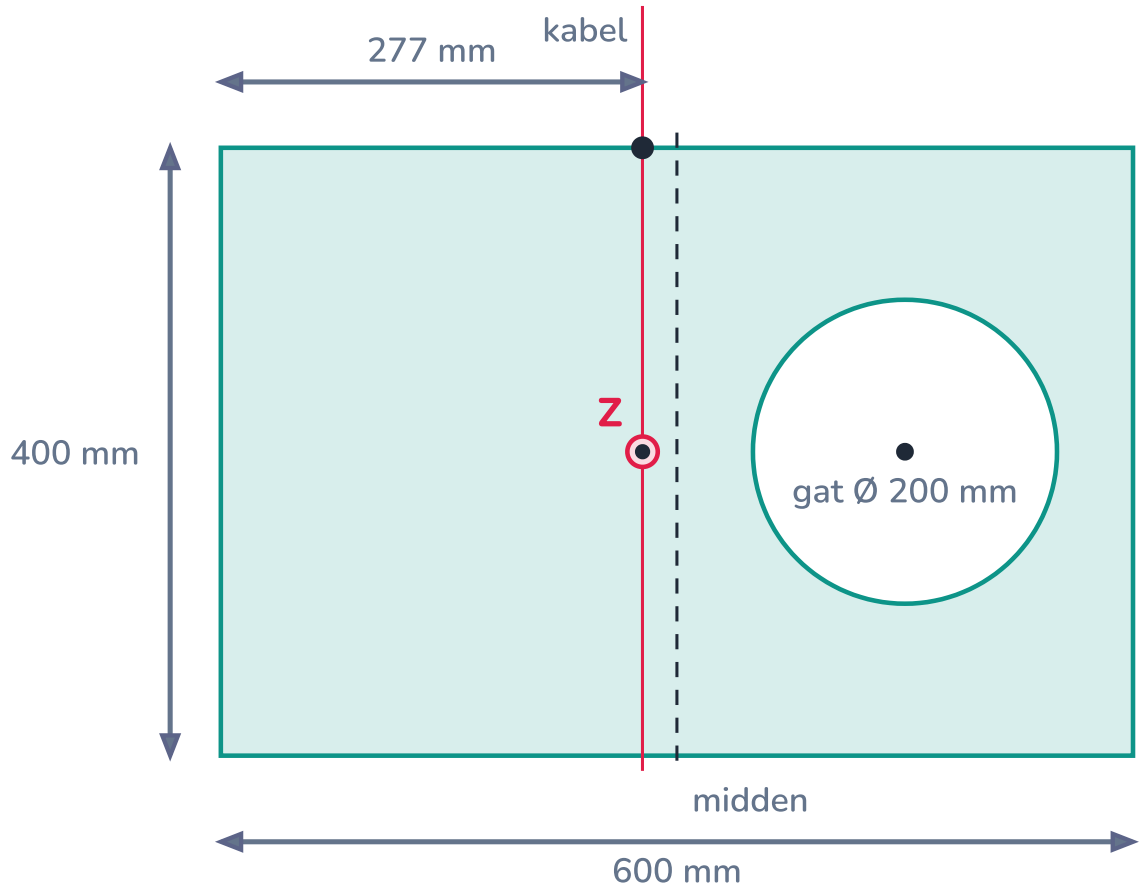
Twee rechthoeken, twee zwaartepunten; het gezamenlijke zwaartepunt ligt op de verbindingslijn, buiten het materiaal.

Elke steen staat zo dat het zwaartepunt van alles erboven recht boven zijn kleine steunvlak ligt.

UITGEWERKT VOORBEELD · 10.14

Een bord met een venster aan één kabel

Bij de ingang van het station komt een informatiebord van aluminium (3 mm dik, 600 mm × 400 mm) met een rond venster van 200 mm diameter voor een scherm. Het midden van het venster ligt 450 mm van de linkerrand, op halve hoogte. Het bord hangt aan één kabel. Waar moet de kabel vast, zodat het bord horizontaal hangt? Hoeveel weegt het bord (aluminium: 2,70 g/cm³)?



De kabel hangt recht boven Z, 277 mm van links; de stippellijn is het midden van het bord.

GEGEVEN

- rechthoek $600 \times 400\,\text{mm}$, zwaartepunt op $x = 300\,\text{mm}$
- gat: straal $100\,\text{mm}$, midden op $x = 450\,\text{mm}$
- dikte 3 mm, $\rho = 2{,}70\,\text{g/cm}^3$

GEVRAAGD

- x_Z
- massa

OPLOSSING

1. Het bord hangt horizontaal als de kabel recht boven het zwaartepunt zit. In de hoogte ligt Z op halve hoogte (symmetrie); we zoeken x_Z.

2. Oppervlaktes: rechthoek $A_1 = 600 \cdot 400 = 240\,000\,\text{mm}^2$; gat $A_2 = -\pi \cdot 100^2 = -31\,416\,\text{mm}^2$ (negatief). Samen $208\,584\,\text{mm}^2$.

3. $$x_Z = \frac{240\,000 \cdot 300 - 31\,416 \cdot 450}{208\,584} = \frac{72\,000\,000 - 14\,137\,000}{208\,584} = 277\,\text{mm}.$$

4. Massa: volume $208\,584\,\text{mm}^2 \cdot 3\,\text{mm} = 625\,800\,\text{mm}^3 = 626\,\text{cm}^3$, dus $m = 626 \cdot 2{,}70 = 1690\,\text{g} \approx 1{,}7\,\text{kg}$.

ANTWOORD

De kabel komt op 277 mm van de linkerrand, 23 mm links van het midden. Het bord weegt ongeveer 1,7 kg.

Klopt dit? Het gat haalt materiaal weg aan de rechterkant, dus het zwaartepunt schuift naar links: dat klopt. En de verschuiving is klein, omdat het gat maar 13 % van de oppervlakte is. Hang je het bord in het midden op, dan kantelt het tot Z recht onder de kabel hangt: de rechterkant komt omhoog.

PROEF · 10.15

Het zwaartepunt van een karton zoeken met een schietlood

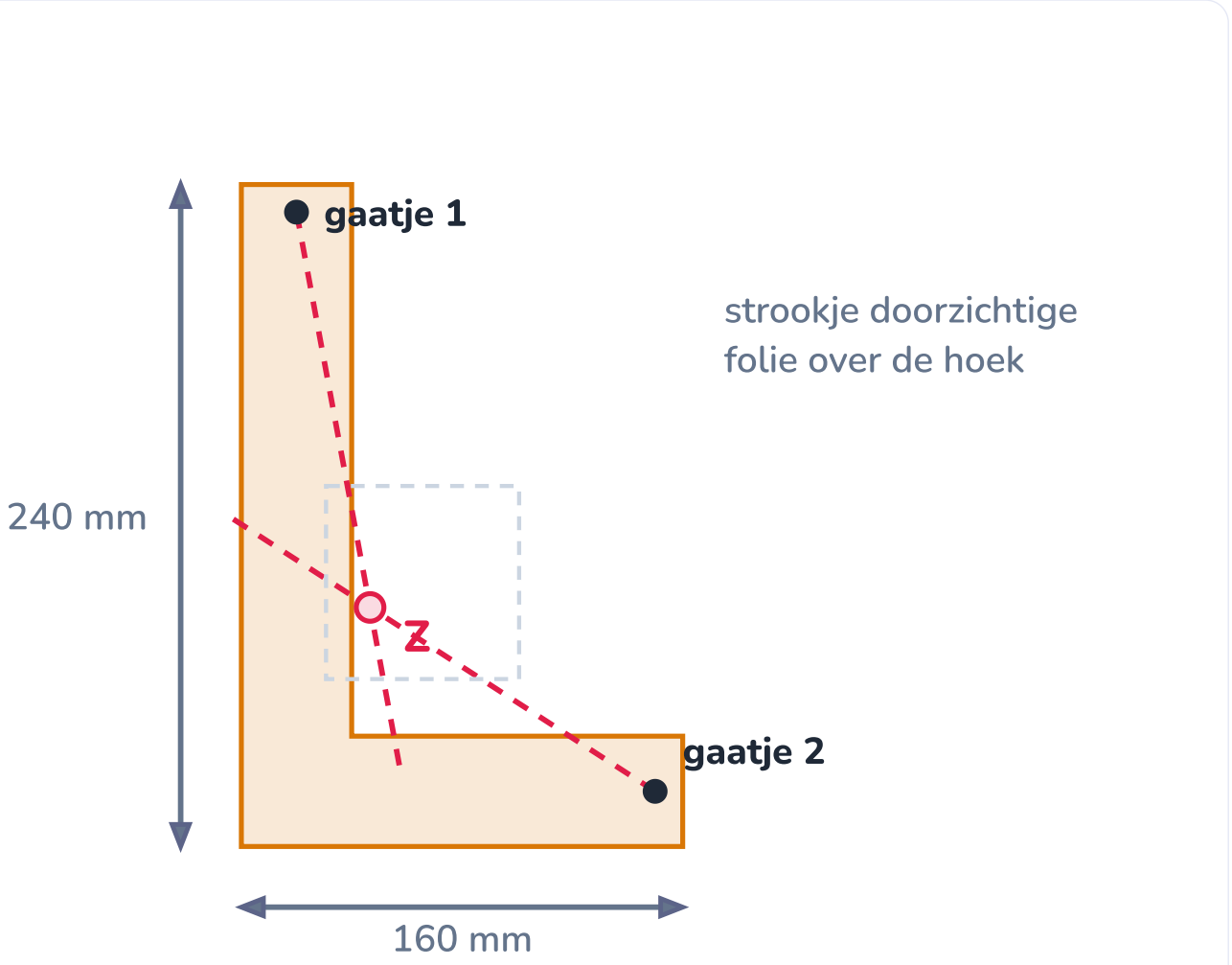
15/22

ONDERZOEKSVRAAG

Kan je het zwaartepunt van een vorm vinden zonder te rekenen? En klopt het met de berekening, ook als het buiten het karton ligt?

MATERIAAL

- stevig karton, schaar, geodriehoek
- een punaise en een stuk isomo of kurk om ze in te prikken
- touwtje met een klein gewichtje (schietlood)
- strookje doorzichtige folie, potlood



Twee schietloodlijnen door twee gaatjes; hun snijpunt is het zwaartepunt, op de folie naast het karton.

WERKWIJZE

1. Knip een L-vorm uit karton: 160 mm breed, 240 mm hoog, benen van 40 mm breed (de L uit de figuur hiervoor, twee keer zo groot).
2. Plak een strookje doorzichtige folie over de binnenhoek: daar verwacht je het zwaartepunt.

3. Prik de punaise door een gaatje vlak bij een hoek, zodat het karton vrij kan draaien. Hang het schietlood aan dezelfde punaise.

4. Wacht tot alles stilhangt en trek met potlood de lijn van het touwtje over karton en folie.

5. Herhaal met een tweede gaatje ver van het eerste. Het snijpunt van de twee lijnen is het zwaartepunt. Controleer met een derde gaatje.

6. Meet de afstand van het snijpunt tot de linker- en de onderrand.

WAARNEMING

De drie lijnen snijden elkaar (bijna) in één punt, op de folie, buiten het karton: ongeveer 47 mm van de linkerrand en 87 mm boven de onderrand. Een leerlingengroep mat 46 en 88 mm.

VERKLARING

Een vrij hangend voorwerp is in evenwicht als het moment van zijn gewicht om het ophangpunt nul is: het zwaartepunt hangt dan recht onder het ophangpunt, op de lijn van het schietlood. Twee zulke lijnen snijden in het zwaartepunt. De berekening geeft voor de dubbele L precies twee keer de waarden van de figuur: $x_Z = 46{,}7\ \text{mm}$ en $y_Z = 86{,}7\ \text{mm}$. Kleine afwijkingen komen van een scheef geknipte rand, de wrijving rond de punaise en de dikte van de potloodlijn.

UITGEWERKT VOORBEELD · 10.16

Hoe hoog ligt het zwaartepunt van de meetkas?

16/22

De meetkas uit Fundamental (1,20 × 0,60 × 0,80 m) staat buiten. Op de plantentafel (30 cm hoog) staan twaalf bakjes met vochtige potgrond; eronder staat het waterreservoir, 20 cm hoog gevuld. In F13 kreeg je de hoogte van het zwaartepunt gegeven; nu reken je ze zelf uit. Hoe hoog ligt het zwaartepunt van de volle kas, en van de lege? Gebruik de massa's uit F7, F10 en F11 en de hoogtes van de zwaartepunten van de delen in de tabel.

deel	massa m (kg)	hoogte zwaartepunt h (m)	$m \cdot h$ (kg·m)
12 bakjes (PP)	3,6	0,360	1,296
tafelblad	8,4	0,291	2,444
frame	3,1	0,400	1,240
wanden en dak	2,9	0,480	1,392
lucht in de kas	0,7	0,400	0,276
water (20 cm hoog)	24,0	0,100	2,400
vochtige potgrond	43,0	0,350	15,050
planten	1,2	0,450	0,540
totaal	**86,9**		**24,638**

GEGEVEN

- acht delen met massa m_i en hoogte h_i van hun zwaartepunt (tabel)
- leeg: zonder water, potgrond en planten (de eerste vijf rijen)

GEVRAAGD

- h_Z vol en leeg

OPLOSSING

1. De formule voor het zwaartepunt met massa's, in de verticale richting: $h_Z = \dfrac{\sum m_i\, h_i}{\sum m_i}$.

2. Vol: $\sum m_i = 86{,}9\,\text{kg}$ en $\sum m_i h_i = 24{,}64\,\text{kg}\cdot\text{m}$, dus $h_Z = \dfrac{24{,}64}{86{,}9} = 0{,}284\,\text{m}$.

3. Leeg (de eerste vijf rijen): $\sum m_i = 18{,}7\,\text{kg}$ en $\sum m_i h_i = 6{,}65\,\text{kg}\cdot\text{m}$, dus $h_Z = \dfrac{6{,}65}{18{,}7} = 0{,}356\,\text{m}$.

4. Horizontaal ligt het zwaartepunt (bij benadering) in het midden: de kas is symmetrisch ingericht.

ANTWOORD

Het zwaartepunt van de volle kas ligt 0,28 m hoog, dat van de lege 0,36 m: dezelfde waarden als in F13. De volle kas weegt 86,9 kg, de lege 18,7 kg.

Klopt dit? Het water ligt laag (10 cm) en trekt het zwaartepunt omlaag, de potgrond ligt boven de tafel en trekt het omhoog. Het resultaat ligt tussen de laagste en de hoogste waarde in de tabel, dichter bij de zwaarste delen: zo hoort het. De lucht in de kas (0,69 kg) mag je eigenlijk weglaten: de buitenlucht duwt de kas met evenveel kracht omhoog (Archimedes, hoofdstuk 11). Laat je ze weg, dan verandert h_Z pas in de derde decimaal.

UITLEG · 10.17

Kantelen en kantelveiligheid

17/22

Een voorwerp dat op een vlak staat, blijft staan zolang de verticale lijn door zijn zwaartepunt binnen het **steunvlak** valt (F13). Kantel je het een beetje, dan draait het rond een **kantelrand**. Het gewicht heeft dan een moment om die rand dat het voorwerp terugduwt, zolang Z nog aan de binnenkant van de rand ligt. Voorbij de **kantelhoek** ligt Z aan de buitenkant en duwt het gewicht het voorwerp verder om.

KANTELHOEK, STABILITEITSMOMENT EN KANTELVEILIGHEID

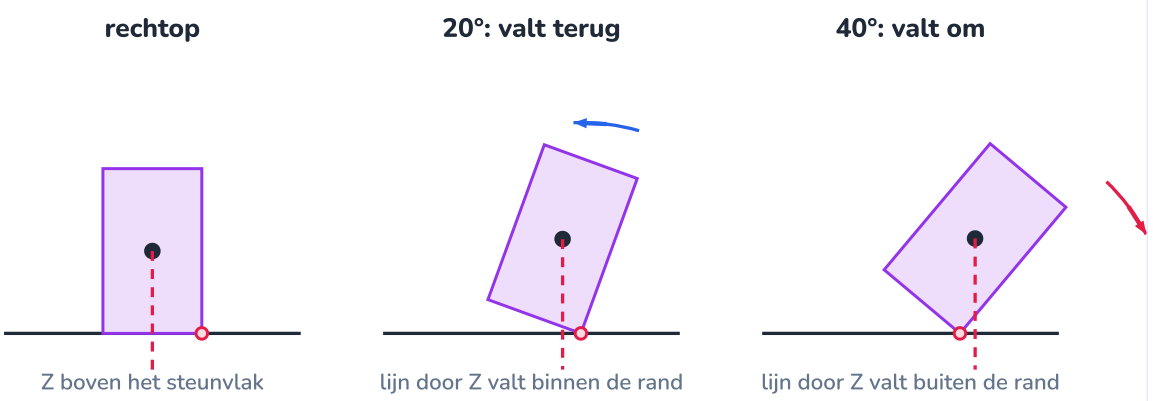
$$\tan\alpha_k = \frac{a}{h_Z} \qquad M_{\text{stab}} = m\,g\,a \qquad S = \frac{M_{\text{stab}}}{M_{\text{kantel}}}$$

a de horizontale afstand van het zwaartepunt tot de kantelrand, h_Z de hoogte van het zwaartepunt. M_{kantel} is het moment van de krachten die het voorwerp willen omduwen (wind, iemand die trekt), om dezelfde rand.

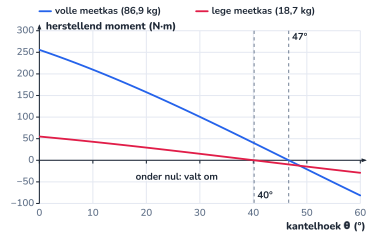
Is de kantelveiligheid S kleiner dan 1, dan valt het voorwerp om. Ontwerpers eisen meestal minstens $S = 1{,}5$, omdat krachten onzeker zijn en een windstoot of een duw groter kan uitvallen dan gerekend. Drie manieren om iets stabieler te maken:

- **Het zwaartepunt verlagen**: zware delen onderaan (ballast), zoals het reservoir in de meetkas.
- **Het steunvlak vergroten**: bredere poten, een bredere voet, uitschuifbare steunpoten bij een hoogtewerker.
- **Verankeren**: een kracht toevoegen aan de andere kant van de kantelrand, met een grote arm: een grondanker, een muuranker bij een kast.

De grafiek toont het moment dat de meetkas terugduwt als je ze over haar lange rand kantelt, $M = m\,g\,(a\cos\theta - h_Z\sin\theta)$ met $a = 0{,}30\,\text{m}$. Bij een hoek van 47° (vol) of 40° (leeg) wordt het nul: daar ligt het zwaartepunt recht boven de rand.



Zolang de lijn door Z binnen de kantelrand valt, draait het blok terug; daarbuiten valt het om.



De volle kas duwt veel harder terug en kantelt pas bij een grotere hoek: lager zwaartepunt, meer massa.

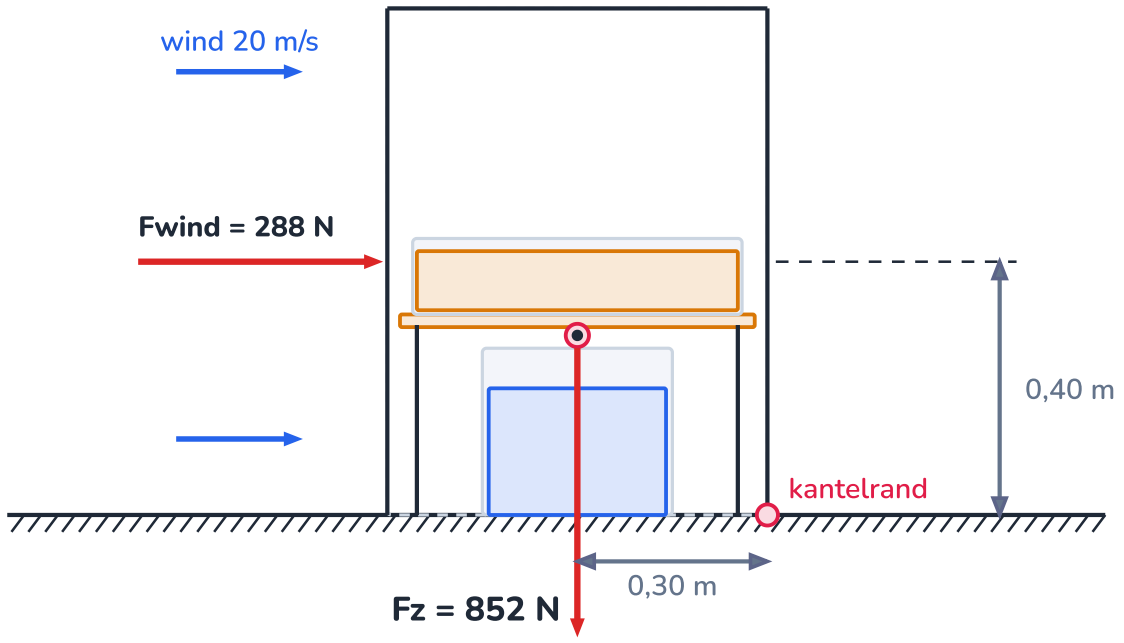
UITGEWERKT VOORBEELD · 10.18

Habitat: kantelt de meetkas in een storm?

18/22

De meetkas staat los op een terras. Bij een storm met windstoten van 20 m/s (72 km/h) duwt de wind met ongeveer 300 Pa op de lange zijwand van 1,20 × 0,80 m (een rekenwaarde; hoe je die uit de windsnelheid haalt, zie je in hoofdstuk E18). Kantelt de volle kas? En de lege, bijvoorbeeld tijdens de bouw of na het schoonmaken? Hoeveel kracht moeten grondankers leveren om de lege kas een kantelveiligheid van 1,5 te geven?

vooraanzicht; rood punt: zwaartepunt Z; gewichtspijl niet op schaal



De wind duwt op 0,40 m hoogte, het gewicht grijpt aan in Z; beide momenten om de kantelrand rechts onderaan.

GEGEVEN

- winddruk $300\,\mathrm{Pa}$ op $1{,}20 \times 0{,}80\,\mathrm{m}$, resultante op halve hoogte (0,40 m)
- vol: 86,9 kg; leeg: 18,7 kg; zwaartepunt in het midden, $a = 0{,}30\,\mathrm{m}$ van de kantelrand
- ankers aan de windkant, 0,60 m van de kantelrand

GEVRAAGD

- S vol en leeg
- ankerkracht voor $S = 1{,}5$

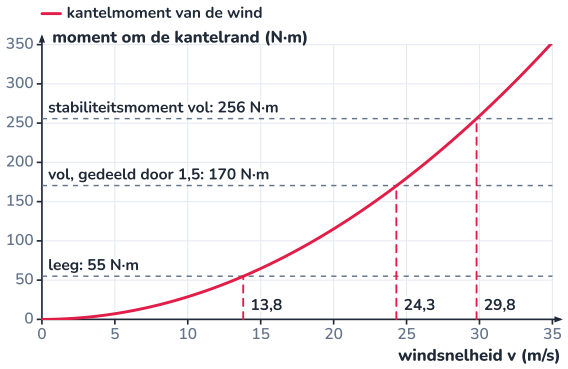
OPLOSSING

1. Windkracht: druk maal oppervlakte (hoofdstuk 11), $F = 300 \cdot 1{,}20 \cdot 0{,}80 = 288\,\mathrm{N}$. Om de kantelrand aan de lijzijde: $M_{\text{kantel}} = 288 \cdot 0{,}40 = 115{,}2\,\mathrm{N \cdot m}$.
2. Vol: $M_{\text{stab}} = 86{,}9 \cdot 9{,}81 \cdot 0{,}30 = 255{,}7\,\mathrm{N \cdot m}$, dus $S = 255{,}7 / 115{,}2 = 2{,}2$.
3. Leeg: $M_{\text{stab}} = 18{,}7 \cdot 9{,}81 \cdot 0{,}30 = 55{,}0\,\mathrm{N \cdot m}$, dus $S = 0{,}48$: de lege kas **kantelt**.
4. De winddruk groeit met het kwadraat van de windsnelheid. De lege kas kantelt al vanaf $20 \cdot \sqrt{55{,}0/115{,}2} = 13{,}8\,\mathrm{m/s}$ (50 km/h), een gewone winderige dag.
5. Ankers aan de windkant trekken omlaag met arm 0,60 m: $55{,}0 + F_{\text{anker}} \cdot 0{,}60 \geq 1{,}5 \cdot 115{,}2$, dus $F_{\text{anker}} \geq \dfrac{172{,}8 - 55{,}0}{0{,}60} = 196\,\mathrm{N}$.

ANTWOORD

Vol staat de kas veilig ($S = 2{,}2$), leeg niet ($S = 0{,}48$): ze waait om vanaf ongeveer 14 m/s. Grondankers aan de windkant moeten samen minstens 196 N kunnen houden. Omdat de wind van beide kanten kan komen: twee ankers van elk 100 N aan elke lange zijde.

Klopt dit? Ook vol haalt de kas $S = 1{,}5$ maar tot ongeveer 24 m/s (88 km/h), en ze kantelt bij 30 m/s (107 km/h): zulke windstoten komen in België bij zware stormen voor. Verankeren is dus altijd verstandig. En het model is optimistisch: de wind zuigt ook aan het dak en tilt de kas een beetje op, wat de kantelveiligheid nog verlaagt.



Het kantelmoment groeit met v²; waar het een horizontale lijn snijdt, is de kantelveiligheid 1.

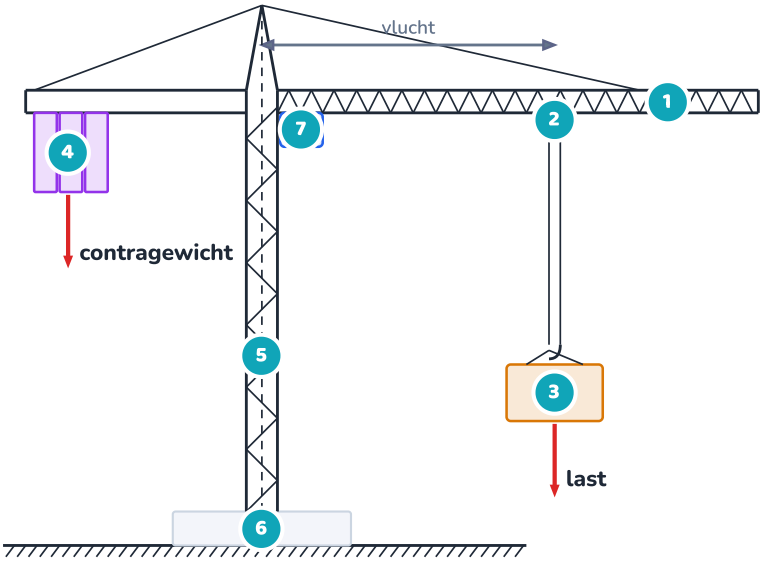
Twee ankers aan elke lange zijde: de lege kas staat pas dan veilig in de wind.

ONTLEED · 10.19

De torenkraan: een wip van zestig meter

19/22

Een torenkraan is een reusachtig momentenevenwicht. Aan de ene kant van de mast hangt de last aan de giek, aan de andere kant ligt het contragewicht op de contragiek. Wat het niet opvangt, moet de mast dragen, en uiteindelijk de fundering. De grafiek eronder is een typische hijstabel.



Schematische torenkraan: giek met loopkat rechts, contragiek met contragewicht links.

1 **Giek**
Een lichte vakwerkarm van tientallen meters. Vakwerk (driehoeken, F13) geeft veel stijfheid voor weinig gewicht: elke kilogram giek ver van de mast kost veel moment.

2 **Loopkat en vlucht**
De loopkat rijdt over de giek. De horizontale afstand van de haak tot de as van de mast heet de **vlucht**: de arm van de last.

3 **Last en lastmoment**
Het **lastmoment** is last maal vlucht. Deze kraan mag 5,0 t heffen tot 16 m; verder weg beslist het lastmoment van 80 t·m: 4,0 t op 20 m, 2,0 t op 40 m, 1,6 t op 50 m (grafiek).

4 **Contragewicht**
Betonblokken op de contragiek. Ze compenseren de giek en ongeveer de helft van het grootste lastmoment. Te zwaar mag niet: zonder last zou de kraan dan achterover willen kantelen (opdracht 10.16).

5 **Mast**
De mast draagt het gewicht van alles en het moment dat het contragewicht niet opheft, afwisselend naar voor en naar achter. Daarom is ook de mast een vakwerk.

6 **Fundering**
Een betonblok of ballast onder een brede voet. Voor de kantelveiligheid telt de afstand van de mast tot de rand van de voet: dat is de arm van het stabiliserende gewicht.

7 **Lastmomentbegrenzer**
Een sensor meet de kracht in de kabel en de plaats van de loopkat, en rekent het lastmoment uit. Wordt het te groot, dan blokkeert de kraan het heffen en het uitrijden van de kat.

CODE · 10.20

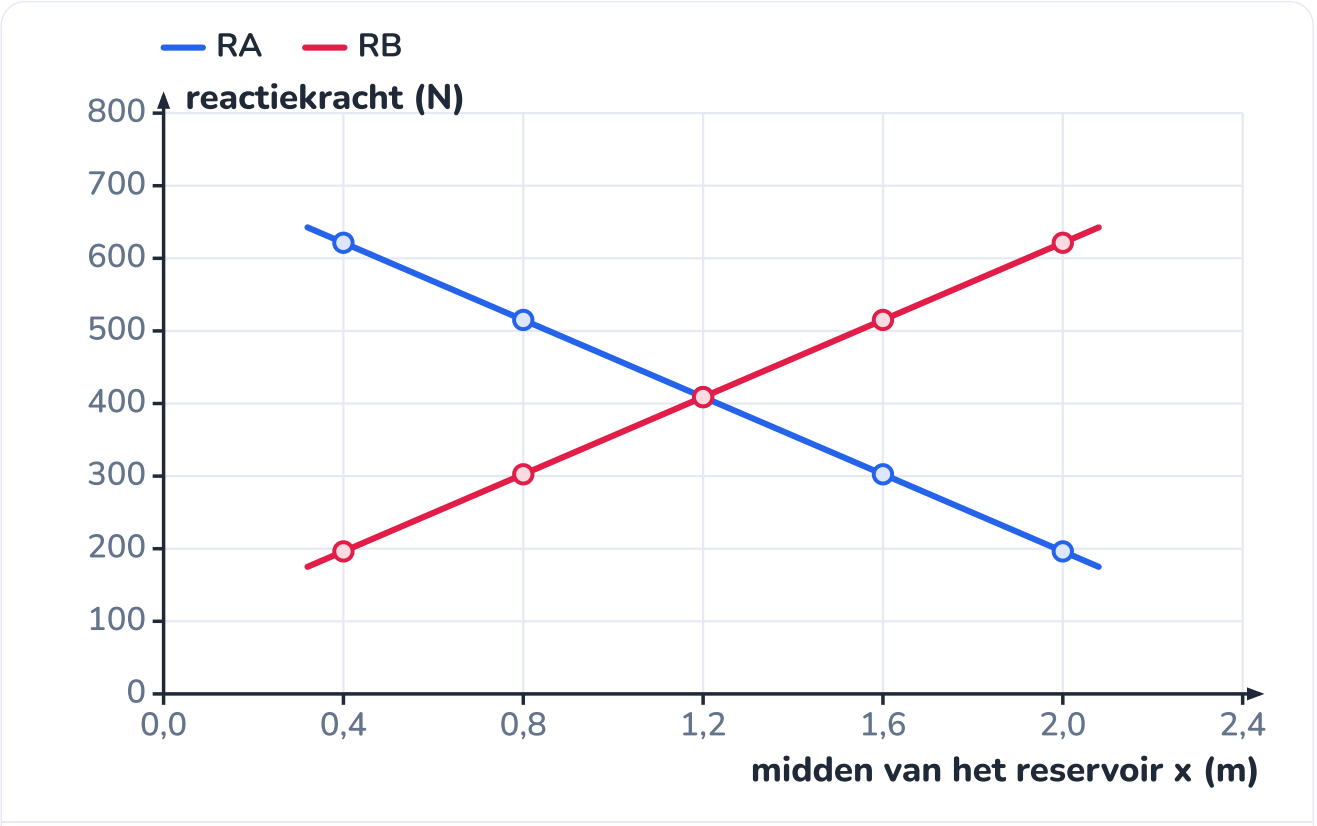
Python: steunpuntsreacties voor veel lastgevallen

20/22

Een ingenieur rekent zelden één lastgeval. Waar mag het reservoir staan, wat als er iemand bij staat, wat als er later een tweede reservoir komt? Het programma zet het rekenwerk van het voorbeeld in een functie: elke verdeelde last wordt een resultante in het midden, de som van de momenten om A geeft R_B en de som van de krachten geeft R_A. Daarna verschuift het het reservoir langs de balk.

Python **steunpuntsreacties.py** 23 regels

```python
# steunpuntsreacties.py - een balk op twee steunpunten: A bij x = 0, B bij x = L
L = 2.40                                  # overspanning van de vloerbalk (m)


def reacties(puntlasten, verdeeld):
    """puntlasten: lijst van (F in N, x in m); verdeeld: lijst van (q in N/m, x1, x2)."""
    krachten = list(puntlasten)
    for q, x1, x2 in verdeeld:            # verdeelde last: resultante in het midden
        krachten.append((q * (x2 - x1), (x1 + x2) / 2))
    som_F = sum(F for F, x in krachten)
    som_M_A = sum(F * x for F, x in krachten)   # moment van elke last om A
    R_B = som_M_A / L                     # som van de momenten om A is nul
    R_A = som_F - R_B                     # som van de verticale krachten is nul
    return R_A, R_B


eigengewicht = [(75.0, 0.0, L)]           # balk en vloerstrook: 75 N/m
print(" x reservoir (m)   R_A (N)   R_B (N)")
for i in range(5):
    x = 0.40 + 0.40 * i                   # midden van het reservoir
    reservoir = [(996.3, x - 0.32, x + 0.32)]   # 65 kg op 0,64 m balk
    R_A, R_B = reacties([], eigengewicht + reservoir)
    print(f"{x:10.2f}      {R_A:8.1f}  {R_B:8.1f}")
```



RA daalt en RB stijgt lineair als het reservoir van A naar B schuift; de punten zijn de uitvoer van het programma.

Uitvoer (gecontroleerd door het programma te draaien):

```
 x reservoir (m)   R_A (N)   R_B (N)
      0.40         621.4     196.3
      0.80         515.1     302.5
      1.20         408.8     408.8
      1.60         302.5     515.1
      2.00         196.3     621.4
```

Beide reacties veranderen **lineair** met de plaats van het reservoir, zoals bij de weegschalen in de proef. In het midden (1,20 m) dragen A en B elk 408,8 N; de som is altijd $637{,}6 + 180 = 817{,}6\,\mathrm{N}$. Zo'n grafiek, 'hoe verandert een reactie als een last over de constructie rijdt', heet een invloedslijn; bruggenbouwers gebruiken ze om te bepalen waar een vrachtwagen het gevaarlijkst staat (hoofdstuk E13).

NASLAG · 10.21

Overzicht: moment en evenwicht

21/22

moment (tegen de klok in positief)

$$M = F \cdot d = F \cdot r \cdot \sin\theta$$

symbool	betekenis	eenheid
M	moment	N·m
F	kracht	N
d	arm: loodrechte afstand tot de werklijn	m
θ	hoek tussen kracht en de lijn naar het draaipunt	°

evenwicht (momenten om een vrij te kiezen punt)

$$\sum F_x = 0 \qquad \sum F_y = 0 \qquad \sum M = 0$$

balk op twee steunpunten met één last op a van A en b van B

$$R_A = \frac{F \cdot b}{L} \qquad R_B = \frac{F \cdot a}{L}$$

resultante van een verdeelde last (in het midden, of op een derde aan de zware kant)

$$Q = q \cdot l \qquad Q_{\triangle} = \tfrac{1}{2}\, q_{\max} \cdot l$$

symbool	betekenis	eenheid
q	verdeelde belasting	N/m

zwaartepunt (gat: negatieve oppervlakte)

$$x_Z = \frac{\sum A_i\, x_i}{\sum A_i} = \frac{\sum m_i\, x_i}{\sum m_i}$$

kantelhoek en kantelveiligheid

$$\tan \alpha_k = \frac{a}{h_Z} \qquad S = \frac{M_{\text{stab}}}{M_{\text{kantel}}} \geq 1{,}5$$

vorm	oppervlakte	zwaartepunt
rechthoek $b \times h$	$b \cdot h$	in het midden: $b/2$ en $h/2$
driehoek met basis b en hoogte h	$\frac{1}{2}\, b \cdot h$	op $h/3$ boven de basis (snijpunt van de zwaartelijnen)
cirkel met straal r	πr^2	in het middelpunt
halve cirkel met straal r	$\frac{1}{2}\, \pi r^2$	op $\frac{4r}{3\pi} \approx 0{,}42\, r$ van de rechte rand

stappenplan evenwicht

1. Kies het voorwerp en teken het vrijlichaamsdiagram, met elke kracht op haar aangrijpingspunt. Vervang steunpunten door reacties (rol: 1, scharnier: 2, inklemming: 3 onbekenden).
2. Vervang verdeelde lasten door hun resultante; kies assen en een positieve draaizin.
3. Neem momenten om een punt waar onbekenden aangrijpen: $\sum M = 0$.
4. Vul aan met $\sum F_x = 0$ en $\sum F_y = 0$.
5. Controleer met momenten om een ander punt, en met gezond verstand: draagt het steunpunt het dichtst bij de last het meest?

Habitat: rekenwaarden uit dit hoofdstuk

vloerbalk module	vuren 45 × 145 mm, h.o.h. 40 cm, overspanning 2,40 m; balk + vloerstrook 75 N/m
reservoir op de vloer	65 kg (met bak, deksel en pomp), 0,64 × 0,44 m; midden op 0,50 m van A: $R_A = 595\,\text{N}$, $R_B = 223\,\text{N}$
meetkas	leeg 18,7 kg, zwaartepunt 0,36 m; vol 86,9 kg, zwaartepunt 0,28 m; kantelhoek 40° en 47° (F13)
meetkas in de storm	300 Pa op de zijwand: 288 N, $S = 2{,}2$ vol, 0,48 leeg; ankers 196 N aan de windkant

HET GEHEEL · 10.22

Waar momenten terugkomen

22/22

Met dit hoofdstuk is de statica compleet: een voorwerp in rust voldoet aan $\sum F = 0$ en $\sum M = 0$. Elke brug, elk dak, elke kraan en elke robotarm wordt daarmee doorgerekend. De momenten komen ook terug als iets wél draait: het koppel van een motor, de hoekversnelling van een vliegwiel. En het zwaartepunt is de sleutel tot alles wat moet blijven staan of drijven.

F11 Krachten
Een kracht kan doen draaien, en heeft een aangrijpingspunt: nu weet je hoeveel ze draait.

F12 Hefbomen en mechanismen
De hefboomwet is een momentenevenwicht met twee krachten; de drie soorten hefbomen komen hier terug.

F13 Constructies: sterk en stabiel
Zwaartepunt, steunvlak en kantelen, nu met getallen; de driehoek van het vakwerk in de giek van de kraan.

I6 Goniometrie en vectoren

I8 De wetten van Newton

I9 Arbeid, energie en vermogen
De gulden regel van de hefboom: wat je wint aan kracht, verlies je aan weg.

Een schuine kracht ontbinden, en de arm $r \sin\theta$ uitrekenen; de 1-op-4-regel van de ladder.

Het vrijlichaamsdiagram en de wrijving aan de voet van de ladder; nu met de krachten op hun echte plaats.

I11 Druk en hydraulica

Kracht verdeeld over een oppervlak; de waterdruk op de wand van het reservoir als driehoekige last; drijven en stabiliteit.

I26 Motoren en actuatoren

Het koppel van een motor in N·m, en hoeveel moment de dakmotor moet leveren.

E5 Vectoren en matrices

Het moment als vectorproduct $\vec{M} = \vec{r} \times \vec{F}$, ook in drie dimensies.

E10 Rotatie

Een moment dat niet nul is, geeft een hoekversnelling: de 'tweede wet' voor draaiende voorwerpen.

E12 Sterkteleer

Wat er ín de balk gebeurt: buigmoment, spanning en doorbuiging van de vloerbalk.

E13 Vakwerken en constructieanalyse

Evenwicht in elk knooppunt en de snedemethode: de krachten in de staven van een dak of een kraan.

Habitat

Voor Habitat leggen we vast: de modulevloer rust op vuren vloerbalken 45 × 145 mm om de 40 cm, met een overspanning van 2,40 m; balk en vloerstrook wegen 75 N/m. Het voedingsreservoir weegt vol 65 kg en staat met zijn midden op 0,50 m van ligger A: A draagt 595 N, B 223 N. De meetkas (leeg 18,7 kg, zwaartepunt 0,36 m; vol 86,9 kg, 0,28 m, zoals in F13) waait leeg om vanaf ongeveer 14 m/s, dus komen er twee grondankers van minstens 100 N aan elke lange zijde. In opdracht 10.14 reken je de draagbalk na met iemand erbij, in opdracht 10.18 de ballast voor het zonnepaneel.

SAMENVATTING

In het kort

- Het moment van een kracht om een punt is $M = F \cdot d$, met d de loodrechte afstand tot de werklijn. Een schuine kracht: $d = r \sin\theta$, of ontbind en neem alleen $F_\perp$.
- Tegen de klok in is positief, met de klok mee negatief. Een moment schrijf je in N·m, nooit in J.
- De hefboomwet is een momentenevenwicht. Een voorwerp is in evenwicht als $\sum F_x = 0$, $\sum F_y = 0$ én $\sum M = 0$; een koppel toont waarom de krachten alleen niet volstaan.
- De som van de momenten is nul om élk punt: kies het punt waar een onbekende aangrijpt.
- Steunpunten vervang je door reacties: een rol levert één kracht, een scharnier twee, een inklemming ook nog een moment. Het steunpunt het dichtst bij de last draagt het meest.
- Een verdeelde last q (N/m) vervang je voor het evenwicht door $Q = q \cdot l$ in het zwaartepunt van het lastvlak.
- Het zwaartepunt van een samengestelde vorm: $x_Z = \sum A_i x_i / \sum A_i$, met een gat als negatieve oppervlakte. Het kan buiten het voorwerp liggen; je vindt het ook met een schietlood.
- Een voorwerp kantelt als de lijn door zijn zwaartepunt buiten de kantelrand valt. Kantelveiligheid $S = M_{\text{stab}} / M_{\text{kantel}}$, liefst minstens 1,5: laag zwaartepunt, breed steunvlak, of verankeren.

BEGRIPPEN

Wat je nu kent

moment (van een kracht)

Maat voor hoe sterk een kracht een voorwerp rond een punt doet draaien: $M = F \cdot d$, in N·m.

arm

De loodrechte afstand van het draaipunt tot de werklijn van de kracht.

werklijn
De rechte waarop een kracht ligt, door haar aangrijpingspunt en in haar richting.

koppel
Twee even grote, tegengestelde, evenwijdige krachten die niet op één lijn liggen: samen geen resulterende kracht, wel een moment.

evenwichtsvoorwaarden
Een voorwerp is in evenwicht als de som van de krachten nul is én de som van de momenten om elk punt nul is.

steunpuntsreactie
De kracht die een steunpunt (oplegging) op een balk of constructie uitoefent.

verdeelde belasting
Een last die over een lengte gespreid is, uitgedrukt in N/m; ze werkt voor het evenwicht als haar resultante in het zwaartepunt van het lastvlak.

zwaartepunt
Het punt waar je de hele zwaartekracht op een voorwerp mag laten aangrijpen.

kantelhoek
De hoek waarover je een voorwerp kan kantelen voor het uit zichzelf omvalt: de lijn door het zwaartepunt gaat dan net door de kantelrand.

kantelveiligheid
Stabiliteitsmoment gedeeld door kantelmoment; ontwerpers eisen meestal minstens 1,5.

I INTERMEDIATE · DEEL I3 · MECHANICA

HANDBOEK · HOOFDSTUK 11

Druk en hydraulica

Waarom een punaise prikt, een dam onderaan dikker is en een klein pompje water 1,5 m hoog krijgt

Een kracht die over een oppervlak verdeeld is, heet **druk**. Druk verklaart waarom sneeuwschoenen werken en een mes snijdt, waarom je oren pijn doen onderaan een zwembad, hoe een klein pedaal een auto van 100 km/h stilzet, en waarom een stalen schip drijft. In dit hoofdstuk reken je met **druk in vloeistoffen** ($p = \rho g h$), gebruik je de **wet van Pascal** in hydraulische systemen, verklaar je met **Archimedes** wat je in F7 al zag bij zinken en drijven, en ontdek je dat we op de bodem van een zee van lucht leven. Voor Habitat meet je het niveau in het reservoir met een druksensor en kies je de pomp.

10× uitleg · 5× uitgewerkt voorbeeld · 3× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

K Krachten, beweging en constructies · M Materie en chemie · W Wiskundig modelleren

Een paar vingerdikke slangen met olie onder hoge druk: zo tilt een graafmachine tonnen.

Na dit hoofdstuk

- Je kan druk berekenen als kracht per oppervlakte en omrekenen tussen Pa, hPa, kPa, bar en MPa.
- Je kan de hydrostatische druk berekenen met $p = \rho g h$ en afleiden uit het gewicht van een vloeistofkolom.
- Je kan uitleggen waarom de druk alleen van de diepte afhangt en niet van de vorm van het vat (communicerende vaten).
- Je kan de wet van Pascal toepassen op een hydraulische pers, een krik en een remkring, en nagaan dat de arbeid behouden blijft.
- Je kan de opwaartse kracht berekenen met de wet van Archimedes en bepalen of en hoe diep een voorwerp drijft.
- Je kan de atmosferische druk verklaren, schatten hoe ze met de hoogte afneemt, en overdruk en absolute druk uit elkaar houden.
- Je kan uitleggen wat een pomp doet, waarom zuigen maar tot ongeveer 10 m werkt, en een pomp kiezen met een pompkarakteristiek.
- Je kan een druksensor gebruiken om een vloeistofniveau te meten, en de omrekening programmeren.

UITLEG · 11.1

Wat je al weet, en wat nu komt

1/23

In F7 zag je dat een voorwerp zinkt, zweeft of drijft naargelang zijn dichtheid, en dat een drijvend voorwerp zo ver wegzakt dat het evenveel vloeistof verplaatst als zijn eigen massa. Waarom dat zo is, bleef open. In F6 botsten gasdeeltjes tegen de wand van hun vat. In hoofdstuk 9 pompte je water 1,5 m hoog, en in hoofdstuk 10 vervang je een verdeelde last door één resultante.

begrip	formule of regel	waar
dichtheid	$\rho = m/V$, water 1,00 g/cm³ = 1000 kg/m³	F7
zinken, zweven, drijven	vergelijk de dichtheid van voorwerp en vloeistof	F7
gasdeeltjes botsen tegen de wand	deeltjesmodel	F6
zwaartekracht, normaalkracht	$F_z = m\,g$	F11, hoofdstuk 8
opvoerhoogte van een pomp	$E_p = m\,g\,h$	hoofdstuk 9
resultante van een driehoekige last, hefboom, koppel	$\frac{1}{2}\,q_{\max}\,l$ op een derde	hoofdstuk 10

In dit hoofdstuk verdeel je een kracht over een oppervlak. Concreet:

- reken je met **druk** en haar eenheden;
- bereken je de **druk in een vloeistof**, onderaan het voedingsreservoir en op zijn wand;
- laat je een kleine kracht een grote maken met **hydrauliek** (wet van Pascal);
- verklaar je **drijven** met de opwaartse kracht van Archimedes;
- ontdek je de **luchtdruk**, en het verschil tussen overdruk en absolute druk;
- zoek je uit wat een **pomp** doet, en welke pomp en welke slang Habitat nodig heeft.

UITLEG · 11.2

Druk: kracht per oppervlakte

2/23

Leg een baksteen plat op zand, dan op zijn kant, dan rechtop. Het gewicht is telkens hetzelfde, maar hoe kleiner het steunvlak, hoe dieper de steen wegzakt. Wat telt, is hoe de kracht over het oppervlak verdeeld is: de **druk**.

DRUK

$$p = \frac{F}{A}$$

F de kracht loodrecht op het oppervlak, in N; A de oppervlakte, in m²; p in pascal: $1\,\mathrm{Pa} = 1\,\mathrm{N/m^2}$.

Eén pascal is heel weinig: een blad papier op tafel geeft ongeveer 1 Pa. Daarom gebruik je vaak veelvouden:

eenheid	in pascal	waar je ze ziet
1 Pa = 1 N/m²	1	SI-eenheid
1 hPa = 1 mbar	100	weerkaarten (1013 hPa)
1 kPa	1000	techniek, sensoren

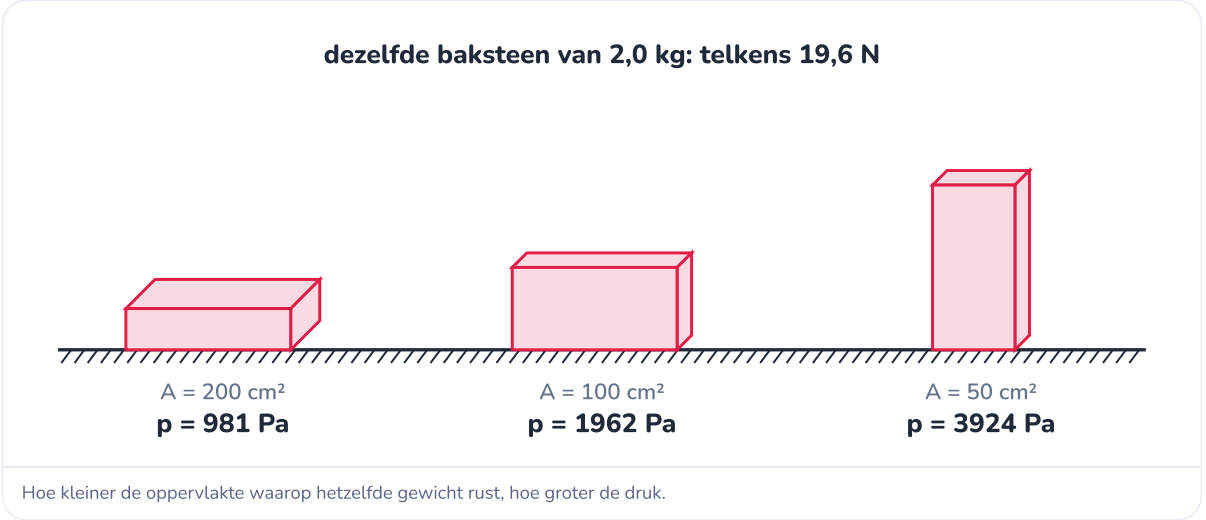
eenheid	in pascal	waar je ze ziet
1 bar	100 000	banden, leidingwater, duikflessen
1 MPa	1 000 000	hydrauliek, sterkte van materialen
1 mmHg (millimeter kwik)	133,3	bloeddruk (120/80 mmHg)
1 standaardatmosfeer	101 325	= 1013,25 hPa = 760 mmHg

Wil je veel druk, dan maak je de oppervlakte klein: een mes, een naald, de punt van een punaise. Wil je weinig druk, dan maak je ze groot: sneeuwschoenen, de rupsbanden van een tank of een graafmachine, de brede riemen van een rugzak.

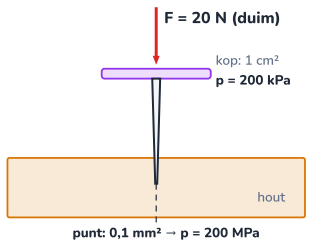
situatie	kracht	oppervlakte	druk (orde van grootte)
winddruk op een wand bij storm (20 m/s)			300 Pa
iemand van 70 kg op twee sneeuwschoenen	687 N	0,40 m²	1,7 kPa
iemand van 60 kg op twee schoenen	589 N	0,030 m²	20 kPa
olifant van 5 t op vier poten	49 kN	ongeveer 0,5 m²	100 kPa
luchtdruk op zeeniveau			101 kPa
autoband (overdruk)			220 kPa = 2,2 bar
iemand van 60 kg op één naaldhak	589 N	1 cm²	6 MPa
punt van een punaise	20 N	0,1 mm²	200 MPa

Druk en kracht zijn hetzelfde

Een naaldhak en een olifantenpoot: de olifant oefent tachtig keer zoveel kracht uit, maar de naaldhak geeft zestig keer zoveel druk. Kracht is een vector met een richting (in N); druk is kracht per oppervlakte (in Pa) en werkt in een vloeistof of gas in alle richtingen tegelijk.



Hoe kleiner de oppervlakte waarop hetzelfde gewicht rust, hoe groter de druk.



Je duim voelt 200 kPa, het hout onder de punt 200 MPa: duizend keer meer.

UITGEWERKT VOORBEELD · 11.3

Habitat: drukt het reservoir te hard op de vloer?

3/23

Het volle voedingsreservoir (65 kg met bak, deksel en pomp, buitenmaten 0,64 m × 0,44 m) staat op de vloer van de klimaatmodule. Vloeren in Belgische woningen worden berekend op een gelijkmatig verdeelde nuttige belasting van 2,0 kN/m² (en daarnaast op een puntlast van 2 kN). Hoe groot is de druk van het reservoir op de vloer? Wat verandert er als het reservoir op vier pootjes van 3 × 3 cm staat?

GEGEVEN

- $m = 65\,\text{kg}$
- $A = 0{,}64 \cdot 0{,}44 = 0{,}2816\,\text{m}^2$; pootjes $4 \cdot 0{,}03 \cdot 0{,}03 = 0{,}0036\,\text{m}^2$
- rekenwaarde vloer: $2{,}0\,\text{kN/m}^2$ verdeeld, $2\,\text{kN}$ als puntlast

GEVRAAGD

- p op de volle bodem en op de pootjes

OPLOSSING

1. Kracht: $F = m\,g = 65 \cdot 9{,}81 = 637{,}65\,\text{N}$.
2. Op de volle bodem: $p = \dfrac{637{,}65}{0{,}2816} = 2264\,\text{Pa} = 2{,}26\,\text{kPa}$, of $2{,}26\,\text{kN/m}^2$.
3. Dat is iets meer dan de 2,0 kN/m² van de norm, maar alleen op 0,28 m². De totale kracht, 0,64 kN, is veel kleiner dan de puntlast van 2 kN waarmee elke vloer ook berekend wordt. In hoofdstuk 10 berekende je welke krachten de liggers daardoor krijgen: 595 N en 223 N.
4. Op vier pootjes: $p = \dfrac{637{,}65}{0{,}0036} = 177\,000\,\text{Pa} = 177\,\text{kPa}$: bijna tachtig keer zoveel.

ANTWOORD

Op zijn volle bodem drukt het reservoir met 2,3 kPa op de vloer, de orde van grootte waarvoor een vloer ontworpen is. Op pootjes wordt dat 177 kPa, genoeg om deuken in een zachte vloerbekleding te maken.

Klopt dit? Dezelfde 637,65 N, een tachtig keer kleinere oppervlakte, dus tachtig keer meer druk: $0{,}2816/0{,}0036 = 78$. Een plaat onder de pootjes spreidt de kracht weer. Zo werken ook de platen onder de steunpoten van een kraan of een hoogtewerker.

UITLEG · 11.4

Druk in een vloeistof

4/23

Duik naar de bodem van een zwembad en je voelt het in je oren: hoe dieper, hoe groter de druk. De oorzaak is het gewicht van het water boven je. Neem een denkbeeldige kolom water met oppervlakte A

en hoogte h. Haar massa is $m = \rho V = \rho A h$ en haar gewicht $\rho A h g$. Verdeeld over de oppervlakte A geeft dat

HYDROSTATISCHE DRUK

$$p = \rho \cdot g \cdot h \qquad p_{\text{abs}} = p_0 + \rho g h$$

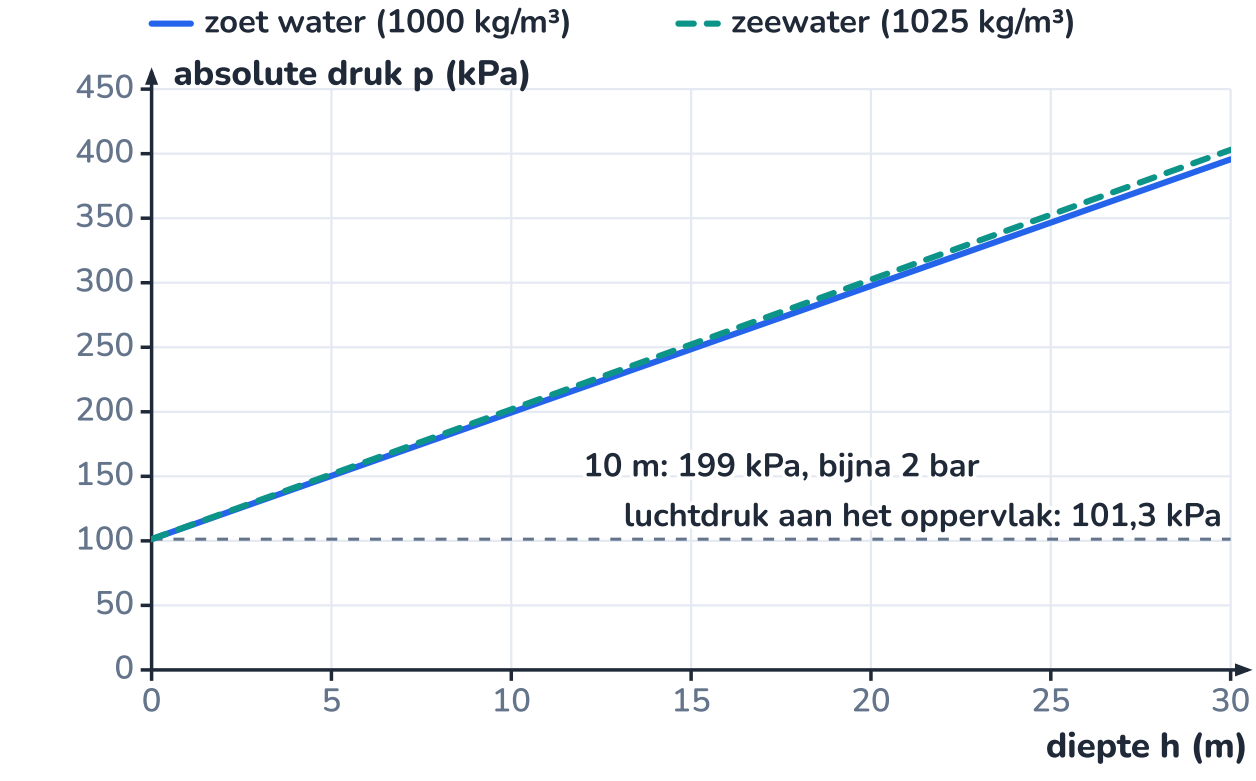
ρ de dichtheid van de vloeistof (kg/m³), h de diepte onder het oppervlak (m). Staat er lucht boven de vloeistof, dan komt de luchtdruk p_0 er nog bij (verderop).

Drie gevolgen die je niet meteen verwacht:

- **De vorm doet er niet toe.** A viel weg uit de formule: de druk hangt alleen af van de diepte. Onderaan een smalle buis van 10 m hoog is de druk even groot als onderaan een meer van 10 m diep.
- **Druk werkt in alle richtingen.** Op een bepaalde diepte duwt het water even hard opzij en omhoog als omlaag. Daarom staat er ook druk op de wanden van een reservoir, en is een stuwdam onderaan dikker.
- **In water: ongeveer 1 bar per 10 m.** $1000 \cdot 9{,}81 \cdot 10 = 98\,100\,\text{Pa}$. Een duiker op 10 m diepte in zee ondervindt dus 1 bar overdruk, 2 bar absoluut: twee keer de druk aan het oppervlak.

Een breed vat drukt harder op zijn bodem

Het water in een breed vat weegt meer, maar het verdeelt zich ook over een grotere bodem. De druk hangt alleen af van de hoogte van de vloeistof. Zet je vaten van heel verschillende vorm op dezelfde bodemplaat, dan is bij gelijke waterhoogte de kracht op die bodemplaat telkens gelijk. Pascal beschreef deze hydrostatische paradox al in de zeventiende eeuw.

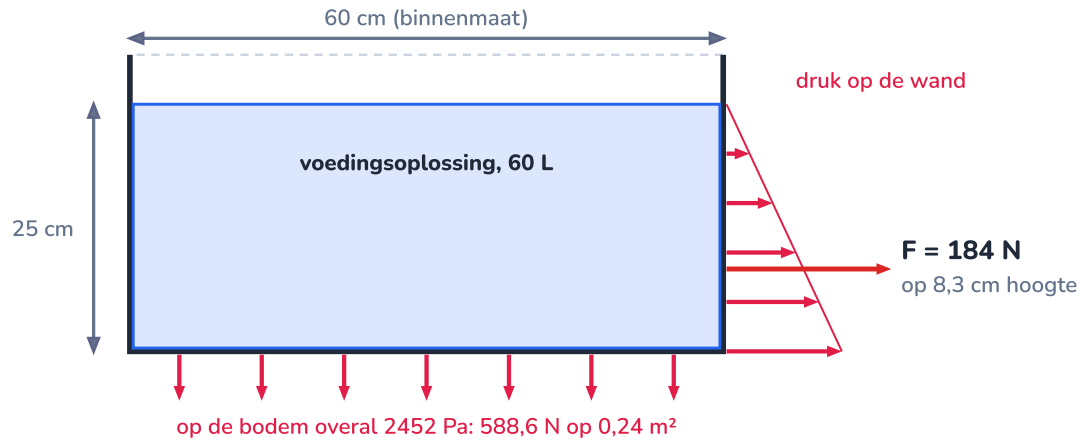


De druk stijgt lineair met de diepte, in zeewater iets sneller dan in zoet water.

UITGEWERKT VOORBEELD · 11.5

Habitat: de druk onderaan het voedingsreservoir

Het voedingsreservoir heeft binnenmaten van 60 cm × 40 cm × 30 cm en is gevuld tot 25 cm (60 L). Bereken de overdruk en de absolute druk op de bodem, de kracht op de bodem, en de kracht die het water op een lange zijwand uitoefent. Neem voor de voedingsoplossing $\rho = 1000\,\mathrm{kg/m^3}$ (1 g zout per liter verandert daar bijna niets aan).



De pijlen op de wand groeien met de diepte (driehoek), op de bodem is de druk overal gelijk.

GEGEVEN

- $h = 0{,}25\,\mathrm{m}$, bodem $0{,}60 \times 0{,}40\,\mathrm{m}$
- $\rho = 1000\,\mathrm{kg/m^3}$, $p_0 = 101\,325\,\mathrm{Pa}$

GEVRAAGD

- p en p_{abs} op de bodem
- F_{bodem}
- F op een lange zijwand

OPLOSSING

1. Overdruk op de bodem: $p = 1000 \cdot 9{,}81 \cdot 0{,}25 = 2452{,}5\,\mathrm{Pa} \approx 2{,}45\,\mathrm{kPa}$. Absoluut: $101\,325 + 2452{,}5 = 103\,778\,\mathrm{Pa} \approx 103{,}8\,\mathrm{kPa}$.
2. Kracht op de bodem (door het water): $F = p \cdot A = 2452{,}5 \cdot 0{,}24 = 588{,}6\,\mathrm{N}$.
3. Controle: dat is precies het gewicht van 60 kg water, $60 \cdot 9{,}81 = 588{,}6\,\mathrm{N}$. Bij een vat met rechte wanden draagt de bodem al het water.
4. Op de zijwand groeit de druk lineair van 0 (bovenaan) tot 2452,5 Pa (onderaan): een driehoekige belasting (hoofdstuk 10). De gemiddelde druk is de helft: $F = \tfrac{1}{2} \cdot 2452{,}5 \cdot (0{,}60 \cdot 0{,}25) = 183{,}9\,\mathrm{N}$.
5. Die resultante grijpt aan op een derde van de waterhoogte boven de bodem: $0{,}25/3 = 8{,}3\,\mathrm{cm}$.

ANTWOORD

Onderaan heerst 2,45 kPa overdruk (103,8 kPa absoluut); het water drukt met 588,6 N op de bodem en met 184 N op elke lange zijwand, op 8,3 cm boven de bodem.

Klopt dit? Daarom buikt een kunststof reservoir onderaan het meest uit, en zit een aftapkraantje daar onder de grootste druk. De overdruk is maar 2,4 % van de luchtdruk: een druksensor voor dit reservoir moet dus verschillen in overdruk meten, niet de absolute druk (codeblad verderop). Een kleinere kracht op de korte wand: $\tfrac{1}{2} \cdot 2452{,}5 \cdot 0{,}40 \cdot 0{,}25 = 123\,\mathrm{N}$.

PROEF · 11.6

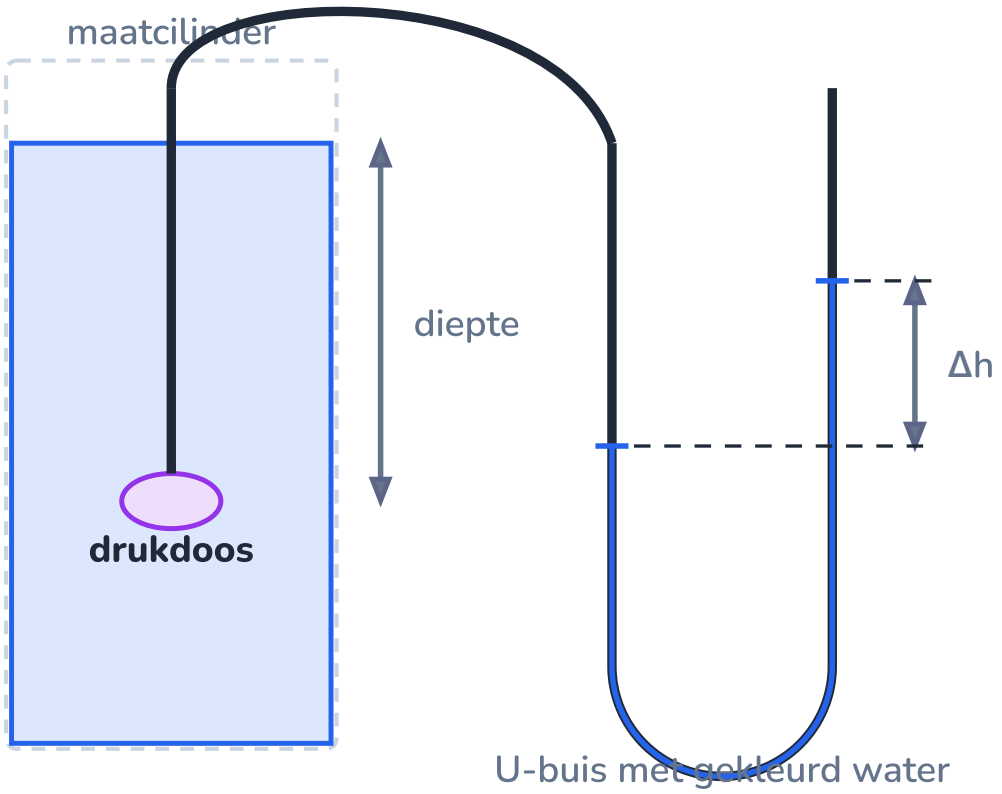
De druk in water meten met een drukdoos

ONDERZOEKSVRAAG

Hoe hangt de druk in water af van de diepte? En hangt ze af van de richting waarin je meet?

MATERIAAL

- hoge maatcilinder of vaas, gevuld met water
- drukdoos: kleine trechter met een strak gespannen ballonvlies
- slangetje en U-buis met gekleurd water
- meetlat



Het water duwt op het vlies; de lucht in het slangetje geeft die druk door aan de U-buis.

WERKWIJZE

1. Verbind de drukdoos met het slangetje met één been van de U-buis. Controleer dat beide waterniveaus in de U-buis gelijk staan.
2. Laat de drukdoos zakken tot 5 cm onder het wateroppervlak (gemeten tot het vlies). Lees het hoogteverschil Δh tussen de twee benen af.
3. Herhaal om de 5 cm tot 25 cm diepte.
4. Draai op 15 cm diepte het vlies eens opzij en eens naar boven, zonder de diepte te veranderen. Noteer telkens Δh.

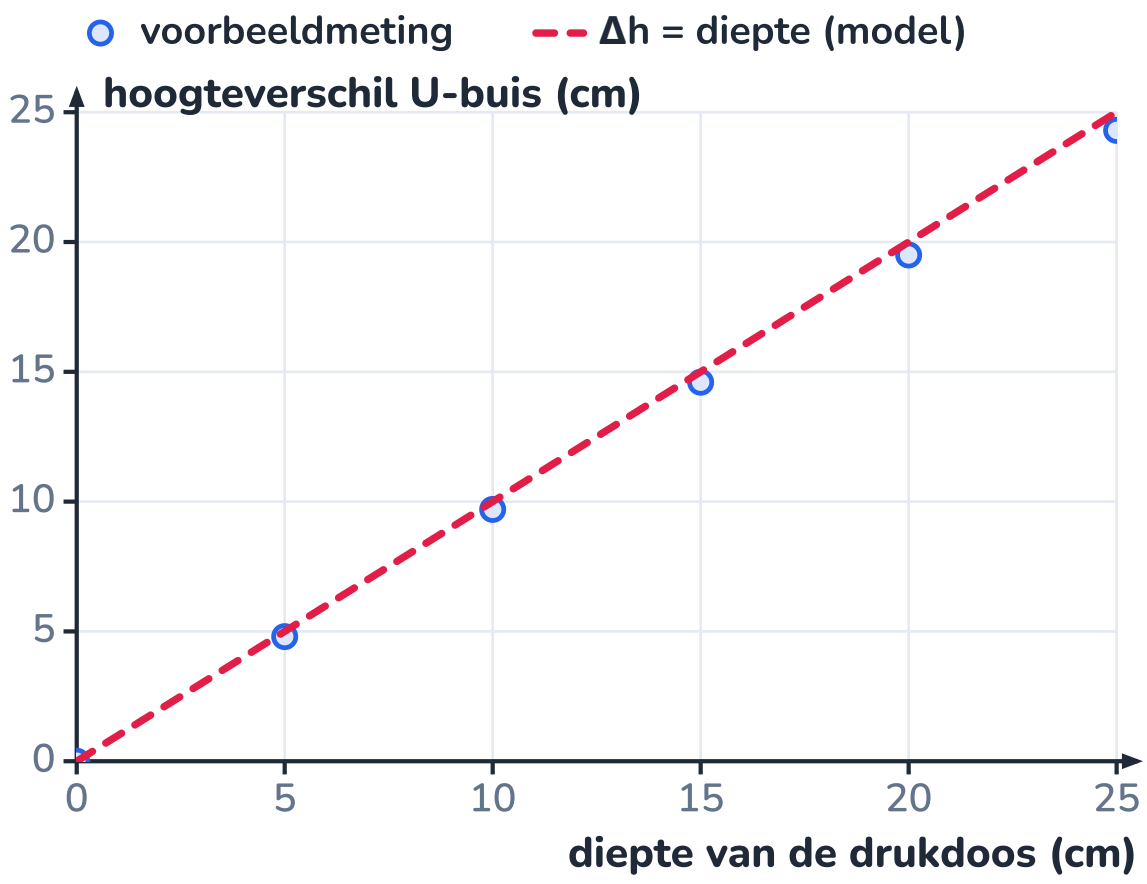
WAARNEMING

diepte (cm)	0	5	10	15	20	25
Δh (cm)	0,0	4,8	9,7	14,6	19,5	24,3

Δh groeit evenredig met de diepte, net iets minder dan de diepte zelf. Op 15 cm blijft Δh 14,6 cm, of het vlies nu naar onder, opzij of naar boven wijst.

VERKLARING

De U-buis vergelijkt de druk in de drukdoos met de luchtdruk: een hoogteverschil Δh in water betekent een overdruk $\rho\, g\, \Delta h$. De drukdoos meet dus $\rho\, g\cdot$ diepte: de druk is recht evenredig met de diepte, zoals $p = \rho\, g\, h$ voorspelt. Dat Δh iets kleiner is dan de diepte (helling 0,97), komt doordat het vlies zelf een beetje spanning opvangt en de lucht in het slangetje iets samengedrukt wordt. Dat het vlies in elke richting hetzelfde meet, toont dat druk in een vloeistof in alle richtingen even groot is.



Voorbeeldmeting: bijna op de rechte Δh = diepte.

UITLEG · 11.7

Communicerende vaten

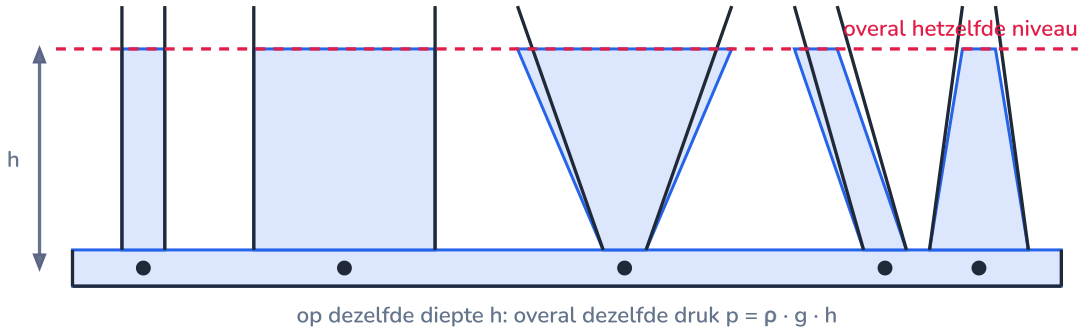
7/23

Verbind vaten onderaan met elkaar en giet er water in: het staat overal even hoog, hoe verschillend de vaten ook zijn. Stond het in één vat hoger, dan was de druk op de bodem daar groter ($p = \rho\, g\, h$), en stroomde het water weg naar de andere vaten tot de drukken gelijk zijn. Zulke vaten heten **communicerende vaten**. Je gebruikt ze vaker dan je denkt:

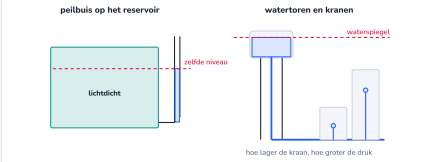
- **De peilbuis.** Het voedingsreservoir van Habitat is lichtdicht (algen, hoofdstuk 5): je ziet het water niet. Een doorzichtig buisje dat onderaan op het reservoir aangesloten is, toont het niveau.
- **De slangwaterpas.** Een doorzichtige slang vol water: de twee uiteinden staan altijd even hoog, ook om een hoek of tien meter verder. Bouwvakkers zetten er peilen mee uit.

- **De watertoren.** Een reservoir hoog boven de stad levert de druk in het leidingnet. Een kraan 32 m onder de waterspiegel krijgt $1000 \cdot 9{,}81 \cdot 32 = 3{,}1\ \text{bar}$ overdruk, zonder dat er een pomp draait. Hoog in een flatgebouw is de druk lager, en daar zijn soms drukverhogingspompen nodig.
- **Een sifon** onder een gootsteen houdt altijd wat water vast, zodat er geen rioollucht naar boven komt.

> ⚠ **Alleen voor één vloeistof in rust**
>
> Gelijke niveaus gelden voor dezelfde vloeistof, die stilstaat, en met dezelfde luchtdruk boven elk oppervlak. Twee vloeistoffen met een verschillende dichtheid staan niet even hoog (dat gebruik je in een U-buis om een dichtheid te meten), en in een leiding waar water stroomt, gaat druk verloren aan wrijving (verderop).



Hoe verschillend de vorm ook is, het water staat overal op hetzelfde niveau en de bodemdruk is overal gelijk.



Links toont de peilbuis het niveau in een reservoir dat je niet kan inkijken; rechts levert de watertoren de druk aan de kranen.

UITLEG · 11.8

De wet van Pascal en de hydraulische pers

8/23

Druk je op de zuiger van een injectiespuit vol water met een dichtgeknepen opening, dan stijgt de druk overal in het water even veel, ook in de hoeken. Blaise Pascal beschreef het in de zeventiende eeuw:

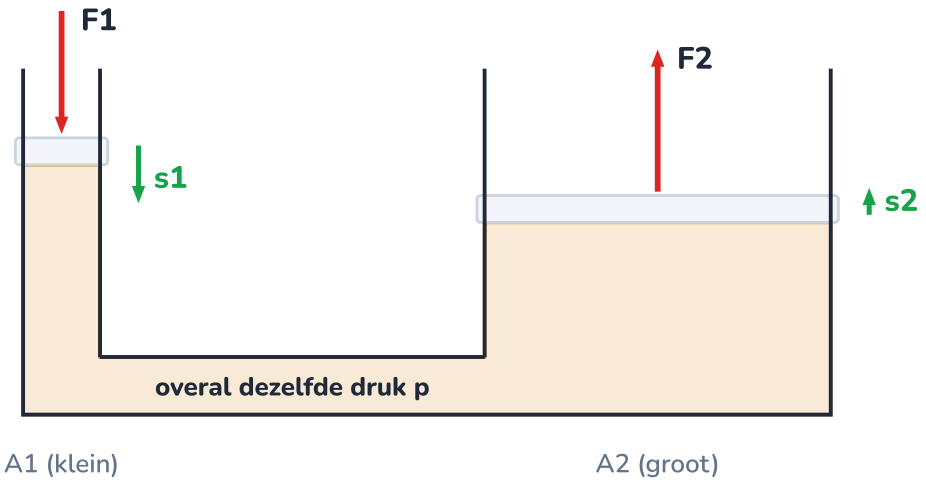
WET VAN PASCAL EN DE HYDRAULISCHE PERS

$$\Delta p \text{ is overal gelijk} \qquad p = \frac{F_1}{A_1} = \frac{F_2}{A_2} \quad \Rightarrow \quad F_2 = F_1 \cdot \frac{A_2}{A_1}$$

Een drukverandering in een ingesloten vloeistof plant zich onverminderd voort. Verbind je een kleine en een grote zuiger, dan is de kracht op de grote zuiger zoveel keer groter als haar oppervlakte.

Een kracht vermenigvuldigen zonder iets te verliezen? Dat kan niet, en dat klopt ook: de olie die de kleine zuiger wegduwt, verdeelt zich onder de grote zuiger over een veel grotere oppervlakte. Het volume blijft gelijk: $A_1 \cdot s_1 = A_2 \cdot s_2$. De grote zuiger beweegt dus zoveel keer *minder*. De arbeid aan beide kanten is dezelfde: $F_1 \cdot s_1 = F_2 \cdot s_2$. Dat is de gulden regel van hoofdstuk 9, net als bij de hefboom (hoofdstuk 10) en het hellend vlak.

Waarom een **vloeistof**? Omdat die (bijna) niet samendrukbaar is. Zit er lucht in een hydraulisch systeem, dan drukt je zuiger eerst die lucht samen voor er iets gebeurt: het pedaal voelt 'sponzig' aan. Hydrauliek zit in graafmachines, hefbruggen, kraanwagens, de landingsgestellen van vliegtuigen, kantpersen die staalplaat plooien en de remmen van elke auto.



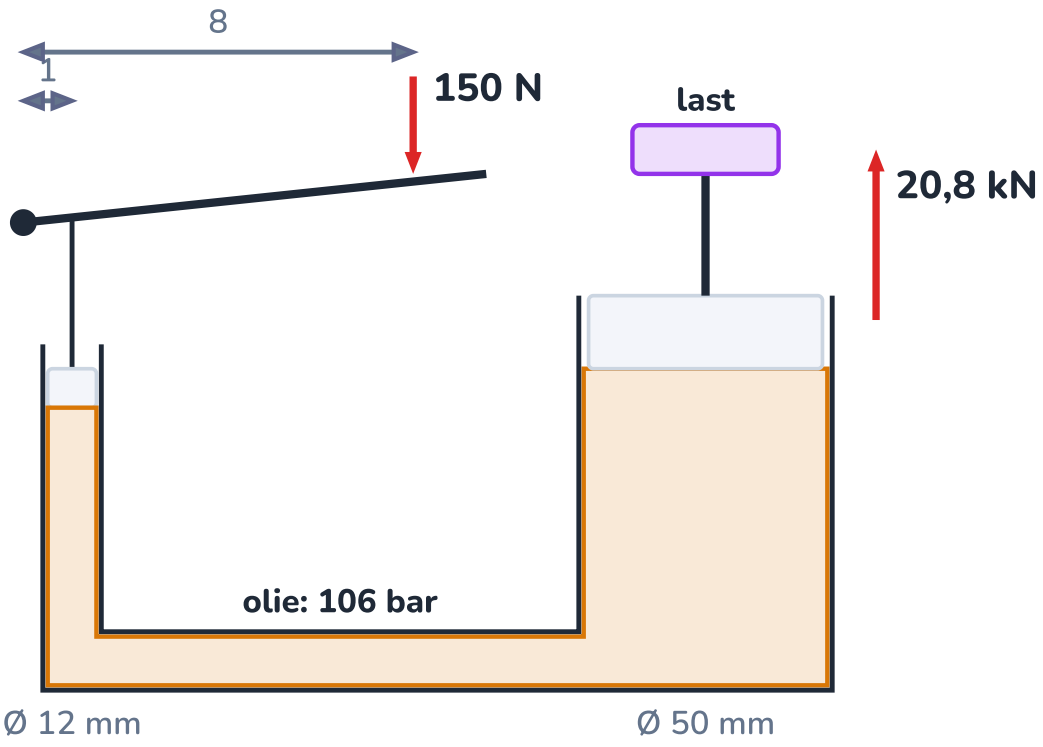
Zelfde druk onder beide zuigers: kleine kracht en grote slag links, grote kracht en kleine slag rechts.

UITGEWERKT VOORBEELD · 11.9

Een hydraulische krik

9/23

Een flessenkrik heeft een pompzuiger van 12 mm diameter en een hefzuiger van 50 mm. Je duwt met 150 N op de hendel, die als hefboom de kracht 8 keer vergroot (hoofdstuk 10). Welke druk heerst er in de olie, en welke last kan de krik heffen? Hoeveel keer moet je pompen om de last 10 cm op te krikken als de pompzuiger per slag 20 mm zakt?



Hendel (hefboom), pompzuiger en hefzuiger: twee krachtvermenigvuldigers achter elkaar.

GEGEVEN

- $d_1 = 12\,\text{mm}$, $d_2 = 50\,\text{mm}$
- handkracht 150 N, hefboom 8 : 1
- slag pompzuiger 20 mm

GEVRAAGD

- p, F_2 en het aantal slagen voor 10 cm

OPLOSSING

1. Kracht op de pompzuiger: $F_1 = 8 \cdot 150 = 1200\,\text{N}$.
2. Oppervlaktes: $A_1 = \pi \cdot 0{,}006^2 = 1{,}131 \cdot 10^{-4}\,\text{m}^2$ en $A_2 = \pi \cdot 0{,}025^2 = 1{,}963 \cdot 10^{-3}\,\text{m}^2$.
3. Druk: $p = \dfrac{1200}{1{,}131 \cdot 10^{-4}} = 1{,}06 \cdot 10^7\,\text{Pa} = 106\,\text{bar}$.
4. Kracht op de hefzuiger: $F_2 = p \cdot A_2 = 1{,}06 \cdot 10^7 \cdot 1{,}963 \cdot 10^{-3} = 20{,}8\,\text{kN}$: het gewicht van 2,1 ton. (Of korter: $F_2 = F_1 \cdot (d_2/d_1)^2 = 1200 \cdot (50/12)^2$.)
5. Per slag verplaatst de pompzuiger $A_1 \cdot 20\,\text{mm}$ olie; de hefzuiger stijgt $20 \cdot \dfrac{A_1}{A_2} = 20 \cdot 0{,}0576 = 1{,}15\,\text{mm}$. Voor 100 mm: $100/1{,}15 = 87$ slagen.

ANTWOORD

De olie staat onder 106 bar en de krik kan 20,8 kN heffen, ruim 2 ton. Daarvoor pomp je 87 keer om 10 cm te winnen.

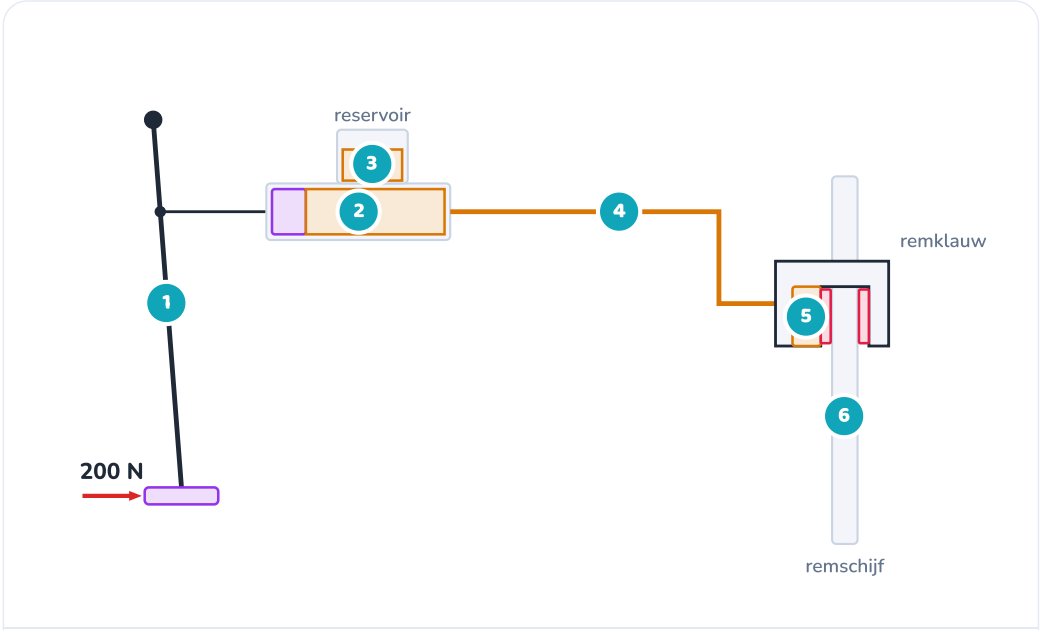
Klopt dit? Arbeid per slag: aan de pompzuiger $1200 \cdot 0{,}020 = 24\,\text{J}$, aan de hefzuiger $20\,833 \cdot 0{,}00115 = 24\,\text{J}$. Wat je wint aan kracht (een factor 17 in de olie, 139 met de hendel erbij), verlies je aan weg. Kleppen zorgen ervoor dat de olie bij het terugtrekken van de pompzuiger niet terugstroomt.

ONTLEED · 11.10

De hydraulische rem van een auto

10/23

Met je voet stop je een auto van anderhalve ton. Dat lukt omdat twee principes uit dit en het vorige hoofdstuk samenwerken: de hefboom en de wet van Pascal. (Bij een echte auto helpt nog een rembekrachtiger, die de pedaalkracht enkele keren vergroot.)



Schematische remkring voor één wiel; in een auto vertakt de remleiding naar de vier wielen.

1 Rempedaal: een hefboom
Het pedaal draait rond een as bovenaan; de duwstang zit op een vierde van de afstand tot je voet. Een voet die met 200 N duwt, duwt de stang met 800 N (hefboom 4 : 1).

2 Hoofdcilinder
Een zuiger van 22 mm diameter ($3{,}80\,\mathrm{cm}^2$): $p = 800/3{,}80 \cdot 10^{-4} = 2{,}1 \cdot 10^6\,\mathrm{Pa}$, ongeveer 21 bar.

3 Reservoir met remvloeistof
Vult de kring bij als de remblokken slijten en de zuigers verder uitschuiven. Remvloeistof neemt vocht op; daarom wordt ze om de twee jaar vervangen.

4 Remleiding
Een dunne stalen leiding. Volgens Pascal staat overal in de kring dezelfde druk. Zit er lucht in, dan drukt die eerst samen: het pedaal zakt door, en de remmen werken slecht. Daarom wordt een remkring na herstellingen 'ontlucht'.

5 Zuiger in de remklauw
Een zuiger van 54 mm diameter ($22{,}9\,\mathrm{cm}^2$): $F = 2{,}1 \cdot 10^6 \cdot 22{,}9 \cdot 10^{-4} = 4{,}8\,\mathrm{kN}$, zes keer de kracht op de stang, 24 keer die van je voet.

6 Remblokken en remschijf
De zuiger klemt de blokken tegen de draaiende schijf. De wrijving (hoofdstuk 8) zet de bewegingsenergie van de auto om in warmte (hoofdstuk 9): een remschijf kan na een lange afdaling honderden graden heet worden.

UITLEG · 11.11

De wet van Archimedes

11/23

Een steen onder water lijkt lichter; een strandbal duw je nauwelijks onder. Het water duwt omhoog, met de **opwaartse kracht**. Waar komt die vandaan? Neem een blok met bovenvlak A en hoogte h onder water. De druk op de zijkanten heft zich op. Op het bovenvlak duwt het water omlaag met $\rho\, g\, h_1 \cdot A$, op het ondervlak omhoog met $\rho\, g\, h_2 \cdot A$. Onderaan is het dieper, dus de druk groter. Het verschil:

$$F_A = \rho_{\mathrm{vl}}\, g\, (h_2 - h_1)\, A = \rho_{\mathrm{vl}}\, g\, V$$

WET VAN ARCHIMEDES

$$F_A = \rho_{\text{vl}} \cdot g \cdot V_{\text{ond}}$$

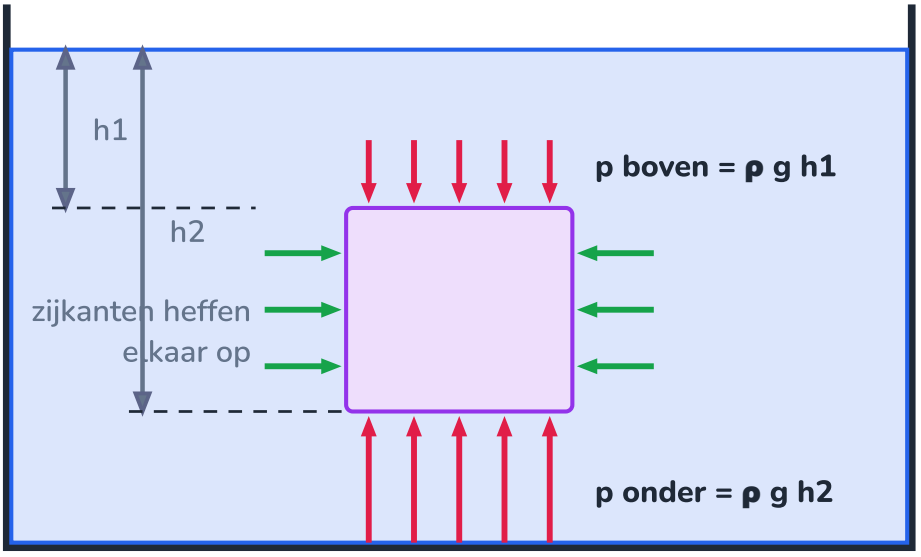
Een voorwerp in een vloeistof (of gas) ondervindt een opwaartse kracht die even groot is als het gewicht van de verplaatste vloeistof. ρ_{vl} de dichtheid van de **vloeistof**, V_{ond} het volume **onder** het oppervlak.

Archimedes beschreef die wet in de derde eeuw voor onze tijdrekening, in zijn boek *Over drijvende lichamen*. Nu kan je F7 verklaren. Op een voorwerp in een vloeistof werken twee krachten: het gewicht $\rho_{\text{v}}\, g\, V$ omlaag en de opwaartse kracht $\rho_{\text{vl}}\, g\, V$ omhoog (helemaal ondergedompeld).

- $\rho_{\text{v}} > \rho_{\text{vl}}$: het gewicht wint, het voorwerp **zinkt**.
- $\rho_{\text{v}} = \rho_{\text{vl}}$: evenwicht, het voorwerp **zweeft**.
- $\rho_{\text{v}} < \rho_{\text{vl}}$: het voorwerp stijgt tot een deel boven komt, en **drijft** als $F_A = F_z$: $\rho_{\text{vl}}\, V_{\text{ond}} = \rho_{\text{v}}\, V$, dus $\frac{V_{\text{ond}}}{V} = \frac{\rho_{\text{v}}}{\rho_{\text{vl}}}$. Precies de regel uit F7.

De opwaartse kracht hangt af van hoe zwaar het voorwerp is

Een loden en een aluminium blokje van hetzelfde volume krijgen onder water dezelfde opwaartse kracht: het water 'weet' niet waarvan het blokje gemaakt is, alleen hoeveel plaats het inneemt. Het loden blokje zinkt toch, omdat zijn gewicht veel groter is.



De druk op het ondervlak is groter dan op het bovenvlak: het verschil is de opwaartse kracht.

PROEF · 11.12

Een steen wegen onder water

12/23

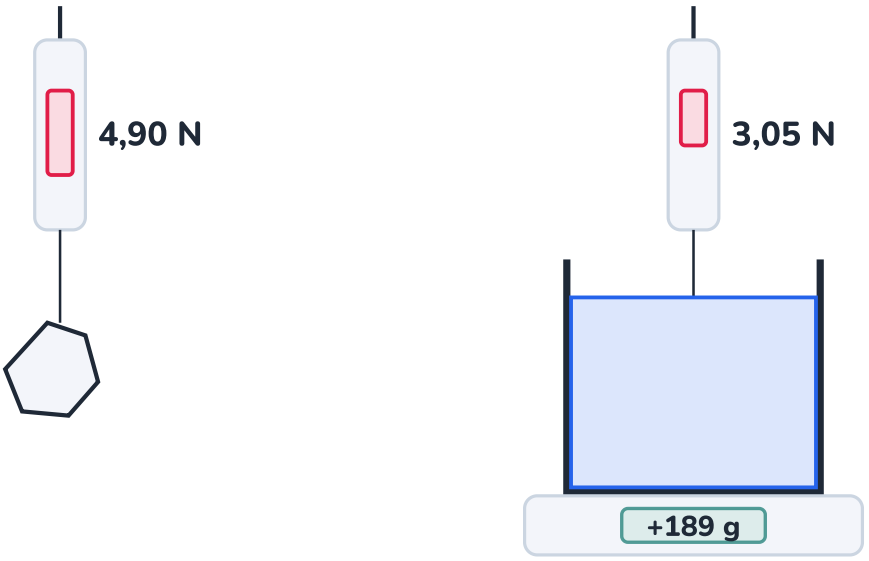
ONDERZOEKSVRAAG

Hoe groot is de opwaartse kracht op een steen, en kan je er het volume en de dichtheid van de steen mee bepalen? Wat toont een weegschaal onder de beker als je de steen onderdompelt?

MATERIAAL

dynamometer van 5 N | steen of metalen blok aan een dun touwtje (ongeveer 500 g) | beker met water op een keukenweegschaal

zout (voor de tweede meting)



in de lucht

onder water, beker op een weegschaal

De dynamometer toont 1,85 N minder, de weegschaal 189 g meer: dat is de opwaartse kracht, twee keer gemeten.

WERKWIJZE

1. Hang de steen aan de dynamometer en lees zijn gewicht in de lucht af.
2. Zet de weegschaal met de beker water op nul.
3. Laat de steen helemaal onder water zakken, zonder de bodem of de wand te raken. Lees de dynamometer en de weegschaal af.
4. Los 100 g zout op per liter water en herhaal.
5. Bereken de opwaartse kracht, het volume en de dichtheid van de steen.

WAARNEMING

	dynamometer	weegschaal onder de beker
in de lucht	4,90 N	0 g
onder in water	3,05 N	+189 g
onder in zout water	2,86 N	+208 g

VERKLARING

De opwaartse kracht in water is $4{,}90 - 3{,}05 = 1{,}85\,\mathrm{N}$. Dat is het gewicht van de verplaatste vloeistof, dus $V = \frac{F_A}{\rho\, g} = \frac{1{,}85}{1000 \cdot 9{,}81} = 1{,}89 \cdot 10^{-4}\,\mathrm{m^3} = 189\,\mathrm{cm^3}$, en de dichtheid van de steen is $500\,\mathrm{g} / 189\,\mathrm{cm^3} = 2{,}64\,\mathrm{g/cm^3}$: ongeveer die van kwarts of graniet. Het volume van een onregelmatig voorwerp meten zonder overloopvat: zo deed Archimedes het volgens Galilei met de krans (F7).

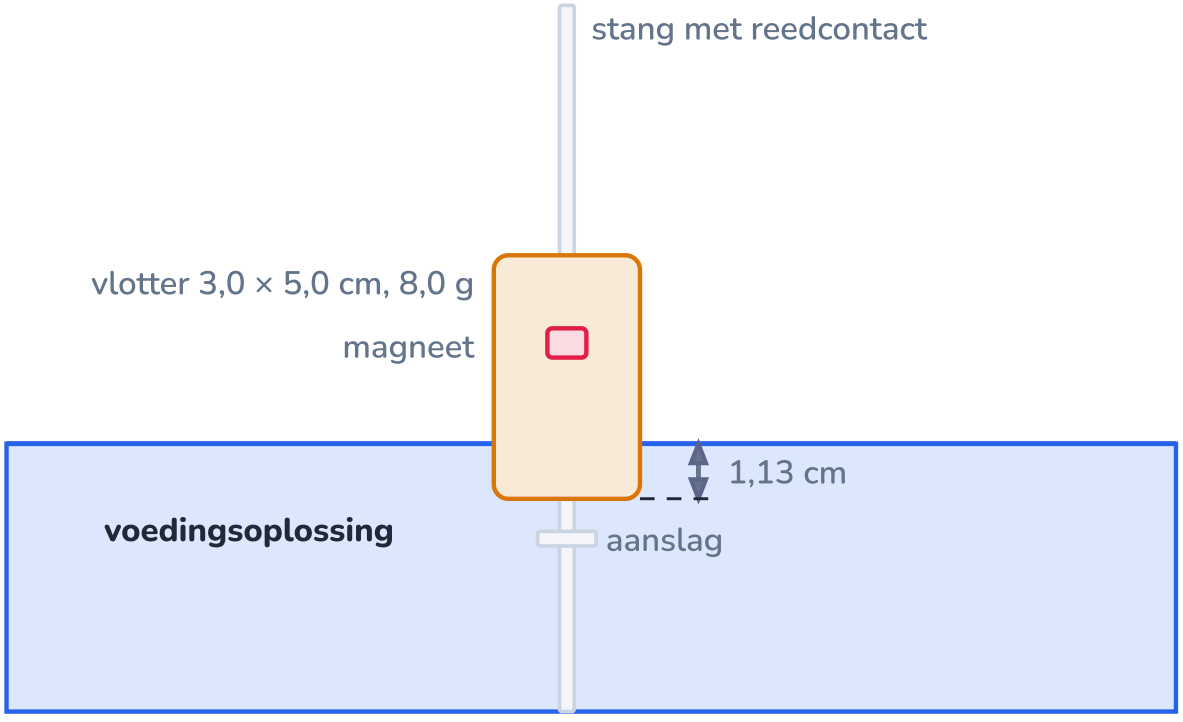
In zout water (ongeveer 1,10 g/cm³) is de opwaartse kracht groter: $1100 \cdot 9{,}81 \cdot 1{,}89 \cdot 10^{-4} = 2{,}04\,\mathrm{N}$. En de weegschaal? Het water duwt de steen omhoog, dus duwt de steen het water omlaag (derde wet, hoofdstuk 8): de weegschaal stijgt met precies $F_A / g = 189\,\mathrm{g}$ (in zout water 208 g). Wat de dynamometer minder toont, toont de weegschaal meer.

UITGEWERKT VOORBEELD · 11.13

Habitat: de vlotter in het reservoir

13/23

Als tweede beveiliging naast de druksensor krijgt het voedingsreservoir een vlotterschakelaar: een holle kunststof cilinder (diameter 3,0 cm, hoogte 5,0 cm, massa 8,0 g) met een magneetje, die langs een stang schuift en een reedcontact in de stang schakelt. Hoe diep zakt de vlotter in de voedingsoplossing? Met welke kracht duwt hij tegen de bovenste aanslag als het water te hoog komt en hem helemaal onderdompelt?



De vlotter drijft met 1,13 cm onder water; het magneetje schakelt het reedcontact in de stang.

GEGEVEN

- cilinder: $r = 1{,}5\,\text{cm}$, $h = 5{,}0\,\text{cm}$, $m = 8{,}0\,\text{g}$
- $\rho_{\text{vl}} = 1{,}00\,\text{g/cm}^3$

GEVRAAGD

- diepgang d
- kracht bij volledige onderdompeling

OPLOSSING

1 Volume: $V = \pi\, r^2\, h = \pi \cdot 1{,}5^2 \cdot 5{,}0 = 35{,}3\,\text{cm}^3$; gemiddelde dichtheid $8{,}0/35{,}3 = 0{,}23\,\text{g/cm}^3$: de vlotter drijft.

2 Drijven: $F_A = F_z$, dus de vlotter verplaatst 8,0 g oplossing, $8{,}0\,\text{cm}^3$. Diepgang: $d = \dfrac{8{,}0}{\pi \cdot 1{,}5^2} = \dfrac{8{,}0}{7{,}07} = 1{,}13\,\text{cm}$.

3 Helemaal onder: $F_A = 1000 \cdot 9{,}81 \cdot 35{,}3 \cdot 10^{-6} = 0{,}346\,\text{N}$, gewicht $0{,}008 \cdot 9{,}81 = 0{,}078\,\text{N}$. De vlotter duwt tegen de aanslag met $0{,}346 - 0{,}078 = 0{,}268\,\text{N}$.

ANTWOORD

De vlotter steekt 1,13 cm in de oplossing en kan, helemaal ondergedompeld, met 0,27 N tegen de aanslag duwen.

Klopt dit? Omdat de vlotter maar 1,13 cm diep zit, volgt hij het wateroppervlak op ongeveer een centimeter na: genoeg voor een alarm. De voedingszouten veranderen de dichtheid met maar 0,1 %: dat verschuift de diepgang minder dan een honderdste millimeter. In olie (0,9 g/cm³) zou hij 1,26 cm diep zakken.

UITLEG · 11.14

Drijven, zweven en stabiel blijven

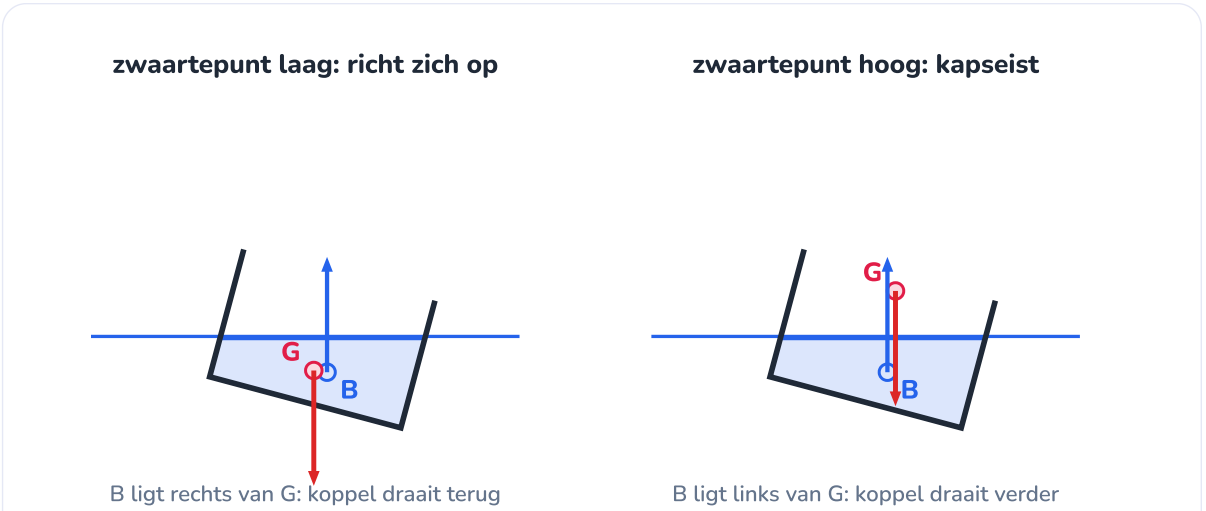
14/23

Een stalen schip drijft omdat zijn romp veel lucht omsluit: de gemiddelde dichtheid van schip en lucht is kleiner dan die van water (F7). Hoe zwaarder het geladen wordt, hoe dieper het zakt, tot het evenveel water verplaatst als het weegt. Een **duikboot** regelt zijn gemiddelde dichtheid met ballasttanks: laat hij er water in, dan zinkt hij; blaast hij ze leeg met perslucht, dan stijgt hij. Een vis doet hetzelfde met zijn zwemblaas. Ook in lucht werkt Archimedes: een heliumballon stijgt omdat de lucht die hij verplaatst zwaarder is dan de ballon met het helium.

Drijven is niet genoeg: je moet ook rechtop blijven. Het gewicht grijpt aan in het zwaartepunt G van het schip, de opwaartse kracht in het zwaartepunt B van het verplaatste water (het drukkingspunt). Helt een schip over, dan verschuift B naar de kant die dieper ligt. Gewicht en opwaartse kracht vormen dan een **koppel** (hoofdstuk 10). Ligt G laag, dan draait dat koppel het schip terug. Ligt G te hoog, dan draait het het schip verder om.

De Vasa

Op 10 augustus 1628 vertrok het Zweedse oorlogsschip Vasa op zijn eerste tocht, in de haven van Stockholm. Na ruim een kilometer helde het in een windvlaag over, water stroomde door de open geschutspoorten naar binnen, en het schip zonk. Het was te hoog gebouwd, met twee dekken vol zware bronzen kanonnen, en had te weinig ballast onderin: het zwaartepunt lag te hoog. In 1961 werd het wrak geborgen; het staat nu in een museum in Stockholm.



Twee dezelfde rompen, 15° overgeheld: alleen de hoogte van G verschilt, en die beslist.

Een schip vol kanonnen boven de waterlijn: indrukwekkend, maar met een te hoog zwaartepunt.

VERHAAL · 11.15

Een zee van lucht

15/23

Waarom komt water omhoog in een rietje of in de buis van een zuigpomp? Tweeduizend jaar lang was het antwoord: de natuur heeft een afkeer van leegte. Maar in 1638 schreef Galilei dat een zuigpomp water niet hoger dan ongeveer 18 Florentijnse ellen (zowat 10 m) kan optrekken. Waarom juist daar zou die afkeer ophouden?

In 1643 vulde zijn leerling Evangelista Torricelli een glazen buis van ongeveer een meter met kwik, sloot ze af, en zette ze omgekeerd in een schaal kwik. Het kwik zakte tot ongeveer 76 cm boven de schaal, en bleef daar staan; erboven was niets. Torricelli begreep waarom: de lucht weegt, en duwt op het kwik in de schaal. Die druk houdt een kolom van 76 cm kwik in evenwicht, of zowat 10 m water, want water is 13,6 keer lichter. In een brief schreef hij dat we op de bodem van een zee van lucht leven.

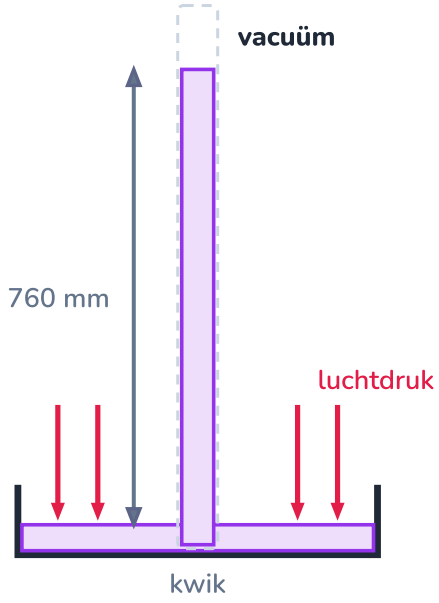
Als de lucht weegt, redeneerde Blaise Pascal, moet de kolom op een berg lager zijn: daar ligt er minder lucht boven je. Op 19 september 1648 beklom zijn schoonbroer Florin Périer de Puy de Dôme, een vulkaan van 1465 m bij Clermont. Beneden stond het kwik op 711 mm, bovenaan zo'n 85 mm lager. De eenheid van druk draagt vandaag de naam van Pascal.

Otto von Guericke, burgemeester van Maagdenburg, bouwde een luchtpomp en toonde ze in 1654 aan keizer Ferdinand III. Hij pompte de lucht uit twee koperen halve bollen die tegen elkaar lagen. Volgens zijn boek uit 1672 kregen zestien paarden, acht aan elke kant, ze niet uit elkaar. Liet hij de lucht terug binnen, dan vielen ze vanzelf los.

Wat je uit dit verhaal meeneemt

Zuigen bestaat eigenlijk niet: je maakt de druk aan één kant kleiner, en de luchtdruk aan de andere kant duwt. Die luchtdruk is enorm, 10 ton per vierkante meter, maar je merkt ze niet omdat ze van alle kanten even hard duwt, ook van binnenuit.

Twee halve bollen, een pomp en geen lucht ertussen: de luchtdruk houdt ze tegen elkaar.



Boven het kwik is het (bijna) leeg; de luchtdruk op de schaal draagt 760 mm kwik.

UITLEG · 11.16

De atmosferische druk

De lucht boven ons weegt. Op zeeniveau is de gemiddelde luchtdruk

STANDAARDATMOSFEER

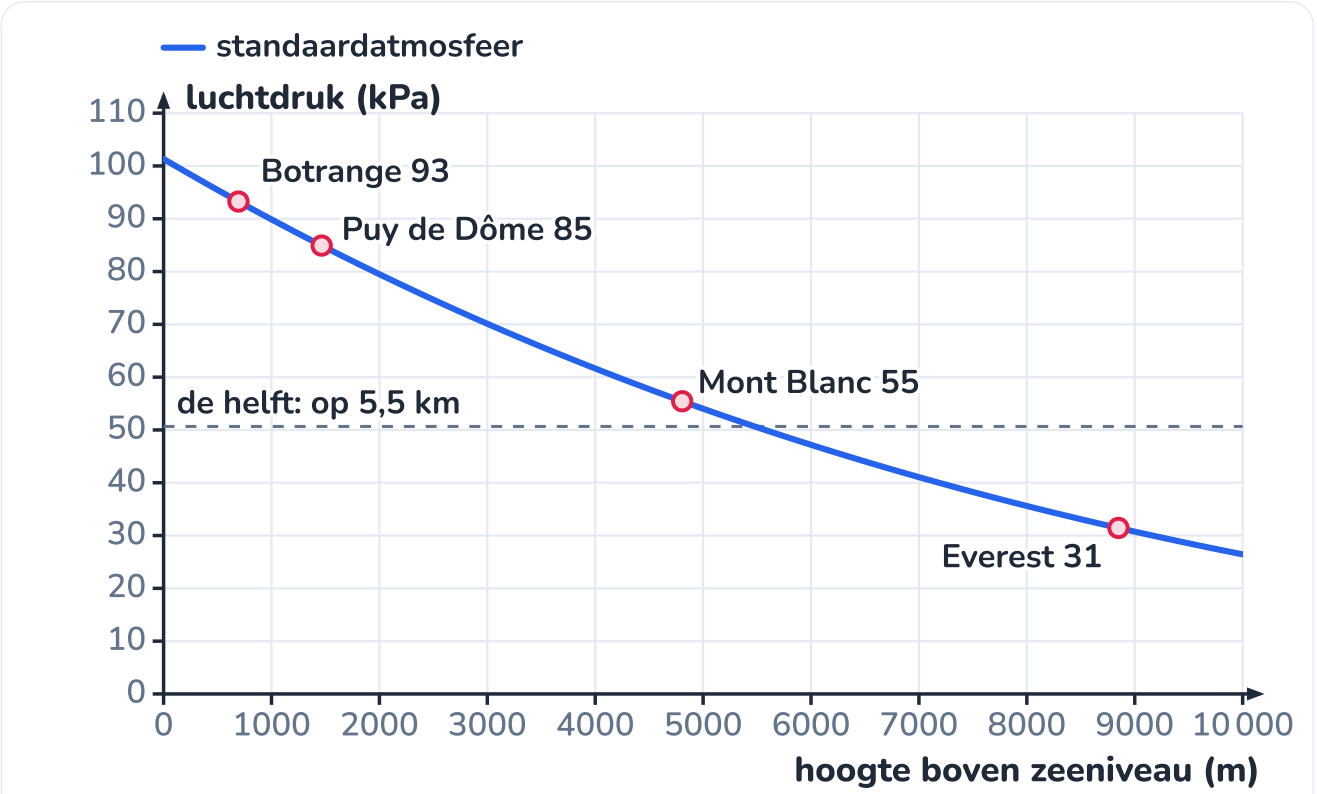
$$p_0 = 101\,325\,\text{Pa} = 1013{,}25\,\text{hPa} \approx 1\,\text{bar}$$

De druk van een kwikkolom van 760 mm, of een waterkolom van 10,3 m. Op elke vierkante meter drukt het gewicht van 10,3 ton lucht.

Waarom word je dan niet platgedrukt? Omdat de druk in een vloeistof of gas in alle richtingen werkt, en omdat de lucht en de vloeistoffen in je lichaam met dezelfde druk terugduwen. Pas als de druk aan één kant verandert, merk je iets: je oren 'ploppen' in een lift of een vliegtuig, een plastic fles die je op een berg leegdrinkt en sluit, wordt beneden ingedeukt.

Het weer. De luchtdruk schommelt met het weer, meestal tussen ongeveer 980 en 1040 hPa. Een barometer toont dat; op een weerkaart verbinden isobaren punten met dezelfde druk. Hogedrukgebieden brengen meestal droog weer, lagedrukgebieden wind en regen.

De hoogte. Hoe hoger, hoe minder lucht er boven je ligt, en hoe dunner de lucht zelf. Onderin daalt de druk ongeveer 12 hPa per 100 m. De afname is niet lineair: op 5,5 km is de druk gehalveerd, op de top van de Mount Everest is ze nog een derde. De grafiek toont de internationale standaardatmosfeer, een model met 15 °C op zeeniveau; de echte druk wijkt daar met het weer een paar procent van af.



Hoe hoger, hoe trager de druk nog daalt: de lucht zelf wordt dunner.

PROEF · 11.17

Luchtdruk meten op de trap

17/23

ONDERZOEKSVRAAG

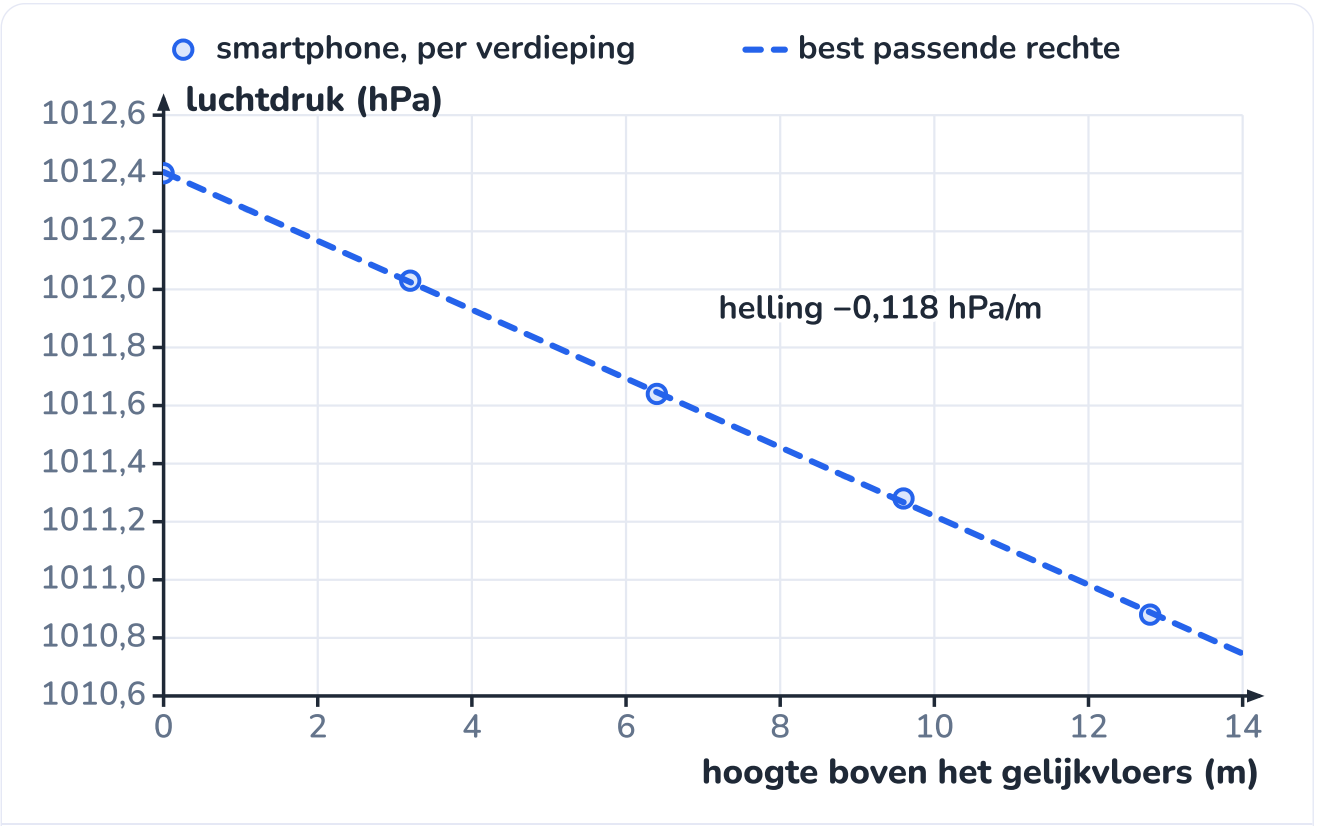
Kan je met een smartphone meten dat de luchtdruk afneemt als je een paar verdiepingen hoger gaat? En kan je er de dichtheid van lucht uit halen?

MATERIAAL

- smartphone met een barometer en een app die hem uitleest (bijvoorbeeld phyphox, 'Druk')
- een trappenhal met minstens vier verdiepingen
- rolmeter

Veiligheid

Loop rustig; de meting heeft tijd nodig, niet snelheid.



Voorbeeldmeting: vijf punten op een rechte; de helling is ρ · g voor lucht.

WERKWIJZE

1. Meet de hoogte van één trede en tel de treden per verdieping; bereken de hoogte van elke verdieping.
2. Leg de smartphone op het gelijkvloers op de trap, wacht 20 s en noteer de gemiddelde druk.
3. Herhaal op elke verdieping, telkens met de telefoon even hoog boven de trede en uit de tocht.
4. Zet de druk uit tegen de hoogte en bepaal de helling van de best passende rechte (hoofdstuk 3).

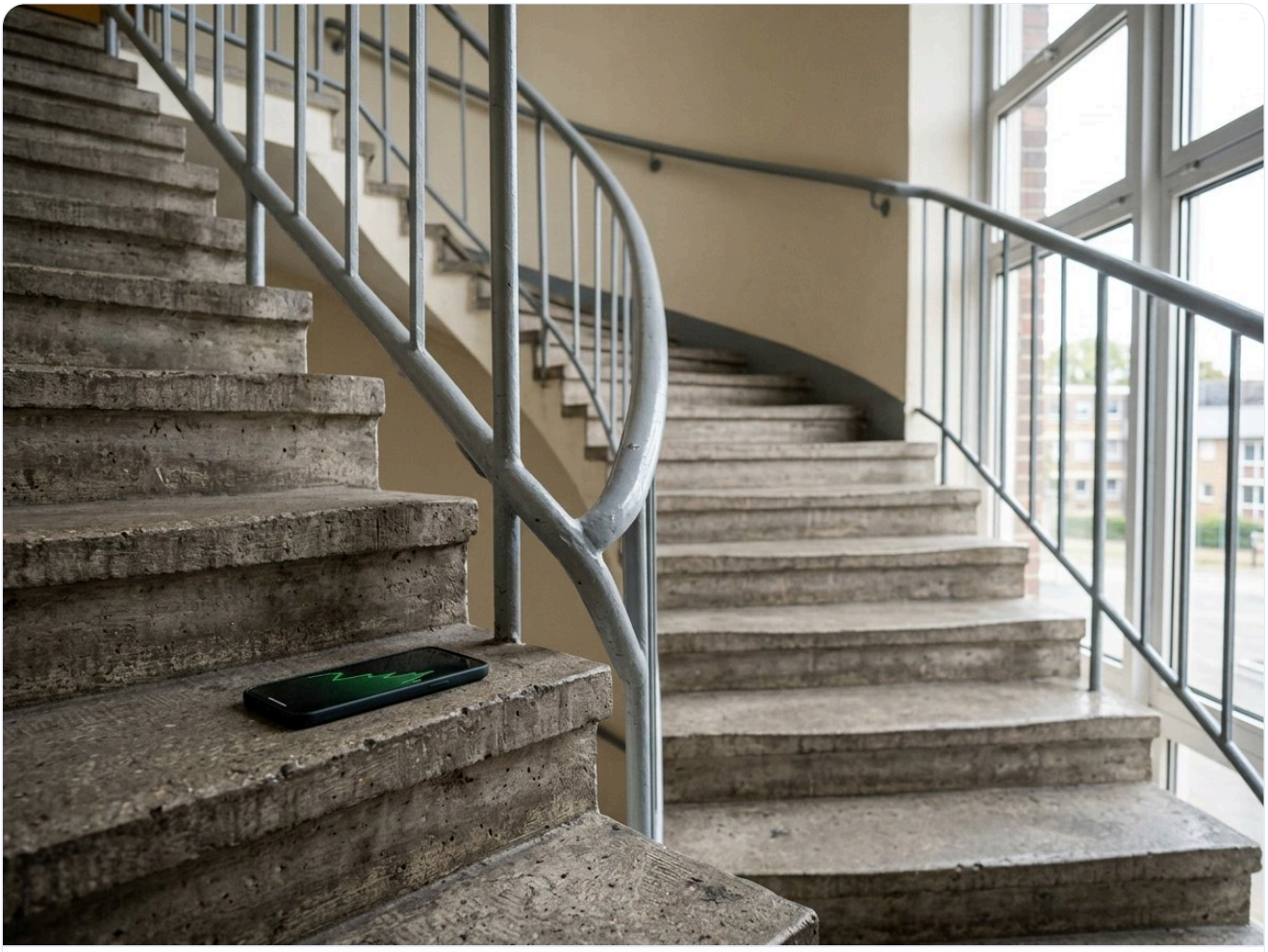
WAARNEMING

hoogte (m)	0,0	3,2	6,4	9,6	12,8
luchtdruk (hPa)	1012,40	1012,03	1011,64	1011,28	1010,88

Per verdieping daalt de druk met ongeveer 0,38 hPa; de punten liggen op een rechte met helling $-0{,}118\,\mathrm{hPa/m} = -11{,}8\,\mathrm{Pa/m}$.

VERKLARING

Ook in lucht geldt $\Delta p = \rho\, g\, \Delta h$: over een paar verdiepingen verandert de dichtheid van de lucht niet merkbaar. Uit de helling volgt $\rho = \dfrac{11{,}8}{9{,}81} = 1{,}21\,\mathrm{kg/m^3}$, dicht bij de tabelwaarde van 1,20 kg/m³ bij 20 °C. Water is ongeveer 830 keer dichter: daarom merk je het drukverschil in een zwembad van 3 m meteen, en op een trap van 13 m niet. Smartphones gebruiken hun barometer om te bepalen op welke verdieping je bent. Wacht je tien minuten en meet je opnieuw, dan is de druk misschien door het weer al een paar tienden hPa veranderd: meet daarom snel na elkaar, of begin en eindig op het gelijkvloers.



De barometer in een smartphone merkt het verschil tussen twee verdiepingen.

UITLEG · 11.18

Overdruk en absolute druk

De meeste drukmeters tonen niet de druk zelf, maar het **verschil met de luchtdruk**. Een bandenspanningsmeter die 2,2 bar toont, meet 2,2 bar **overdruk**: in de band is de **absolute druk** $2{,}2 + 1{,}0 = 3{,}2\,\text{bar}$. Een lege band is niet leeg: er zit lucht in van 1 bar absoluut, en de meter toont 0.

ABSOLUTE DRUK EN OVERDRUK

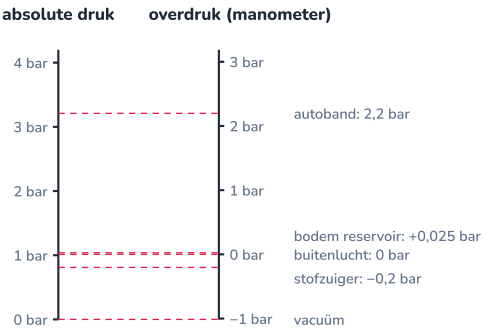
$$p_{\text{abs}} = p_0 + p_{\text{over}}$$

p_0 de luchtdruk. Een negatieve overdruk heet onderdruk: in een stofzuiger, in je mond als je aan een rietje zuigt. De absolute druk kan nooit onder nul: daar is vacuüm.

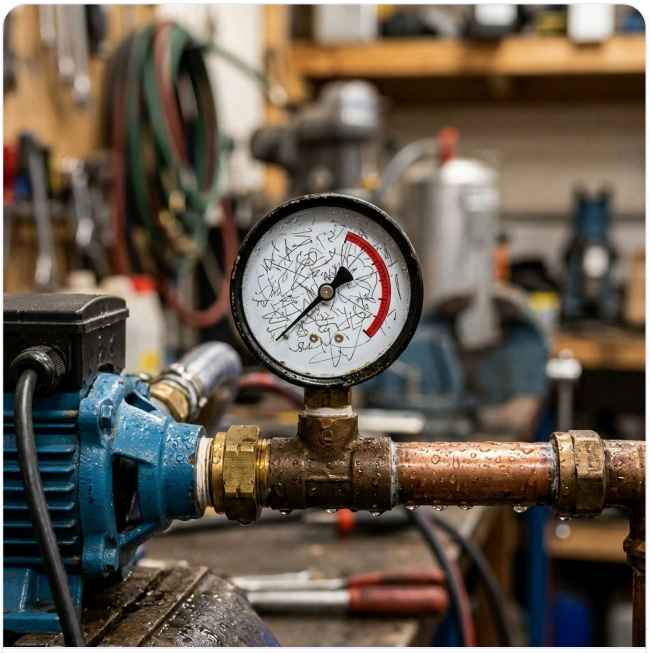
Een eenvoudige manometer is een **U-buis** met water: het ene been hangt aan het gas, het andere staat open. Het hoogteverschil geeft de overdruk, $p_{\text{over}} = \rho\, g\, \Delta h$. Een aardgasleiding in huis heeft een overdruk van ongeveer 20 à 25 mbar: 20 à 25 cm water in de U-buis. De ronde wijzermanometer op een compressor of een pomp bevat een gebogen buisje (de buis van Bourdon) dat zich strekt als de druk erin stijgt, en de wijzer meeneemt. Ook je **bloeddruk** is een overdruk: 120 mmHg is $120 \cdot 133{,}3 = 16\,\text{kPa}$ boven de luchtdruk.

Welke druk staat in de formule?

Voor krachten op een wand (een reservoir, een band, een duikbril) telt het drukverschil tussen binnen en buiten: meestal de overdruk. Voor gassen (hoofdstuk E14) reken je met de absolute druk.



Links vanaf het vacuüm, rechts vanaf de luchtdruk: de twee schalen zijn 1,013 bar verschoven.



Een wijzermanometer toont overdruk: bij open lucht staat hij op nul.

UITLEG · 11.19

Pompen en leidingen

19/23

Een pomp maakt een **drukverschil**: aan de persopening is de druk hoger dan aan de zuigopening. Die overdruk duwt het water door de leiding omhoog. Pompbouwers geven het drukverschil meestal als **opvoerhoogte**: de hoogte van de waterkolom die het kan dragen.

OPVOERHOOGTE EN DRUK

$$\Delta p = \rho \cdot g \cdot H \qquad 1\,\text{bar} \approx 10{,}2\,\text{m waterkolom}$$

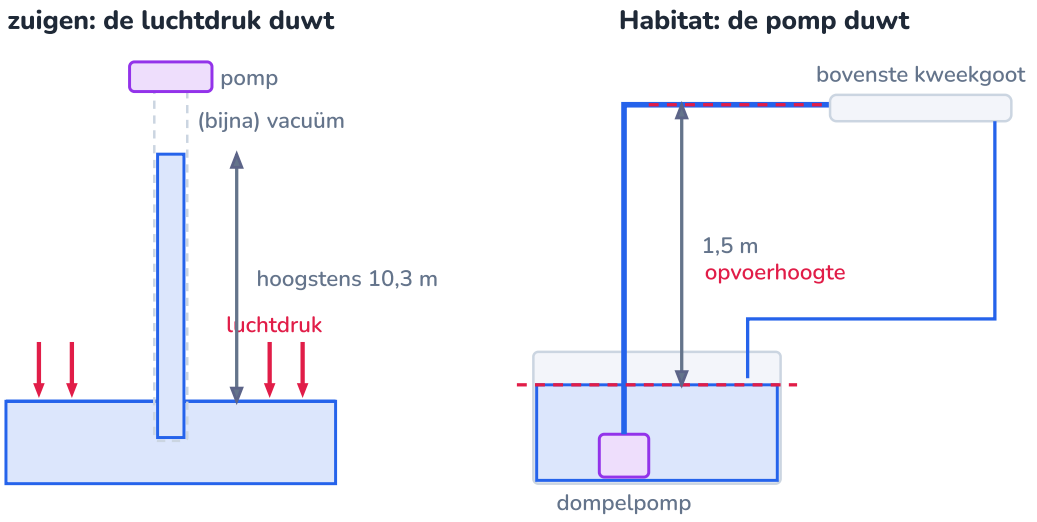
De pomp van Habitat (1,5 m) maakt $1000 \cdot 9{,}81 \cdot 1{,}5 = 14{,}7\,\text{kPa}$ overdruk; een pomp voor een leidingnet van 4 bar heeft een opvoerhoogte van ongeveer 40 m.

Zuigen tot 10 m. Een pomp boven een put 'zuigt' niet: ze maakt de druk in de zuigbuis kleiner, en de luchtdruk op het water in de put duwt het omhoog. Zelfs met een perfect vacuüm draagt de luchtdruk maar 10,3 m water. In de praktijk halen zuigpompen 7 à 8 m: bij lage druk begint het water te koken en vormen er dampbellen. Diepe putten krijgen daarom een **dompelpomp** onderin, die duwt. Ook in het reservoir van Habitat staat een dompelpomp.

Verlies in de leiding. Stromend water wrijft tegen de wand en tegen zichzelf. Daardoor gaat er druk verloren, alsof de pomp het water hoger moest brengen. Dat verlies groeit ongeveer met het kwadraat van de snelheid, en de snelheid met het omgekeerde van de doorsnede: een dunnere slang kost veel meer. Voor de slang van Habitat (2,5 m lang, bij 5,0 L/min, berekend met de formules van hoofdstuk E18):

binnendiameter van de slang	8 mm	10 mm	13 mm	16 mm
snelheid van het water bij 5,0 L/min	1,66 m/s	1,06 m/s	0,63 m/s	0,41 m/s
extra opvoerhoogte (2,5 m slang)	1,57 m	0,56 m	0,17 m	0,07 m

Een pomp levert dus niet één debiet, maar een debiet dat afhangt van de opvoerhoogte die het systeem vraagt: de **pompkarakteristiek**. Hoe hoger ze moet duwen, hoe minder water ze verplaatst.



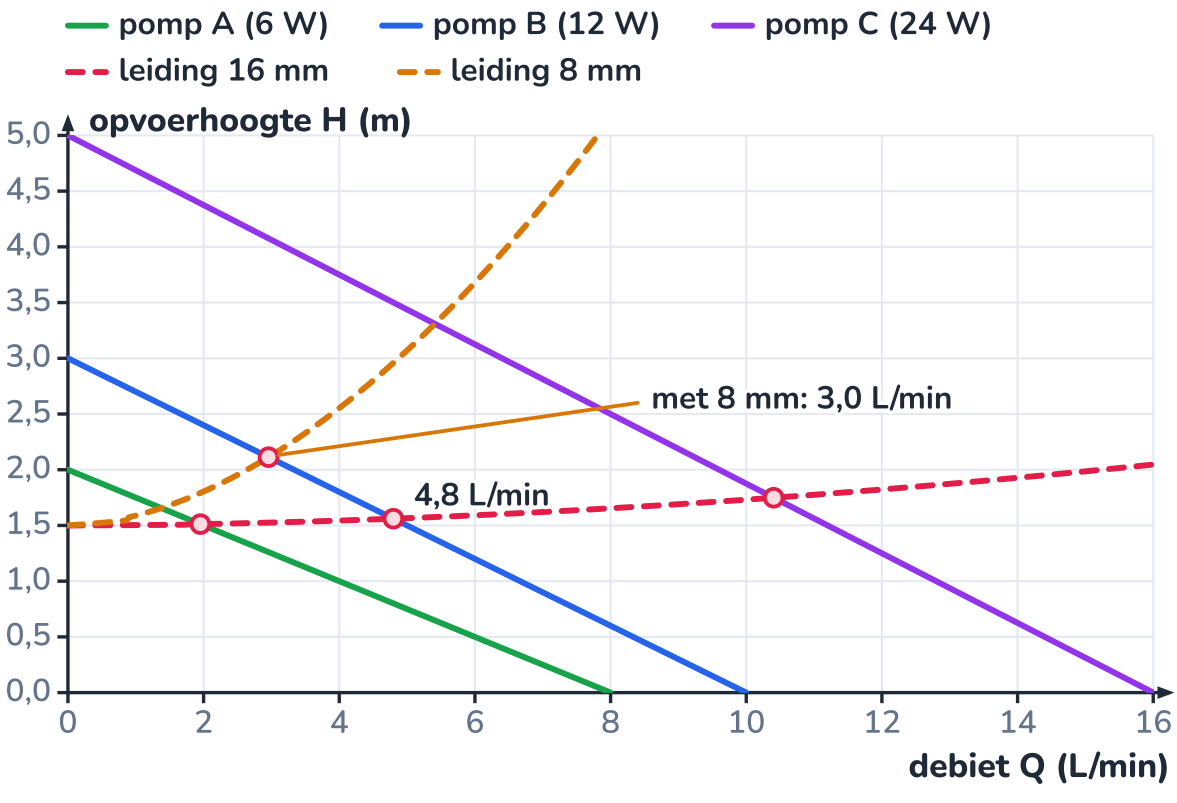
Links draagt de luchtdruk het water omhoog, tot hoogstens 10,3 m. Rechts duwt de pomp; de opvoerhoogte telt vanaf het wateroppervlak.

UITGEWERKT VOORBEELD · 11.20

Habitat: welke pomp, welke slang?

20/23

Voor de waterkringloop liggen drie 12 V-pompen op tafel, met hun karakteristiek (vereenvoudigd als rechte): A (6 W) tot 2,0 m en 8 L/min; B (12 W, de pomp uit hoofdstuk 9) tot 3,0 m en 10 L/min; C (24 W) tot 5,0 m en 16 L/min. Het water moet 1,5 m hoog, van het wateroppervlak in het reservoir tot de bovenste kweekgoot, door een slang van 2,5 m. Het team wil ongeveer 5 L/min. Welke pomp, en welke slang: 8 mm of 16 mm binnendiameter?



Drie pompen (volle lijnen), twee slangen (stippellijnen); de punten zijn de werkpunten.

GEGEVEN

- pompkarakteristieken A, B, C (grafiek)
- hoogteverschil 1,5 m; slang 2,5 m, 8 of 16 mm; leidingverlies uit de tabel van het vorige blad

GEVRAAGD

- het werkpunt (debiet en opvoerhoogte) van elke combinatie
- de keuze

OPLOSSING

1. De **leidingkarakteristiek** is de opvoerhoogte die het systeem vraagt bij elk debiet: 1,5 m plus het leidingverlies. Bij 0 L/min is het 1,5 m; met de slang van 16 mm stijgt ze nauwelijks (0,07 m extra bij 5 L/min), met 8 mm snel (1,6 m extra).
2. Het **werkpunt** is het snijpunt van pomp- en leidingkarakteristiek: daar levert de pomp precies de opvoerhoogte die het systeem bij dat debiet vraagt.
3. Pomp A haalt met 16 mm maar 2,0 L/min: te weinig. Pomp C haalt 10,4 L/min: ruim twee keer te veel, voor twee keer het vermogen.
4. Pomp B met 16 mm: 4,8 L/min bij 1,56 m. Met 8 mm valt ze terug tot 3,0 L/min: de dunne slang 'knijpt' de pomp af.

ANTWOORD

Pomp B met een slang van 16 mm: 4,8 L/min, net onder de 5,0 L/min uit hoofdstuk 9 (dat was het debiet bij 1,5 m zonder leidingverlies). De 60 L één keer rondpompen duurt dan 12,5 minuten in plaats van 12.

Klopt dit? Een dikkere slang kost bijna niets en spaart energie: met 8 mm zou je voor hetzelfde debiet een pomp als C nodig hebben, met het dubbele verbruik. Hoe je pomp- en leidingkarakteristieken zelf berekent, leer je in hoofdstuk E18. In opdracht 11.14 kijk je wat er gebeurt als het niveau in het reservoir zakt.

CODE · 11.21

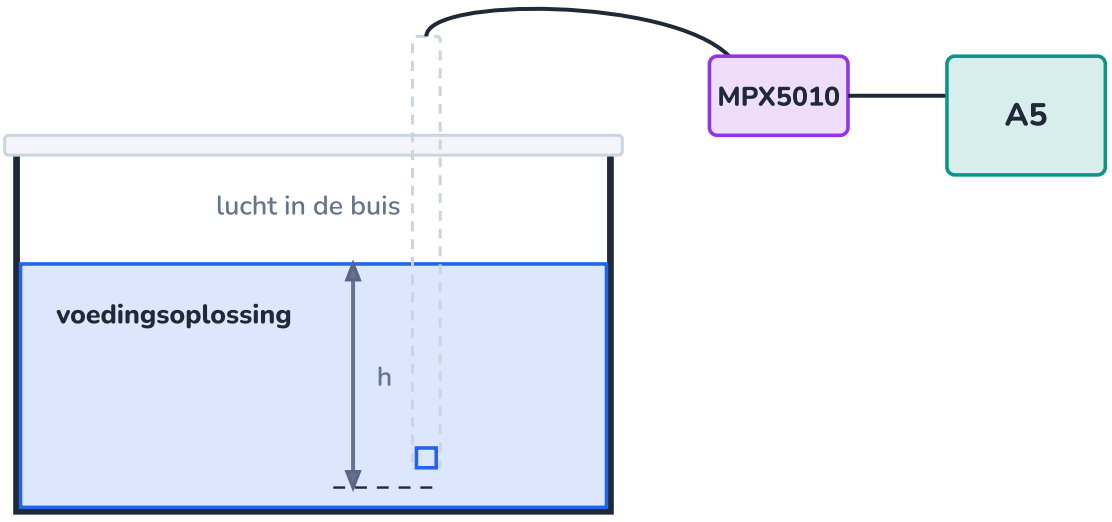
Python en C++: het niveau meten met een druksensor

21/23

Het reservoir is lichtdicht, dus meet je het niveau met de druk. Een stijve buis hangt tot vlak boven de bodem; de lucht erin wordt samengedrukt tot haar druk gelijk is aan de waterdruk op die diepte, en geeft die door naar een droge druksensor op het deksel (Pascal). De sensor MPX5010 meet overdruk van 0 tot 10 kPa en geeft volgens zijn datasheet $U = V_S\,(0{,}09\,p + 0{,}04)$, met p in kPa en $V_S = 5\,\mathrm{V}$: 0,2 V bij 0 kPa, 450 mV per kPa. De microcontroller leest die spanning met een ADC van 10 bit (hoofdstuk 25) als een getal van 0 tot 1023. Het programma rekent terug: getal, spanning, druk, hoogte, volume.

Python **niveau.py** 16 regels

```python
# niveau.py - waterhoogte en volume uit de druksensor onderaan het voedingsreservoir
RHO, G = 1000.0, 9.81          # dichtheid van de oplossing (kg/m3) en valversnelling (m/s2)
VS = 5.0                       # voedingsspanning van de sensor (V)
BODEM = 0.60 * 0.40            # binnenoppervlakte van het reservoir (m2)


def niveau(ruw):
    u = ruw * VS / 1023                  # ADC-waarde (0 tot 1023) naar volt
    p = (u / VS - 0.04) / 0.09 * 1000    # datasheet: druk in kPa; maal 1000 geeft Pa
    h = p / (RHO * G)                    # hydrostatische druk: p = rho * g * h
    return u, p, h, h * BODEM * 1000     # volt, Pa, m, liter


for ruw in (41, 86, 154, 222, 267):
    u, p, h, liter = niveau(ruw)
    print(f"ADC {ruw:4d}   {u:5.3f} V   {p:6.0f} Pa   {h * 100:5.1f} cm   {liter:5.1f} L")
```



De lucht in de buis geeft de waterdruk door; de sensor blijft droog, boven op het deksel.

Uitvoer (gecontroleerd door het programma te draaien):

```
ADC   41   0.200 V        1 Pa     0.0 cm     0.0 L
ADC   86   0.420 V      490 Pa     5.0 cm    12.0 L
ADC  154   0.753 V     1228 Pa    12.5 cm    30.0 L
ADC  222   1.085 V     1967 Pa    20.0 cm    48.1 L
ADC  267   1.305 V     2456 Pa    25.0 cm    60.1 L
```

Eén stap van de ADC is 4,9 mV, dus 0,011 kPa of 1,1 mm water: fijn genoeg. Maar de datasheet belooft maar $\pm 5\,\%$ van het bereik, tot 0,5 kPa of 5 cm water fout. Daarom kalibreer je: je leest de sensor af bij een leeg reservoir en trekt dat getal voortaan af (hoofdstuk 1 en 3; opdracht 11.11). Op de microcontroller ziet dezelfde berekening er zo uit:

```
// niveau.ino - dezelfde berekening op de microcontroller
const float RHO = 1000.0, G = 9.81, VS = 5.0, BODEM = 0.60 * 0.40;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int ruw = analogRead(A5);                    // sensoruitgang op A5 (Mega)
  float u = ruw * VS / 1023.0;                 // volt
  float p = (u / VS - 0.04) / 0.09 * 1000.0;   // overdruk in Pa
  float h = p / (RHO * G);                     // waterhoogte in m
  Serial.print(h * 100.0);
  Serial.print(" cm, ");
  Serial.print(h * BODEM * 1000.0);
  Serial.println(" L");
  delay(1000);
}
```

Overzicht: druk en hydraulica

druk

$$p = \frac{F}{A}$$

symbool	betekenis	eenheid
p	druk	Pa = N/m²
F	kracht loodrecht op het oppervlak	N
A	oppervlakte	m²

hydrostatische druk

$$p = \rho\, g\, h \qquad p_{\text{abs}} = p_0 + \rho\, g\, h$$

symbool	betekenis	eenheid
ρ	dichtheid van de vloeistof	kg/m³
h	diepte onder het oppervlak	m
p_0	luchtdruk, 101 325 Pa (standaard)	Pa

wet van Pascal: hydraulische pers

$$\frac{F_1}{A_1} = \frac{F_2}{A_2} \qquad A_1\, s_1 = A_2\, s_2$$

wet van Archimedes

$$F_A = \rho_{\text{vl}}\, g\, V_{\text{ond}} \qquad \frac{V_{\text{ond}}}{V} = \frac{\rho_{\text{v}}}{\rho_{\text{vl}}}\ \text{(drijven)}$$

overdruk, absolute druk en opvoerhoogte

$$p_{\text{abs}} = p_0 + p_{\text{over}} \qquad \Delta p = \rho\, g\, H$$

eenheid	in pascal	waar je ze ziet
1 Pa = 1 N/m²	1	SI-eenheid
1 hPa = 1 mbar	100	weerkaarten (1013 hPa)
1 kPa	1000	techniek, sensoren
1 bar	100 000	banden, leidingwater, duikflessen
1 MPa	1 000 000	hydrauliek, sterkte van materialen
1 mmHg (millimeter kwik)	133,3	bloeddruk (120/80 mmHg)
1 standaardatmosfeer	101 325	= 1013,25 hPa = 760 mmHg

dichtheid	kg/m³
lucht (20 °C, zeeniveau)	1,20
helium (20 °C)	0,17
zoet water	1000
zeewater	1025
ijs	917
hydraulische olie	ongeveer 870
kwik	13 590

Habitat: rekenwaarden uit dit hoofdstuk	
voedingsreservoir	binnenmaten 60 × 40 × 30 cm; vol 25 cm (60 L): 2,45 kPa overdruk op de bodem, 184 N op een lange wand
niveaumeting	druksensor MPX5010 (0 tot 10 kPa, 0,2 V + 0,45 V/kPa); alarm bij 10 cm (24 L), plus vlotterschakelaar
pomp en slang	pomp B (12 W), slang 16 mm, 2,5 m: werkpunt 4,8 L/min bij 1,56 m
druk op de vloer	2,3 kPa op de volle bodem; nooit op dunne pootjes zonder plaat

HET GEHEEL · 11.23

Waar druk terugkomt

23/23

Druk verbindt de mechanica met de stoffen: ze is kracht per oppervlakte, en ze komt voort uit het gewicht van een vloeistof of het botsen van gasdeeltjes. In Expert wordt ze een van de hoofdrolspelers: in de gaswetten, in de warmtepomp en in de stroming door leidingen.

F6 Het deeltjesmodel
Gasdruk als het botsen van deeltjes tegen de wand.

F7 Massa, volume en dichtheid
Zinken, zweven en drijven, nu verklaard met de opwaartse kracht.

I8 De wetten van Newton
Normaalkracht en derde wet: de weegschaal onder de beker, de reactie van het water.

I9 Arbeid, energie en vermogen
De pomp en haar opvoerhoogte; de gulden regel bij de hydraulische pers.

I10 Moment en evenwicht
De waterdruk op een wand als driehoekige last; het koppel dat een schip terug rechtzet.

I22 De spanningsdeler en sensoren
Een sensor met een analoge uitgang inlezen en kalibreren, zoals de druksensor.

I36 Hydroponie en voedselproductie
Het hydroponiesysteem waarvoor de pomp en de niveaumeting dienen.

E14 Gassen en de gaswetten
Druk, volume en temperatuur van gassen; absolute druk in de ideale gaswet.

E16 Warmtepompen en koelen
Een warmtepomp werkt met een koelmiddel onder hoge en lage druk.

E18 Stroming van vloeistoffen en gassen
Bernoulli, drukverlies in leidingen en het werkpunt van een pomp, nu echt berekend.

Habitat

Voor Habitat leggen we vast: het voedingsreservoir heeft binnenmaten van 60 × 40 × 30 cm en is vol bij 25 cm (60 L); onderaan heerst dan 2,45 kPa overdruk. Het niveau wordt gemeten met een druksensor MPX5010 via een luchtbuis; onder 10 cm (24 L) gaat het alarm af, met een vlotterschakelaar als tweede beveiliging. De pomp uit hoofdstuk 9 (pomp B, 12 W) krijgt een slang van 16 mm binnendiameter en 2,5 m: werkpunt 4,8 L/min bij 1,56 m. In opdracht 11.14 reken je het hele systeem na.

SAMENVATTING

In het kort

- Druk is kracht per oppervlakte: $p = F/A$, in pascal ($1\,\mathrm{Pa} = 1\,\mathrm{N/m^2}$); $1\,\mathrm{bar} = 10^5\,\mathrm{Pa}$, $1\,\mathrm{hPa} = 100\,\mathrm{Pa}$.
- In een vloeistof in rust is $p = \rho\, g\, h$: ze hangt alleen af van de diepte, niet van de vorm van het vat, en werkt in alle richtingen. In water ongeveer 1 bar per 10 m.
- Communicerende vaten: dezelfde vloeistof in rust staat overal even hoog (peilbuis, slangwaterpas, watertoren).
- Wet van Pascal: een drukverandering in een ingesloten vloeistof plant zich overal voort. Een hydraulische pers vermenigvuldigt de kracht met A_2/A_1, en deelt de weg door hetzelfde getal.
- Wet van Archimedes: $F_A = \rho_{\mathrm{vl}}\, g\, V_{\mathrm{ond}}$. Een drijvend voorwerp verplaatst zijn eigen gewicht aan vloeistof; stabiel drijven vraagt een laag zwaartepunt.
- De luchtdruk is op zeeniveau gemiddeld 101 325 Pa; ze daalt met de hoogte (de helft op 5,5 km). Zuigen werkt daarom tot hoogstens 10,3 m water.
- Een manometer toont overdruk: $p_{\mathrm{abs}} = p_0 + p_{\mathrm{over}}$.

- Een pomp maakt een drukverschil $\rho\, g\, H$. Het werkpunt is het snijpunt van pomp- en leidingkarakteristiek; een te dunne leiding kost veel debiet.

BEGRIPPEN

Wat je nu kent

druk
Kracht loodrecht op een oppervlak, gedeeld door die oppervlakte: $p = F/A$, in pascal.

pascal
De SI-eenheid van druk: $1\,\mathrm{Pa} = 1\,\mathrm{N/m^2}$. $1\,\mathrm{bar} = 100\,000\,\mathrm{Pa}$, $1\,\mathrm{hPa} = 100\,\mathrm{Pa}$.

hydrostatische druk
De druk in een vloeistof in rust door het gewicht van de vloeistof erboven: $p = \rho\, g\, h$.

communicerende vaten
Vaten die onderaan verbonden zijn; een vloeistof in rust staat erin overal even hoog.

wet van Pascal
Een drukverandering in een ingesloten vloeistof plant zich onverminderd voort naar elk punt van de vloeistof.

opwaartse kracht
De kracht omhoog van een vloeistof of gas op een voorwerp erin: $F_A = \rho_{\mathrm{vl}}\, g\, V_{\mathrm{ond}}$ (wet van Archimedes).

atmosferische druk
De druk van de lucht rondom ons, door het gewicht van de lucht erboven; op zeeniveau gemiddeld 101 325 Pa.

overdruk
Het verschil tussen de druk en de atmosferische druk; dat is wat een manometer meestal toont.

absolute druk
De druk gemeten vanaf het vacuüm: overdruk plus atmosferische druk.

opvoerhoogte
De hoogte van een waterkolom die met het drukverschil van een pomp overeenkomt: $H = \Delta p/(\rho\, g)$.

werkpunt
Het snijpunt van de pompkarakteristiek en de leidingkarakteristiek: het debiet en de opvoerhoogte waarbij het systeem echt draait.

HANDBOEK · HOOFDSTUK 12

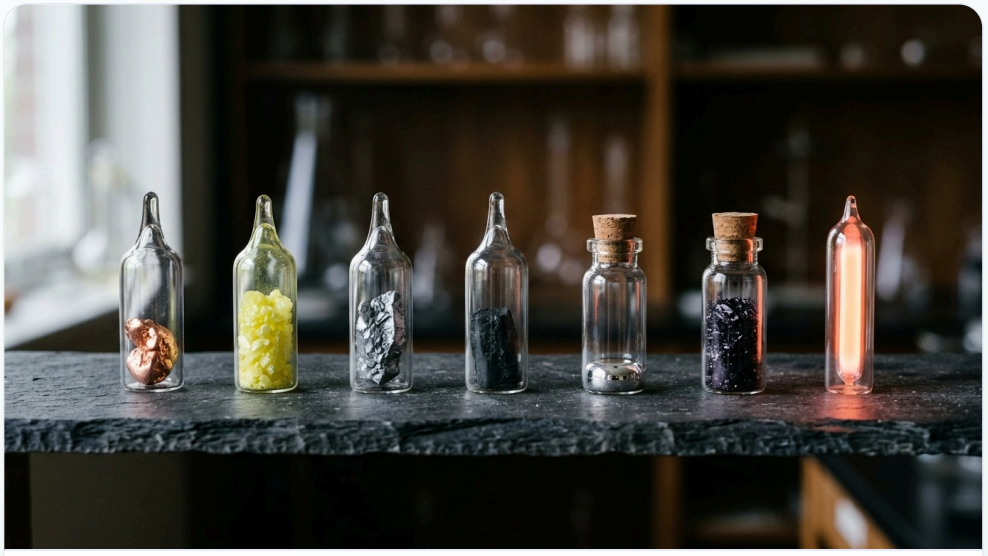
Atomen en het periodiek systeem

Wat er in een atoom zit, en waarom 118 elementen in één tabel passen

In F6 bestond alles uit deeltjes. Nu kijk je erin: een atoom is een piepkleine kern met protonen en neutronen, en daaromheen elektronen. Het aantal protonen bepaalt welk element het is, de elektronen in de buitenste schil bepalen hoe het reageert. Daarmee begrijp je de bouw van het **periodiek systeem**, de tabel die Mendelejev in 1869 opstelde en waarmee hij elementen voorspelde die nog niemand gezien had. Voor Habitat zoek je uit welke elementen een plant nodig heeft, en waar ze in de voedingsoplossing zitten.

9× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 2× verhaal · 1× het geheel · 1× code · ± 8 lestijden

M Materie en chemie



Koper, zwavel, silicium, koolstof, gallium, jood en neon: zeven elementen, zeven soorten atomen.

Na dit hoofdstuk

- Je kan de bouw van een atoom beschrijven met protonen, neutronen en elektronen, hun lading, massa en plaats.
- Je kan uit het atoomnummer en het massagetal het aantal protonen, neutronen en elektronen afleiden, ook voor ionen.
- Je kan uitleggen wat isotopen zijn en de gemiddelde atoommassa berekenen uit isotoopmassa's en hun aandeel.
- Je kan de elektronen van de eerste twintig elementen over de schillen verdelen en de grenzen van dat model benoemen.
- Je kan het aantal valentie-elektronen afleiden en in verband brengen met de groep en de ionlading.
- Je kan de opbouw van het periodiek systeem uitleggen: perioden, groepen, families, metalen, niet-metalen, metalloïden en edelgassen.
- Je kan met het periodiek systeem eigenschappen van een element voorspellen, zoals Mendelejev deed.
- Je kan de elementen noemen die een plant nodig heeft en ze terugvinden in een voedingsoplossing.

UITLEG · 12.1

Wat je al weet, en wat nu komt

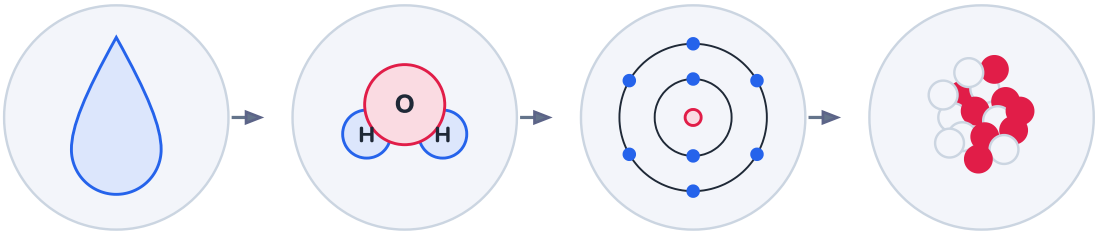
1/21

In F6 leerde je dat alles uit deeltjes bestaat, en in F8 dat een zuivere stof uit één soort deeltjes bestaat. Met de scheidingstechnieken uit F9 kan je een zuivere stof niet verder opsplitsen. Met een chemische reactie soms wel: stuur een stroom door water, en je krijgt twee gassen, waterstof en zuurstof. Water is dus een **samengestelde stof** (een **verbinding**): de deeltjes, de **moleculen**, zijn opgebouwd uit twee soorten **atomen**. Waterstof en zuurstof kan je chemisch niet verder ontbinden: het zijn **enkelvoudige stoffen**, opgebouwd uit één soort atomen. Zo'n atoomsoort heet een **element**. Er zijn er 118, waarvan er ongeveer 90 in de natuur voorkomen.

wat je al weet	waar	wat er nu bij komt
alles bestaat uit deeltjes die bewegen	F6	wat die deeltjes zijn: atomen, moleculen, ionen
een zuivere stof bestaat uit één soort deeltjes	F8	enkelvoudige en samengestelde stoffen, elementen
zout water geleidt, suikerwater niet: er zijn geladen deeltjes	F8	waar die lading vandaan komt: elektronen
halveringstijd en exponentieel verval	hoofdstuk 5	radioactieve isotopen zoals koolstof-14

In dit hoofdstuk

- open je het atoom: **kern** en **elektronen**, protonen en neutronen;
- leer je de **notatie** van een atoomsoort lezen, en ontdek je **isotopen**;
- verdeel je de elektronen over **schillen** en zoek je de **valentie-elektronen**;
- zie je de kleuren die elementen in een vlam geven, en waarom;
- volg je Mendelejev bij het bouwen van het **periodiek systeem**, en voorspel je zelf een element;
- zoek je uit welke elementen de planten van Habitat nodig hebben.



een druppel water
± 1,7 · 10^{21} moleculen

een watermolecule
± 0,3 nm

een zuurstofatoom
± 0,1 nm = 10^{-10} m

de kern: 8 p + 8 n
± 6 · 10^{-15} m

Elke stap is een zoom van duizenden keren: van een druppel naar een molecule, een atoom en de kern.

VERHAAL · 12.2

Een kogel die terugkaatst op vloeipapier

2/21

Rond 1800 dacht John Dalton dat atomen harde, ondeelbare bolletjes waren, voor elk element een andere soort. Een eeuw later bleek dat niet te kloppen. In 1897 ontdekte J.J. Thomson het **elektron**: een negatief deeltje, veel lichter dan het lichtste atoom, dat uit elk metaal kon komen. Atomen hadden dus

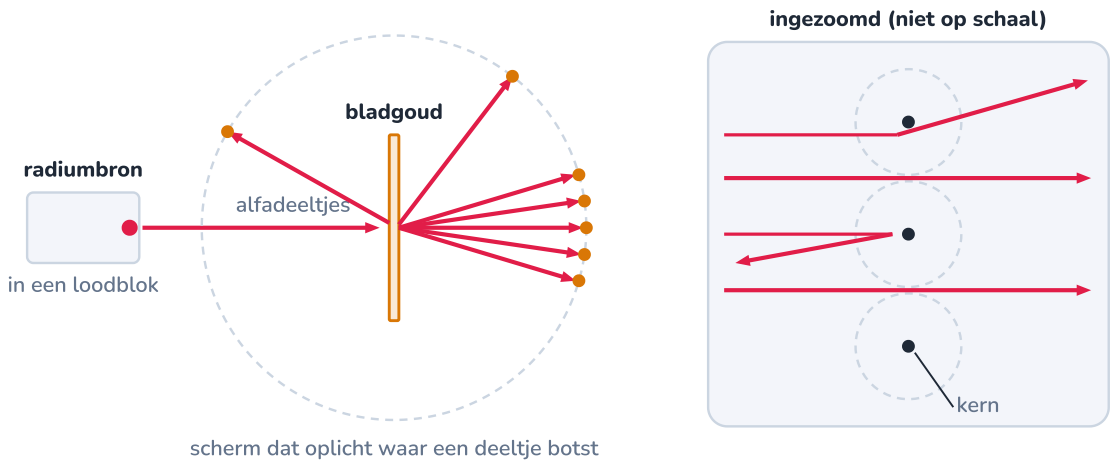
onderdelen. Thomson stelde zich een atoom voor als een positieve bol met de elektronen erin verspreid, als rozijnen in een cake.

In Manchester liet Ernest Rutherford vanaf 1909 zijn medewerkers Hans Geiger en Ernest Marsden alfadeeltjes, snelle positieve deeltjes uit radium, afvuren op een flinterdun blaadje goud. Rond het blaadje stond een scherm dat een lichtflitsje gaf waar een deeltje insloeg. In een donkere kamer telden ze de flitsjes, urenlang, door een microscoop. Bijna alle deeltjes gingen rechtdoor, alsof het goud er niet was. Maar een heel klein deel boog sterk af, en af en toe kaatste er een terug. Rutherford vergeleek het later met een granaat die je afvuurt op een vel vloeipapier en die terugkomt.

In 1911 gaf hij de verklaring. Een atoom is bijna helemaal leeg. Alle positieve lading en bijna alle massa zitten in een piepkleine **kern**. Een alfadeeltje dat de kern net mist, gaat rechtdoor; alleen een deeltje dat vlak bij een kern komt, wordt door de afstoting sterk afgebogen. In 1932 vond James Chadwick het laatste bouwsteentje van de kern: het ongeladen **neutron**.

Hoe weet je iets over iets wat je niet kan zien?

Niemand heeft de kern van een goudatoom ooit gezien. Rutherford besloot tot haar bestaan uit een telling van flitsjes en een berekening: welk model voorspelt hoeveel deeltjes onder welke hoek afbuigen? Zo werkt wetenschap vaak: een model voorspelt iets meetbaars, en de meting beslist.



Bijna alle alfadeeltjes vliegen tussen de kernen door; een enkel deeltje dat een kern bijna raakt, kaatst terug.

UITLEG · 12.3

Protonen, neutronen en elektronen

3/21

Een atoom bestaat uit drie soorten deeltjes:

deeltje	symbool	waar	lading	massa (kg)	massa (u)
proton	p^+	kern	+1	$1{,}673 \cdot 10^{-27}$	1,007
neutron	n	kern	0	$1{,}675 \cdot 10^{-27}$	1,009
elektron	e^-	rond de kern	−1	$9{,}109 \cdot 10^{-31}$	0,000 549

Lading. Een proton heeft een positieve lading die precies even groot is als de negatieve lading van een elektron: $1{,}602 \cdot 10^{-19}$ C. We noemen dat kortweg +1 en −1. Een atoom heeft **evenveel elektronen als protonen** en is dus neutraal.

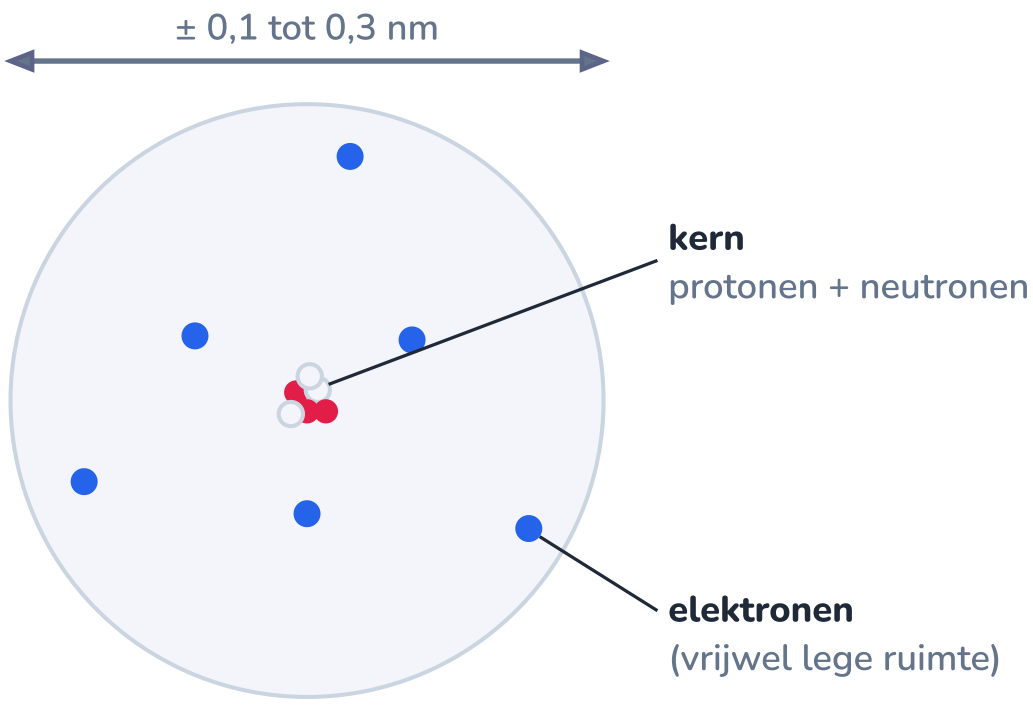
Massa. Een proton en een neutron zijn bijna even zwaar; een elektron is 1836 keer lichter dan een proton. Meer dan 99,9 % van de massa van een atoom zit dus in de kern. Omdat kilogrammen hier onhandig zijn, gebruik je de **atomaire massa-eenheid** u: één twaalfde van de massa van een koolstof-12-atoom, $1\,\mathrm{u} = 1{,}6605 \cdot 10^{-27}\,\mathrm{kg}$. Een proton en een neutron wegen elk ongeveer 1 u.

Grootte. Een atoom is ongeveer 0,1 tot 0,3 nm groot (10^{-10} m): tien miljoen atomen op een rij zijn samen maar 1 à 3 mm lang. De kern is nog eens ongeveer 60 000 keer kleiner. Was de kern een zandkorrel van 1 mm, dan was het atoom een bol van 60 m, zo groot als een voetbalveld, met daarin alleen een paar elektronen.

Waarom vliegen de protonen in de kern niet uit elkaar, als gelijke ladingen elkaar afstoten? Op heel korte afstand trekken protonen en neutronen elkaar aan met de **sterke kernkracht**, veel sterker dan de elektrische afstoting. Neutronen zijn daarbij een soort lijm: zonder neutronen houdt alleen de waterstofkern het uit.

Een atoom is een klein, vol bolletje

Een atoom is bijna helemaal lege ruimte. Dat je toch niet door een tafel zakt, komt door de elektronen: de elektronen van jouw hand en die van de tafel stoten elkaar af lang voor de kernen elkaar ooit naderen.



Niet op schaal: op deze tekening zou de kern kleiner zijn dan een honderdste van een pixel.

UITLEG · 12.4

Atoomnummer en massagetal

4/21

Wat een atoom tot koolstof of tot goud maakt, is het aantal protonen in de kern. Dat getal heet het **atoomnummer** Z. Elk koolstofatoom heeft 6 protonen, elk goudatoom 79. Het **massagetal** A is het aantal protonen en neutronen samen, de kerndeeltjes. Het aantal neutronen is dus

AANTAL NEUTRONEN

$$N = A - Z$$

Z atoomnummer (protonen), A massagetal (protonen + neutronen). In een neutraal atoom is het aantal elektronen gelijk aan Z.

Je schrijft een atoomsoort als ${}^{A}_{Z}\mathrm{X}$, met het symbool van het element: ${}^{35}_{17}\mathrm{Cl}$ is een chlooratoom met 17 protonen en $35 - 17 = 18$ neutronen. Omdat het symbool al zegt welk element het is, laat men Z vaak weg: ${}^{35}\mathrm{Cl}$, of in woorden chloor-35.

atoomsoort	Z (protonen)	A	neutronen N = A − Z	elektronen (neutraal)
${}^{1}_{1}\mathrm{H}$	1	1	0	1
${}^{4}_{2}\mathrm{He}$	2	4	2	2
${}^{12}_{6}\mathrm{C}$	6	12	6	6
${}^{16}_{8}\mathrm{O}$	8	16	8	8
${}^{23}_{11}\mathrm{Na}$	11	23	12	11
${}^{56}_{26}\mathrm{Fe}$	26	56	30	26
${}^{238}_{92}\mathrm{U}$	92	238	146	92

Ionen. Bij een chemische reactie kan een atoom elektronen afgeven of opnemen, nooit protonen: die zitten diep in de kern. Het wordt dan een **ion** met een lading. Een natriumatoom dat één elektron afgeeft, wordt een natriumion Na^{+}: 11 protonen, 10 elektronen. Een chlooratoom dat er één opneemt, wordt een chloride-ion Cl^{-}: 17 protonen, 18 elektronen. De lading is altijd het aantal protonen min het aantal elektronen.

Links boven het massagetal, links onder het atoomnummer: chloor-35 heeft 17 protonen en 18 neutronen.

UITGEWERKT VOORBEELD · 12.5

Deeltjes tellen in atomen en ionen

5/21

Hoeveel protonen, neutronen en elektronen hebben ${}^{31}\mathrm{P}$, ${}^{40}\mathrm{Ca}^{2+}$ en ${}^{32}\mathrm{S}^{2-}$? En welk atoom heeft 26 protonen en 30 neutronen?

GEGEVEN

GEVRAAGD

- massagetallen 31, 40 en 32; ladingen 0, 2+ en 2−
- het periodiek systeem (atoomnummers)

- protonen, neutronen en elektronen
- symbool en massagetal van het vierde atoom

OPLOSSING

1 **Fosfor-31.** In het periodiek systeem heeft fosfor $Z = 15$: 15 protonen. Neutronen: $31 - 15 = 16$. Neutraal, dus 15 elektronen.

2 **Calciumion.** Calcium: $Z = 20$, dus 20 protonen en $40 - 20 = 20$ neutronen. De lading 2+ betekent twee elektronen *minder* dan protonen: $20 - 2 = 18$ elektronen.

3 **Sulfide-ion.** Zwavel: $Z = 16$, dus 16 protonen en $32 - 16 = 16$ neutronen. De lading 2− betekent twee elektronen *meer*: 18 elektronen.

4 **Het onbekende atoom.** 26 protonen: $Z = 26$, dat is ijzer (Fe). Massagetal $A = 26 + 30 = 56$: $^{56}_{26}\mathrm{Fe}$, ijzer-56.

ANTWOORD

$^{31}\mathrm{P}$: 15 p, 16 n, 15 e; $^{40}\mathrm{Ca}^{2+}$: 20 p, 20 n, 18 e; $^{32}\mathrm{S}^{2-}$: 16 p, 16 n, 18 e; het vierde atoom is ijzer-56.

Klopt dit? Het calciumion en het sulfide-ion hebben allebei 18 elektronen, evenveel als het edelgas argon. Dat is geen toeval: verderop in dit hoofdstuk zie je waarom atomen zo graag de elektronenverdeling van een edelgas krijgen.

UITLEG · 12.6

Isotopen

6/21

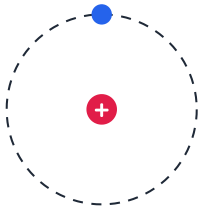
Alle atomen van een element hebben evenveel protonen, maar niet altijd evenveel neutronen. Atomen van hetzelfde element met een ander massagetal heten **isotopen** (Grieks voor 'op dezelfde plaats': ze staan in hetzelfde vakje van het periodiek systeem). Waterstof heeft er drie: gewone waterstof zonder neutron, deuterium met één en tritium met twee neutronen.

Isotopen hebben **dezelfde elektronen**, en dus dezelfde chemische eigenschappen: ze vormen dezelfde verbindingen. Ze verschillen in **massa**, en soms in **stabiliteit**. Water met deuterium in plaats van gewone waterstof, 'zwaar water' D_2O, is 11 % zwaarder: een ijsblokje van zwaar water zinkt in gewoon water. Sommige isotopen zijn instabiel: hun kern valt na een tijd uiteen, ze zijn **radioactief**.

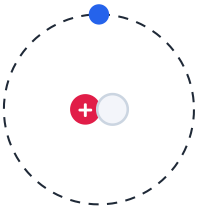
element	isotopen in de natuur (aandeel van de atomen)	opmerking
waterstof	H-1: 99,985 %; H-2 (deuterium): 0,015 %	H-3 (tritium) is radioactief (halveringstijd 12,3 jaar)
koolstof	C-12: 98,93 %; C-13: 1,07 %	C-14: ongeveer 1 op 10^{12}, radioactief, halveringstijd 5730 jaar
stikstof	N-14: 99,636 %; N-15: 0,364 %	N-15 gebruiken plantenonderzoekers om meststof te 'volgen'
chloor	Cl-35: 75,76 %; Cl-37: 24,24 %	daarom is de atoommassa 35,45 en geen geheel getal
kalium	K-39: 93,26 %; K-41: 6,73 %	K-40 (0,0117 %) is radioactief: ook in bananen en in je lichaam
uraan	U-238: 99,27 %; U-235: 0,72 %	kernbrandstof wordt verrijkt tot 3 à 5 % U-235

Koolstof-14 ontstaat voortdurend hoog in de atmosfeer. Planten nemen het op met de koolstofdioxide die ze gebruiken voor fotosynthese, dieren eten de planten: zolang een organisme leeft, heeft het dezelfde verhouding C-14/C-12 als de lucht. Na de dood komt er geen C-14 meer bij, en halveert de

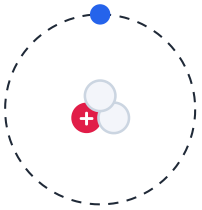
hoeveelheid om de 5730 jaar. Uit wat er overblijft, bereken je hoe lang geleden het organisme stierf: de **koolstofdatering**, met de halveringstijd uit hoofdstuk 5.



waterstof-1
1 p, 0 n · 99,985 %



waterstof-2 (deuterium)
1 p, 1 n · 0,015 %



waterstof-3 (tritium)
1 p, 2 n · radioactief

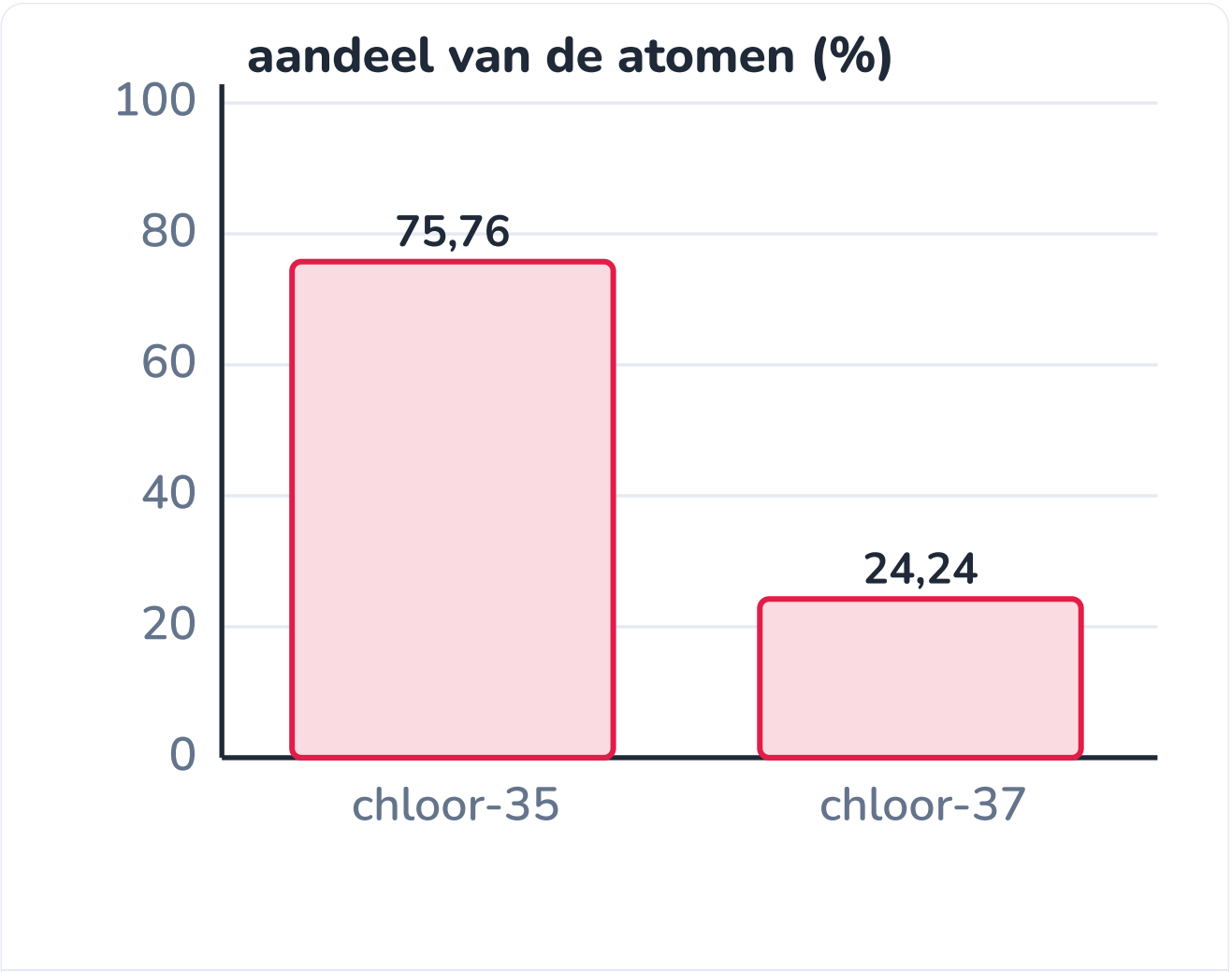
Eén proton en één elektron, en 0, 1 of 2 neutronen: chemisch dezelfde stof, maar een andere massa.

UITGEWERKT VOORBEELD · 12.7

De gemiddelde atoommassa van chloor

7/21

In het periodiek systeem staat voor chloor een atoommassa van 35,45 u, terwijl geen enkel chlooratoom 35,45 u weegt. Een massaspectrometer, een toestel dat atomen sorteert op massa, meet in natuurlijk chloor twee isotopen: chloor-35 (34,969 u, 75,76 % van de atomen) en chloor-37 (36,966 u, 24,24 %). Toon aan dat de gemiddelde atoommassa 35,45 u is.



Het massaspectrum van chloor: twee pieken, bij 35 en bij 37, in de verhouding 3 : 1.

GEGEVEN

- $m_1 = 34{,}969\ \text{u}$, aandeel $f_1 = 0{,}7576$
- $m_2 = 36{,}966\ \text{u}$, aandeel $f_2 = 0{,}2424$

GEVRAAGD

- de gemiddelde atoommassa $\bar{m}$

OPLOSSING

1. Neem in gedachten 10 000 chlooratomen: 7576 wegen elk 34,969 u, 2424 wegen elk 36,966 u.
2. Hun totale massa: $7576 \cdot 34{,}969 + 2424 \cdot 36{,}966 = 264\,925 + 89\,606 = 354\,531\ \text{u}$.
3. Gemiddeld per atoom: $354\,531 / 10\,000 = 35{,}45\ \text{u}$.
4. Korter, met de aandelen als decimale getallen (een **gewogen gemiddelde**): $\bar{m} = f_1\, m_1 + f_2\, m_2 = 0{,}7576 \cdot 34{,}969 + 0{,}2424 \cdot 36{,}966 = 35{,}45\ \text{u}$.

ANTWOORD

De gemiddelde atoommassa van chloor is $35{,}45\ \text{u}$: het gewogen gemiddelde van zijn twee isotopen.

Klopt dit? Het gemiddelde ligt tussen 35 en 37, veel dichter bij 35: er is drie keer zoveel chloor-35. Reken je met de massagetallen (35 en 37) in plaats van de echte isotoopmassa's, dan krijg je 35,48: dicht bij, maar niet exact, omdat protonen en neutronen niet precies 1 u wegen en er bij het samenvoegen van een kern een beetje massa verdwijnt als bindingsenergie.

UITGEWERKT VOORBEELD · 12.8

Hoeveel koper-63 zit er in koper?

8/21

Koper heeft twee stabiele isotopen: koper-63 (62,930 u) en koper-65 (64,928 u). De gemiddelde atoommassa is 63,546 u. Welk percentage van de koperatomen is koper-63?

GEGEVEN

- $m_{63} = 62{,}930\,\text{u}$, $m_{65} = 64{,}928\,\text{u}$
- $\bar{m} = 63{,}546\,\text{u}$

GEVRAAGD

- het aandeel x van koper-63

OPLOSSING

1. Noem het aandeel van koper-63 x (als decimaal getal). Er zijn maar twee isotopen, dus het aandeel van koper-65 is $1 - x$.
2. Gewogen gemiddelde: $x \cdot 62{,}930 + (1 - x) \cdot 64{,}928 = 63{,}546$.
3. Haakjes uitwerken: $62{,}930\,x + 64{,}928 - 64{,}928\,x = 63{,}546$, dus $-1{,}998\,x = -1{,}382$.
4. $x = \dfrac{1{,}382}{1{,}998} = 0{,}692$.

ANTWOORD

69,2 % van de koperatomen is koper-63, 30,8 % is koper-65.

Klopt dit? De gemiddelde massa 63,546 ligt dichter bij 63 dan bij 65, dus er moet meer koper-63 zijn: klopt. Invullen: $0{,}692 \cdot 62{,}930 + 0{,}308 \cdot 64{,}928 = 63{,}545$. De gemeten waarde is 69,15 %: het verschil komt van het afronden van de isotoopmassa's.

UITLEG · 12.9

Elektronen in schillen

9/21

Hoe zitten de elektronen rond de kern? In 1913 stelde Niels Bohr een model voor waarin de elektronen in vaste **schillen** rond de kern zitten, op een steeds grotere afstand en met een steeds hogere energie. De schillen krijgen een nummer $n = 1, 2, 3, \ldots$ of een letter K, L, M, N. Een schil kan hoogstens $2n^2$ elektronen bevatten: 2, 8, 18, 32.

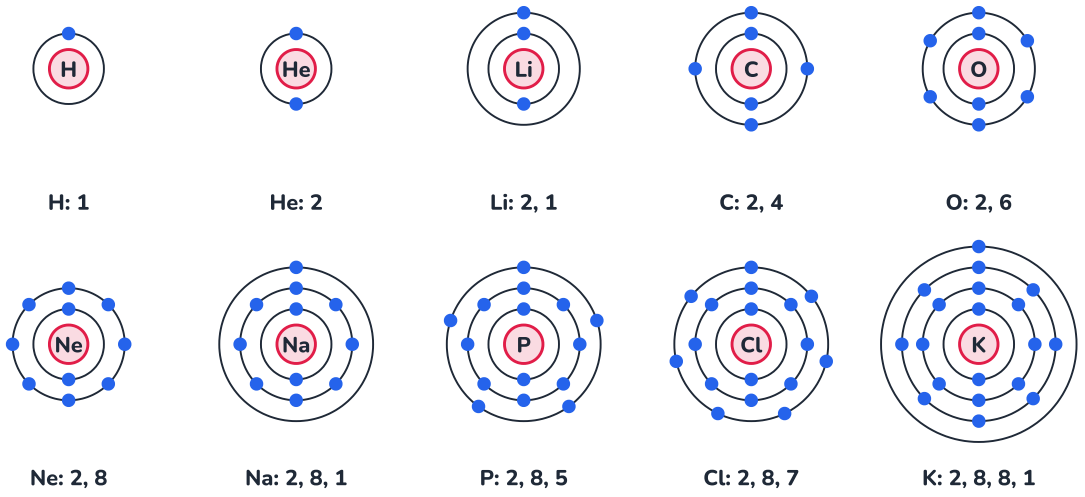
De elektronen vullen eerst de binnenste schil, die het dichtst bij de kern zit en de laagste energie heeft. Voor de eerste twintig elementen geldt de eenvoudige regel **2, 8, 8, 2**:

- de eerste schil (K): hoogstens 2 elektronen;
- de tweede schil (L): hoogstens 8;
- de derde schil (M): tot 8 elektronen voor het volgende element al de vierde schil begint;
- de vierde schil (N): 1 elektron bij kalium, 2 bij calcium.

Natrium (11 elektronen) wordt zo 2, 8, 1; in de letternotatie K2 L8 M1. Chloor (17) wordt 2, 8, 7, en argon (18) 2, 8, 8.

⚠ Wat het schillenmodel niet kan

De derde schil kan 18 elektronen aan, en toch begint bij kalium al de vierde schil. Het schillenmodel verklaart dat niet: elke schil bestaat in werkelijkheid uit *subschillen* met een iets andere energie, en een deel van de vierde schil ligt lager dan de rest van de derde. Pas vanaf scandium (Z = 21) vult de derde schil zich verder tot 18. Ook de beweging klopt niet: elektronen draaien niet in nette banen zoals planeten. Een elektron dat in een cirkel draait, zou straling uitzenden en in de kern vallen. In werkelijkheid is een elektron eerder een 'wolk' die aangeeft waar het zich waarschijnlijk bevindt (het orbitaalmodel, Expert). Voor de eerste twintig elementen en voor het begrijpen van bindingen (hoofdstuk 13) is het schillenmodel wel een uitstekend gereedschap.



De blauwe bolletjes zijn de elektronen, de rode kern draagt het symbool. Kalium begint een vierde schil.

UITGEWERKT VOORBEELD · 12.10

De schillen van fosfor, kalium en het calciumion

10/21

Verdeel de elektronen over de schillen voor fosfor (Z = 15), kalium (Z = 19) en het calciumion Ca^{2+}.

GEGEVEN

- Z = 15, 19 en 20; lading van het calciumion: 2+
- regel 2, 8, 8, 2

GEVRAAGD

- de verdeling over de schillen
- het aantal valentie-elektronen

OPLOSSING

1. **Fosfor**: 15 elektronen. Eerste schil 2 (rest 13), tweede schil 8 (rest 5), derde schil 5. Verdeling 2, 8, 5: vijf elektronen in de buitenste schil.
2. **Kalium**: 19 elektronen. 2 en 8 (rest 9). De derde schil neemt er 8, dan begint de vierde schil met het laatste elektron: 2, 8, 8, 1. Niet 2, 8, 9: in dit model is de derde schil 'voorlopig vol' bij 8.
3. **Calciumion**: calcium heeft 20 elektronen (2, 8, 8, 2). Het ion Ca^{2+} heeft er twee minder, en die gaan uit de buitenste schil: 2, 8, 8. De vierde schil is weg.

ANTWOORD

P: 2, 8, 5 (5 valentie-elektronen); K: 2, 8, 8, 1 (1); Ca^{2+}: 2, 8, 8, dezelfde verdeling als argon.

Klopt dit? Tel na: 2 + 8 + 5 = 15; 2 + 8 + 8 + 1 = 19; 2 + 8 + 8 = 18 = 20 − 2. De som van de schillen is altijd het aantal elektronen.

PROEF · 12.11

Vlamkleuren: elk element zijn eigen licht

11/21

ONDERZOEKSVRAAG

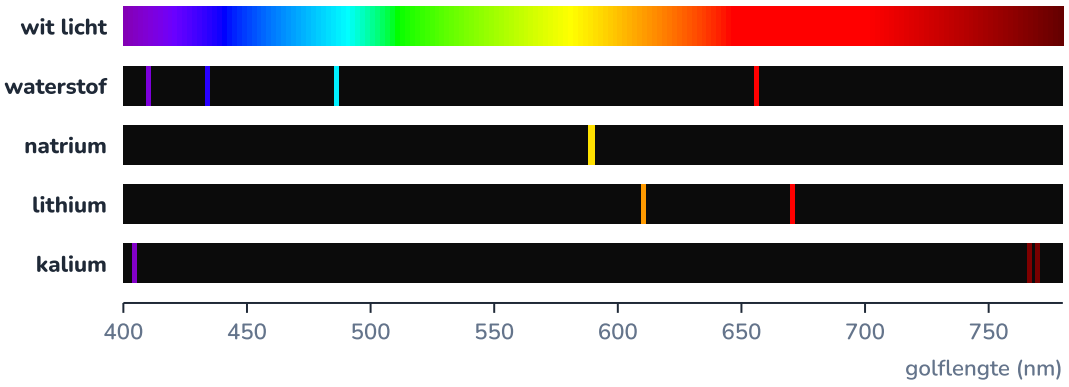
Kleuren zouten van verschillende metalen een vlam op een eigen manier? Kan je een zout herkennen aan zijn vlamkleur?

MATERIAAL

- bunsenbrander of gasbrander op een hittebestendige plaat
- houten satéprikkers, een nacht geweekt in gedemineraliseerd water
- kleine hoeveelheden (een mespuntje) natriumchloride, kaliumchloride, calciumchloride, lithiumchloride en kopersulfaat, elk in een eigen schaaltje
- blauw kobaltglas
- bekerglas met water om de prikkers te doven
- veiligheidsbril

Veiligheid

Veiligheidsbril, haar vast, geen losse kleding. Werk met kleine hoeveelheden, houd de prikker schuin zodat er niets in de brander valt, en doof hem daarna in het water. Lithiumchloride, calciumchloride en kopersulfaat: niet inslikken, niet in de ogen. Kopersulfaatresten bij het chemisch afval (GHS09).



Wit licht bevat alle kleuren. Een element geeft maar enkele scherpe lijnen: kalium rood en violet samen, dat zie je als lila.

WERKWIJZE

1. Regel de brander tot een blauwe, bijna kleurloze vlam (luchtregeling open).
2. Druk de natte punt van een prikker in het eerste zout, zodat er wat korreltjes aan blijven kleven.
3. Houd de punt in de rand van de vlam, niet in de blauwe kern. Noteer de kleur die de vlam krijgt.
4. Neem voor elk zout een nieuwe prikker. Bekijk de kaliumvlam ook door het kobaltglas.
5. Laat een klasgenoot een van de vijf zouten kiezen zonder te zeggen welk. Herken het aan zijn vlamkleur.

WAARNEMING

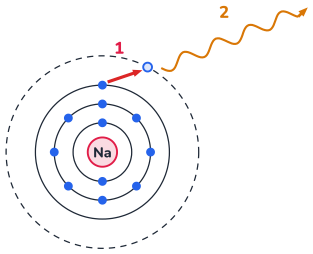
zout	metaal	vlamkleur
natriumchloride	natrium	fel geeloranje, overstemt alle andere kleuren
kaliumchloride	kalium	lila, zwak; door het kobaltglas roze-violet
calciumchloride	calcium	oranjerood (baksteenrood)
lithiumchloride	lithium	karmijnrood
kopersulfaat	koper	blauwgroen

De kleur hoort bij het metaal, niet bij het chloride: natriumchloride en natriumsulfaat kleuren de vlam allebei geel. Een klein beetje natrium (zweet op je vingers, stof) kleurt elke vlam al geel. Het kobaltglas houdt geel licht tegen: daardoor zie je het zwakke kaliumlicht, ook als er wat natrium bij zit.

VERKLARING

De hete vlam geeft de atomen energie. Een elektron springt daardoor naar een hogere, lege schil. Daar blijft het niet: het valt terug, en geeft de energie af als een flitsje licht. Omdat de schillen vaste energieën hebben, is het energieverschil, en dus de **kleur** van dat licht, voor elk element anders. Natrium geeft vooral licht van 589 nm: geel. Stuur je dat licht door een prisma, dan zie je geen regenboog maar een paar scherpe lijnen: een **lijnenspectrum**, een vingerafdruk van het element.

In 1860 en 1861 ontdekten Robert Bunsen en Gustav Kirchhoff zo twee nieuwe elementen, die ze naar de kleur van hun lijnen noemden: cesium (hemelsblauw) en rubidium (dieprood). In 1868 zagen sterrenkundigen in het licht van de zon een gele lijn die bij geen enkel bekend element paste: helium, zo genoemd naar de Griekse zon, *helios*, en pas in 1895 op aarde gevonden. Vuurwerk gebruikt dezelfde natuurkunde: strontium voor rood, barium voor groen, natrium voor geel. En de oranje verlichting die decennialang de Belgische autosnelwegen verlichtte, was het licht van natriumdamplampen.



1 energie uit de vlam tilt een elektron naar een hogere, lege schil
2 terugvallen: geel licht van 589 nm

De sprong omhoog kost energie, de val terug geeft ze als licht van één welbepaalde kleur.

Lees het etiket van elk potje: deze pictogrammen staan erop.

Natrium, kalium, calcium, lithium en koper: vijf metalen, vijf vlamkleuren.

UITLEG · 12.12

Valentie-elektronen en de edelgasconfiguratie

12/21

Bij een chemische reactie komen atomen met hun buitenkant in contact. De elektronen in de buitenste schil, de **valentie-elektronen**, bepalen daarom hoe een atoom reageert. Natrium heeft er 1, chloor 7, koolstof 4.

Bij de **edelgassen** is de buitenste schil vol: 2 bij helium, 8 bij neon en argon. Die elementen reageren bijna nooit. Andere atomen gaan chemische verbindingen aan zodat ze zelf ook zo'n volle buitenste schil krijgen: de **edelgasconfiguratie**. Omdat dat (behalve bij helium) acht elektronen zijn, heet dat ook de **octetregel**.

- **Metalen** hebben weinig valentie-elektronen. Ze geven ze makkelijk af en worden een positief ion: natrium (2, 8, 1) wordt Na^+ (2, 8), met de verdeling van neon. Magnesium wordt Mg^{2+}, aluminium Al^{3+}.

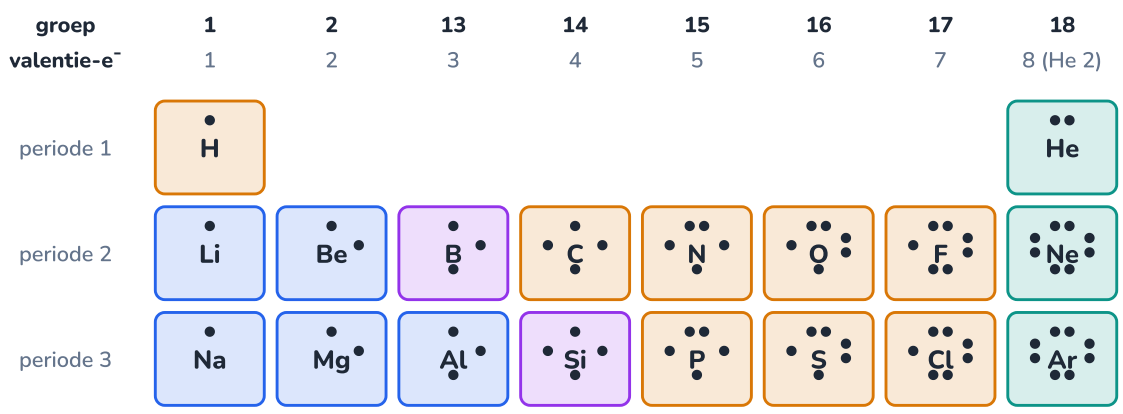
- **Niet-metalen** hebben veel valentie-elektronen. Ze nemen er op of delen er (hoofdstuk 13): chloor (2, 8, 7) wordt Cl^- (2, 8, 8), met de verdeling van argon. Zuurstof wordt O^{2-}, stikstof N^{3-}.

Je tekent de valentie-elektronen als stippen rond het symbool: het **Lewissymbool**. Eerst één stip aan elke kant, dan paren. Zo zie je in één oogopslag hoeveel elektronen een atoom kan afgeven, opnemen of delen.

groep	1	2	13	14	15	16	17	18
valentie-elektronen	1	2	3	4	5	6	7	8
meest voorkomend ion	+1	+2	+3	(deelt)	−3	−2	−1	geen
voorbeeld	K^+	Ca^{2+}	Al^{3+}	C	N^{3-}	O^{2-}	Cl^-	Ar

Van groep naar lading

Groep 1 en 2: lading +1 en +2. Groep 13: +3. Groep 15, 16 en 17: acht min het aantal valentie-elektronen, met een minteken: −3, −2, −1. De planten van Habitat nemen hun kalium, calcium en magnesium op als precies zulke ionen: K^+, Ca^{2+} en Mg^{2+}.



Het aantal stippen is het groepsnummer (min tien vanaf groep 13). Links geven de atomen elektronen af, rechts nemen ze op.

VERHAAL · 12.13

Mendelejev en de lege vakjes

13/21

In 1869 waren 63 elementen bekend. Chemici wisten van elk de atoommassa en een hele reeks eigenschappen, maar geen orde. De Russische scheikundige Dmitri Mendelejev, professor in Sint-Petersburg, schreef aan een leerboek en zocht een logische volgorde voor de hoofdstukken. Hij schreef elk element met zijn eigenschappen op een kaartje en schoof ermee tot er een patroon verscheen: in de volgorde van de atoommassa keren dezelfde eigenschappen *periodiek* terug. Lithium, natrium en kalium, zachte metalen die heftig met water reageren, kwamen onder elkaar te staan; fluor, chloor, broom en jood ook.

Mendelejev deed twee gedurfde dingen. Hij liet **vakjes open** waar geen bekend element paste, en hij voorspelde de eigenschappen van die ontbrekende elementen uit hun buren. En hij zette telluur (127,6) vóór jood (126,9), tegen de volgorde van de massa in, omdat jood bij de halogenen hoort. Pas in 1913 bleek waarom dat klopte: Henry Moseley toonde met röntgenstraling aan dat de volgorde in de tabel die van het **atoomnummer** is, niet die van de massa.

In 1875 vond de Fransman Paul-Émile Lecoq de Boisbaudran in zinkerts een nieuw metaal, dankzij twee violette lijnen in zijn spectrum: **gallium**. Mendelejev herkende er zijn 'eka-aluminium' in. Toen Lecoq een dichtheid van 4,7 g/cm³ meldde, schreef Mendelejev hem dat die ongeveer 5,9 moest zijn. Lecoq zuiverde zijn staal en mat opnieuw: 5,9. In 1879 volgde scandium (eka-boor), en in 1886 vond Clemens Winkler **germanium**, Mendelejevs eka-silicium:

eigenschap	eka-silicium, voorspeld (1871)	germanium, gemeten
atoommassa	72	72,63
dichtheid	5,5 g/cm³	5,32 g/cm³
kleur	grijs	grijs
formule van het oxide	EsO_2	GeO_2
chloride	$EsCl_4$, vloeibaar, kookt onder 100 °C	$GeCl_4$, kookt bij 86,5 °C
dichtheid van het chloride	1,9 g/cm³	1,88 g/cm³

Vanaf dan twijfelde niemand nog aan het periodiek systeem. Element 101 draagt zijn naam: mendelevium.

Een goede theorie steekt haar nek uit

Anderen, zoals de Duitser Lothar Meyer, hadden ook ontdekt dat eigenschappen periodiek terugkeren. Wat Mendelejev onderscheidde, waren de voorspellingen die fout konden blijken. Ze bleken juist.

Gallium smelt al bij 29,8 °C: in je hand. Mendelejev voorspelde het metaal vier jaar voor het gevonden werd.

Het periodiek systeem: perioden en groepen

In het periodiek systeem staan de elementen in de volgorde van hun atoomnummer, van links naar rechts en rij per rij. Vandaag telt het 118 elementen; de laatste vier kregen in 2016 hun naam.

- Een horizontale rij heet een **periode**. Het nummer van de periode is het **aantal schillen**: natrium (2, 8, 1) staat in periode 3. Bij elke nieuwe periode begint een nieuwe schil.
- Een kolom heet een **groep**. Elementen in een groep hebben **evenveel valentie-elektronen**, en daardoor gelijkaardige chemische eigenschappen: ze vormen een familie. In de hoofdgroepen is het aantal valentie-elektronen het groepsnummer (groep 1 en 2) of het groepsnummer min tien (groep 13 tot 18).

groep	familienaam	kenmerk
1 (behalve H)	alkalimetalen	zachte metalen, reageren heftig met water; hoe lager in de groep, hoe heftiger
2	aardalkalimetalen	metalen, minder heftig; calcium en magnesium in kalk, beenderen en bladgroen
3 tot 12	overgangsmetalen	de gewone gebruiksmetalen: ijzer, koper, zink, nikkel; vaak gekleurde verbindingen
16	zuurstofgroep	zuurstof en zwavel
17	halogenen	reactieve niet-metalen: fluor, chloor, broom, jood; vormen zouten met metalen
18	edelgassen	volle buitenste schil, reageren bijna niet

Onderaan staan twee losse rijen van veertien à vijftien elementen, de **lanthaniden** en de **actiniden**. Ze horen eigenlijk in periode 6 en 7, maar zo past de tabel op een blad. Bijna alles na uraan (Z = 92) wordt door mensen gemaakt, vaak maar enkele atomen tegelijk die na een fractie van een seconde uiteenvallen.

> ⚠ **Waterstof, een buitenbeentje**
>
> Waterstof staat in groep 1 omdat het één valentie-elektron heeft, maar het is geen metaal: het is een gas dat elektronen liever deelt dan afgeeft. Helium staat in groep 18 bij de edelgassen, al heeft het maar twee valentie-elektronen: zijn enige schil is vol.

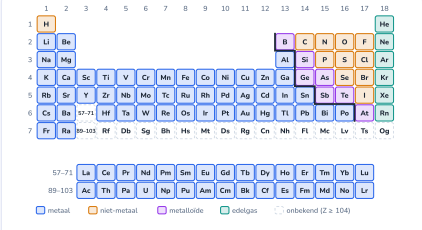
periode	groep 1	groep 2	groep 13	groep 14	groep 15	groep 16	groep 17	groep 18
1	1 **H** waterstof 1,008 1							2 **He** helium 4,003 2
2	3 **Li** lithium 6,94 2, 1	4 **Be** beryllium 9,012 2, 2	5 **B** boor 10,81 2, 3	6 **C** koolstof 12,011 2, 4	7 **N** stikstof 14,007 2, 5	8 **O** zuurstof 15,999 2, 6	9 **F** fluor 18,998 2, 7	10 **Ne** neon 20,180 2, 8
3	11 **Na** natrium 22,990 2, 8, 1	12 **Mg** magnesium 24,305 2, 8, 2	13 **Al** aluminium 26,982 2, 8, 3	14 **Si** silicium 28,085 2, 8, 4	15 **P** fosfor 30,974 2, 8, 5	16 **S** zwavel 32,06 2, 8, 6	17 **Cl** chloor 35,45 2, 8, 7	18 **Ar** argon 39,95 2, 8, 8
4	19 **K** kalium 39,098 2, 8, 8, 1	20 **Ca** calcium 40,078 2, 8, 8, 2						

metaal · niet-metaal · metalloïde · edelgas

in elk vakje: atoomnummer, symbool, naam, gemiddelde atoommassa (u) en de schillen

In periode 4 volgen na calcium eerst tien overgangsmetalen (scandium tot zink), daarna gallium tot krypton.

Periode = aantal schillen, groep = aantal valentie-elektronen. Onder elkaar staan dus atomen met evenveel elektronen in de buitenste schil.



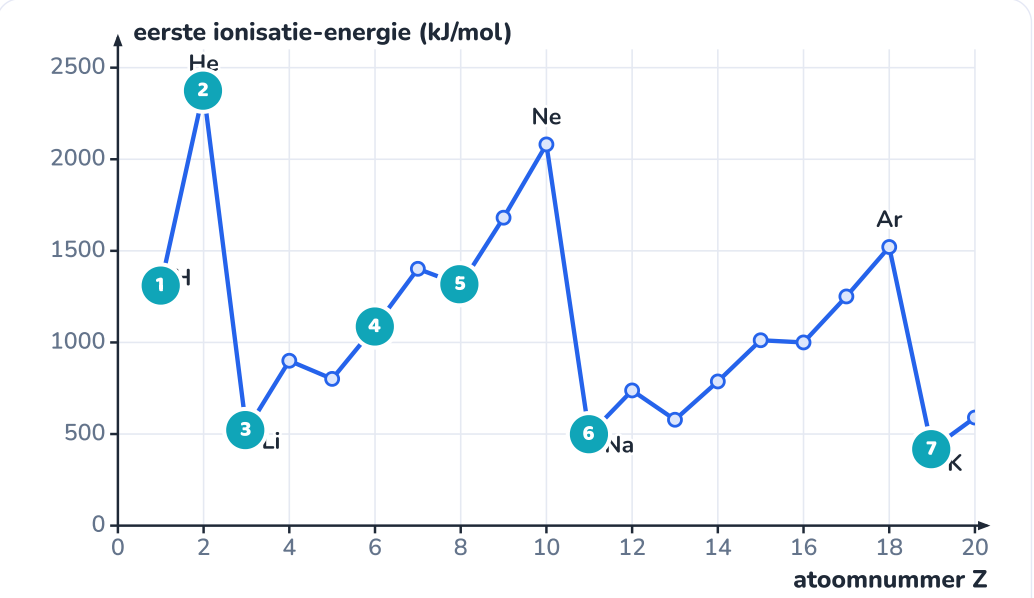
De dikke trap scheidt de metalen (links) van de metalloïden en niet-metalen. Voor polonium en astaat is de indeling onzeker.

ONTLEED · 12.15

Ionisatie-energie: het ritme van het periodiek systeem

15/21

De **ionisatie-energie** is de energie die nodig is om het zwakst gebonden elektron uit een atoom te halen, gemeten voor een mol atomen (een vast, heel groot aantal: hoofdstuk 14). Hoe hoger ze is, hoe steviger het atoom zijn elektronen vasthoudt. De grafiek toont de gemeten waarden voor de eerste twintig elementen. Het periodieke ritme springt eruit.



Gemeten waarden (NIST). Pieken bij de edelgassen, dalen bij de alkalimetalen: de schillen zijn in de grafiek te zien.

1 Waterstof: 1312 kJ/mol
Eén elektron, vlak bij de kern en zonder andere elektronen ertussen.

2 Helium: de hoogste van allemaal
Twee protonen trekken aan twee elektronen in een volle eerste schil. Daarom reageert helium niet.

3 Lithium: een diepe val
Het derde elektron zit in een nieuwe, verder gelegen schil, afgeschermd door de twee binnenste. Het laat makkelijk los: lithium is een metaal dat graag Li^+ wordt.

4 Binnen een periode: stijgend
Van lithium naar neon komen er protonen bij, maar geen schillen: de kern trekt steeds harder aan dezelfde schil.

5 Kleine knikken bij B en O
Boor en zuurstof liggen iets lager dan de trend. Het schillenmodel verklaart dat niet: daarvoor heb je de subschillen nodig (Expert).

6 Natrium: opnieuw een val
Na neon begint de derde schil. Het patroon van periode 2 herhaalt zich in periode 3: daarom heet de tabel periodiek.

7 Kalium: de laagste
Het valentie-elektron zit in de vierde schil, het verst van de kern. Kalium reageert nog heftiger met water dan natrium.

UITLEG · 12.16

Metalen, niet-metalen, metalloïden en edelgassen

16/21

Ongeveer vier op de vijf elementen zijn **metalen**. Ze staan links en in het midden van het periodiek systeem. Rechtsboven staan de **niet-metalen**, en langs de trap ertussen de **metalloïden** (halfmetalen), met eigenschappen van beide. De laatste kolom zijn de **edelgassen**.

	metalen	niet-metalen	metalloïden	edelgassen
waar	links en midden (ongeveer vier op vijf)	rechtsboven, plus waterstof	langs de trap	groep 18
uitzicht	glanzend	dof; gassen of brosse vaste stoffen, broom is vloeibaar	glanzend maar bros	kleurloze gassen
geleiding	goed (stroom en warmte)	slecht (grafiet is een uitzondering)	matig: halfgeleiders	niet

	metalen	niet-metalen	metalloïden	edelgassen
vervormen	pletbaar en buigzaam	bros	bros	
valentie-elektronen	meestal 1 tot 3	4 tot 7 (waterstof 1)	3 tot 6	8 (helium 2)
in verbindingen	geven elektronen af: positieve ionen	nemen elektronen op of delen ze	delen elektronen	reageren bijna nooit
voorbeelden	Na, Mg, Al, Fe, Cu, K, Ca	H, C, N, O, P, S, Cl	B, Si, Ge, As	He, Ne, Ar

Het verschil volgt uit de valentie-elektronen. Metaalatomen houden hun paar valentie-elektronen losjes vast (lage ionisatie-energie): in een stuk metaal zwerven ze vrij rond, en dat maakt metalen geleidend en glanzend (hoofdstuk 13). Niet-metalen houden hun elektronen stevig vast. Metalloïden zitten ertussen: **silicium** geleidt een beetje, en hoe goed hangt af van de temperatuur en van piepkleine hoeveelheden andere atomen. Daarom is het de grondstof van chips en zonnecellen (F8, hoofdstuk 23).

Ook binnen een groep zie je een trend. De alkalimetalen reageren heftiger naarmate je lager gaat: lithium bruist rustig in water, natrium schiet sissend over het water, kalium vat vlam. De halogenen worden net minder reactief naar onder: fluor is het meest reactieve element dat bestaat, jood is een vaste stof die je (opgelost) als ontsmettingsmiddel op een wonde smeert. Chloor zit in kleine hoeveelheden in leidingwater om bacteriën te doden (F9).

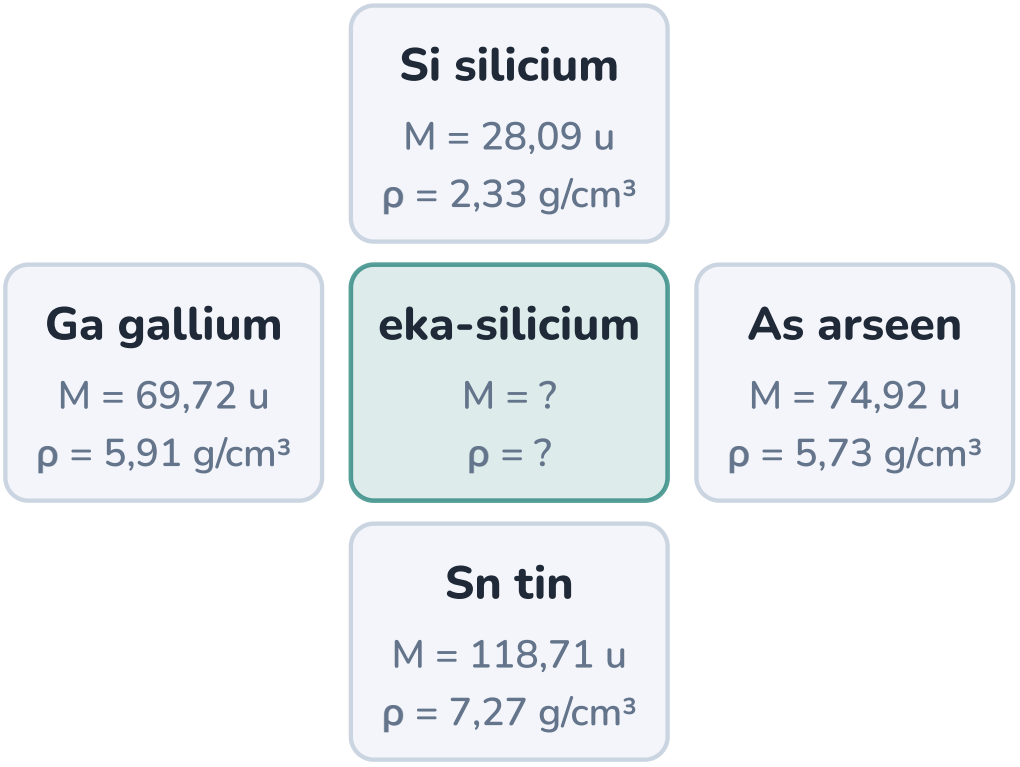
De edelgassen zijn nuttig omdat ze niet reageren. Argon (bijna 1 % van de lucht) beschermt het gesmolten metaal bij het lassen en de gloeidraad in een lamp; helium vult ballonnen en koelt de magneten van een MRI-scanner; neon geeft het rode licht van reclameborden.

Natrium op water: een zacht metaal met één valentie-elektron dat het maar al te graag afgeeft.

UITGEWERKT VOORBEELD · 12.17

Germanium voorspellen, zoals Mendelejev

Stel dat germanium nog onbekend is. In het periodiek systeem staat het tussen silicium (erboven) en tin (eronder), en tussen gallium (links) en arseen (rechts). Schat zijn atoommassa en zijn dichtheid uit die vier buren, en vergelijk met de gemeten waarden 72,63 u en 5,32 g/cm³.



Het ontbrekende element in het midden erft iets van elk van zijn buren.

GEGEVEN

- Si: 28,09 u, 2,33 g/cm³; Sn: 118,71 u, 7,27 g/cm³
- Ga: 69,72 u, 5,91 g/cm³; As: 74,92 u, 5,73 g/cm³

GEVRAAGD

- geschatte atoommassa en dichtheid van germanium

OPLOSSING

1. Eigenschappen veranderen geleidelijk langs een groep en langs een periode. Een eerste schatting is het gemiddelde van de buren.
2. Alleen boven en onder: $(28{,}09 + 118{,}71)/2 = 73{,}4\,\text{u}$ en $(2{,}33 + 7{,}27)/2 = 4{,}80\,\text{g/cm}^3$.
3. Met alle vier de buren: $(28{,}09 + 118{,}71 + 69{,}72 + 74{,}92)/4 = 72{,}9\,\text{u}$ en $(2{,}33 + 7{,}27 + 5{,}91 + 5{,}73)/4 = 5{,}31\,\text{g/cm}^3$.
4. Vergelijk: de atoommassa is 72,63 u (0,4 % verschil), de dichtheid 5,32 g/cm³ (0,2 % verschil).

ANTWOORD

Uit de vier buren schat je 72,9 u en 5,31 g/cm³, heel dicht bij de gemeten 72,63 u en 5,32 g/cm³. Met alleen de verticale buren zit je er bij de dichtheid 10 % naast.

Klopt dit? Mendelejev voorspelde 72 u en 5,5 g/cm³, met de kennis van 1871. Dat het gemiddelde zo goed werkt, is geen wet maar een vuistregel: het werkt goed als de eigenschap geleidelijk verloopt. Voor het smeltpunt werkt het slecht (gallium smelt al bij 30 °C, silicium pas bij 1414 °C).

CODE · 12.18

Python: schillen, periode en groep uitrekenen

18/21

De regels van dit hoofdstuk zijn precies genoeg om een computer te laten rekenen: verdeel de elektronen volgens 2, 8, 8, 2, tel de schillen voor de periode, en leid de groep af uit het aantal elektronen in de buitenste schil. De functie `schillen(z)` vult de schillen een voor een; `groep(verdeling)` houdt rekening met de uitzonderingen van periode 1.

Python **schillen.py** 31 regels

```python
# schillen.py - schillen, valentie-elektronen, periode en groep voor Z = 1 tot 20
SYMBOLEN = ["H", "He", "Li", "Be", "B", "C", "N", "O", "F", "Ne",
            "Na", "Mg", "Al", "Si", "P", "S", "Cl", "Ar", "K", "Ca"]


def schillen(z):
    """Verdeling over de schillen volgens het schillenmodel (2, 8, 8, 2): geldig tot calcium."""
    verdeling = []
    for plaatsen in (2, 8, 8, 2):
        aantal = min(z, plaatsen)
        if aantal > 0:
            verdeling.append(aantal)
        z = z - aantal
    return verdeling


def groep(verdeling):
    """Groep 1, 2 of 13 tot 18, uit het aantal elektronen in de buitenste schil."""
    buiten = verdeling[-1]
    if len(verdeling) == 1:             # periode 1: waterstof in groep 1, helium in groep 18
        return 1 if buiten == 1 else 18
    if buiten <= 2:
        return buiten
    return buiten + 10


for z in range(1, 21):
    s = schillen(z)
    tekst = ", ".join(str(a) for a in s)
    print(f"{z:2d} {SYMBOLEN[z - 1]:2s}   schillen {tekst:11s}  valentie {s[-1]}"
          f"   periode {len(s)}   groep {groep(s):2d}")
```

Uitvoer (het programma werd gedraaid bij het bouwen van deze pagina):

```
 1 H    schillen 1           valentie 1   periode 1   groep  1
 2 He   schillen 2           valentie 2   periode 1   groep 18
 3 Li   schillen 2, 1        valentie 1   periode 2   groep  1
 4 Be   schillen 2, 2        valentie 2   periode 2   groep  2
 5 B    schillen 2, 3        valentie 3   periode 2   groep 13
 6 C    schillen 2, 4        valentie 4   periode 2   groep 14
 7 N    schillen 2, 5        valentie 5   periode 2   groep 15
 8 O    schillen 2, 6        valentie 6   periode 2   groep 16
```

```
 9 F    schillen 2, 7         valentie 7   periode 2   groep 17
10 Ne   schillen 2, 8         valentie 8   periode 2   groep 18
11 Na   schillen 2, 8, 1      valentie 1   periode 3   groep  1
12 Mg   schillen 2, 8, 2      valentie 2   periode 3   groep  2
13 Al   schillen 2, 8, 3      valentie 3   periode 3   groep 13
14 Si   schillen 2, 8, 4      valentie 4   periode 3   groep 14
15 P    schillen 2, 8, 5      valentie 5   periode 3   groep 15
16 S    schillen 2, 8, 6      valentie 6   periode 3   groep 16
17 Cl   schillen 2, 8, 7      valentie 7   periode 3   groep 17
18 Ar   schillen 2, 8, 8      valentie 8   periode 3   groep 18
19 K    schillen 2, 8, 8, 1   valentie 1   periode 4   groep  1
20 Ca   schillen 2, 8, 8, 2   valentie 2   periode 4   groep  2
```

Het programma is zo goed als het model: probeer $z = 26$ (ijzer) en je krijgt 2, 8, 8, 2 met 6 elektronen te weinig, want de lus stopt na vier schillen. Echt ijzer heeft 2, 8, 14, 2. Een model in code heeft hetzelfde geldigheidsgebied als het model op papier (hoofdstuk 4): tot en met calcium. In opdracht 12.11 schrijf je zelf een functie die de gemiddelde atoommassa berekent.

UITLEG · 12.19

Habitat: welke elementen heeft een plant nodig?

19/21

Een plant bestaat voor meer dan 90 % (van de droge massa) uit koolstof, waterstof en zuurstof. Die haalt ze uit de lucht (koolstofdioxide) en uit water. Daarnaast heeft ze nog veertien andere elementen nodig, die ze als ionen opneemt uit het water rond haar wortels. In de hydroponie van de klimaatmodule zit er geen aarde rond de wortels: alles wat de plant nodig heeft, moet in de voedingsoplossing zitten.

- **Hoofdelementen** (tientallen tot honderden mg per liter oplossing): stikstof, fosfor, kalium, calcium, magnesium en zwavel.
- **Sporenelementen** (minder dan enkele mg per liter): ijzer, mangaan, zink, koper, boor, molybdeen, chloor en nikkel.

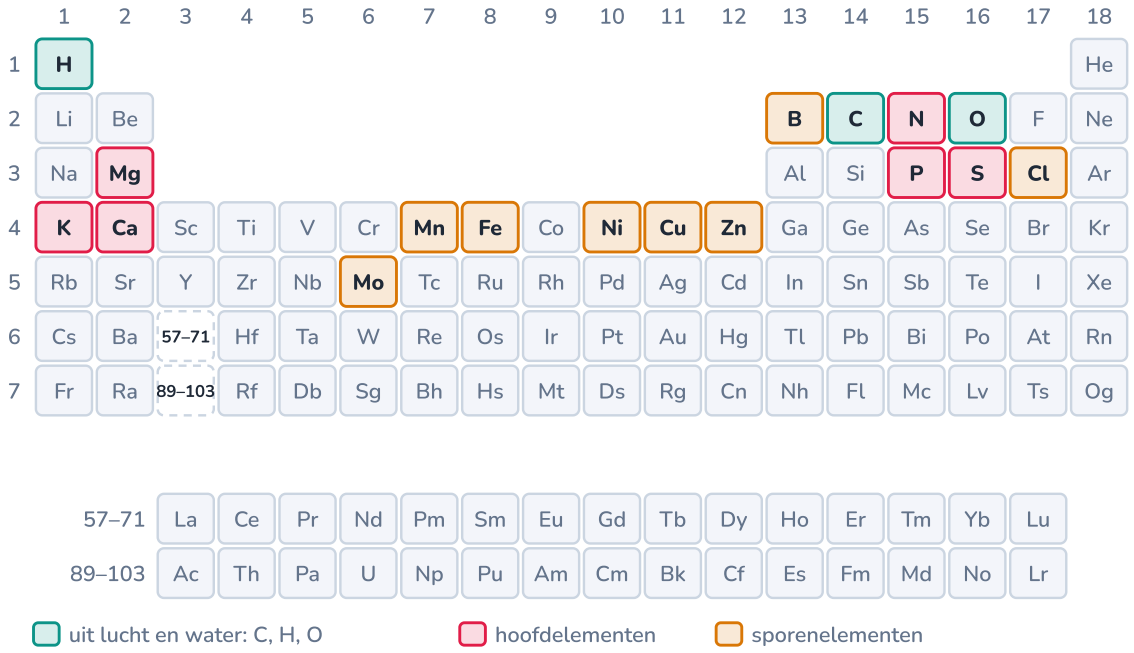
element	opgenomen als	waarvoor de plant het gebruikt	tekort zie je aan
stikstof N	NO_3^-, NH_4^+	eiwitten, bladgroen (chlorofyl), DNA	oude bladeren worden egaal geel
fosfor P	$H_2PO_4^-$	DNA, energiedrager ATP, wortelgroei	trage groei, paarsige bladeren
kalium K	K^+	waterhuishouding, openen en sluiten van de huidmondjes	bruine bladranden bij oude bladeren
calcium Ca	Ca^{2+}	stevige celwanden	jonge bladeren met bruine randjes (randjesbrand bij sla)
magnesium Mg	Mg^{2+}	het centrale atoom van elke chlorofylmolecule	geel tussen de nerven van oude bladeren
zwavel S	SO_4^{2-}	eiwitten (twee aminozuren bevatten zwavel)	jonge bladeren lichtgroen
ijzer Fe	Fe^{2+}, ijzerchelaat	aanmaak van chlorofyl, enzymen	jonge bladeren geel met groene nerven
Mn, Zn, Cu, B, Mo, Cl, Ni	als ionen, in sporen	enzymen, celwanden (B), nitraat verwerken (Mo)	vlekken, misvormde jonge bladeren

Voor de klimaatmodule mengen we de zouten zelf. Per liter regenwater: 0,59 g calciumnitraat ($Ca(NO_3)_2{\cdot}4H_2O$), 0,30 g kaliumnitraat (KNO_3), 0,25 g magnesiumsulfaat ($MgSO_4{\cdot}7H_2O$) en 0,14 g

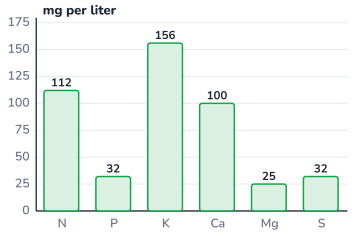
kaliumdiwaterstoffosfaat (KH_2PO_4), samen 1,28 g, plus een kant-en-klaar mengsel met ijzer en de andere sporenelementen. Dat levert de hoeveelheden in de grafiek. Hoe je die milligrammen uit de formules berekent, leer je in hoofdstuk 14; waarom zouten als ionen oplossen, in hoofdstuk 13.

Wat betekent NPK 20-20-20?

Op een fles plantenvoeding staan drie getallen: het massapercentage stikstof, fosfor en kalium. Fosfor en kalium staan er volgens een oude afspraak niet als element op, maar als de oxiden P_2O_5 en K_2O, ook al zitten die oxiden helemaal niet in de fles. In hoofdstuk 14 reken je uit hoeveel fosfor en kalium er echt in zitten.



Zeventien elementen zijn onmisbaar. Op één na (molybdeen) staan ze allemaal in de eerste vier perioden.



Het recept van de klimaatmodule: vooral kalium, stikstof en calcium. Sla heeft er genoeg aan.

NASLAG · 12.20

Overzicht: atoombouw en periodiek systeem

20/21

aantal neutronen

$$N = A - Z$$

symbool	betekenis	eenheid
A	massagetal: protonen + neutronen	–
Z	atoomnummer: aantal protonen	–
N	aantal neutronen	–

lading van een ion

$$\text{lading} = Z - \text{aantal elektronen}$$

gemiddelde atoommassa

$$\bar{m} = f_1\, m_1 + f_2\, m_2 + \ldots$$

symbool	betekenis	eenheid
m_i	massa van isotoop i	u
f_i	aandeel van isotoop i (samen 1)	–

schillen (tot calcium: 2, 8, 8, 2)

$$\text{hoogstens } 2n^2 \text{ elektronen in schil } n$$

atomaire massa-eenheid

$$1\,\text{u} = 1{,}6605 \cdot 10^{-27}\,\text{kg}$$

deeltje	symbool	waar	lading	massa (kg)	massa (u)
proton	p^+	kern	+1	$1{,}673 \cdot 10^{-27}$	1,007
neutron	n	kern	0	$1{,}675 \cdot 10^{-27}$	1,009
elektron	e^-	rond de kern	−1	$9{,}109 \cdot 10^{-31}$	0,000 549

in het periodiek systeem	zegt je
atoomnummer (rangnummer)	aantal protonen, en elektronen in het neutrale atoom
periode (rij)	aantal schillen
groep (kolom)	aantal valentie-elektronen: groep 1, 2 → 1, 2; groep 13 tot 18 → 3 tot 8
links van de trap	metaal: geeft elektronen af, positief ion
rechts van de trap	niet-metaal: neemt elektronen op of deelt ze
groep 18	edelgas: volle buitenste schil, reageert bijna niet

Habitat: rekenwaarden uit dit hoofdstuk	
voedingsoplossing klimaatmodule (per liter regenwater)	0,59 g $Ca(NO_3)_2{\cdot}4H_2O$, 0,30 g KNO_3, 0,25 g $MgSO_4{\cdot}7H_2O$, 0,14 g KH_2PO_4 + sporenelementen
hoofdelementen (mg/L)	N 112, P 32, K 156, Ca 100, Mg 25, S 33
voor het reservoir van 60 L	35,4 g calciumnitraat, 18,0 g kaliumnitraat, 15,0 g magnesiumsulfaat, 8,4 g kaliumdiwaterstoffosfaat

HET GEHEEL · 12.21

Waar het atoom terugkomt

21/21

Het atoommodel is de basis van de hele leerlijn materie en chemie. Het deeltjesmodel uit F6 krijgt hier een binnenkant; in de volgende hoofdstukken verklaar je daarmee bindingen, reacties, zuren en batterijen. Maar het atoom duikt ook op in de elektronica (vrije elektronen, halfgeleiders), in de biologie (koolstofdatering, plantvoeding) en in de energie (kernenergie, zonnecellen).

F6 Het deeltjesmodel

De deeltjes van F6 blijken atomen, moleculen en ionen te zijn.

F8 Zuivere stoffen en mengsels

Zuivere stoffen, oplossingen en de geleidingsproef: ionen zijn geladen atomen.

F19 Spanning, stroom en weerstand

Stroom is een stroom van elektronen, de deeltjes uit dit hoofdstuk.

I5 Groei en verval: exponentiële modellen

Radioactief verval en de halveringstijd van koolstof-14.

I13 Chemische bindingen en stofeigenschappen

Valentie-elektronen en de octetregel verklaren ionbinding, atoombinding en metaalbinding.

I14 Reacties en stoichiometrie

Met de atoommassa's reken je molaire massa's en de massa's bij een reactie.

I16 Redox en batterijen

Een batterij verplaatst elektronen van het ene atoom naar het andere.

I23 Halfgeleiders: diode, led en transistor

Silicium, de metalloïde tussen geleider en isolator, in diodes en transistors.

I36 Hydroponie en voedselproductie

De voedingsoplossing samenstellen en bewaken.

E27 Elektrochemie: batterijen, brandstofcellen en elektrolyse

Elektronen overdragen, kwantitatief: celspanning en elektrolyse.

E28 Materiaalkunde

Kristalroosters, dotering van halfgeleiders: het atoom als bouwsteen van materialen.

Habitat

Voor Habitat leggen we het recept van de voedingsoplossing van de klimaatmodule vast: per liter regenwater 0,59 g calciumnitraat, 0,30 g kaliumnitraat, 0,25 g magnesiumsulfaat en 0,14 g kaliumdiwaterstoffosfaat, plus sporenelementen. Dat geeft 112 mg stikstof, 32 mg fosfor, 156 mg kalium, 100 mg calcium, 25 mg magnesium en 33 mg zwavel per liter. In opdracht 12.12 zoek je de elementen erin terug; in hoofdstuk 13 meet je de sterkte van de oplossing met de EC, in hoofdstuk 14 reken je de milligrammen zelf na.

SAMENVATTING

In het kort

- Een atoom bestaat uit een piepkleine kern met protonen (+1) en neutronen (0), en elektronen (−1) eromheen. De kern bevat meer dan 99,9 % van de massa.
- Het atoomnummer Z (aantal protonen) bepaalt het element. Het massagetal A telt protonen en neutronen: $N = A - Z$.
- Een ion heeft meer of minder elektronen dan protonen: Na^+ heeft 10 elektronen, Cl^- heeft er 18.
- Isotopen zijn atomen van hetzelfde element met een ander aantal neutronen. De atoommassa in het periodiek systeem is het gewogen gemiddelde van de isotopen (chloor: 35,45 u).
- In het schillenmodel zitten de elektronen in schillen: tot calcium volgens 2, 8, 8, 2. Het model verklaart niet waarom de vierde schil al begint bij kalium, en elektronen draaien niet echt in banen.
- Valentie-elektronen (buitenste schil) bepalen hoe een atoom reageert. Atomen streven naar een edelgasconfiguratie: metalen worden positieve ionen, niet-metalen negatieve ionen of ze delen elektronen.
- Het periodiek systeem ordent de elementen volgens atoomnummer: periode = aantal schillen, groep = aantal valentie-elektronen. Mendelejev voorspelde er gallium en germanium mee.
- Metalen staan links, niet-metalen rechtsboven, metalloïden langs de trap, edelgassen in groep 18.

- Een plant heeft naast C, H en O nog zes hoofdelementen (N, P, K, Ca, Mg, S) en acht sporenelementen nodig, die ze als ionen uit de voedingsoplossing haalt.

BEGRIPPEN

Wat je nu kent

atoom
Het kleinste deeltje van een element: een kern van protonen en neutronen met elektronen eromheen.

proton
Positief geladen deeltje in de kern; het aantal protonen bepaalt het element.

neutron
Ongeladen deeltje in de kern, met ongeveer dezelfde massa als een proton.

elektron
Negatief geladen deeltje rond de kern, ongeveer 1836 keer lichter dan een proton.

atoomnummer
Het aantal protonen in de kern, symbool Z; het rangnummer in het periodiek systeem.

massagetal
Het aantal protonen plus neutronen in de kern, symbool A.

isotopen
Atomen van hetzelfde element (evenveel protonen) met een verschillend aantal neutronen.

atomaire massa-eenheid
Eén twaalfde van de massa van een koolstof-12-atoom: $1\,\mathrm{u} = 1{,}6605 \cdot 10^{-27}\,\mathrm{kg}$.

gemiddelde atoommassa
Het gewogen gemiddelde van de isotoopmassa's van een element, met hun aandeel in de natuur als gewicht.

schillenmodel
Model waarin de elektronen in schillen rond de kern zitten, met hoogstens 2, 8, 18, 32 elektronen per schil.

valentie-elektronen
De elektronen in de buitenste schil; ze bepalen hoe een atoom reageert.

edelgasconfiguratie
Een volle buitenste schil (8 elektronen, bij helium 2), zoals bij de edelgassen: bijzonder stabiel.

ion
Een atoom (of groepje atomen) met meer of minder elektronen dan protonen, en dus een lading.

periode
Een horizontale rij in het periodiek systeem; het nummer is het aantal schillen.

groep
Een kolom in het periodiek systeem; elementen in een groep hebben evenveel valentie-elektronen en lijken op elkaar.

metalloïde
Element met eigenschappen tussen metaal en niet-metaal in, zoals silicium: een halfgeleider.

ionisatie-energie
De energie die nodig is om het zwakst gebonden elektron uit een atoom te halen, in kJ per mol atomen.

HANDBOEK · HOOFDSTUK 13

Chemische bindingen en stofeigenschappen

Waarom zout breekt en koper buigt, waarom ijs drijft en olie niet oplost

Atomen blijven zelden alleen. Ze geven elektronen af, nemen ze op, delen ze of laten ze vrij rondzwerven, en zo ontstaan **ionbindingen**, **atoombindingen** en **metaalbindingen**. Die bindingen verklaren bijna alles wat je in F8 en F10 over stofeigenschappen leerde: waarom zout pas bij 801 °C smelt en water al bij 0 °C, waarom zout water stroom geleidt en suikerwater niet, waarom ijs drijft. Voor Habitat leer je de sterkte van de voedingsoplossing meten met haar elektrische geleidbaarheid, de **EC**.

11× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 9 lestijden

M Materie en chemie

Kubusvormige zoutkristallen: het ionrooster van natriumchloride, zichtbaar met het blote oog.

Na dit hoofdstuk

- Je kan ionbinding, atoombinding en metaalbinding beschrijven en herkennen aan de soorten atomen.
- Je kan de formule van een zout opstellen uit de ionladingen en eenvoudige stoffen correct benoemen, ook met samengestelde ionen.
- Je kan Lewisstructuren tekenen van eenvoudige moleculen en structuurformules lezen.
- Je kan met de elektronegativiteit voorspellen of een binding apolair, polair of ionair is, en of een molecule polair is.
- Je kan van der Waalskrachten en waterstofbruggen gebruiken om kookpunten te vergelijken.
- Je kan uitleggen waarom water bijzonder is: hoog kookpunt, ijs dat drijft, goed oplosmiddel.
- Je kan smeltpunt, geleiding en oplosbaarheid van een stof verklaren uit haar binding.
- Je kan uitleggen waarom een zoutoplossing geleidt, de EC van een voedingsoplossing meten, omrekenen naar 25 °C en gebruiken om bij te sturen.

UITLEG · 13.1

Wat je al weet, en wat nu komt

1/21

In hoofdstuk 12 zag je dat atomen streven naar een volle buitenste schil, de edelgasconfiguratie. Alleen de edelgassen hebben die al; alle andere atomen gaan **bindingen** aan om er een te krijgen. Dat kan op drie manieren, en elke manier geeft stoffen met eigen eigenschappen.

wat je al weet	waar	wat er nu bij komt
valentie-elektronen, edelgasconfiguratie, ionen zoals Na^+ en Cl^-	hoofdstuk 12	hoe atomen daardoor aan elkaar vastzitten
zout water geleidt, suikerwater niet	F8	waarom: vrije ionen tegenover moleculen
olie lost niet op in water; ijs drijft	F7, F8	polariteit en waterstofbruggen
smeltpunt en kookpunt als stofeigenschap	F6, F8	waarom ijzer bij 1538 °C smelt en ijs bij 0 °C
metalen geleiden, zijn pletbaar	F10	de metaalbinding

In dit hoofdstuk

- leer je de drie soorten binding kennen en herkennen;
- stel je formules en namen op van zouten en eenvoudige moleculen;
- teken je Lewisstructuren en lees je structuurformules;
- ontdek je waarom sommige moleculen een positieve en een negatieve kant hebben, en wat dat doet met hun kookpunt en oplosbaarheid;
- verklaar je de eigenaardigheden van water;
- meet je voor Habitat de sterkte van de voedingsoplossing met de EC.

Overdragen, delen of vrijlaten: wat er met de valentie-elektronen gebeurt, bepaalt het soort binding.

VERHAAL · 13.2

Arrhenius: zout valt uiteen, ook zonder stroom

2/21

In 1834 noemde Michael Faraday de geladen deeltjes die in een oplossing naar de elektroden bewegen **ionen**, naar het Griekse woord voor 'gaand'. Hij dacht, zoals iedereen toen, dat de elektrische stroom die ionen maakte: de spanning zou de zoutmoleculen uit elkaar trekken.

Een halve eeuw later mat de jonge Zweed Svante Arrhenius in Uppsala de geleidbaarheid van zoutoplossingen die steeds verder verdund werden. Hij kwam tot een gewaagd besluit: een zout valt in

water *vanzelf* uiteen in ionen, ook als er geen stroom loopt. De stroom zet de ionen alleen in beweging. Zijn doctoraat (1884) overtuigde de professoren in Uppsala nauwelijks: hij kreeg de laagste graad waarmee je nog slaagde. Maar enkele bekende chemici buiten Zweden, zoals Wilhelm Ostwald en Jacobus van 't Hoff, zagen dat zijn idee heel wat verklaarde, onder meer waarom zout water bij een lagere temperatuur bevriest dan suikerwater met evenveel moleculen per liter.

In 1903 kreeg Arrhenius de Nobelprijs voor scheikunde voor zijn theorie van de elektrolytische dissociatie. Vandaag is ze zo vanzelfsprekend dat een hydroponieteler er elke dag op vertrouwt: een EC-meter meet hoeveel ionen er vrij in de voedingsoplossing zweven. Arrhenius is trouwens ook bekend om een heel ander werk: in 1896 berekende hij als een van de eersten hoeveel de aarde zou opwarmen als de hoeveelheid koolstofdioxide in de lucht zou toenemen (hoofdstuk 14).

Met zulke opstellingen mat Arrhenius hoe goed steeds meer verdunde zoutoplossingen geleidden.

UITLEG · 13.3

De ionbinding

3/21

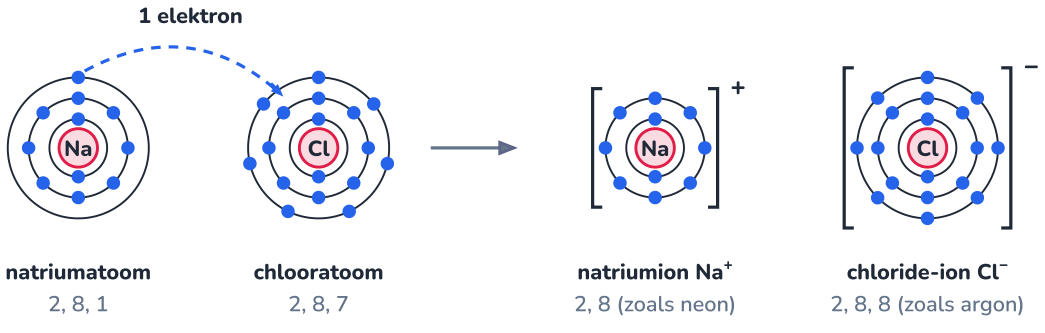
Breng natrium (2, 8, 1) en chloor (2, 8, 7) samen, en het ene valentie-elektron van natrium springt over naar chloor. Allebei krijgen ze een edelgasconfiguratie: het natriumion Na^+ die van neon, het chloride-ion Cl^- die van argon. Tegengestelde ladingen trekken elkaar aan: dat is de **ionbinding**. Zo'n binding ontstaat tussen een **metaal** (dat makkelijk elektronen afgeeft) en een **niet-metaal** (dat ze graag opneemt).

Er ontstaan geen losse 'moleculen NaCl'. Elk Na^+ trekt alle Cl^- in zijn buurt aan en omgekeerd: miljarden ionen stapelen zich in een regelmatig **ionrooster**, waarin elk ion zes buren met de tegengestelde lading

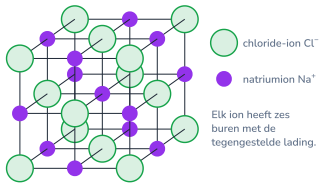
heeft. Daarom zijn zoutkristallen kubusjes. De formule NaCl zegt alleen in welke verhouding de ionen voorkomen: 1 op 1.

Uit dat rooster volgen de eigenschappen van zouten:

- **Hoog smeltpunt**: om het rooster af te breken, moet je veel sterke aantrekkingen tegelijk overwinnen. NaCl smelt bij 801 °C, magnesiumoxide MgO (ionen 2+ en 2−, dubbel zo sterk geladen) pas bij 2852 °C.
- **Hard maar bros**: een klap verschuift een laag ionen; dan komen gelijke ladingen tegenover elkaar en scheurt het kristal (figuur bij de metaalbinding).
- **Vast geleidt niet, gesmolten of opgelost wel**: in het rooster zitten de ionen vast. In een smelt of een oplossing bewegen ze vrij en dragen ze de stroom (F8).



Na de overdracht hebben beide ionen een volle buitenste schil; de haakjes met de lading tonen dat het ionen zijn.



Een stukje van het rooster, niet op schaal: in werkelijkheid raken de ionen elkaar.

UITLEG · 13.4

Formules en namen van zouten

4/21

Een zout is elektrisch neutraal: de positieve en negatieve ladingen heffen elkaar op. Dat is de hele regel om de formule te vinden. Met de ionladingen uit hoofdstuk 12 (groep 1: +1, groep 2: +2, groep 13: +3, groep 15: −3, groep 16: −2, groep 17: −1) en de tabel hieronder kan je elke eenvoudige formule opstellen.

positieve ionen		negatieve ionen (één atoom)		samengestelde ionen	
H^+	waterstofion	F^-	fluoride	OH^-	hydroxide
Na^+, K^+	natrium-, kaliumion	Cl^-	chloride	NO_3^-	nitraat
NH_4^+	ammoniumion	Br^-	bromide	SO_4^{2-}	sulfaat
Mg^{2+}, Ca^{2+}	magnesium-, calciumion	I^-	jodide	CO_3^{2-}	carbonaat
Zn^{2+}	zinkion	O^{2-}	oxide	HCO_3^-	waterstofcarbonaat

positieve ionen		negatieve ionen (één atoom)		samengestelde ionen	
Cu^{2+}	koper(II)ion	S^{2-}	sulfide	PO_4^{3-}	fosfaat
Fe^{2+}, Fe^{3+}	ijzer(II)-, ijzer(III)ion	N^{3-}	nitride	$H_2PO_4^-$	diwaterstoffosfaat
Al^{3+}	aluminiumion			CH_3COO^-	acetaat

De naam is de naam van het positieve ion, gevolgd door die van het negatieve: natriumchloride (NaCl), calciumcarbonaat ($CaCO_3$, kalk en krijt), kaliumnitraat (KNO_3), magnesiumsulfaat ($MgSO_4$), ammoniumnitraat (NH_4NO_3). Hoeveel ionen van elke soort er zijn, staat in de formule, niet in de naam. Alleen bij metalen die meer dan één lading kunnen hebben, zet je die lading erbij in Romeinse cijfers: ijzer(II)oxide FeO, ijzer(III)oxide Fe_2O_3 (roest).

Samengestelde ionen zoals nitraat of sulfaat zijn groepjes atomen die met atoombindingen aan elkaar vastzitten en samen een lading dragen. Heb je er meer dan één nodig, dan zet je ze tussen haakjes: calciumnitraat is $Ca(NO_3)_2$.

Kristalwater

Veel zouten nemen bij het kristalliseren watermoleculen mee in hun rooster. Magnesiumsulfaat uit de tuincentrum-zak (Epsom-zout, bitterzout) is $MgSO_4 \cdot 7H_2O$: magnesiumsulfaat met zeven moleculen kristalwater, de 'heptahydraat'. Dat water weegt mee als je afweegt (hoofdstuk 14), maar lost gewoon op als water.

UITGEWERKT VOORBEELD · 13.5

Formules en namen van zouten opstellen

5/21

(a) Stel de formule op van calciumchloride, aluminiumoxide, calciumnitraat en ammoniumsulfaat. (b) Geef de naam van $MgSO_4$, K_2CO_3 en KH_2PO_4, drie zouten die in meststoffen zitten.

GEGEVEN

- de ionladingen uit de groepen en de tabel met samengestelde ionen

GEVRAAGD

- formules en namen

OPLOSSING

1. **Calciumchloride**: Ca^{2+} en Cl^-. Om +2 te compenseren heb je twee chloride-ionen nodig: $CaCl_2$.
2. **Aluminiumoxide**: Al^{3+} en O^{2-}. Het kleinste veelvoud van 3 en 2 is 6: twee Al^{3+} (+6) en drie O^{2-} (−6): Al_2O_3. (Snelle weg, de *kruisregel*: de lading van het ene ion wordt de index van het andere.)
3. **Calciumnitraat**: Ca^{2+} en NO_3^-: twee nitraationen, tussen haakjes: $Ca(NO_3)_2$. Zonder haakjes zou $CaNO_{32}$ iets heel anders betekenen.
4. **Ammoniumsulfaat**: NH_4^+ en SO_4^{2-}: twee ammoniumionen: $(NH_4)_2SO_4$.
5. **Namen**: $MgSO_4$ = magnesiumsulfaat; K_2CO_3 = kaliumcarbonaat; KH_2PO_4 = kaliumdiwaterstoffosfaat (in de handel ook 'monokaliumfosfaat').

ANTWOORD

$CaCl_2$, Al_2O_3, $Ca(NO_3)_2$, $(NH_4)_2SO_4$; magnesiumsulfaat, kaliumcarbonaat, kaliumdiwaterstoffosfaat.

Klopt dit? Tel de ladingen na: $CaCl_2$: +2 − 2 = 0; Al_2O_3: +6 − 6 = 0; $(NH_4)_2SO_4$: +2 − 2 = 0. Is de som niet nul, dan is de formule fout.

UITLEG · 13.6

De atoombinding: elektronen delen

6/21

Twee niet-metalen trekken allebei hard aan elektronen: geen van beide geeft er een af. Ze **delen** er dan. Twee waterstofatomen leggen hun elektron samen in een **gemeenschappelijk elektronenpaar** tussen de twee kernen, en elk atoom 'telt' dat paar mee: beide hebben dan de configuratie van helium. Dat gedeelde paar is een **atoombinding** (ook covalente binding). Zo ontstaan **moleculen**: een vast groepje atomen, zoals H_2, H_2O of CO_2.

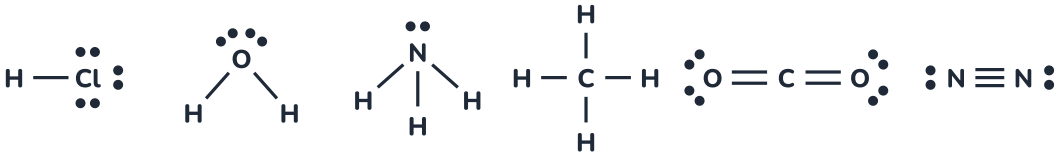
Hoeveel bindingen een atoom aangaat, volgt uit hoeveel elektronen het nog mist voor een octet (waterstof: voor twee):

atoom	H	F, Cl, Br, I	O, S	N, P	C, Si
valentie-elektronen	1	7	6	5	4
aantal bindingen	1	1	2	3	4
vrije elektronenparen	0	3	2	1	0

In een **Lewisstructuur** teken je elk gedeeld paar als een streepje en elk vrij paar als twee stippen. Soms delen twee atomen twee paren (een **dubbele binding**, zoals in CO_2) of drie (een **drievoudige binding**, zoals in N_2: die is zo sterk dat stikstofgas bijna niet reageert, en planten de stikstof uit de lucht niet kunnen gebruiken).

Namen van moleculen uit twee niet-metalen tellen met Griekse voorvoegsels: mono (1), di (2), tri (3), tetra (4), penta (5). Het eerste element krijgt geen 'mono'.

formule	naam	waar je het tegenkomt
CO	koolstofmonoxide	onvolledige verbranding, giftig (CO-melder)
CO_2	koolstofdioxide	ademhaling, verbranding, fotosynthese
SO_2	zwaveldioxide	vroeger zure regen; bewaarmiddel in wijn
N_2O	distikstofmonoxide	lachgas, ook een sterk broeikasgas uit mest
NO_2	stikstofdioxide	uitlaatgassen, luchtvervuiling
P_2O_5	difosforpentaoxide	het oxide waarmee men fosfor op een meststoffenetiket uitdrukt
CCl_4	koolstoftetrachloride	vroeger een ontvettingsmiddel, nu verboden
H_2O, NH_3, CH_4	water, ammoniak, methaan	triviale namen die iedereen gebruikt



waterstofchloride	water	ammoniak	methaan	koolstofdioxide	stikstof
HCl	H_2O	NH_3	CH_4	CO_2	N_2

Elk streepje is een gedeeld elektronenpaar, elk stippenpaar een vrij paar. Tel na: elk atoom behalve H heeft er acht rond zich.

UITGEWERKT VOORBEELD · 13.7

Lewisstructuren tekenen in vier stappen

7/21

Teken de Lewisstructuur van ammoniak NH_3 en van koolstofdioxide CO_2.

GEGEVEN

- valentie-elektronen: H 1, C 4, N 5, O 6

GEVRAAGD

- de Lewisstructuren

OPLOSSING

1. **Stap 1: tel alle valentie-elektronen.** NH_3: $5 + 3 \cdot 1 = 8$, dus 4 paren. CO_2: $4 + 2 \cdot 6 = 16$, dus 8 paren.
2. **Stap 2: zet het atoom dat de meeste bindingen maakt in het midden** (nooit H), en verbind het met de andere atomen met één streepje. NH_3: N in het midden, drie N–H (3 paren gebruikt). CO_2: O–C–O (2 paren gebruikt).
3. **Stap 3: verdeel de overige paren als vrije paren**, eerst op de buitenste atomen. NH_3: één paar over, dat gaat naar N. CO_2: nog 6 paren, drie op elke O.
4. **Stap 4: controleer het octet**, en maak zo nodig van een vrij paar een extra binding. NH_3: N heeft 3 bindingen + 1 paar = 8 elektronen, elke H 2: klaar. CO_2: elke O heeft er 8, maar C heeft er maar 4. Schuif van elke O één vrij paar naar een binding met C: O=C=O, met twee vrije paren op elke O. Nu heeft ook C er 8.

ANTWOORD

Ammoniak: N met drie enkele bindingen naar H en één vrij paar. Koolstofdioxide: O=C=O met twee dubbele bindingen en twee vrije paren op elke zuurstof (zie de figuur bij het vorige blad).

Klopt dit? Tel de elektronen in de tekening: NH_3 3 × 2 + 2 = 8; CO_2 4 × 2 (bindingen) + 4 × 2 (vrije paren) = 16. Klopt het totaal niet met stap 1, dan heb je een paar vergeten of te veel getekend.

UITLEG · 13.8

Structuurformules lezen

8/21

Een **molecuulformule** zoals C_2H_6O zegt welke atomen er in een molecule zitten, niet hoe ze aan elkaar vastzitten. Dat doet de **structuurformule**. En dat maakt uit: met C_2H_6O kan je twee heel verschillende stoffen bouwen. In **ethanol** (de alcohol in wijn en in handgel) zit een O–H-groep; ethanol is een vloeistof die kookt bij 78 °C en in elke verhouding met water mengt. In **dimethylether** zit de zuurstof tussen de

twee koolstofatomen; het is een gas dat kookt bij −24 °C. Stoffen met dezelfde molecuulformule maar een andere structuur heten **isomeren**.

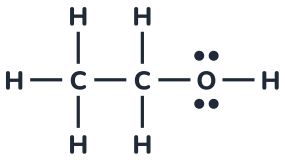
Zo lees je een structuurformule:

- elk symbool is een atoom, elk streepje een gedeeld elektronenpaar;
- tel de streepjes per atoom: C heeft er altijd 4, N 3, O 2, H en Cl 1;
- vrije elektronenparen laat men vaak weg: je weet dat O er twee heeft en N één;
- een verkorte schrijfwijze groepeert: $CH_3–CH_2–OH$ is ethanol, $CH_3–O–CH_3$ dimethylether.

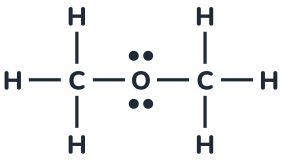
Ureum, $CO(NH_2)_2$, is de stikstofverbinding in urine en een van de meest gebruikte meststoffen ter wereld: 46 % van zijn massa is stikstof. In de bodem zetten bacteriën het om tot ammonium en nitraat, de ionen die een plant opneemt. In de hydroponie van Habitat gebruiken we het niet: daar zit geen bodemleven om het om te zetten.

Een bouwdoos helpt

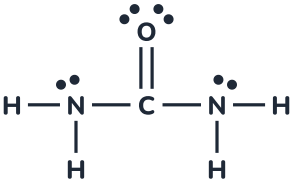
Een molecuulbouwdoos (bolletjes met gaatjes en staafjes) dwingt je om de regels te volgen: een koolstofbolletje heeft vier gaten, zuurstof twee. Bouw C_2H_6O op alle manieren die je kan vinden: je vindt er precies twee.



ethanol
C_2H_6O · kookt bij 78 °C



dimethylether
C_2H_6O · kookt bij −24 °C



ureum (meststof)
CH_4N_2O · smelt bij 133 °C

Ethanol en dimethylether hebben dezelfde atomen, anders geschikt: de ene kookt bij 78 °C, de andere bij −24 °C.

UITLEG · 13.9

Polariteit: wie trekt het hardst?

9/21

In H_2 trekken twee gelijke atomen even hard aan het gedeelde paar. In HCl niet: chloor trekt harder, en het paar ligt dichter bij chloor. Chloor krijgt een beetje negatieve lading (δ−, 'delta min'), waterstof een beetje positieve (δ+). Zo'n binding heet **polair**.

Hoe hard een atoom in een binding aan elektronen trekt, heet zijn **elektronegativiteit** (EN). Linus Pauling stelde er in 1932 een schaal voor. Fluor trekt het hardst (3,98); de alkalimetalen het zwakst. In het periodiek systeem stijgt de EN van links naar rechts en van onder naar boven.

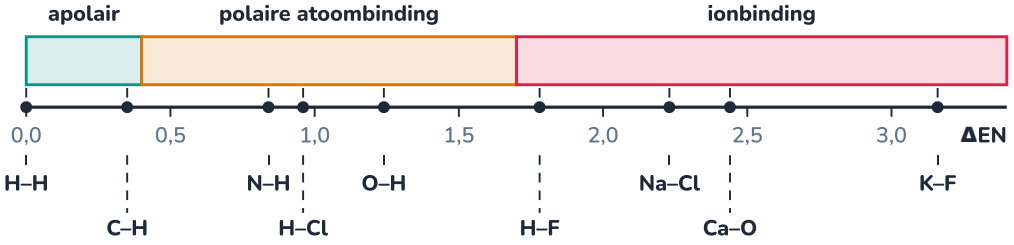
	groep 1	2	13	14	15	16	17
periode 1	H 2,20						
periode 2	Li 0,98	Be 1,57	B 2,04	C 2,55	N 3,04	O 3,44	F 3,98
periode 3	Na 0,93	Mg 1,31	Al 1,61	Si 1,90	P 2,19	S 2,58	Cl 3,16
periode 4	K 0,82	Ca 1,00					Br 2,96

Het **verschil** in EN (ΔEN) tussen twee atomen voorspelt het soort binding:

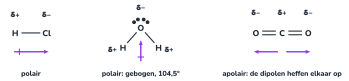
- ΔEN kleiner dan 0,4: **apolaire atoombinding** (H–H, C–H, Cl–Cl);
- ΔEN van 0,4 tot 1,7: **polaire atoombinding** (O–H 1,24; H–Cl 0,96);
- ΔEN groter dan 1,7: **ionbinding** (Na–Cl 2,23; Mg–O 2,13).

Een molecule met polaire bindingen is zelf alleen polair (een **dipool**) als de bindingen elkaar niet opheffen. Dat hangt af van de **vorm**. Koolstofdioxide is recht: de twee C=O-dipolen trekken even hard in tegengestelde richting, de molecule is apolair. Water is gebogen (104,5°): beide O–H-dipolen wijzen een beetje naar dezelfde kant, en de molecule heeft een negatieve kant (O) en een positieve kant (de H's). Methaan is een regelmatige piramide met vier gelijke C–H-bindingen: apolair.

> ⚠ **Een vuistregel, geen wet**
>
> De grenzen 0,4 en 1,7 zijn afspraken; de overgang is geleidelijk. Waterstoffluoride HF heeft een ΔEN van 1,78, maar is bij kamertemperatuur een gas van moleculen, geen zout. Gebruik de regel om te voorspellen, en controleer met de eigenschappen.



Van gelijk delen via ongelijk delen tot overdragen: een glijdende schaal.



De paarse pijl (de dipool) wijst van δ+ naar δ–. In CO_2 heffen twee even grote, tegengestelde dipolen elkaar op.

PROEF · 13.10

Van binding naar eigenschap

10/21

ONDERZOEKSVRAAG

Vijf vaste stoffen: keukenzout, suiker, kaarsvet, koper en zand. Kan je uit een paar eenvoudige tests afleiden welk soort binding elk heeft?

MATERIAAL

- een spatelpunt keukenzout (natriumchloride), kristalsuiker (sacharose), kaarsvet (paraffine), een stukje koperdraad, fijn zand (siliciumdioxide)
- vijf reageerbuizen met 5 mL gedemineraliseerd water, vijf met 5 mL zonnebloemolie
- geleidingstester (9 V-batterij, led met weerstand, twee elektroden) of multimeter
- kookplaat met een aluminium bakje (tot ongeveer 250 °C), pincet, veiligheidsbril

Veiligheid

Veiligheidsbril. Gesmolten suiker en kaarsvet zijn heet en kleven aan de huid: niets aanraken, bakje met een pincet verplaatsen. Geen open vlam bij het kaarsvet. Werk met de 9 V-batterij, nooit met het net.

Vijf stoffen, vier soorten bouw: het gedrag bij een paar eenvoudige tests verraadt de binding.

WERKWIJZE

1. Bekijk en druk elke stof met de spatel: is ze hard, bros, zacht, buigzaam?
2. Test de geleiding van de vaste stof met de elektroden.
3. Doe een spatelpunt van elke stof in water en in olie, schud en noteer of ze oplost.
4. Test de geleiding van de oplossingen in water.
5. Leg een korreltje van elke stof in het bakje op de kookplaat (stand hoog, 5 minuten). Noteer wat smelt.

WAARNEMING

	zout	suiker	kaarsvet	koper	zand
mechanisch	hard, bros	hard, bros	zacht	buigzaam	hard
geleidt vast	nee	nee	nee	ja	nee
in water	lost op	lost op	niet	niet	niet
in olie	niet	niet	lost (langzaam) op	niet	niet
oplossing in water geleidt	ja	nee	–	–	–
op de kookplaat	niets	smelt, wordt bruin	smelt snel	niets	niets

VERKLARING

Zout: hard en bros, smelt niet op de plaat, lost op in water en de oplossing geleidt: een **ionrooster**. **Suiker** en **kaarsvet**: smelten laag, geleiden nooit: **molecuulstoffen**. Suiker is polair (veel O–H-groepen) en lost op in water; kaarsvet (koolwaterstoffen) is apolair en lost op in olie. Suiker lost wel op, maar als ongeladen moleculen: de oplossing geleidt niet. **Koper**: geleidt als vaste stof, buigt: een **metaal**. **Zand**: hard, smelt niet, lost nergens in op, geleidt niet: een **atoomrooster** waarin elk siliciumatoom met vier zuurstofatomen verbonden is, en elk zuurstofatoom met twee siliciumatomen, door het hele kristal heen. Geen losse moleculen dus, en een smeltpunt van ongeveer 1710 °C.

UITLEG · 13.11

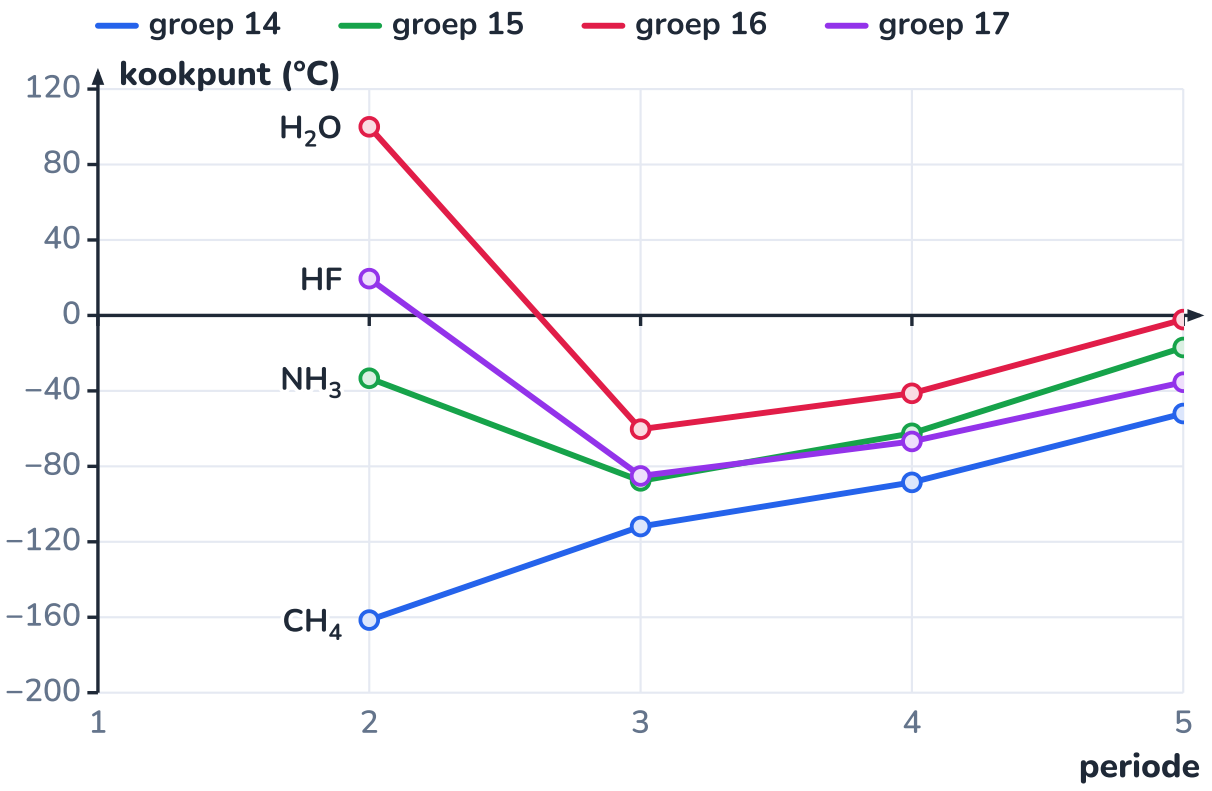
Krachten tussen moleculen: van der Waals en waterstofbruggen

11/21

Binnen een watermolecule houden atoombindingen de atomen stevig bij elkaar. Maar water kookt al bij 100 °C: dan gaan niet de atomen uit elkaar, maar de **moleculen**. Het kookpunt van een molecuulstof hangt dus af van de krachten *tussen* de moleculen, en die zijn veel zwakker.

- **Van der Waalskrachten** werken tussen alle moleculen. De elektronen van een molecule bewegen: op elk moment is de ene kant een tikje negatiever, en dat trekt de buren een beetje aan. Hoe groter de molecule (meer elektronen), hoe sterker. Daarom stijgt het kookpunt in de reeks methaan CH_4 (−161,5 °C), ethaan (−88,6 °C), propaan (−42,1 °C), butaan (−0,5 °C): lichte moleculen zijn gassen, zware worden vloeistoffen (benzine) en vaste stoffen (kaarsvet).
- **Dipoolkrachten**: polaire moleculen trekken elkaar aan met hun δ+ en δ−-kant.
- **Waterstofbruggen** zijn de sterkste: een H-atoom dat aan N, O of F gebonden is, is sterk δ+ en wordt aangetrokken door een vrij elektronenpaar van een N, O of F van een andere molecule. Een waterstofbrug is ongeveer twintig keer zwakker dan een O–H-binding (± 20 tegenover 460 kJ/mol), maar veel sterker dan de gewone krachten tussen kleine moleculen.

De grafiek toont het effect. In groep 14 stijgt het kookpunt mooi met de grootte van de molecule: van CH_4 naar SnH_4. In groep 15, 16 en 17 doet de kleinste molecule niet mee: NH_3, HF en vooral H_2O koken veel hoger dan de trend voorspelt. Zonder waterstofbruggen zou water ongeveer bij −80 °C koken, en was er op aarde geen vloeibaar water.



Gemeten kookpunten bij 1013 hPa. De eerste stof van groep 15, 16 en 17 springt eruit: waterstofbruggen.

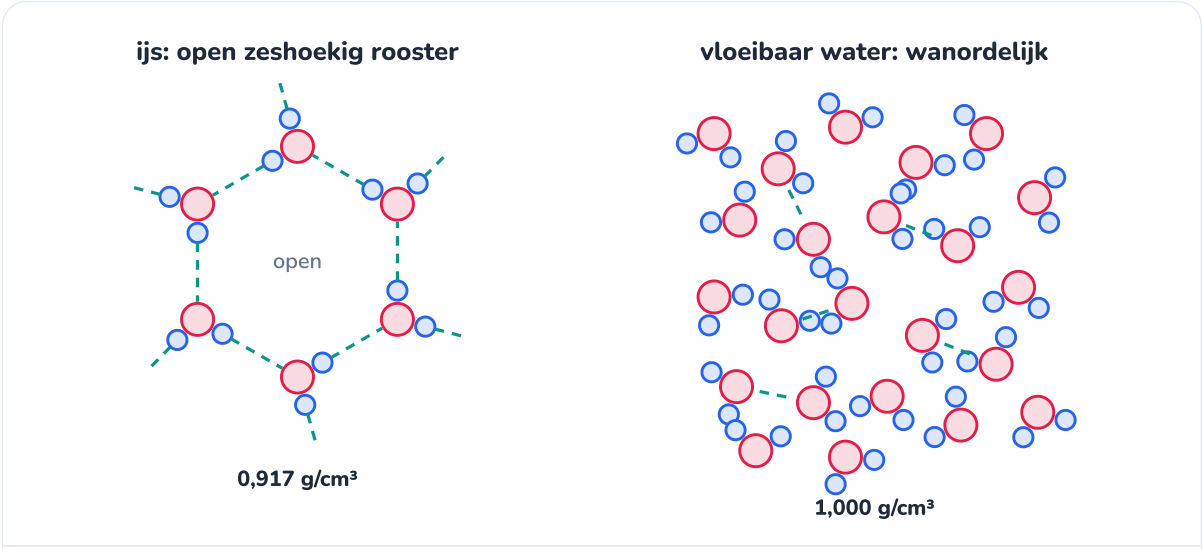
UITLEG · 13.12

Waarom water zo bijzonder is

12/21

Een watermolecule is klein, sterk polair, en kan tegelijk twee waterstofbruggen geven (met zijn twee H's) en twee ontvangen (met zijn twee vrije paren). Daaruit volgt een hele reeks eigenaardigheden, waarvan het leven afhangt:

- **Vloeibaar bij kamertemperatuur.** Moleculen van dezelfde grootte, zoals CH_4 of H_2S, zijn gassen.
- **IJs drijft.** In ijs zit elke molecule met vier waterstofbruggen vast in een open, zeshoekig rooster met veel lege ruimte. Bij het smelten breekt een deel van de bruggen en schuiven de moleculen dichter bij elkaar: vloeibaar water (bijna 1,000 g/cm³) is dichter dan ijs (0,917 g/cm³). Een vijver bevriest van boven naar onder, en de vissen overleven onder het ijs (F7).
- **Het dichtst bij 4 °C.** Onder 4 °C vormen zich al kleine stukjes ijsrooster: water zet dan uit als het kouder wordt.
- **Veel energie om op te warmen en te verdampen.** Om water 1 °C op te warmen, heb je 4,18 J per gram nodig, meer dan voor bijna elke andere stof; om het te verdampen 2,26 kJ per gram. Een kas met water als warmtebuffer (hoofdstuk 18) en een plant die zich koelt door water te verdampen, maken daar gebruik van.
- **Een grote oppervlaktespanning.** De moleculen aan het oppervlak worden naar binnen getrokken: schaatsenrijders (insecten) lopen erop. En de moleculen hangen zo sterk aan elkaar dat een plant een ononderbroken waterkolom van haar wortels tot in haar bladeren kan optrekken.
- **Het beste oplosmiddel dat we kennen** voor zouten en polaire stoffen (volgend blad).



Plat getekend: in ijs vormt elke molecule vier waterstofbruggen (stippellijnen), in drie dimensies, met open zeshoeken ertussen.

Een schaatsenrijder zakt niet door: waterstofbruggen maken het wateroppervlak stevig.

UITGEWERKT VOORBEELD · 13.13

Welke binding, welke eigenschappen?

13/21

Voorspel voor kaliumchloride KCl, koolstoftetrachloride CCl_4 en ethanol C_2H_5OH: het soort binding, een hoog of laag smeltpunt, of de stof in water oplost, en of ze (opgelost) stroom geleidt. Vergelijk daarna met de gemeten waarden.

GEGEVEN

- EN: K 0,82; Cl 3,16; C 2,55; O 3,44; H 2,20
- CCl_4: regelmatige piramide (zoals methaan)
- ethanol: $CH_3–CH_2–OH$

GEVRAAGD

- binding en eigenschappen van de drie stoffen

OPLOSSING

1. **KCl**: metaal + niet-metaal, ΔEN = 3,16 − 0,82 = 2,34: ionbinding, een ionrooster. Verwacht: hoog smeltpunt, hard en bros, oplosbaar in water met vrije ionen die geleiden. Gemeten: smelt bij 770 °C, lost goed op en geleidt.
2. **CCl_4**: twee niet-metalen, ΔEN(C–Cl) = 0,61: polaire atoombindingen. Maar de vier C–Cl-dipolen wijzen naar de hoeken van een regelmatige piramide en heffen elkaar op: een **apolaire** molecuulstof. Verwacht: laag kookpunt, lost niet op in water, wel vetten. Gemeten: kookt bij 76,7 °C, mengt niet met water (vormt een aparte laag).
3. **Ethanol**: atoombindingen, met een polaire O–H-groep (ΔEN = 1,24) die waterstofbruggen vormt. Verwacht: vloeistof met een kookpunt rond water, mengt met water, geleidt niet (geen ionen). Gemeten: kookt bij 78,4 °C, mengt in elke verhouding met water, geleidt niet.

ANTWOORD

KCl: ionrooster, hoog smeltpunt, oplosbaar en geleidend. CCl_4: apolaire moleculen, laag kookpunt, niet oplosbaar in water. Ethanol: polaire moleculen met waterstofbruggen, mengt met water, geleidt niet.

Klopt dit? Ethanol kookt zelfs iets lager dan water, hoewel de molecule zwaarder is: ze heeft maar één O–H-groep en vormt dus minder waterstofbruggen per molecule dan water.

UITLEG · 13.14

Oplossen: soort zoekt soort

14/21

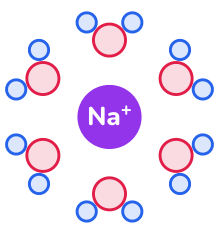
Waarom lost zout wel op in water, en olie niet? Bij het oplossen moeten de deeltjes van de opgeloste stof loskomen van elkaar, en tussen de watermoleculen gaan zitten. Dat lukt als ze aan water minstens even goed vastgehouden worden als aan elkaar.

- **Zouten**: watermoleculen draaien hun δ−-kant (de zuurstof) naar een positief ion en hun δ+-kant (een waterstof) naar een negatief ion. Omringd door water, **gehydrateerd**, laten de ionen het rooster los en zweven ze vrij in de oplossing. Daarom geleidt een zoutoplossing: de ionen kunnen naar de elektroden bewegen. Niet elk zout lost goed op: bij calciumcarbonaat (kalk) en calciumfosfaat wint het rooster.
- **Polaire moleculen** zoals suiker en ethanol vormen waterstofbruggen met water en lossen op, maar als hele, ongeladen moleculen: geen geleiding.
- **Apolaire moleculen** zoals olie, vet en kaarsvet worden door water niet aangetrokken. De watermoleculen blijven liever aan elkaar hangen en duwen de olie weg. Olie lost wel op in andere apolaire stoffen, zoals wasbenzine.

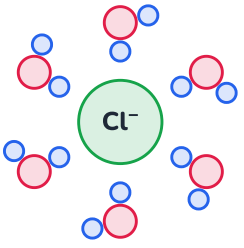
Kort: **soort zoekt soort**. Polair en ionair lossen op in polair (water, *hydrofiel*), apolair in apolair (*hydrofoob*). Zeep is een molecule met een geladen kop die in water wil en een lange apolaire staart die in vet wil: zo sleept ze vetvuil het water in.

Waarom de voedingsoplossing in twee bakken zit

Calcium en sulfaat of fosfaat vormen samen zouten die slecht oplossen (gips en calciumfosfaat). In de verdunde voedingsoplossing blijft alles opgelost, maar in een geconcentreerde stamoplossing zou het neerslaan. Telers maken daarom twee stamoplossingen: **A** met calciumnitraat (en ijzer), **B** met de sulfaten en fosfaten. Pas in het reservoir komen ze samen.



O (δ−) naar het positieve ion



één H (δ+) naar het negatieve ion

Rond een positief ion wijst de negatieve kant van water naar binnen, rond een negatief ion de positieve kant.

UITLEG · 13.15

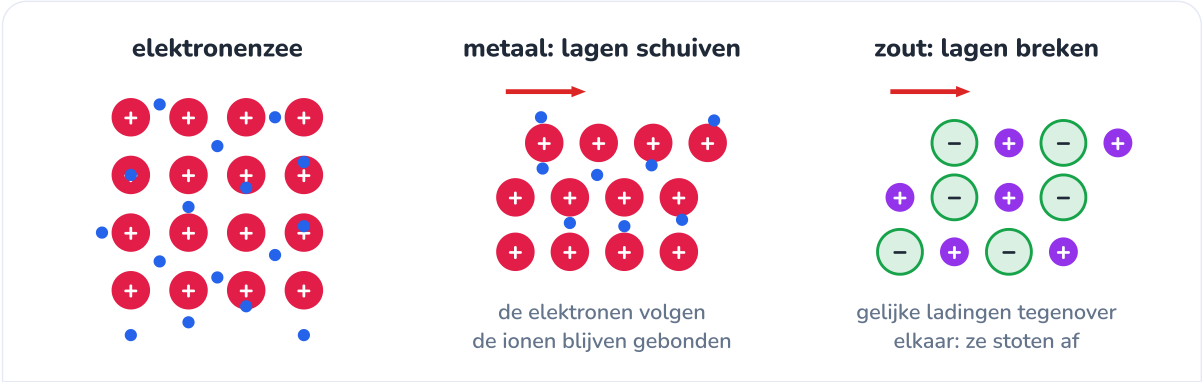
De metaalbinding

15/21

Metaalatomen hebben weinig valentie-elektronen en houden die losjes vast. In een stuk metaal geven alle atomen hun valentie-elektronen af aan het geheel: er blijft een rooster van positieve **metaalionen** over, met daartussen een **zee van vrije elektronen** die van niemand in het bijzonder zijn. De aantrekking tussen de ionen en de elektronenzee is de **metaalbinding**. Daaruit volgen de eigenschappen van metalen (F10):

- **Elektrische geleiding**: sluit je een spanning aan, dan schuift de hele elektronenzee op (F19).
- **Warmtegeleiding**: de snelle vrije elektronen dragen ook warmte mee. Een metalen lepel in hete thee wordt snel warm.
- **Pletbaar en buigzaam**: duw je een laag ionen opzij, dan volgen de elektronen mee en blijven de ionen gebonden. Bij een zout komen dan gelijke ladingen tegenover elkaar en breekt het kristal.
- **Glans**: de vrije elektronen kaatsen licht terug.

Hoe sterk de metaalbinding is, verschilt veel: natrium (één valentie-elektron) smelt al bij 98 °C en is zacht als boter; ijzer bij 1538 °C; wolfraam, het metaal van gloeidraden, pas bij 3422 °C. In een **legering** (F8) verstoren andere atomen de regelmatige lagen: die schuiven moeilijker, en het metaal wordt harder. Daarom is staal sterker dan zuiver ijzer.



Links de elektronenzee; midden schuiven de lagen van een metaal zonder te breken; rechts komen in een zout gelijke ladingen tegenover elkaar.

ONTLEED · 13.16

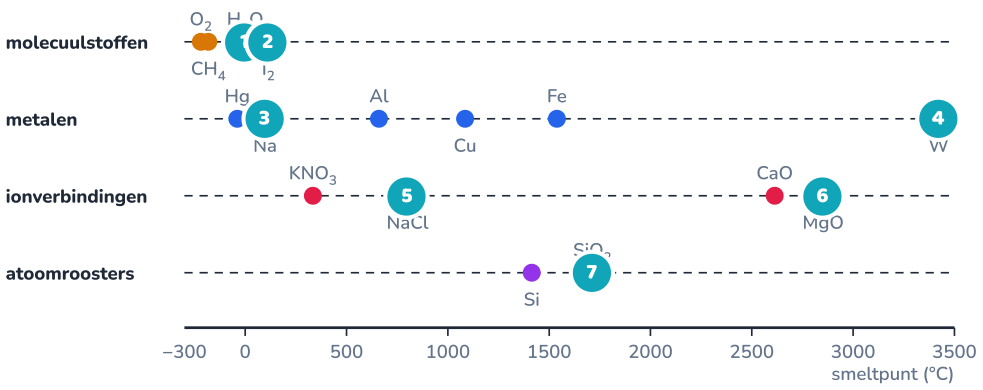
Van binding naar smeltpunt

16/21

Om een stof te laten smelten, moet je de deeltjes loswrikken uit hun vaste plaats. Hoe sterker ze aan elkaar vastzitten, hoe hoger het smeltpunt. De figuur zet gemeten smeltpunten naast elkaar, per soort binding. Het overzicht van alle eigenschappen staat in de tabel.

	ionverbinding	molecuulstof	metaal	atoomrooster
deeltjes	positieve en negatieve ionen	moleculen	metaalionen + vrije elektronen	atomen
wat houdt ze samen	ionbinding (sterk)	binnen de molecule sterk; tussen de moleculen zwak	metaalbinding	atoombindingen door het hele rooster
smeltpunt	hoog (NaCl 801 °C)	laag (H_2O 0 °C, CH_4 −182 °C)	meestal hoog (Fe 1538 °C)	zeer hoog (SiO_2 ± 1710 °C)
geleiding vast	nee	nee	ja	nee (grafiet wel)

	ionverbinding	molecuulstof	metaal	atoomrooster
geleiding gesmolten of opgelost	ja: vrije ionen	nee (behalve zuren, hoofdstuk 15)	ja (gesmolten)	lost niet op
in water	vaak oplosbaar	polair: vaak; apolair: nee	lost niet op	lost niet op
mechanisch	hard en bros	zacht	pletbaar, buigzaam	zeer hard



Gemeten smeltpunten bij 1013 hPa. Molecuulstoffen zitten bijna allemaal links, ionen en atoomroosters rechts.

1 Molecuulstoffen: laag
Smelten maakt alleen de zwakke krachten tussen de moleculen los, niet de bindingen erbinnen. Water (0 °C) ligt hoger dan zuurstof (–219 °C) door zijn waterstofbruggen.

2 Jood: groot en toch apolair
Grote moleculen met veel elektronen: sterke van der Waalskrachten. Jood is bij kamertemperatuur een vaste stof (114 °C).

3 Natrium: een zwak metaal
Eén valentie-elektron per atoom in de elektronenzee: een zwakke metaalbinding, smelt bij 98 °C.

4 Wolfraam: het hoogste
Veel elektronen in de zee en kleine ionen: 3422 °C. Daarom maakte men er gloeidraden van.

5 Natriumchloride: ionen 1+ en 1–
Het hele rooster moet loskomen: 801 °C.

6 Magnesiumoxide: ionen 2+ en 2–
Dubbele ladingen trekken bij dezelfde afstand vier keer zo hard aan: 2852 °C. MgO bekleedt daarom de binnenkant van smeltovens.

7 Kwarts: een atoomrooster
Smelten betekent hier echte atoombindingen breken, door het hele kristal: ongeveer 1710 °C.

UITLEG · 13.17

Habitat: EC, de sterkte van de voedingsoplossing meten

17/21

Hoeveel voedingszouten zitten er nog in het reservoir van 60 L? Een plant neemt niet alle ionen even snel op, en water verdampt of wordt opgedronken. Je kan niet elke dag een labo-analyse doen. Wel kan je meten hoe goed de oplossing **stroom geleidt**: hoe meer vrije ionen, hoe beter. Dat is de **elektrische geleidbaarheid**, in de hydroponie kortweg **EC** (*electrical conductivity*).

Zo werkt een EC-meter. Twee plaatjes met oppervlakte A staan op een vaste afstand d in de oplossing. De meter legt een spanning aan en meet de stroom, dus de weerstand R van de vloeistof ertussen (F19). Een kortere afstand of grotere plaatjes geven een kleinere weerstand; om een getal te krijgen dat alleen van de oplossing afhangt, rekent de meter om met de **celconstante** $K = d/A$:

ELEKTRISCHE GELEIDBAARHEID

$$\mathrm{EC} = \frac{K}{R} \qquad K = \frac{d}{A}$$

R in ohm, K in cm^{-1}, EC in siemens per cm (S/cm); in de hydroponie in mS/cm. De siemens is het omgekeerde van de ohm: $1\,\mathrm{S} = 1/\Omega$.

Drie praktische punten:

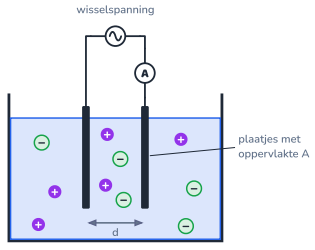
- **Wisselspanning.** Met gelijkspanning zouden de ionen zich ophopen aan de plaatjes en ontstonden er gasbelletjes (de proef in F8): de meting zou verlopen. Een EC-meter wisselt de spanning daarom honderden keren per seconde om.
- **Temperatuur.** Warmere ionen bewegen vlotter: de EC stijgt ongeveer 2 % per °C. Meters rekenen daarom om naar 25 °C (temperatuurcompensatie): $\mathrm{EC}_{25} = \mathrm{EC}_T / (1 + 0{,}02\,(T - 25))$.
- **IJken.** Met een ijkvloeistof met een bekende EC (0,01 mol/L kaliumchloride: 1,413 mS/cm) controleer je of de meter juist meet.

water of oplossing	EC bij 25 °C
zuiver water (theoretisch)	0,000 055 mS/cm
gedemineraliseerd water	0,001 à 0,01 mS/cm
regenwater	0,01 à 0,05 mS/cm
kraanwater (België, Nederland)	meestal 0,3 à 0,8 mS/cm
voedingsoplossing voor sla en kruiden	1,2 à 1,8 mS/cm
voedingsoplossing voor tomaten	2 à 5 mS/cm
ijkvloeistof: 0,01 mol/L kaliumchloride	1,413 mS/cm
zeewater	± 53 mS/cm

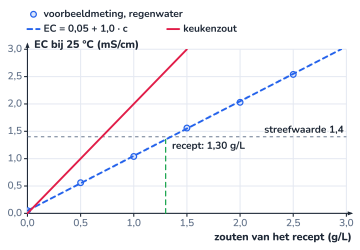
Van EC naar gram per liter. Voor een vast zoutmengsel stijgt de EC evenredig met de concentratie. De ijklijn toont het voor het recept van Habitat (hoofdstuk 12), opgelost in regenwater: ongeveer 1,0 mS/cm per gram zouten per liter, bovenop de 0,05 mS/cm van het regenwater. Het volledige recept (1,30 g/L met de sporenelementen) geeft dus 1,35 mS/cm. Als **streefwaarde** voor sla en kruiden nemen we 1,4 mS/cm; onder 1,2 of boven 1,8 stuur je bij. Keukenzout geeft per gram twee keer zoveel EC: elk zout heeft zijn eigen ijklijn.

⚠ Wat de EC niet zegt

De EC telt alle ionen, ook die de plant niet nodig heeft. Vul je steeds bij met kraanwater, dan hopen natrium, chloride en kalk zich op: de EC blijft netjes op 1,4, terwijl er steeds minder voeding in zit. Daarom gebruikt Habitat regenwater (F9), en ververs je de oplossing om de paar weken helemaal. Welk ion ontbreekt, zegt de EC ook niet: daarvoor heb je een analyse nodig.



Hoe meer ionen tussen de plaatjes, hoe groter de stroom en hoe kleiner de weerstand: hoe hoger de EC.



Voorbeeldmeting voor het recept van Habitat: een rechte met helling 1,0 mS/cm per g/L. Keukenzout ter vergelijking.

Elke dag even meten: de EC zegt hoe sterk de voedingsoplossing nog is.

UITGEWERKT VOORBEELD · 13.18

Habitat: de EC van het reservoir bijsturen

18/21

Het reservoir werd met 60 L voedingsoplossing van EC 1,40 mS/cm gevuld. Een paar warme dagen later staat er nog 51 L in, en de meter toont (na temperatuurcompensatie) 1,58 mS/cm. (a) Waarom is de EC gestegen? (b) Je vult bij tot 60 L met regenwater (EC 0,05 mS/cm). Welke EC verwacht je? (c) Hoeveel gram zouten moet je dan nog toevoegen om weer op 1,40 mS/cm te komen, en hoeveel mL van de stamoplossingen A en B (elk 100 keer geconcentreerd) is dat?

GEGEVEN

- $V_1 = 51\,\text{L}$, $\text{EC}_1 = 1{,}58\,\text{mS/cm}$; bijvullen tot $V_2 = 60\,\text{L}$ met regenwater (0,05 mS/cm)
- ijklijn: $\text{EC} = 0{,}05 + 1{,}0 \cdot c$ (c in g/L)
- stamoplossingen: 10 mL A + 10 mL B per liter geeft het volledige recept, 1,30 g/L

GEVRAAGD

- verklaring
- EC_2
- gram zouten en mL stamoplossing

OPLOSSING

1. (a) De planten namen meer water op dan zouten (verdamping via de bladeren), en er verdampte water uit het reservoir. De ionen bleven in minder water achter: de concentratie en dus de EC stegen.
2. (b) Het deel van de EC dat van de zouten komt, is $1{,}58 - 0{,}05 = 1{,}53\,\text{mS/cm}$, evenredig met de hoeveelheid zout per liter. Dezelfde zouten in 60 L in plaats van 51 L: $1{,}53 \cdot \frac{51}{60} = 1{,}30$. Met het regenwater erbij: $\text{EC}_2 = 1{,}30 + 0{,}05 = 1{,}35\,\text{mS/cm}$.
3. (c) Er ontbreekt $1{,}40 - 1{,}35 = 0{,}05\,\text{mS/cm}$, dat is $0{,}05\,\text{g/L}$ zouten (helling 1,0). Voor 60 L: $0{,}05 \cdot 60 = 3{,}0\,\text{g}$.
4. Het volledige recept is 1,30 g per 20 mL stamoplossing (10 mL A + 10 mL B). Voor 3,0 g: $20 \cdot 3{,}0/1{,}30 = 46\,\text{mL}$, dus 23 mL A en 23 mL B. Roer, wacht even en meet opnieuw.

ANTWOORD

De EC steeg omdat er water verdween en de ionen achterbleven. Na het bijvullen verwacht je 1,35 mS/cm; 3,0 g zouten (23 mL A en 23 mL B) brengen de oplossing terug op 1,40 mS/cm.

Klopt dit? Voeg nooit rechtstreeks stamoplossing A bij B: dan slaat calciumsulfaat neer. Giet ze na elkaar in het reservoir, en roer tussendoor. En let op: bijvullen met kraanwater (0,55 mS/cm) in plaats van regenwater zou de EC al tot 1,43 brengen, zonder een gram voeding extra: met natrium en kalk.

CODE · 13.19

Python: het soort binding en de EC bij 25 °C

19/21

Twee kleine functies die de regels van dit hoofdstuk toepassen. `bindingstype` gebruikt het verschil in elektronegativiteit, `ec25` de temperatuurcompensatie van een EC-meter. In de datalogger van Habitat (hoofdstuk 30) komt die tweede functie terug.

Python **binding.py** 28 regels

```python
# binding.py - het soort binding uit de elektronegativiteit, en de EC omrekenen naar 25 graden
EN = {"H": 2.20, "Li": 0.98, "C": 2.55, "N": 3.04, "O": 3.44, "F": 3.98, "Na": 0.93, "Mg": 1.31,
      "Al": 1.61, "Si": 1.90, "P": 2.19, "S": 2.58, "Cl": 3.16, "K": 0.82, "Ca": 1.00}


def bindingstype(a, b):
    verschil = abs(EN[a] - EN[b])
    if verschil < 0.4:
        soort = "apolaire atoombinding"
```

```
    elif verschil <= 1.7:
        soort = "polaire atoombinding"
    else:
        soort = "ionbinding"
    return verschil, soort


for a, b in [("H", "H"), ("C", "H"), ("H", "Cl"), ("O", "H"), ("Al", "Cl"), ("Mg", "O"), ("Na", "Cl")]:
    verschil, soort = bindingstype(a, b)
    print(f"{a}-{b}: verschil {verschil:.2f} -> {soort}")


def ec25(ec_gemeten, temperatuur, alfa=0.02):
    """Geleidbaarheid omrekenen naar 25 graden: ze stijgt ongeveer 2 % per graad."""
    return ec_gemeten / (1 + alfa * (temperatuur - 25))


for ec, t in [(1.21, 18), (1.40, 25), (1.62, 31)]:
    print(f"gemeten {ec:.2f} mS/cm bij {t} graden -> {ec25(ec, t):.2f} mS/cm bij 25 graden")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
H-H: verschil 0.00 -> apolaire atoombinding
C-H: verschil 0.35 -> apolaire atoombinding
H-Cl: verschil 0.96 -> polaire atoombinding
O-H: verschil 1.24 -> polaire atoombinding
Al-Cl: verschil 1.55 -> polaire atoombinding
Mg-O: verschil 2.13 -> ionbinding
Na-Cl: verschil 2.23 -> ionbinding
gemeten 1.21 mS/cm bij 18 graden -> 1.41 mS/cm bij 25 graden
gemeten 1.40 mS/cm bij 25 graden -> 1.40 mS/cm bij 25 graden
gemeten 1.62 mS/cm bij 31 graden -> 1.45 mS/cm bij 25 graden
```

Al–Cl (ΔEN 1,55) valt volgens de regel net onder de ionbinding, en inderdaad: aluminiumchloride sublimeert al bij 180 °C, heel laag voor een 'zout'. Een meting bij 18 °C (1,21 mS/cm) en een bij 31 °C (1,62 mS/cm) blijken omgerekend bijna dezelfde oplossing te zijn als de 1,40 mS/cm bij 25 °C: zonder compensatie zou je bij warm weer onterecht verdunnen. Op de microcontroller van Habitat:

```
// dezelfde temperatuurcompensatie op de microcontroller
float ec25(float ecGemeten, float temperatuur) {
  const float ALFA = 0.02;                      // 2 % per graad
  return ecGemeten / (1.0 + ALFA * (temperatuur - 25.0));
}
```

NASLAG · 13.20

Overzicht: bindingen en eigenschappen

20/21

soort binding (vuistregel)

$\Delta EN < 0{,}4$ apolair $0{,}4 \le \Delta EN \le 1{,}7$ polair $\Delta EN > 1{,}7$ ion

elektrische geleidbaarheid

$$\mathrm{EC} = \frac{K}{R} \qquad K = \frac{d}{A}$$

symbool	betekenis	eenheid
K	celconstante	$\mathrm{cm^{-1}}$
R	weerstand van de oplossing tussen de plaatjes	Ω
EC	geleidbaarheid	S/cm, mS/cm

temperatuurcompensatie

$$\mathrm{EC}_{25} = \frac{\mathrm{EC}_T}{1 + 0{,}02\,(T - 25)}$$

symbool	betekenis	eenheid
T	temperatuur van de oplossing	°C

	ionverbinding	molecuulstof	metaal	atoomrooster
deeltjes	positieve en negatieve ionen	moleculen	metaalionen + vrije elektronen	atomen
wat houdt ze samen	ionbinding (sterk)	binnen de molecule sterk; tussen de moleculen zwak	metaalbinding	atoombindingen door het hele rooster
smeltpunt	hoog (NaCl 801 °C)	laag (H_2O 0 °C, CH_4 −182 °C)	meestal hoog (Fe 1538 °C)	zeer hoog (SiO_2 ± 1710 °C)
geleiding vast	nee	nee	ja	nee (grafiet wel)
geleiding gesmolten of opgelost	ja: vrije ionen	nee (behalve zuren, hoofdstuk 15)	ja (gesmolten)	lost niet op
in water	vaak oplosbaar	polair: vaak; apolair: nee	lost niet op	lost niet op
mechanisch	hard en bros	zacht	pletbaar, buigzaam	zeer hard

positieve ionen		negatieve ionen (één atoom)		samengestelde ionen	
H^+	waterstofion	F^-	fluoride	OH^-	hydroxide
Na^+, K^+	natrium-, kaliumion	Cl^-	chloride	NO_3^-	nitraat
NH_4^+	ammoniumion	Br^-	bromide	SO_4^{2-}	sulfaat
Mg^{2+}, Ca^{2+}	magnesium-, calciumion	I^-	jodide	CO_3^{2-}	carbonaat
Zn^{2+}	zinkion	O^{2-}	oxide	HCO_3^-	waterstofcarbonaat
Cu^{2+}	koper(II)ion	S^{2-}	sulfide	PO_4^{3-}	fosfaat
Fe^{2+}, Fe^{3+}	ijzer(II)-, ijzer(III)ion	N^{3-}	nitride	$H_2PO_4^-$	diwaterstoffosfaat
Al^{3+}	aluminiumion			CH_3COO^-	acetaat

Habitat: rekenwaarden uit dit hoofdstuk	
EC-streefwaarde voedingsoplossing	1,4 mS/cm bij 25 °C; bijsturen onder 1,2 of boven 1,8
ijklijn recept (regenwater)	EC = 0,05 + 1,0 · c (c in g/L); volledig recept 1,30 g/L → 1,35 mS/cm
stamoplossingen	A (calciumnitraat, ijzer) en B (sulfaten, fosfaat), elk 100 ×: 10 mL A + 10 mL B per liter
ijken	0,01 mol/L KCl = 1,413 mS/cm bij 25 °C

HET GEHEEL · 13.21

Waar bindingen terugkomen

21/21

Bindingen zijn het scharnier tussen het atoom en de stof: met de valentie-elektronen uit hoofdstuk 12 verklaar je hier de stofeigenschappen uit F6 tot F10. Zo goed als elk volgend chemie- en materiaalhoofdstuk bouwt erop voort: reacties breken en maken bindingen, zuren zijn polaire moleculen die ionen vormen, polymeren zijn reuzenmoleculen, halfgeleiders zijn atoomroosters.

F6 Het deeltjesmodel
Smelten en koken: de krachten tussen de deeltjes, nu met een naam.

F7 Massa, volume en dichtheid
Waarom ijs drijft: het open rooster van waterstofbruggen.

F8 Zuivere stoffen en mengsels
Oplossen en de geleidingsproef: ionen tegenover moleculen.

F10 Materialen en hun eigenschappen

Metalen, keramiek en kunststoffen: hun eigenschappen volgen uit de binding.

I12 Atomen en het periodiek systeem

Valentie-elektronen, ionladingen en de edelgasconfiguratie.

I14 Reacties en stoichiometrie

Een reactie breekt bindingen en maakt nieuwe; met formules reken je.

I15 Zuren, basen en pH

Zuren zijn polaire moleculen die in water ionen vormen.

I17 Polymeren en composieten

Reuzenmoleculen en de krachten tussen hun ketens.

I18 Warmte berekenen

De grote soortelijke warmte van water: waterstofbruggen.

I36 Hydroponie en voedselproductie

EC en pH samen bewaken in het hydroponiesysteem.

E26 Chemisch evenwicht

Oplosbaarheid en neerslag, zoals gips in de stamoplossing, kwantitatief.

E28 Materiaalkunde

Kristalroosters, legeringen en halfgeleiders in detail.

Habitat

Voor Habitat leggen we vast: de voedingsoplossing heeft een streefwaarde van 1,4 mS/cm (bij 25 °C), bijsturen onder 1,2 of boven 1,8. Het recept uit hoofdstuk 12 op regenwater geeft 1,35 mS/cm; per gram zouten per liter stijgt de EC met ongeveer 1,0 mS/cm. De zouten gaan in twee stamoplossingen, A en B, om neerslag van gips te vermijden. In opdracht 13.12 ontwerp je de EC-bewaking; in hoofdstuk 36 komt de pH erbij.

SAMENVATTING

In het kort

- Atomen binden om een edelgasconfiguratie te krijgen: door elektronen over te dragen (ionbinding), te delen (atoombinding) of vrij te laten (metaalbinding).
- Metaal + niet-metaal geeft een ionrooster: hoog smeltpunt, hard en bros, geleidt alleen gesmolten of opgelost.
- De formule van een zout volgt uit de ionladingen: de som is nul ($CaCl_2$, Al_2O_3, $Ca(NO_3)_2$). De naam: positief ion + negatief ion (calciumnitraat).
- Niet-metalen delen elektronenparen: moleculen. In een Lewisstructuur heeft elk atoom (behalve H) acht elektronen rond zich. Isomeren hebben dezelfde formule maar een andere structuur.
- Het verschil in elektronegativiteit bepaalt of een binding apolair, polair of ionair is; de vorm bepaalt of een molecule een dipool is (H_2O wel, CO_2 niet).
- Tussen moleculen werken van der Waalskrachten (groter bij grotere moleculen) en waterstofbruggen (bij H aan N, O of F): ze bepalen smelt- en kookpunten van molecuulstoffen.
- Water is bijzonder door zijn waterstofbruggen: vloeibaar bij kamertemperatuur, ijs drijft, grote warmtecapaciteit, uitstekend oplosmiddel.
- Soort zoekt soort: ionen en polaire stoffen lossen op in water, apolaire in apolaire oplosmiddelen. Een zoutoplossing geleidt omdat de ionen vrij bewegen.
- De EC meet hoeveel ionen er in een oplossing zitten (mS/cm, omgerekend naar 25 °C). Voor Habitat: streefwaarde 1,4 mS/cm.

BEGRIPPEN

Wat je nu kent

ionbinding
De aantrekking tussen positieve en negatieve ionen in een ionrooster; ontstaat als een metaal elektronen overdraagt aan een niet-metaal.

atoombinding
Een gemeenschappelijk elektronenpaar tussen twee atomen (meestal niet-metalen); ook covalente binding.

metaalbinding
De aantrekking tussen positieve metaalionen en de vrije elektronen (elektronenzee) ertussen.

Lewisstructuur
Tekening van een molecule met alle valentie-elektronen: bindingen als streepjes, vrije elektronenparen als stippen.

elektronegativiteit
Hoe sterk een atoom in een binding de elektronen naar zich toe trekt (schaal van Pauling, 0,7 tot 4,0).

polaire binding
Atoombinding waarin de elektronen dichter bij het meest elektronegatieve atoom liggen: dat atoom wordt δ−, het andere δ+.

dipool
Molecule met een positieve en een negatieve kant, doordat de polaire bindingen elkaar niet opheffen.

waterstofbrug
Aantrekking tussen een H-atoom dat gebonden is aan N, O of F en een vrij elektronenpaar van een N, O of F van een andere molecule.

van der Waalskracht
Zwakke aantrekking tussen alle moleculen, groter naarmate de moleculen groter zijn.

isomeren
Stoffen met dezelfde molecuulformule maar een andere structuur, en dus andere eigenschappen.

hydratatie
Het omringen van opgeloste ionen door watermoleculen, met de passende kant naar het ion gericht.

EC (elektrische geleidbaarheid)
Hoe goed een oplossing stroom geleidt, in mS/cm; een maat voor de hoeveelheid opgeloste ionen.

siemens
Eenheid van geleiding: het omgekeerde van de ohm, $1\,\mathrm{S} = 1/\Omega$.

HANDBOEK · HOOFDSTUK 14

Reacties en stoichiometrie

Atomen tellen door te wegen: van de gesloten kolf van Lavoisier tot de CO_2 van een liter benzine

Bij een chemische reactie verdwijnen er geen atomen en komen er geen bij: ze worden alleen anders gebonden. Daarom kan je een reactie in een kloppende vergelijking schrijven, en daarom kan je ermee rekenen. Omdat atomen te klein zijn om te tellen, tel je ze door te wegen: met de **mol**. Zo bereken je hoeveel zuurstof een gasvlam verbruikt, waarom een liter benzine meer dan twee kilogram koolstofdioxide oplevert, en voor Habitat hoeveel CO_2 de slaplanten van de kweekruimte opnemen, en waar die vandaan moet komen.

9× uitleg · 7× uitgewerkt voorbeeld · 2× proef · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

Materie en chemie · Wiskundig modelleren

Methaan en zuurstof worden koolstofdioxide en water: de druppeltjes op de koude lepel zijn reactieproduct.

Na dit hoofdstuk

- Je kan een reactievergelijking kloppend maken en het verschil tussen coëfficiënt en index uitleggen.
- Je kan de wet van behoud van massa toepassen en verklaren met het deeltjesmodel.
- Je kan de molaire massa van een stof berekenen, ook met haakjes en kristalwater.
- Je kan omrekenen tussen massa, hoeveelheid stof (mol), aantal deeltjes en het volume van een gas.
- Je kan met een kloppende vergelijking de massa of het volume van een reactieproduct of een beginstof berekenen.
- Je kan berekenen hoeveel koolstofdioxide ontstaat bij het verbranden van een brandstof, en volledige en onvolledige verbranding onderscheiden.
- Je kan exotherme en endotherme reacties herkennen en in een energiediagram voorstellen.
- Je kan berekenen hoeveel CO2 de planten van de kweekruimte nodig hebben, en bronnen en ventilatie beoordelen.

UITLEG · 14.1

Wat je al weet, en wat nu komt

In F8 zag je het verschil tussen een mengsel en een reactie: als je zout oplost, blijft het zout bestaan; als je hout verbrandt, ontstaan nieuwe stoffen. In hoofdstuk 12 en 13 leerde je welke atomen er bestaan en hoe ze aan elkaar vastzitten. Nu breng je het samen: bij een reactie breken bindingen en vormen zich nieuwe, en je gaat er precies mee rekenen.

wat je al weet	waar	wat er nu bij komt
bij een mengsel blijven de stoffen bestaan; bij een reactie ontstaan nieuwe stoffen	F8	reacties in formules schrijven
fotosynthese: koolstofdioxide + water → suiker + zuurstof	F27	dezelfde reactie, kloppend en met getallen
atoommassa's in u	hoofdstuk 12	de mol: van atomen tellen naar grammen wegen
formules van zouten en moleculen	hoofdstuk 13	molaire massa's berekenen
energie in joule, energie blijft behouden	F15, hoofdstuk 9	energie die vrijkomt of opgenomen wordt bij een reactie
koolstofdioxide en het klimaat	F17	hoeveel CO_2 er uit een liter brandstof komt

In dit hoofdstuk

- schrijf je reacties als **kloppende vergelijkingen**, en toon je aan dat de massa behouden blijft;
- leer je de **mol** kennen, de eenheid waarmee chemici atomen tellen door te wegen;
- reken je met reacties: hoeveel gram, hoeveel liter gas;
- bereken je hoeveel CO_2 er uit een liter benzine of diesel komt;
- onderscheid je reacties die warmte afgeven en reacties die warmte opnemen;
- reken je voor Habitat uit hoeveel CO_2 de planten van de kweekruimte nodig hebben.

Brandende staalwol wordt zwaarder: het ijzer bindt zuurstof uit de lucht. Een reactie, geen verdwijntruc.

VERHAAL · 14.2

Lavoisier en de gesloten kolf

2/22

Wie in de achttiende eeuw een stuk metaal lang verhitte, zag het dof worden en een beetje zwaarder. Maar hout dat verbrandde, werd lichter. De meeste chemici verklaarden dat met een onzichtbare stof, het *flogiston*, dat bij het branden zou ontsnappen.

De Fransman Antoine Lavoisier geloofde alleen wat hij kon wegen. Hij liet de nauwkeurigste balansen van zijn tijd bouwen en deed zijn proeven in **gesloten** glazen vaten. In 1774 verhitte hij tin in een dichtgesmolten glazen kolf. Het tin werd grijs, maar de kolf met alles erin woog na afloop precies evenveel als ervoor. Pas toen hij de kolf opende, stroomde er lucht naar binnen en werd het geheel zwaarder. Het metaal had dus iets uit de lucht opgenomen. Dat bestanddeel van de lucht, dat Joseph Priestley en Carl Scheele kort voordien apart gemaakt hadden, noemde Lavoisier in 1777 **oxygène**: zuurstof. Branden en roesten zijn reacties met zuurstof; flogiston bestond niet.

In 1783 toonde hij dat water geen element is maar een verbinding van waterstof (dat hij zo noemde: 'watermaker') en zuurstof. In zijn leerboek van 1789 schreef hij de regel op die sindsdien de basis is van alle scheikunde: bij een reactie gaat geen massa verloren en komt er geen bij. Zijn vrouw, Marie-Anne Paulze Lavoisier, werkte mee in het labo, hield de metingen bij en tekende de dertien platen met toestellen in dat boek.

Lavoisier was ook belastingpachter voor de koning. Tijdens de Franse Revolutie werd hij daarvoor veroordeeld, en op 8 mei 1794 onthoofd.

Waarom 'gesloten' het verschil maakte

Robert Boyle had al een eeuw eerder metalen in een afgesloten vat verhit, maar hij opende het vat vóór het wegen. Hij zag de massa stijgen en dacht dat vuurdeeltjes door het glas waren gedrongen. Eén detail in de werkwijze, het wegen van het hele gesloten systeem, maakte het verschil tussen een foute en een juiste verklaring.

Balansen, gesloten kolven en veel geduld: het gereedschap waarmee Lavoisier de scheikunde tot een meetwetenschap maakte.

UITLEG · 14.3

Wat gebeurt er bij een reactie?

3/22

Bij een chemische reactie verdwijnen stoffen, de **beginstoffen**, en ontstaan er nieuwe, de **reactieproducten**, met andere eigenschappen. Je herkent een reactie vaak aan een gas dat ontstaat, een kleur die verandert, een vaste stof die neerslaat, warmte of licht die vrijkomt. Op het niveau van de deeltjes gebeurt er telkens hetzelfde: **bindingen breken en nieuwe bindingen vormen zich**. De atomen zelf blijven bestaan; ze worden alleen anders gegroepeerd.

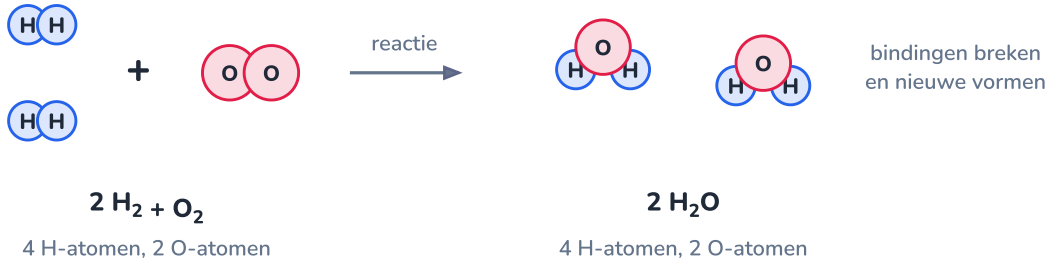
Waterstof en zuurstof reageren tot water. Als woordvergelijking: waterstof + zuurstof → water. Met formules, en met de aggregatietoestand erbij (s: vast, l: vloeibaar, g: gas, aq: opgelost in water):

WATERSTOF EN ZUURSTOF REAGEREN TOT WATER

$$2\,H_2\,(g) + O_2\,(g) \longrightarrow 2\,H_2O\,(l)$$

Links van de pijl de beginstoffen, rechts de reactieproducten. Het getal vóór een formule is de coëfficiënt: twee moleculen waterstof en één molecule zuurstof geven twee moleculen water.

Omdat er geen atomen verloren gaan, staan links en rechts van de pijl evenveel atomen van elke soort: 4 H en 2 O. Dat is de **wet van behoud van massa** van Lavoisier, gezien vanuit de atomen: elk atoom heeft zijn eigen massa, dus als alle atomen blijven, blijft ook de totale massa.



$2\ H_2 + O_2$

4 H-atomen, 2 O-atomen

$2\ H_2O$

4 H-atomen, 2 O-atomen

Twee waterstofmoleculen en een zuurstofmolecule worden twee watermoleculen: tel de bolletjes links en rechts.

UITLEG · 14.4

Reactievergelijkingen kloppend maken

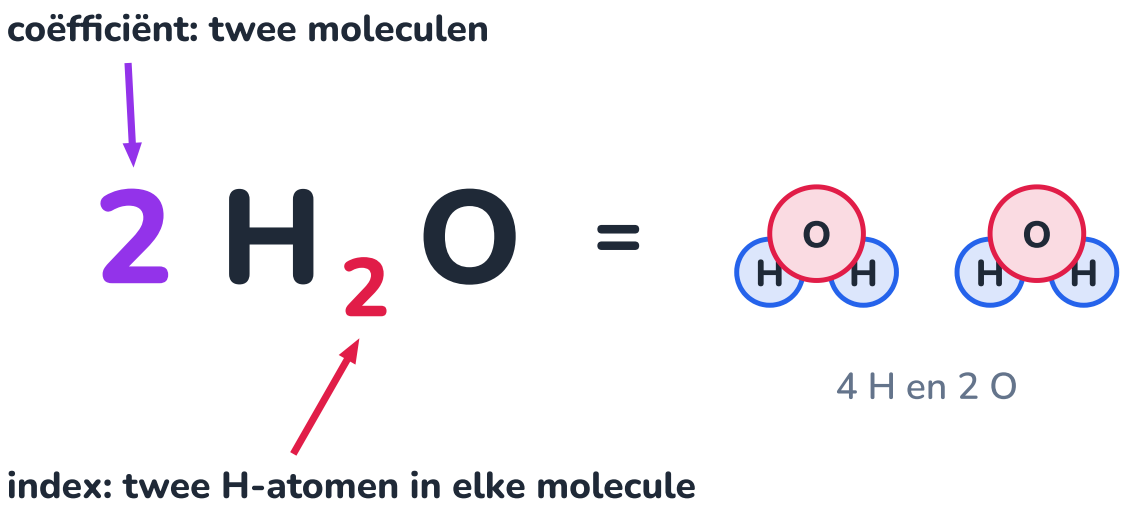
Een reactievergelijking **klopt** als er links en rechts evenveel atomen van elke soort staan. Je maakt ze kloppend door **coëfficiënten** (de getallen vóór de formules) te kiezen. De **indices** (de kleine getallen in de formules) blijf je van af: die bepalen welke stof het is.

Een werkwijze die bijna altijd lukt:

1. Schrijf de juiste formules van alle beginstoffen en producten.
2. Begin met het element dat in de ingewikkeldste formule zit, en maak het in orde met een coëfficiënt.
3. Doe daarna de andere elementen. Elementen die als enkelvoudige stof voorkomen (O_2, H_2, Fe), en meestal ook H en O, spaar je voor het laatst: die kan je nog aanpassen zonder de rest te storen.
4. Kom je op een breuk uit (bijvoorbeeld 7/2 O_2), vermenigvuldig dan alle coëfficiënten met de noemer.
5. Controleer: tel elk element links en rechts, en kijk of de coëfficiënten niet nog te vereenvoudigen zijn.

Een index veranderen om het kloppend te maken

Van H_2O maken H_2O_2, omdat je rechts een zuurstofatoom tekortkomt? Dan schrijf je een andere stof op: waterstofperoxide, een bleekmiddel. De formule ligt vast; alleen het aantal moleculen (de coëfficiënt) kan je kiezen.



De coëfficiënt telt moleculen, de index telt atomen in één molecule.

UITGEWERKT VOORBEELD · 14.5

Drie vergelijkingen kloppend maken

5/22

Maak kloppend: (a) de verbranding van methaan (aardgas), (b) het roesten van ijzer tot ijzer(III)oxide, (c) de verbranding van octaan C_8H_{18}, een bestanddeel van benzine.

GEGEVEN

- $CH_4 + O_2 \longrightarrow CO_2 + H_2O$
- $Fe + O_2 \longrightarrow Fe_2O_3$
- $C_8H_{18} + O_2 \longrightarrow CO_2 + H_2O$

GEVRAAGD

- de coëfficiënten

OPLOSSING

1. (a) C: 1 links, 1 rechts: in orde. H: 4 links, dus 2 H_2O rechts. O: rechts nu $2 + 2 = 4$, dus 2 O_2 links. $CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O$.
2. (b) O: 2 links, 3 rechts. Het kleinste gemene veelvoud is 6: 3 O_2 en 2 Fe_2O_3. Nu staan er rechts 4 Fe, dus 4 Fe links. $4\,Fe + 3\,O_2 \longrightarrow 2\,Fe_2O_3$.
3. (c) C: 8 CO_2. H: 18 H, dus 9 H_2O. O rechts: $8 \cdot 2 + 9 = 25$ atomen, dus $25/2$ O_2 links. Een breuk: alles maal 2. $2\,C_8H_{18} + 25\,O_2 \longrightarrow 16\,CO_2 + 18\,H_2O$.

ANTWOORD

$CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O$; $4\,Fe + 3\,O_2 \longrightarrow 2\,Fe_2O_3$; $2\,C_8H_{18} + 25\,O_2 \longrightarrow 16\,CO_2 + 18\,H_2O$.

Klopt dit? Tel bij (c): links 16 C, 36 H, 50 O; rechts 16 C, 36 H, 32 + 18 = 50 O. Klopt. Merk op hoeveel zuurstof een brandstof als octaan nodig heeft: 12,5 moleculen per molecule octaan.

PROEF · 14.6

Blijft de massa behouden als er een gas ontstaat?

6/22

ONDERZOEKSVRAAG

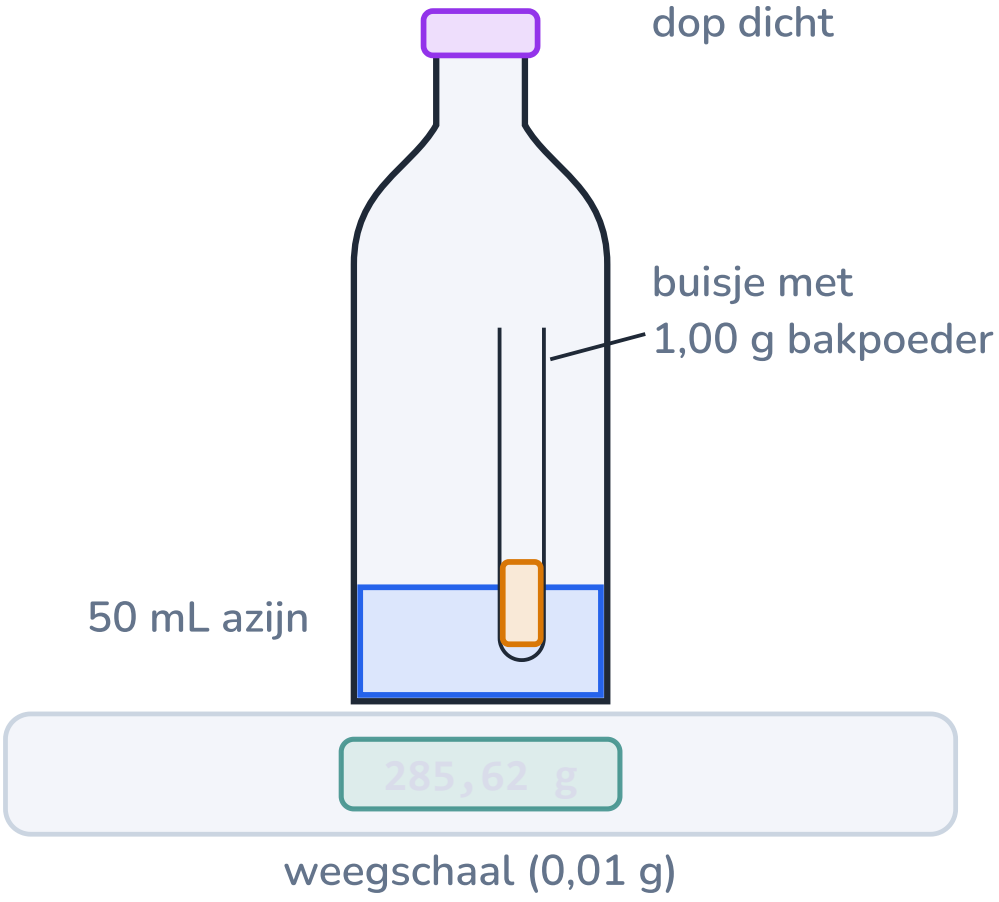
Bakpoeder en azijn bruisen: er ontstaat een gas. Wordt het geheel daardoor lichter? En wat als je de reactie in een gesloten fles laat verlopen?

MATERIAAL

- PET-fles van 1,5 L met dop
- 50 mL azijn (5 %)
- 1,00 g natriumwaterstofcarbonaat (bakpoeder) in een klein reageerbuisje
- weegschaal tot op 0,01 g
- veiligheidsbril

Veiligheid

Veiligheidsbril. Gebruik niet meer dan 1 g bakpoeder in een gesloten fles van 1,5 L: dan stijgt de druk maar met ongeveer 0,2 bar. Nooit in een glazen fles, nooit met meer poeder. Open de dop langzaam, met de opening van je weg.



Zolang de dop dicht is, weegt de weegschaal alle atomen: voor, tijdens en na de reactie.

WERKWIJZE

1. Giet de azijn in de fles. Zet het buisje met bakpoeder rechtop in de fles, zodat het nog niet in de azijn valt.
2. Draai de dop dicht. Weeg de fles met alles erin.
3. Kantel de fles zodat het bakpoeder in de azijn valt. Wacht tot het bruisen stopt en weeg opnieuw.
4. Draai de dop langzaam open. Weeg meteen, en nog eens na 10 minuten (af en toe zacht walsen).

WAARNEMING

moment	massa (g)
voor de reactie, dop dicht	285,62
na de reactie, dop dicht (fles voelt hard aan)	285,62
meteen na het openen	285,24
10 minuten na het openen	285,13

Voorbeeldmeting. In de gesloten fles verandert de massa niet, op de laatste decimaal na. Na het openen daalt ze met ongeveer een halve gram.

VERKLARING

De reactie is $\mathrm{NaHCO_3 + CH_3COOH \longrightarrow CH_3COONa + H_2O + CO_2}$. In de gesloten fles blijven alle atomen binnen: de massa blijft gelijk, ook al is er een gas ontstaan. Het gas duwt wel tegen de wand: de fles wordt hard. Open je de fles, dan ontsnapt de koolstofdioxide en daalt de massa. Uit 1,00 g bakpoeder ontstaat 0,52 g koolstofdioxide (dat reken je verderop in dit hoofdstuk zelf uit). Na tien minuten is er 0,49 g weg: een klein deel van het gas blijft opgelost in de vloeistof of hangt nog in de fles, want koolstofdioxide is zwaarder dan lucht.

De fles wordt hard, de weegschaal blijft staan.

UITLEG · 14.7

De mol: atomen tellen door te wegen

7/22

In de vergelijking $\mathrm{CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O}$ reageert één molecule methaan met twee moleculen zuurstof. Maar hoeveel gram zuurstof heb je nodig voor een gram methaan? Moleculen kan je niet tellen: een druppel water bevat er al ongeveer $2 \cdot 10^{21}$. Wel kan je wegen, zoals een bank munten niet telt maar weegt.

Chemici spraken daarom een 'pakje' af met een vast, gigantisch aantal deeltjes: de **mol**. Eén mol bevat precies

DE CONSTANTE VAN AVOGADRO

$$N_A = 6{,}022\,140\,76 \cdot 10^{23}\ \text{mol}^{-1}$$

Sinds 2019 is dit getal exact vastgelegd: het definieert de mol. Het is zo gekozen dat 1 mol koolstof-12-atomen (vrijwel precies) 12 g weegt.

Het slimme aan dat getal: de massa van één mol van een stof in gram is dezelfde als de massa van één deeltje in u (hoofdstuk 12). Een koolstofatoom weegt gemiddeld 12,011 u, dus één mol koolstof weegt 12,011 g. Een watermolecule weegt $2 \cdot 1{,}008 + 15{,}999 = 18{,}015\ \text{u}$, dus één mol water weegt 18,015 g, een slokje van 18 mL. Die massa per mol heet de **molaire massa** M, in g/mol.

HOEVEELHEID STOF

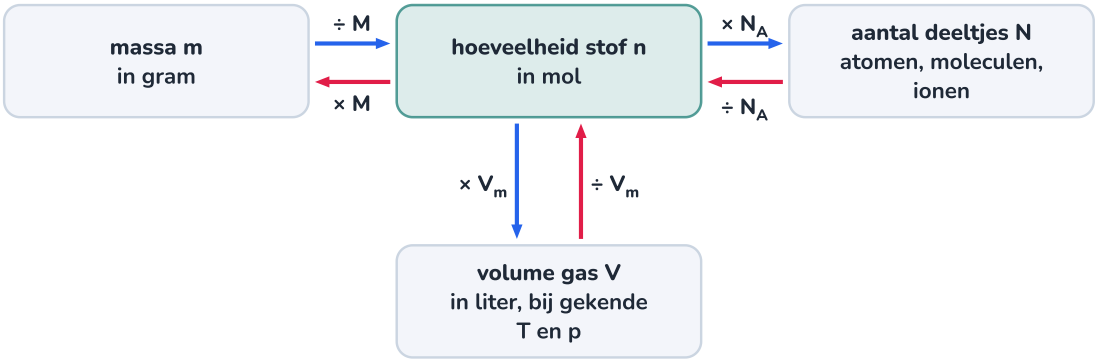
$$n = \frac{m}{M} \qquad N = n \cdot N_A$$

n de hoeveelheid stof in mol, m de massa in g, M de molaire massa in g/mol, N het aantal deeltjes.

Het schema hieronder vat alle omrekeningen van dit hoofdstuk samen. De mol staat in het midden: wie met reacties rekent, rekent altijd via de mol.

Een gevoel voor de mol

Een mol zandkorrels van 0,3 mm zou heel België bedekken onder een laag van ongeveer een halve kilometer. Een mol watermoleculen is maar 18 mL: een slokje. Zo klein zijn moleculen.



Gram ↔ mol met de molaire massa M, mol ↔ deeltjes met de constante van Avogadro, mol ↔ liter gas met het molair volume.

UITGEWERKT VOORBEELD · 14.8

Molaire massa's berekenen

8/22

Bereken de molaire massa van koolstofdioxide CO_2, glucose $C_6H_{12}O_6$ en calciumnitraat-tetrahydraat $Ca(NO_3)_2{\cdot}4H_2O$, het calciumzout van de voedingsoplossing.

GEGEVEN

- atoommassa's: H 1,008; C 12,011; N 14,007; O 15,999; Ca 40,078 (u, of g/mol)

GEVRAAGD

- M in g/mol

OPLOSSING

1. Tel de atomen in de formule en tel hun atoommassa's op. CO_2: $12{,}011 + 2 \cdot 15{,}999 = 44{,}009\ \text{g/mol}$.

2. $C_6H_{12}O_6$: $6 \cdot 12{,}011 + 12 \cdot 1{,}008 + 6 \cdot 15{,}999 = 72{,}066 + 12{,}096 + 95{,}994 = 180{,}156\ \text{g/mol}$.

3. Haakjes: $(NO_3)_2$ betekent 2 N en 6 O. Het kristalwater $4H_2O$ telt mee: 8 H en 4 O. Samen: 1 Ca, 2 N, 10 O, 8 H.

4. $40{,}078 + 2 \cdot 14{,}007 + 10 \cdot 15{,}999 + 8 \cdot 1{,}008 = 40{,}078 + 28{,}014 + 159{,}99 + 8{,}064 = 236{,}15\ \text{g/mol}$.

ANTWOORD

$M(CO_2) = 44{,}01\ \text{g/mol}$; $M(C_6H_{12}O_6) = 180{,}16\ \text{g/mol}$; $M(Ca(NO_3)_2 \cdot 4H_2O) = 236{,}15\ \text{g/mol}$.

Klopt dit? Zonder kristalwater zou calciumnitraat 164,09 g/mol wegen. Wie het kristalwater vergeet, rekent met een verkeerde molaire massa en doseert 30 % te weinig calcium. Op het etiket van een meststof staat daarom altijd welke vorm het is.

UITGEWERKT VOORBEELD · 14.9

Van gram naar mol naar deeltjes

9/22

(a) Hoeveel mol en hoeveel moleculen water zitten er in een glas van 250 g? (b) Een theelepel keukenzout weegt ongeveer 6,0 g. Hoeveel mol NaCl is dat, en hoeveel natriumionen?

GEGEVEN

- $m(H_2O) = 250\ \text{g}$, $M = 18{,}015\ \text{g/mol}$
- $m(NaCl) = 6{,}0\ \text{g}$, $M = 22{,}990 + 35{,}45 = 58{,}44\ \text{g/mol}$
- $N_A = 6{,}022 \cdot 10^{23}\ \text{mol}^{-1}$

GEVRAAGD

- n en N

OPLOSSING

1. (a) $n = \dfrac{m}{M} = \dfrac{250\ \text{g}}{18{,}015\ \text{g/mol}} = 13{,}88\ \text{mol}$.

2. $N = n \cdot N_A = 13{,}88 \cdot 6{,}022 \cdot 10^{23} = 8{,}36 \cdot 10^{24}$ moleculen.

3. (b) $n = \dfrac{6{,}0}{58{,}44} = 0{,}103\ \text{mol}$ NaCl. Elke 'formule-eenheid' NaCl bevat één Na^+, dus $0{,}103\ \text{mol}$ natriumionen: $0{,}103 \cdot 6{,}022 \cdot 10^{23} = 6{,}2 \cdot 10^{22}$ ionen, en evenveel chloride-ionen.

ANTWOORD

(a) 13,9 mol, $8{,}4 \cdot 10^{24}$ watermoleculen; (b) 0,103 mol NaCl, $6{,}2 \cdot 10^{22}$ natriumionen.

Klopt dit? Snelle schatting: 18 g water is 1 mol, dus 250 g is ongeveer 250/18 ≈ 14 mol. De eenheden kloppen: g gedeeld door g/mol geeft mol; mol maal 1/mol geeft een aantal.

CODE · 14.10

Python: een molaire-massarekenaar

10/22

Molaire massa's uitrekenen is telwerk, en telwerk laat je aan een computer over. Het moeilijkste is de formule lezen: een element is een hoofdletter met soms een kleine letter erachter (Ca, Cl), een getal erna is het aantal, en haakjes vermenigvuldigen alles ertussen. De functie `tel_atomen` leest de formule teken per teken en houdt voor elk niveau van haakjes een woordenboek bij, op een *stapel*. Kristalwater schrijf je met een sterretje: `MgSO4*7H2O`.

Python **molmassa.py** 53 regels

```python
# molmassa.py - molaire massa uit een formule, ook met haakjes en kristalwater
ATOOMMASSA = {"H": 1.008, "C": 12.011, "N": 14.007, "O": 15.999, "Na": 22.990, "Mg": 24.305,
              "P": 30.974, "S": 32.06, "Cl": 35.45, "K": 39.098, "Ca": 40.078, "Fe": 55.845, "Cu": 63.546}


def lees_getal(tekst, i):
    """Leest een getal vanaf positie i. Staat er geen getal, dan is het 1."""
    j = i
    while j < len(tekst) and tekst[j].isdigit():
        j += 1
    getal = int(tekst[i:j]) if j > i else 1
    return getal, j


def tel_atomen(formule):
    """Ca(NO3)2 -> {'Ca': 1, 'N': 2, 'O': 6}"""
    stapel = [{}]                           # een woordenboek per niveau van haakjes
    i = 0
    while i < len(formule):
        teken = formule[i]
        if teken == "(":
            stapel.append({})
            i += 1
        elif teken == ")":
            getal, i = lees_getal(formule, i + 1)
            groep = stapel.pop()
            for el, aantal in groep.items():
                stapel[-1][el] = stapel[-1].get(el, 0) + aantal * getal
        else:                               # element: een hoofdletter, soms gevolgd door een kleine letter
            el = teken
            i += 1
            if i < len(formule) and formule[i].islower():
                el += formule[i]
                i += 1
            getal, i = lees_getal(formule, i)
            stapel[-1][el] = stapel[-1].get(el, 0) + getal
    return stapel[0]


def molaire_massa(formule):
    """Molaire massa in g/mol. Kristalwater mag met een sterretje: MgSO4*7H2O."""
    totaal = 0.0
    for stuk in formule.replace("·", "*").split("*"):
        factor, i = lees_getal(stuk, 0)           # de 7 in 7H2O
        for el, aantal in tel_atomen(stuk[i:]).items():
            totaal += factor * aantal * ATOOMMASSA[el]
    return totaal


if __name__ == "__main__":
    for f in ["H2O", "CO2", "C6H12O6", "C8H18", "NaHCO3", "Ca(NO3)2", "Ca(NO3)2*4H2O",
              "MgSO4*7H2O", "KH2PO4", "(NH4)2SO4"]:
        print(f"{f:15s} {molaire_massa(f):8.2f} g/mol")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
H2O              18.02 g/mol
CO2              44.01 g/mol
C6H12O6         180.16 g/mol
C8H18           114.23 g/mol
NaHCO3           84.01 g/mol
Ca(NO3)2        164.09 g/mol
Ca(NO3)2*4H2O   236.15 g/mol
```

```
MgSO4*7H2O      246.47 g/mol
KH2PO4          136.08 g/mol
(NH4)2SO4       132.13 g/mol
```

Deze pagina rekent zelf ook met die functie: de tabel met molaire massa's in het overzicht, de CO_2 per kilogram brandstof en de grafieken van dit hoofdstuk komen er allemaal uit. In opdracht 14.10 breid je het programma uit met het massapercentage van elk element.

UITLEG · 14.11

Rekenen met reacties: de molverhouding

11/22

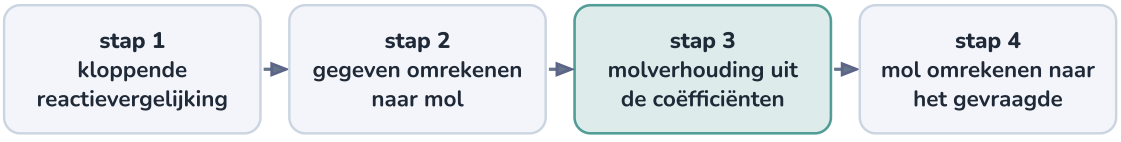
De coëfficiënten van een kloppende vergelijking zeggen in welke verhouding deeltjes reageren. Omdat een mol altijd hetzelfde aantal deeltjes bevat, is dat meteen ook de verhouding in mol: de **molverhouding**. In $\mathrm{CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O}$ reageert 1 mol methaan met 2 mol zuurstof tot 1 mol koolstofdioxide en 2 mol water. In gram is die verhouding niet 1 : 2 : 1 : 2, want elke stof heeft een andere molaire massa. Daarom reken je altijd via de mol, in vier stappen:

1. Schrijf de kloppende reactievergelijking.
2. Reken het gegeven om naar mol: $n = m/M$ (of voor een gas $n = V/V_m$).
3. Gebruik de molverhouding om het aantal mol van de gevraagde stof te vinden.
4. Reken dat om naar wat gevraagd is: $m = n \cdot M$ (of $V = n \cdot V_m$).

Een overzichtelijke manier om dat op te schrijven, is een **rekentabel** onder de vergelijking, met per stof de coëfficiënt, het aantal mol, de molaire massa en de massa (volgend blad). Dit rekenen met hoeveelheden bij reacties heet **stoichiometrie**, van het Griekse *stoicheion*, 'bestanddeel', en *metron*, 'maat'.

Als er van één beginstof te veel is

Beginstoffen zijn zelden precies in de juiste verhouding aanwezig. De stof die als eerste op is, heet de **beperkende stof**: zij bepaalt hoeveel product er ontstaat, de rest blijft over. Bij de fles met bakpoeder was er azijn genoeg voor 3,5 g bakpoeder: het bakpoeder (1,00 g) was beperkend. Reken dus eerst uit welke stof op raakt (opdracht 14.13).



Stap 3 is het hart van elke berekening: de molverhouding komt uit de coëfficiënten.

UITGEWERKT VOORBEELD · 14.12

Hoeveel zuurstof verbruikt een gasvlam?

12/22

Een gasbrander verbrandt 16,0 g methaan volledig. Hoeveel gram zuurstof verbruikt hij, en hoeveel gram koolstofdioxide en water ontstaat er?

GEGEVEN

- $m(\mathrm{CH_4}) = 16{,}0\,\mathrm{g}$

GEVRAAGD

- $m(\mathrm{O_2}), m(\mathrm{CO_2}), m(\mathrm{H_2O})$

- $\mathrm{CH_4 + 2\,O_2 \longrightarrow CO_2 + 2\,H_2O}$

OPLOSSING

1 Stap 1: de vergelijking is kloppend. Molverhouding 1 : 2 : 1 : 2.

2 Stap 2: $n(\mathrm{CH_4}) = 16{,}0/16{,}04 = 0{,}997\,\mathrm{mol}$.

3 Stap 3: $n(\mathrm{O_2}) = 2 \cdot 0{,}997 = 1{,}995\,\mathrm{mol}$; $n(\mathrm{CO_2}) = 0{,}997\,\mathrm{mol}$; $n(\mathrm{H_2O}) = 1{,}995\,\mathrm{mol}$.

4 Stap 4: $m(\mathrm{O_2}) = 1{,}995 \cdot 32{,}00 = 63{,}8\,\mathrm{g}$; $m(\mathrm{CO_2}) = 0{,}997 \cdot 44{,}01 = 43{,}9\,\mathrm{g}$; $m(\mathrm{H_2O}) = 1{,}995 \cdot 18{,}02 = 35{,}9\,\mathrm{g}$. In een rekentabel:

	$\mathrm{CH_4}$	$\mathrm{O_2}$	$\mathrm{CO_2}$	$\mathrm{H_2O}$
coëfficiënt	1	2	1	2
n (mol)	0,997	1,995	0,997	1,995
M (g/mol)	16,04	32,00	44,01	18,02
m (g)	16,0	63,8	43,9	35,9

ANTWOORD

De vlam verbruikt 63,8 g zuurstof en maakt 43,9 g koolstofdioxide en 35,9 g water.

Klopt dit? Massabehoud: links $16{,}0 + 63{,}8 = 79{,}8\,\mathrm{g}$, rechts $43{,}9 + 35{,}9 = 79{,}8\,\mathrm{g}$. Klopt. De vlam verbruikt vier keer zoveel zuurstof als brandstof: een gasvuur in een kleine, slecht verluchte ruimte maakt de lucht snel arm aan zuurstof.

UITLEG · 14.13

Gassen: het molair volume

13/22

Gassen hebben een eigenaardigheid die vaste stoffen en vloeistoffen niet hebben. In 1811 stelde Amedeo Avogadro dat **gelijke volumes van gassen**, bij dezelfde temperatuur en druk, **evenveel deeltjes** bevatten, welk gas het ook is. In een gas zitten de deeltjes zo ver uit elkaar (F6) dat hun eigen grootte er niet toe doet: alleen hun aantal telt. Eén mol van elk gas neemt dus hetzelfde volume in: het **molair volume** V_m.

MOLAIR VOLUME BIJ 1013 HPA

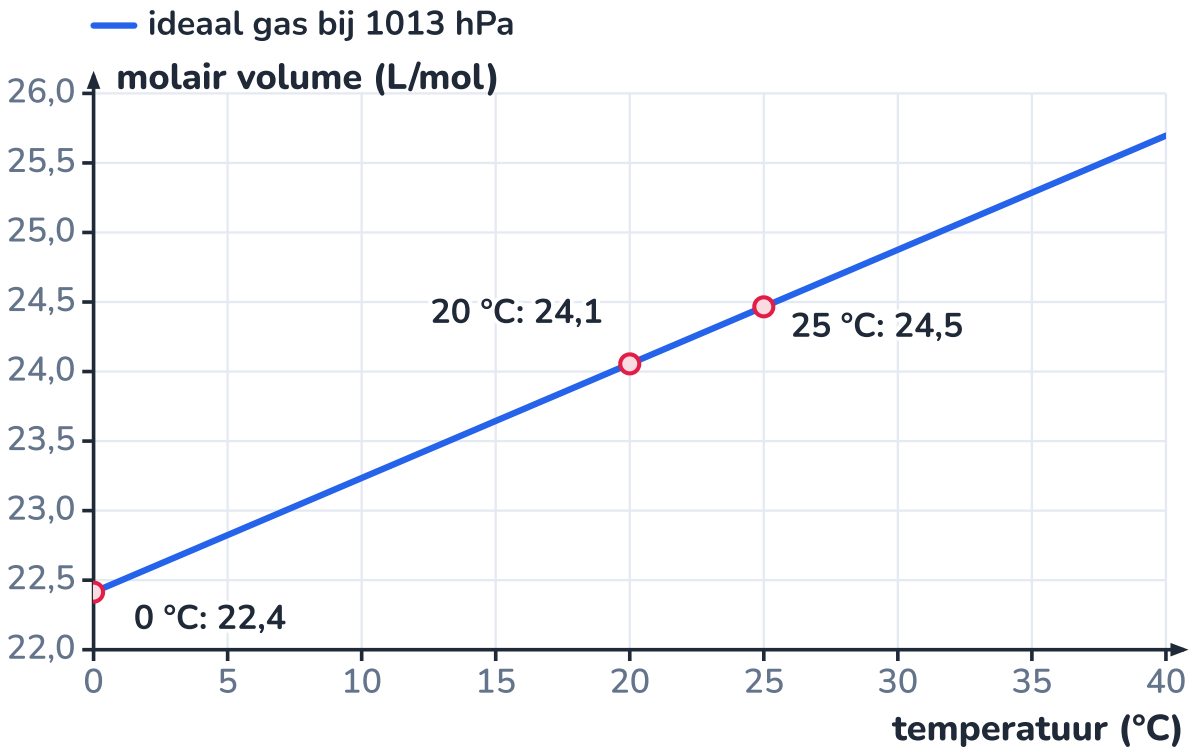
$$n = \frac{V}{V_m} \qquad V_m = 22{,}4\,\mathrm{L/mol}\ (0\,^\circ\mathrm{C}) \quad 24{,}1\,\mathrm{L/mol}\ (20\,^\circ\mathrm{C}) \quad 24{,}5\,\mathrm{L/mol}\ (25\,^\circ\mathrm{C})$$

Warmere gassen zetten uit: V_m is recht evenredig met de absolute temperatuur (in kelvin). Bij een lagere druk is V_m groter. Hoe je dat precies berekent, zie je in hoofdstuk E14 (de ideale gaswet).

In dit hoofdstuk rekenen we bij kamertemperatuur (20 °C) met **24,1 L/mol**, of afgerond 24 L/mol. Eén mol koolstofdioxide (44 g) en één mol waterstof (2 g) nemen allebei ongeveer 24 L in; daarom is koolstofdioxide 22 keer zo zwaar als waterstof, per liter.

Alleen voor gassen

Een mol vloeibaar water is 18 mL, een mol ijzer 7 mL. Het molair volume geldt alleen voor gassen, en je moet er altijd de temperatuur bij zeggen.



Hoe warmer, hoe meer ruimte een mol gas inneemt: van 22,4 L bij 0 °C tot 24,5 L bij 25 °C.

UITGEWERKT VOORBEELD · 14.14

Hoeveel gas uit een gram bakpoeder?

14/22

In de proef met de gesloten fles reageerde 1,00 g natriumwaterstofcarbonaat volledig met azijn. Hoeveel gram en hoeveel liter koolstofdioxide ontstond er bij 20 °C? Met hoeveel procent stijgt de druk in de fles, die 1,45 L lucht bevatte?

GEGEVEN

- $m(\mathrm{NaHCO_3}) = 1{,}00\,\mathrm{g}$, $M = 84{,}01\,\mathrm{g/mol}$
- $\mathrm{NaHCO_3 + CH_3COOH \longrightarrow CH_3COONa + H_2O + CO_2}$
- $V_m = 24{,}1\,\mathrm{L/mol}$ bij 20 °C

GEVRAAGD

- $m(\mathrm{CO_2})$ en $V(\mathrm{CO_2})$
- de drukstijging

OPLOSSING

1. $n(\mathrm{NaHCO_3}) = 1{,}00/84{,}01 = 0{,}0119\,\mathrm{mol}$. Molverhouding 1 : 1, dus $n(\mathrm{CO_2}) = 0{,}0119\,\mathrm{mol}$.
2. $m = 0{,}0119 \cdot 44{,}01 = 0{,}524\,\mathrm{g}$; $V = 0{,}0119 \cdot 24{,}1 = 0{,}286\,\mathrm{L}$ (bij de gewone luchtdruk).
3. In de gesloten fles kan het gas niet uitzetten. In dezelfde ruimte zitten nu $0{,}286/1{,}45 = 20\,\%$ meer deeltjes, die dus 20 % vaker tegen de wand botsen (F6): de druk stijgt met ongeveer 20 %, van 1,0 tot 1,2 bar.

ANTWOORD

0,524 g koolstofdioxide, 0,29 L bij 20 °C; in de gesloten fles stijgt de druk met ongeveer 0,2 bar.

Klopt dit? De gemeten massadaling na tien minuten (0,49 g) ligt net onder de berekende 0,52 g: niet al het gas was al ontsnapt. Met 5 g bakpoeder (en azijn genoeg) zou de druk op ongeveer 2 bar komen: daarom de veiligheidsregel bij de proef.

Verbranding

Een **verbranding** is een reactie van een stof met zuurstof waarbij veel energie vrijkomt, als warmte en licht. Brandstoffen als aardgas, benzine, diesel, hout en kaarsvet bestaan vooral uit koolstof en waterstof. Bij een **volledige verbranding**, met zuurstof genoeg, wordt alle koolstof koolstofdioxide en alle waterstof water:

VOLLEDIGE VERBRANDING VAN PROPAAN (CAMPINGGAS)

$$C_3H_8 + 5\,O_2 \longrightarrow 3\,CO_2 + 4\,H_2O$$

Is er te weinig zuurstof, dan verloopt de verbranding **onvolledig**. Dan ontstaat ook **koolstofmonoxide** CO, en zelfs zuivere koolstof: roet. Een gasvlam met genoeg lucht brandt blauw; een vlam met te weinig lucht wordt geel en walmt.

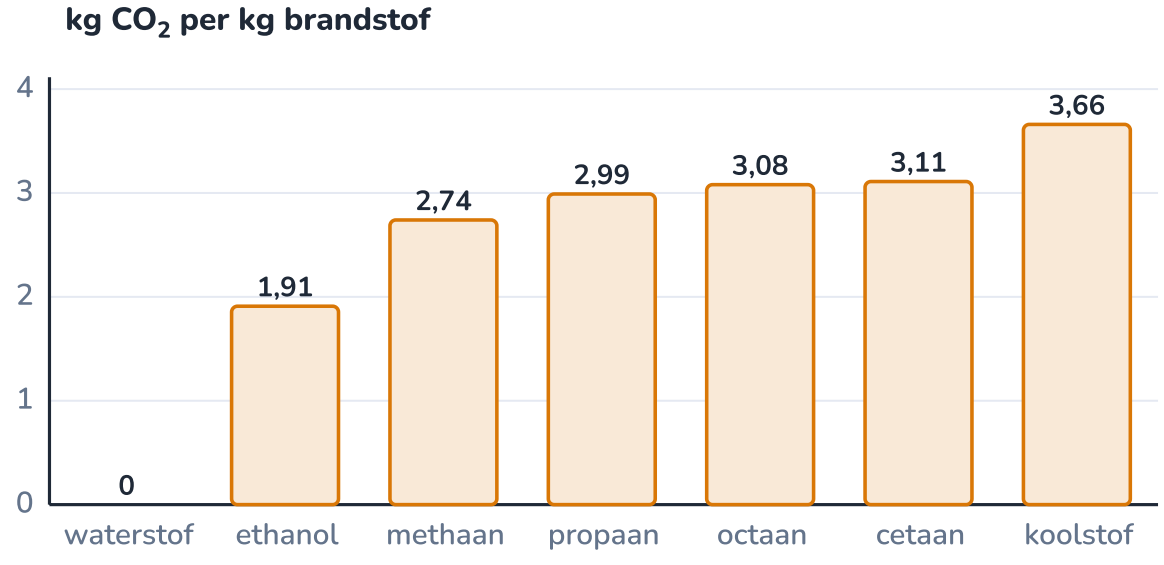
ONVOLLEDIGE VERBRANDING VAN METHAAN

$$2\,CH_4 + 3\,O_2 \longrightarrow 2\,CO + 4\,H_2O$$

Koolstofmonoxide: geurloos en dodelijk

CO bindt zich ongeveer tweehonderd keer sterker aan het hemoglobine in je bloed dan zuurstof, en verdringt zo de zuurstof. Je ruikt het niet en je ziet het niet. Een slecht afgestelde geiser of kachel in een slecht verluchte ruimte kan dodelijk zijn: in België komen er elk jaar mensen om. Daarom hangt er naast elk verbrandingstoestel een CO-melder, en brandt in Habitat niets in een afgesloten ruimte.

Waarom meer CO_2 dan brandstof? Elk koolstofatoom (12 u) haalt twee zuurstofatomen (samen 32 u) uit de lucht en wordt een molecule CO_2 van 44 u: 3,67 keer zo zwaar. Een brandstof die voor 84 % uit koolstof bestaat, zoals octaan, levert zo meer dan drie kilogram CO_2 per kilogram. Waterstof bevat geen koolstof: bij zijn verbranding ontstaat alleen water. Ethanol bevat al zuurstof en levert minder CO_2 per kilogram, maar ook minder energie.

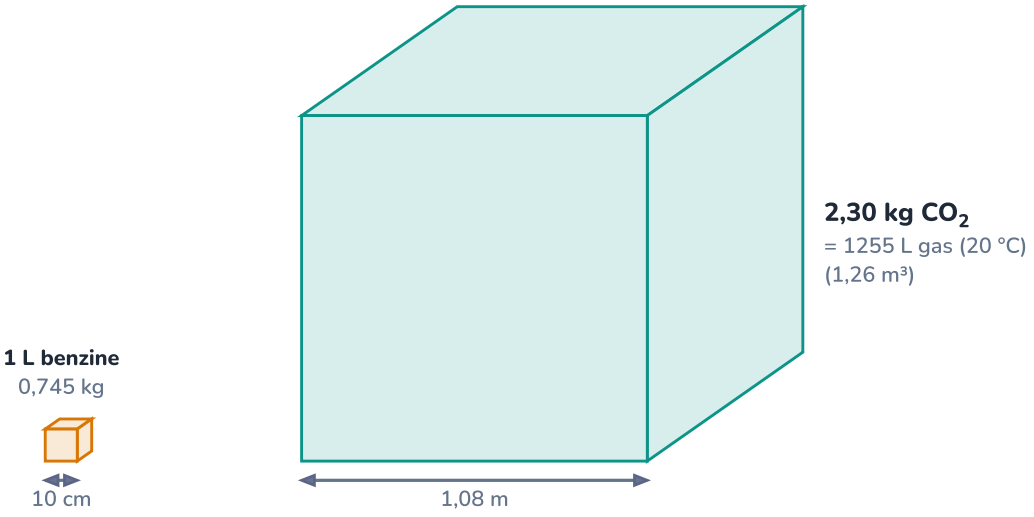


Berekend uit de formules met de molaire-massarekenaar van dit hoofdstuk, bij volledige verbranding.

UITGEWERKT VOORBEELD · 14.16

Hoeveel CO_2 uit een liter benzine en een liter diesel?

Een auto verbruikt benzine. Hoeveel kilogram koolstofdioxide komt er vrij als hij één liter verbrandt? Benzine is een mengsel van vele koolwaterstoffen; reken met octaan C_8H_{18} als model, met de dichtheid van benzine: 0,745 kg/L. Doe hetzelfde voor diesel, met cetaan $C_{16}H_{34}$ als model en 0,835 kg/L. Vergelijk met de officiële omrekenfactoren, ongeveer 2,3 kg/L voor benzine en 2,6 kg/L voor diesel.



Op schaal: de liter benzine (kubus van 10 cm) wordt een kubus CO_2 van meer dan een meter.

GEGEVEN

- benzine: $\rho = 0{,}745\,\text{kg/L}$, model C_8H_{18}, $M = 114{,}23\,\text{g/mol}$
- diesel: $\rho = 0{,}835\,\text{kg/L}$, model $C_{16}H_{34}$, $M = 226{,}45\,\text{g/mol}$
- $M(CO_2) = 44{,}01\,\text{g/mol}$

GEVRAAGD

- $m(CO_2)$ per liter

OPLOSSING

1. Stap 1: $2\,C_8H_{18} + 25\,O_2 \longrightarrow 16\,CO_2 + 18\,H_2O$. Molverhouding octaan : CO_2 = 2 : 16 = 1 : 8.
2. Stap 2: een liter benzine weegt 745 g: $n(C_8H_{18}) = 745/114{,}23 = 6{,}52\,\text{mol}$.
3. Stap 3: $n(CO_2) = 8 \cdot 6{,}52 = 52{,}2\,\text{mol}$.
4. Stap 4: $m(CO_2) = 52{,}2 \cdot 44{,}01 = 2296\,\text{g} = 2{,}30\,\text{kg}$. Als gas bij 20 °C: $52{,}2 \cdot 24{,}1 = 1255\,\text{L}$.
5. Diesel: $2\,C_{16}H_{34} + 49\,O_2 \longrightarrow 32\,CO_2 + 34\,H_2O$, verhouding 1 : 16. $n = 835/226{,}45 = 3{,}69\,\text{mol}$; $n(CO_2) = 59{,}0\,\text{mol}$; $m = 59{,}0 \cdot 44{,}01 = 2597\,\text{g} = 2{,}60\,\text{kg}$.

ANTWOORD

Eén liter benzine levert ongeveer 2,30 kg CO_2 (1,26 m³ gas), één liter diesel ongeveer 2,60 kg. Dat klopt met de officiële waarden van 2,3 en 2,6 kg per liter.

Klopt dit? Een liter diesel geeft meer CO_2 dan een liter benzine, vooral omdat diesel dichter is: per kilogram brandstof zijn ze bijna gelijk (3,08 en 3,11 kg). Echte diesel bevat ook moleculen met relatief meer koolstof (ringen): met de gemiddelde formule $C_{12}H_{23}$ vind je 2,64 kg/L. Om één liter benzine te verbranden, is er 2,61 kg zuurstof nodig, uit 9,4 m³ lucht. Een auto die 5,5 L per 100 km verbruikt, stoot $5{,}5 \cdot 2{,}30 = 12{,}6\,\text{kg}$ per 100 km uit: 126 g per kilometer.

UITLEG · 14.17

Exotherm en endotherm

17/22

Bij elke reactie verandert er ook energie van vorm. Om een binding te breken, moet je energie toevoegen; als er een binding ontstaat, komt er energie vrij. Wat overblijft, bepaalt of de reactie warmte afgeeft of opneemt.

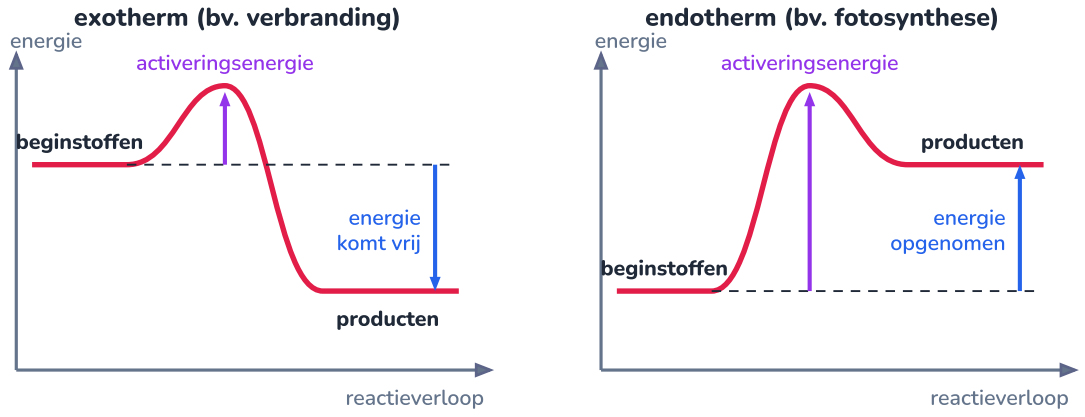
- Bij een **exotherme** reactie komt er meer energie vrij bij het vormen van de nieuwe bindingen dan het breken van de oude kostte. De omgeving wordt warmer. Voorbeelden: elke verbranding, roesten (handwarmers met ijzerpoeder), de celademhaling in je lichaam. Eén mol methaan (16 g) geeft bij verbranding 890 kJ, genoeg om 2,7 L water van 20 °C aan de kook te brengen.
- Bij een **endotherme** reactie is het omgekeerd: ze neemt energie op uit de omgeving, als warmte of als licht. Voorbeelden: **fotosynthese** (de energie komt van het licht), het ontbinden van kalk in een kalkoven, en de reactie van citroenzuur met bakpoeder, die merkbaar afkoelt.

FOTOSYNTHESE: ENDOTHERM

$$6\,CO_2 + 6\,H_2O \longrightarrow C_6H_{12}O_6 + 6\,O_2 \qquad +\,2803\,\text{kJ per mol glucose}$$

Dezelfde hoeveelheid energie komt vrij als de plant (of wie haar opeet) de glucose weer verbrandt, bij de celademhaling.

Waarom brandt papier niet vanzelf, als verbranding toch energie oplevert? Omdat eerst enkele bindingen moeten breken voor de nieuwe zich kunnen vormen. Die drempel heet de **activeringsenergie**. Een lucifer levert ze; daarna houdt de vrijgekomen warmte de reactie zelf op gang. Hoe temperatuur en katalysatoren die drempel beïnvloeden, zie je in hoofdstuk 34 (enzymen) en E25.



Links liggen de producten lager dan de beginstoffen: energie komt vrij. Rechts hoger: energie wordt opgenomen. Beide moeten eerst over de heuvel.

PROEF · 14.18

Warm of koud?

18/22

ONDERZOEKSVRAAG

Welke van twee reacties geeft warmte af aan het water rondom, en welke neemt er warmte uit op?

MATERIAAL

- twee plastic bekers in een isolerende houder (piepschuim)
- thermometer of temperatuursensor met datalogger
- A: 50 mL water, 10 g citroenzuur, 10 g natriumwaterstofcarbonaat (bakpoeder)
- B: 50 mL waterstofperoxide 3 % (apotheek), een koffielepel droge gist
- roerstaafje, veiligheidsbril

Veiligheid

Veiligheidsbril. Citroenzuur irriteert de ogen (GHS07). Waterstofperoxide niet in de ogen of op de huid; beker B schuimt sterk: gebruik een grote beker en zet hem in een schaal.

GHS07

citroenzuur:

ernstige oogirritatie

Citroenzuur draagt het uitroepteken: bril op.

WERKWIJZE

1. Meet de begintemperatuur van het water in beker A. Los het citroenzuur op, voeg het bakpoeder toe en roer. Noteer de laagste temperatuur.
2. Meet de begintemperatuur van het waterstofperoxide in beker B. Voeg de gist toe en roer. Noteer de hoogste temperatuur.
3. Houd in beker B een gloeiende houtspaander net boven het schuim.

WAARNEMING

	begin (°C)	uiterste (°C)	verschil	verder
A: citroenzuur + bakpoeder	20,5	9,8	−10,7 °C	bruist
B: waterstofperoxide + gist	20,4	31,6	+11,2 °C	schuimt, de spaander vlamt op

Voorbeeldmeting.

VERKLARING

Reactie A neemt energie op uit het water: ze is **endotherm**, en het water koelt af. De bubbels zijn koolstofdioxide, zoals bij de fles met azijn. In B ontbindt waterstofperoxide: $2\,H_2O_2 \longrightarrow 2\,H_2O + O_2$. Die reactie is **exotherm**. Zonder gist verloopt ze heel traag; de gist bevat een enzym, katalase, dat de reactie versnelt zonder zelf verbruikt te worden (hoofdstuk 34). Dat het gas zuurstof is, bewijst de gloeiende spaander die weer opvlamt.

Links wordt het koud, rechts warm, en de zuurstof uit het schuim laat de spaander opvlammen.

UITGEWERKT VOORBEELD · 14.19

Habitat: hoeveel CO 2 eet een slaplant?

19/22

In de kweekruimte van de klimaatmodule groeien 48 kroppen sla in 35 dagen tot een krop van ongeveer 200 g. Sla is voor 95 % water; met de wortels erbij houdt een plant ongeveer 12 g droge stof over. Neem als model voor die droge stof $(CH_2O)n$: de bouwsteen van suiker, zetmeel en cellulose. Hoeveel koolstofdioxide heeft één krop opgenomen, hoeveel zuurstof gaf ze af, en hoeveel CO_2 nemen de 48 kroppen samen op per uur licht (16 uur per dag)?

GEGEVEN

- droge stof 12 g per krop, model CH_2O, $M = 30{,}03\,\text{g/mol}$
- fotosynthese per bouwsteen: $CO_2 + H_2O \longrightarrow CH_2O + O_2$
- 48 kroppen, 35 dagen, 16 uur licht per dag; $V_m = 24{,}1\,\text{L/mol}$

GEVRAAGD

- $m(CO_2)$ en $m(O_2)$ per krop
- CO_2 per uur voor de hele teelt

OPLOSSING

1. $n(CH_2O) = 12/30{,}03 = 0{,}400\,\text{mol}$. Molverhouding 1 : 1, dus 0,400 mol CO_2 opgenomen en 0,400 mol O_2 afgegeven.
2. $m(CO_2) = 0{,}400 \cdot 44{,}01 = 17{,}6\,\text{g}$, of $0{,}400 \cdot 24{,}1 = 9{,}6\,\text{L}$ gas. $m(O_2) = 0{,}400 \cdot 32{,}00 = 12{,}8\,\text{g}$.
3. Voor 48 kroppen: $48 \cdot 17{,}6 = 844\,\text{g}$ CO_2 per teelt, gemiddeld $844/35 = 24{,}1\,\text{g}$ per dag.
4. De planten nemen alleen CO_2 op als het licht brandt: $24{,}1/16 = 1{,}51\,\text{g}$ per uur, of $1{,}51/44{,}01 \cdot 24{,}1 = 0{,}82\,\text{L}$ CO_2 per uur.

ANTWOORD

Eén krop sla neemt ongeveer 17,6 g CO_2 op (9,6 L) en geeft 12,8 g zuurstof af. De hele teelt vraagt gemiddeld 1,5 g CO_2 per lichtuur.

Klopt dit? Koolstofcontrole: 17,6 g CO_2 bevat $17{,}6 \cdot 12{,}01/44{,}01 = 4{,}8\,\text{g}$ koolstof, en droge plantenstof bestaat inderdaad voor ongeveer 40 % uit koolstof ($0{,}40 \cdot 12 = 4{,}8\,\text{g}$). Het is een gemiddelde: jonge plantjes nemen weinig op, grote kroppen in de laatste week veel meer.

UITLEG · 14.20

Habitat: waar haalt de kweekruimte haar CO_2?

De kweekruimte van de klimaatmodule is binnen 2,40 m × 2,00 m × 2,40 m: 11,5 m³ lucht. Buitenlucht bevat vandaag ongeveer 425 ppm koolstofdioxide (0,0425 %). In de kweekruimte zit dus $11{,}5 \cdot 1000 \cdot 425 \cdot 10^{-6} = 4{,}9\,\text{L}$ CO_2, of 9,0 g. De planten nemen 0,82 L per uur op: de concentratie daalt met ongeveer 72 ppm per uur. Zonder verse lucht zakt ze in drie uur tot 200 ppm, en dan valt de groei sterk terug. Een gesloten kweekruimte houdt het dus maar een paar uur vol.

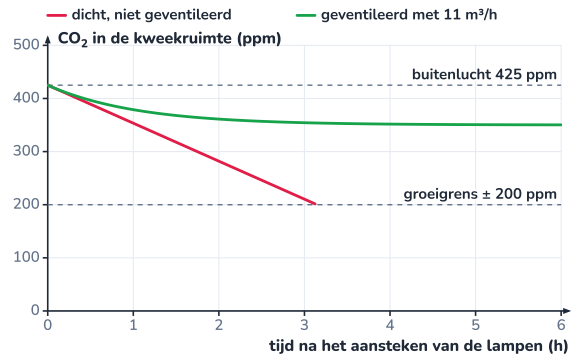
Ventilatie. Wordt er 11 m³ buitenlucht per uur doorgeblazen (ongeveer één keer de ruimte per uur), dan brengt elke kubieke meter die binnenkomt met 425 ppm en buitengaat met 350 ppm $1000 \cdot 75 \cdot 10^{-6} = 0{,}075\,\text{L}$ CO_2 mee: samen 0,82 L per uur, precies wat de planten opnemen. De concentratie zakt dan niet verder dan 350 ppm (grafiek).

Andere bronnen.

- **De bewoners.** Een volwassene ademt ongeveer 1 kg CO_2 per dag uit, 42 g per uur: veertig keer wat de hele teelt opneemt. De lucht uit de leefruimte door de kweekruimte leiden, is dus een goed idee. Maar in een afgesloten kweekruimte van 11,5 m³ laat één persoon de concentratie met bijna 2000 ppm per uur stijgen; boven ongeveer 1000 ppm voelt lucht bedompt en daalt de concentratie van wie er werkt.
- **Compost en gisting.** Bij het afbreken van plantenresten (hoofdstuk 35) en bij gisting van suiker ontstaat CO_2 (opdracht 14.14).
- **Verbranding.** Glastuinbouwers verbranden aardgas in een warmtekrachtkoppeling en blazen de gezuiverde rookgassen in hun serre; in Nederland brengt een pijpleiding CO_2 van een raffinaderij en een bio-ethanolfabriek naar honderden serres. In Habitat verbrandt niets in een gesloten ruimte (koolstofmonoxide).
- **Een fles CO_2.** Tuinders verrijken tot 800 à 1000 ppm, omdat planten dan sneller groeien. De kweekruimte eenmaal van 425 naar 800 ppm brengen kost 4,3 L (7,9 g); een fles van 2 kg volstaat bij 24 g per dag voor ongeveer 80 dagen. Maar met ventilatie waait die extra CO_2 weer naar buiten.

Zuurstof: de rekening klopt niet

De 48 kroppen geven samen 614 g zuurstof af per teelt, 17,5 g per dag. Een mens ademt ongeveer 0,84 kg zuurstof per dag in: de hele teelt levert dus maar 2 % van wat één bewoner nodig heeft. Een station dat zijn lucht met planten wil vernieuwen, heeft tientallen vierkante meter gewassen per bewoner nodig, of een algenreactor (hoofdstuk E31).



Model: 48 planten nemen 1,5 g CO_2 per uur op. Dicht: lineair naar 200 ppm in drie uur. Geventileerd: exponentieel naar 350 ppm.

Een CO_2-sensor en een ventilator: meer is er niet nodig om de planten genoeg CO_2 te geven.

NASLAG · 14.21

Overzicht: reacties en stoichiometrie

21/22

massa en hoeveelheid stof

$$n = \frac{m}{M}$$

symbool	betekenis	eenheid
n	hoeveelheid stof	mol
m	massa	g
M	molaire massa	g/mol

aantal deeltjes

$N = n \cdot N_A \qquad N_A = 6{,}022 \cdot 10^{23}\ \text{mol}^{-1}$

symbool	betekenis	eenheid
N	aantal deeltjes	–
N_A	constante van Avogadro	mol^{-1}

gassen

$V = n \cdot V_m \qquad V_m = 24{,}1\ \text{L/mol}\ (20\,^\circ\text{C},\ 101$

symbool	betekenis	eenheid
V	volume van het gas	L
V_m	molair volume	L/mol

molverhouding bij a A + b B → …

$$\frac{n_A}{a} = \frac{n_B}{b}$$

symbool	betekenis	eenheid
a, b	coëfficiënten in de kloppende vergelijking	–

stof	formule	M (g/mol)
water	H_2O	18,02
zuurstof	O_2	32,00
koolstofdioxide	CO_2	44,01
methaan	CH_4	16,04
propaan	C_3H_8	44,10
octaan	C_8H_{18}	114,23
glucose	$C_6H_{12}O_6$	180,16
ethanol	C_2H_5OH	46,07
natriumchloride	NaCl	58,44
natriumwaterstofcarbonaat	$NaHCO_3$	84,01

stof	formule	M (g/mol)
calciumcarbonaat	$CaCO_3$	100,09
azijnzuur	CH_3COOH	60,05
kaliumnitraat	KNO_3	101,10
calciumnitraat-tetrahydraat	$Ca(NO_3)_2{\cdot}4H_2O$	236,15
magnesiumsulfaat-heptahydraat	$MgSO_4{\cdot}7H_2O$	246,47
kaliumdiwaterstoffosfaat	KH_2PO_4	136,08

brandstof (model)	dichtheid	CO_2 per kg	CO_2 per liter
benzine (octaan C_8H_{18})	0,745 kg/L	3,08 kg	2,30 kg (1,26 m³)
diesel (cetaan $C_{16}H_{34}$)	0,835 kg/L	3,11 kg	2,60 kg
aardgas (methaan CH_4)	0,67 kg/m³ (20 °C)	2,74 kg	1,83 kg per m³

Habitat: rekenwaarden uit dit hoofdstuk	
kweekruimte	2,40 × 2,00 × 2,40 m binnen: 11,5 m³ lucht, 9,0 g CO_2 bij 425 ppm
slaplant	200 g vers, 12 g droge stof; neemt 17,6 g CO_2 op, geeft 12,8 g O_2 af
teelt van 48 kroppen (35 dagen)	844 g CO_2: 24 g per dag, 1,5 g (0,82 L) per lichtuur
zonder ventilatie	−72 ppm per uur: 200 ppm na ongeveer 3 uur
ventilatie	11 m³/h (ongeveer één keer per uur) houdt 350 ppm
een bewoner	ademt ± 1 kg CO_2 per dag uit, ademt ± 0,84 kg O_2 in

HET GEHEEL · 14.22

Waar reacties en de mol terugkomen

22/22

De mol is het gereedschap waarmee je in alle verdere chemie rekent: concentraties in mol per liter (zuren en basen), lading per mol (batterijen), energie per mol (thermochemie). Verbranding en fotosynthese verbinden de chemie met energie, klimaat en leven: de koolstof die een plant uit de lucht haalt, gaat terug de lucht in als iemand haar opeet of verbrandt.

F8 Zuivere stoffen en mengsels
Mengsel of reactie; massabehoud bij oplossen.

F17 Energie uit zon, wind en water
Fossiele brandstoffen en CO_2: hier in kilogram per liter.

F27 Planten: fotosynthese en groei
De woordvergelijking van fotosynthese wordt een kloppende vergelijking met getallen.

I12 Atomen en het periodiek systeem
Atoommassa's in u: de basis van de molaire massa.

I13 Chemische bindingen en stofeigenschappen
Formules van zouten en moleculen; bindingen die breken en ontstaan.

I15 Zuren, basen en pH
Concentraties in mol per liter, neutralisatie en titratie.

I16 Redox en batterijen
Verbranding en roesten als elektronenoverdracht; lading per mol.

I18 Warmte berekenen
De warmte van een reactie opvangen en meten (calorimetrie).

I34 Enzymen en stofwisseling
Fotosynthese en celademhaling als reactievergelijking; enzymen als katalysator.

I38 Duurzaam ontwerpen

CO_2-equivalenten van een product berekenen.

E14 Gassen en de gaswetten

De ideale gaswet: het molair volume bij elke temperatuur en druk.

E25 Thermochemie en reactiesnelheid

Reactie-enthalpie, activeringsenergie en reactiesnelheid, kwantitatief.

E31 Bioprocestechniek

Een algenreactor die CO_2 omzet in zuurstof, met massabalansen.

Habitat

Voor Habitat leggen we vast: de kweekruimte van de klimaatmodule is binnen 2,40 × 2,00 × 2,40 m (11,5 m³). Een teelt van 48 kroppen sla neemt 844 g CO_2 op in 35 dagen, gemiddeld 1,5 g per lichtuur. Zonder ventilatie zakt de CO_2 in drie uur van 425 naar 200 ppm; een ventilatie van 11 m³/h houdt 350 ppm. De uitgeademde lucht van de bewoners (± 1 kg CO_2 per persoon per dag) is een bron om te benutten. In opdracht 14.12 ontwerp je de CO_2-voorziening; in hoofdstuk 19 blijkt wat die ventilatie in de winter aan warmte kost.

SAMENVATTING

In het kort

- Bij een chemische reactie breken bindingen en vormen zich nieuwe; de atomen blijven. Daarom klopt een reactievergelijking: links en rechts evenveel atomen van elke soort.
- Je maakt een vergelijking kloppend met coëfficiënten, nooit door indices te veranderen.
- Wet van behoud van massa (Lavoisier): in een gesloten systeem blijft de massa gelijk, ook als er een gas ontstaat.
- Een mol is $6{,}022 \cdot 10^{23}$ deeltjes. De molaire massa M (g/mol) is de som van de atoommassa's in de formule. $n = m/M$.
- Rekenen met een reactie: kloppende vergelijking, gegeven naar mol, molverhouding, mol naar het gevraagde. De beperkende stof bepaalt hoeveel product er ontstaat.
- Eén mol van elk gas neemt bij 1013 hPa 22,4 L in bij 0 °C en 24,1 L bij 20 °C: $V = n \cdot V_m$.
- Volledige verbranding van koolwaterstoffen geeft CO_2 en water; onvolledige ook het giftige CO. Een liter benzine levert ongeveer 2,3 kg CO_2, een liter diesel 2,6 kg.
- Exotherme reacties geven energie af (verbranding), endotherme nemen energie op (fotosynthese). Elke reactie moet eerst over een drempel: de activeringsenergie.
- De 48 slaplanten van de kweekruimte nemen 1,5 g CO_2 per lichtuur op; zonder ventilatie is de CO_2 in 3 uur grotendeels op.

BEGRIPPEN

Wat je nu kent

chemische reactie

Een proces waarbij stoffen verdwijnen en nieuwe stoffen ontstaan: de atomen worden anders gebonden.

reactievergelijking

De reactie in formules, met evenveel atomen van elke soort links en rechts van de pijl.

coëfficiënt

wet van behoud van massa

Het getal vóór een formule in een reactievergelijking: hoeveel deeltjes (of mol) van die stof.

Bij een reactie in een gesloten systeem is de massa van de producten gelijk aan die van de beginstoffen.

mol

Hoeveelheid stof met precies $6{,}022\,140\,76 \cdot 10^{23}$ deeltjes (de constante van Avogadro N_A).

molaire massa

De massa van één mol van een stof, in g/mol; getalsmatig gelijk aan de molecuul- of formulemassa in u.

molverhouding

De verhouding waarin stoffen reageren en ontstaan, af te lezen uit de coëfficiënten.

beperkende stof

De beginstof die als eerste op is en daardoor bepaalt hoeveel product er ontstaat.

molair volume

Het volume van één mol gas: bij 1013 hPa 22,4 L bij 0 °C en 24,1 L bij 20 °C.

verbranding

Reactie van een stof met zuurstof waarbij energie vrijkomt; van koolwaterstoffen ontstaan CO_2 en water.

exotherme reactie

Reactie waarbij energie vrijkomt naar de omgeving, meestal als warmte.

endotherme reactie

Reactie die energie opneemt uit de omgeving (warmte, licht).

activeringsenergie

De energie die nodig is om een reactie op gang te brengen.

ppm

Parts per million: deeltjes per miljoen. 425 ppm CO_2 betekent 425 CO_2-moleculen per miljoen deeltjes lucht (0,0425 %).

HANDBOEK · HOOFDSTUK 15

Zuren, basen en pH

Van citroensap tot ontstopper: een logaritmische schaal, een kleurstof uit rode kool en een druppel die alles verandert

Citroensap en ontstopper hebben iets gemeen: allebei tasten ze je huid aan, het ene als zuur, het andere als base. Hoe zuur of hoe basisch een oplossing is, druk je uit met één getal: de **pH**. Je leert wat een stof zuur of basisch maakt, waarom de pH-schaal met machten van tien werkt, hoe een zuur en een base elkaar opheffen, en hoe je met een buret tot op een druppel meet hoeveel zuur er in azijn zit. Voor Habitat bewaak je de pH van de voedingsoplossing: met een geijkte pH-meter, en met een proefje op één liter voor je bijstuurt.

9× uitleg · 6× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 2× verhaal · 1× het geheel · 2× code · ± 10 lestijden

M Materie en chemie · B Leven en systemen · O Onderzoeken en meten

Eén kleurstof uit rode kool, acht kleuren: van zuur (rood) tot sterk basisch (geel).

Na dit hoofdstuk

- Je kan zuren en basen uit het dagelijks leven herkennen, hun eigenschappen noemen en er veilig mee werken.
- Je kan met de theorie van Arrhenius (en kort Brønsted) uitleggen wat een stof zuur of basisch maakt, en het verschil tussen sterk en geconcentreerd.
- Je kan de pH berekenen uit de concentratie van H_3O^+ of OH^- (sterke zuren en basen) en omgekeerd, en uitleggen waarom de schaal logaritmisch is.
- Je kan voorspellen wat verdunnen met de pH doet, en waarom een zuur door verdunnen nooit basisch wordt.
- Je kan een indicator kiezen bij een omslaggebied en de kleur voorspellen.
- Je kan een neutralisatie als reactievergelijking schrijven en uitrekenen welke stof overblijft na mengen.
- Je kan een titratie uitvoeren, een titratiecurve lezen en er de concentratie van een zuur mee bepalen.
- Je kan een pH-meter ijken met bufferoplossingen en uit een ijking de pH van een monster berekenen.
- Je kan uitleggen waarom de pH belangrijk is in de bodem, in de hydroponie en in de oceaan, en de pH van de voedingsoplossing bewaken en bijsturen.

UITLEG · 15.1

Wat je al weet, en wat nu komt

Azijn, citroensap en cola smaken zuur; zeep en soda voelen glad aan. Op het etiket van een ontstopper en van ontkalker staan gevarenpictogrammen (F5). In F8 zag je dat azijn stroom geleidt, maar zwak: er zijn ionen in. Nu leer je wat die zure en basische stoffen gemeen hebben, en hoe je hun sterkte meet.

wat je al weet	waar	wat er nu bij komt
gevarenpictogrammen: bijtend, irriterend	F5	waarom zuren en basen bijten, en hoe je veilig verdunt
concentratie in g/L; azijn geleidt zwak; cola is zuur	F8	concentratie in mol/L en de pH
de logaritmische schaal: elke stap is een factor 10	hoofdstuk 5	de pH-schaal is er een
ionen; zouten vallen in water uiteen (Arrhenius)	hoofdstuk 13	zuren en basen maken ionen: H_3O^+ en OH^-
reactievergelijkingen, de mol, $n = m/M$	hoofdstuk 14	neutralisatie en titratie: rekenen met mol per liter
de EC van de voedingsoplossing	hoofdstuk 13	de tweede meting die elke hydroponieteler doet: de pH

In dit hoofdstuk

- leer je wat een stof zuur of basisch maakt, volgens Arrhenius en Brønsted;
- reken je de pH uit, en zie je waarom de pH-schaal met machten van tien werkt;
- kies je een indicator en maak je er zelf een uit rode kool;
- laat je een zuur en een base elkaar opheffen, en bepaal je met een titratie hoeveel zuur er in azijn zit;
- ijk je een pH-meter met bufferoplossingen;
- bekijk je de pH van de bodem, van de voedingsoplossing en van de oceaan;
- bewaak en stuur je voor Habitat de pH van het voedingsreservoir bij.

Boven eetbaar zuur en zacht basisch, onder bijtend: dezelfde chemie, een andere sterkte.

UITLEG · 15.2

Zuren en basen in het dagelijks leven

2/23

Zuren smaken zuur (proef het nooit in het labo!), reageren met kalk en met onedele metalen zoals zink en ijzer (er ontstaat waterstofgas), en kleuren blauw lakmoespapier rood. **Basen** voelen glad of zeepachtig aan, omdat ze de vetten en eiwitten van je huid aantasten, en kleuren rood lakmoes blauw. Een oplossing van een base in water heet ook een **loog**: natronloog, kaliloog. En het belangrijkste: een zuur en een base heffen elkaars werking op.

product	zuur of base	werkzame stof	waarvoor
azijn	zuur	azijnzuur CH_3COOH	smaak, bewaren, ontkalken
citroensap, frisdrank	zuur	citroenzuur, fosforzuur, koolzuur	frisse smaak, bewaren
ontkalker	zuur	citroenzuur, amidosulfonzuur of mierenzuur	kalk $CaCO_3$ oplossen
maagsap	zuur	zoutzuur (HCl in water)	eiwitten verteren, bacteriën doden
accuzuur (autobatterij)	zuur	zwavelzuur H_2SO_4, ± 35 %	elektrolyt van de loodaccu (hoofdstuk 16)
bakpoeder, soda	zwakke base	$NaHCO_3$, Na_2CO_3	deeg laten rijzen, reinigen

product	zuur of base	werkzame stof	waarvoor
zeep	zwakke base	natrium- en kaliumzouten van vetzuren	vet losmaken
allesreiniger	base	ammoniak NH_3 of soda	vet en vuil losmaken
ontstopper	sterke base	natriumhydroxide NaOH	haar en vet in de afvoer afbreken
kalk voor de tuin	base	calciumcarbonaat $CaCO_3$	zure grond minder zuur maken

Zuren én basen kunnen bijten

Geconcentreerde zuren (accuzuur, zoutzuur, sommige ontkalkers) en geconcentreerde basen (ontstopper, natronloog) dragen het pictogram **bijtend** (GHS05): ze vreten huid, ogen en metaal aan. Basen zijn voor de ogen zelfs gevaarlijker dan veel zuren, omdat ze dieper in het weefsel doordringen. Verdunde oplossingen en soda zijn vaak **irriterend** (GHS07). Draag altijd een bril, en spoel een spat meteen minutenlang af met veel water.

Verdun je een geconcentreerd zuur: **eerst het water, dan het zuur**, langzaam en al roerend. Omgekeerd kan de warmte die vrijkomt het water plots doen koken, en spat het zuur eruit.

GHS05 bijtend

GHS07 irriterend

Links: geconcentreerde zuren en basen. Rechts: veel verdunde oplossingen, soda, citroenzuur.

VERHAAL · 15.3

Sørensen, bier en de p van pH

3/23

In 1909 leidde de Deense chemicus Søren Sørensen de chemische afdeling van het Carlsberg Laboratorium in Kopenhagen, het onderzoekslabo van de gelijknamige brouwerij. Hij bestudeerde eiwitten en enzymen, de stoffen die van mout en gist bier maken. Hij merkte dat enzymen heel gevoelig zijn voor de zuurgraad: een beetje zuurder of minder zuur, en ze werkten veel trager.

De zuurgraad zelf was lastig op te schrijven. De concentraties van de waterstofionen in zijn oplossingen waren getallen met veel nullen achter de komma, die bovendien een factor honderd of duizend van elkaar konden verschillen. Sørensen stelde voor om alleen de **exponent** te noteren, zonder het minteken: 0,000 01 mol/L is 10^{-5} mol/L, en dus 'p_H 5'. Zo ontstond de pH-schaal. Waar de p precies voor staat, weet niemand zeker; de meest gehoorde uitleg is *Potenz*, Duits voor macht.

Een kwarteeuw later, in 1934, vroeg een oud-studiegenoot bij de citrusvereniging van Californië aan de chemicus Arnold Beckman om een toestel waarmee je snel en betrouwbaar de zuurgraad van citroensap kon meten. Beckman combineerde een glaselektrode met een versterker met elektronenbuizen, en

bouwde zo de eerste handige pH-meter. Vanaf 1935 maakte zijn bedrijf ze in serie. Vandaag meet elke hydroponieteler er dagelijks mee.

Waarom dit verhaal ertoe doet

De pH is geen natuurconstante maar een slimme schrijfwijze: een logaritme, bedacht om onhandige getallen hanteerbaar te maken. Dat verklaart ook waarom pH 4 niet 'een beetje' zuurder is dan pH 5, maar tien keer.

Het Carlsberg Laboratorium rond 1909: enzymen, bier en de uitvinding van de pH-schaal.

UITLEG · 15.4

Wat maakt een stof zuur of basisch?

In hoofdstuk 13 leerde je van Arrhenius dat zouten in water uiteenvallen in ionen. Hij ging nog een stap verder: een **zuur** is een stof die in water **waterstofionen** vormt, een **base** een stof die in water **hydroxide-ionen** OH^- vormt.

Waterstofchloride is een polair gas (hoofdstuk 13). Leid je het in water, dan geeft elke molecule haar H-atoom zonder elektron, dus een proton H^+, door aan een watermolecule. Zo'n los proton bestaat in water niet: het hangt meteen vast aan een vrij elektronenpaar van zuurstof, en vormt het **oxoniumion** H_3O^+. De oplossing heet **zoutzuur**.

ZOUTZUUR ONTSTAAT

$$\mathrm{HCl\,(g) + H_2O\,(l) \longrightarrow H_3O^+\,(aq) + Cl^-\,(aq)}$$

Vaak schrijft men kortweg H^+ in plaats van H_3O^+: het betekent hetzelfde.

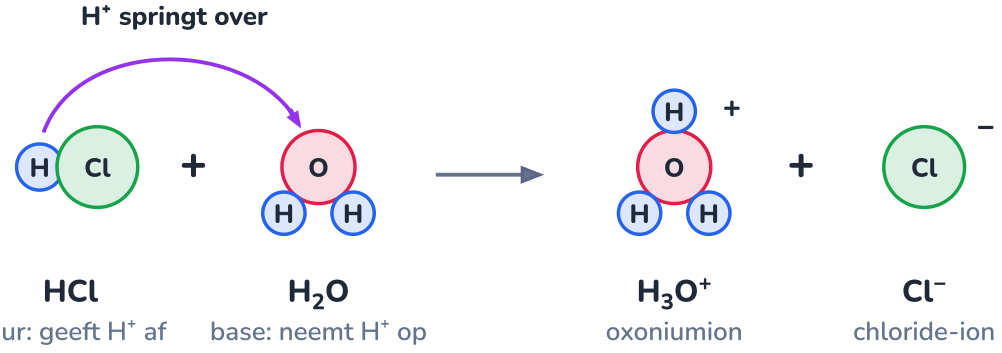
Natriumhydroxide is een zout; het valt in water uiteen in Na^+ en OH^-: $\mathrm{NaOH\,(s) \longrightarrow Na^+\,(aq) + OH^-\,(aq)}$. Ammoniak bevat geen OH^- en is toch een base: het neemt een proton van water over, $\mathrm{NH_3 + H_2O \rightleftharpoons NH_4^+ + OH^-}$.

Dat laatste verklaart de ruimere definitie van de Deen Johannes Brønsted en de Engelsman Thomas Lowry (1923): een **zuur** is een **protondonor**, een **base** een **protonacceptor**. Een zuur-basereactie is dan het doorgeven van een proton. Water kan allebei: bij zoutzuur neemt het een proton op, bij ammoniak geeft het er een af.

Sterk en zwak. Bij zoutzuur en salpeterzuur HNO_3 geven (bijna) alle moleculen hun proton af: het zijn **sterke zuren**. Bij azijnzuur doet maar een kleine fractie dat, zo'n 1 % in een oplossing van 0,1 mol/L: een **zwak zuur**. De dubbele pijl bij ammoniak betekent dat de reactie in beide richtingen verloopt en maar gedeeltelijk gebeurt. In dit hoofdstuk reken je alleen de pH van sterke zuren en basen uit; zwakke volgen in E26.

Sterk is niet hetzelfde als geconcentreerd

Sterk zegt welk deel van de moleculen een proton afgeeft; *geconcentreerd* zegt hoeveel zuur er per liter is. Zoutzuur van 0,001 mol/L is een sterk zuur in een verdunde oplossing (pH 3); azijn is een zwak zuur in een vrij geconcentreerde oplossing (0,83 mol/L, pH ± 2,4). Bij dezelfde concentratie van 0,1 mol/L heeft zoutzuur pH 1,0 en azijnzuur ongeveer pH 2,9.



Brønsted in één beeld: het zuur geeft een proton af, het water neemt het op.

UITLEG · 15.5

Water en de pH-schaal

5/23

Ook zuiver water bevat een piepklein beetje ionen: watermoleculen geven elkaar af en toe een proton door, $\mathrm{2\,H_2O \rightleftharpoons H_3O^+ + OH^-}$. In zuiver water bij 25 °C zijn er van elk $1{,}0 \cdot 10^{-7}$ mol per liter: evenveel H_3O^+ als OH^-, dus **neutraal**. Het product van beide concentraties blijft in *elke* waterige oplossing gelijk: voeg je zuur toe, dan stijgt $[H_3O^+]$ en daalt $[OH^-]$ evenveel keer.

DE WATERCONSTANTE

$$K_w = [\mathrm{H_3O^+}] \cdot [\mathrm{OH^-}] = 1{,}0 \cdot 10^{-14} \qquad (25\,^\circ\mathrm{C})$$

De rechte haken betekenen: concentratie in mol/L. Bij andere temperaturen is K_w anders: bij 37 °C is zuiver water neutraal bij pH 6,8.

Die concentraties lopen van ongeveer 1 mol/L tot 10^{-14} mol/L: veertien machten van tien. Daarom gebruik je, zoals Sørensen, de exponent:

PH EN POH (25 °C)

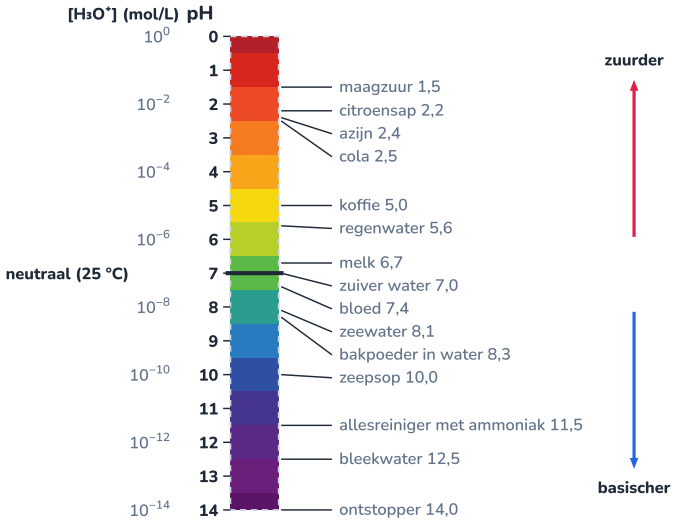
$$\mathrm{pH} = -\log[\mathrm{H_3O^+}] \qquad [\mathrm{H_3O^+}] = 10^{-\mathrm{pH}} \qquad \mathrm{pOH} = -\log[\mathrm{OH^-}] \qquad \mathrm{pH} + \mathrm{pOH} = 14$$

pH onder 7: zuur; pH 7: neutraal; pH boven 7: basisch. De pH heeft geen eenheid, maar de concentratie moet in mol/L staan.

De pH-schaal is **logaritmisch** (hoofdstuk 5): één pH-eenheid lager betekent tien keer meer oxoniumionen, twee eenheden honderd keer. Citroensap (pH 2,2) bevat $10^{5{,}0-2{,}2} = 10^{2{,}8} \approx 630$ keer meer H_3O^+ dan koffie (pH 5,0): het verschil van de pH-waarden is de exponent.

De logaritme op je rekenmachine

$\log(10^{-3}) = -3$: de logaritme van een macht van tien is de exponent. Voor andere getallen gebruik je de toets log: $\log(2{,}0 \cdot 10^{-3}) = -2{,}70$, dus pH 2,70. Terug gaat met 10^x: bij pH 3,40 hoort $10^{-3{,}40} = 4{,}0 \cdot 10^{-4}$ mol/L. Geef een pH met twee decimalen: zoveel cijfers heeft ook een goede pH-meter.



Typische waarden: elk streepje op de schaal is een factor tien in de concentratie oxoniumionen.

UITGEWERKT VOORBEELD · 15.6

De pH van sterke zuren en basen berekenen

6/23

(a) Bereken de pH van zoutzuur met 0,020 mol/L. (b) Bereken de pH van natronloog met 0,0050 mol/L. (c) Een oplossing heeft pH 3,40. Hoe groot is de concentratie oxoniumionen, en hoeveel keer zuurder is ze dan een oplossing met pH 5,40?

GEGEVEN

- (a) $c(\mathrm{HCl}) = 0{,}020\,\mathrm{mol/L}$, sterk zuur
- (b) $c(\mathrm{NaOH}) = 0{,}0050\,\mathrm{mol/L}$, sterke base

GEVRAAGD

- pH
- $[\mathrm{H_3O^+}]$ en de verhouding

- (c) pH 3,40 en pH 5,40

OPLOSSING

1. (a) Zoutzuur is een sterk zuur: elke HCl-molecule wordt een H_3O^+. Dus $[H_3O^+] = 0{,}020\,\text{mol/L}$ en $\text{pH} = -\log(0{,}020) = 1{,}70$.
2. (b) NaOH valt volledig uiteen: $[OH^-] = 0{,}0050\,\text{mol/L}$. $\text{pOH} = -\log(0{,}0050) = 2{,}30$, dus $\text{pH} = 14 - 2{,}30 = 11{,}70$.
3. Of via de waterconstante: $[H_3O^+] = \dfrac{1{,}0 \cdot 10^{-14}}{0{,}0050} = 2{,}0 \cdot 10^{-12}\,\text{mol/L}$, en $-\log(2{,}0 \cdot 10^{-12}) = 11{,}70$. Zelfde uitkomst.
4. (c) $[H_3O^+] = 10^{-3{,}40} = 4{,}0 \cdot 10^{-4}\,\text{mol/L}$. Het verschil is 2,00 pH-eenheden: $10^{2{,}00} = 100$ keer meer oxoniumionen.

ANTWOORD

(a) pH 1,70; (b) pH 11,70; (c) $4{,}0 \cdot 10^{-4}$ mol/L, honderd keer zuurder dan pH 5,40.

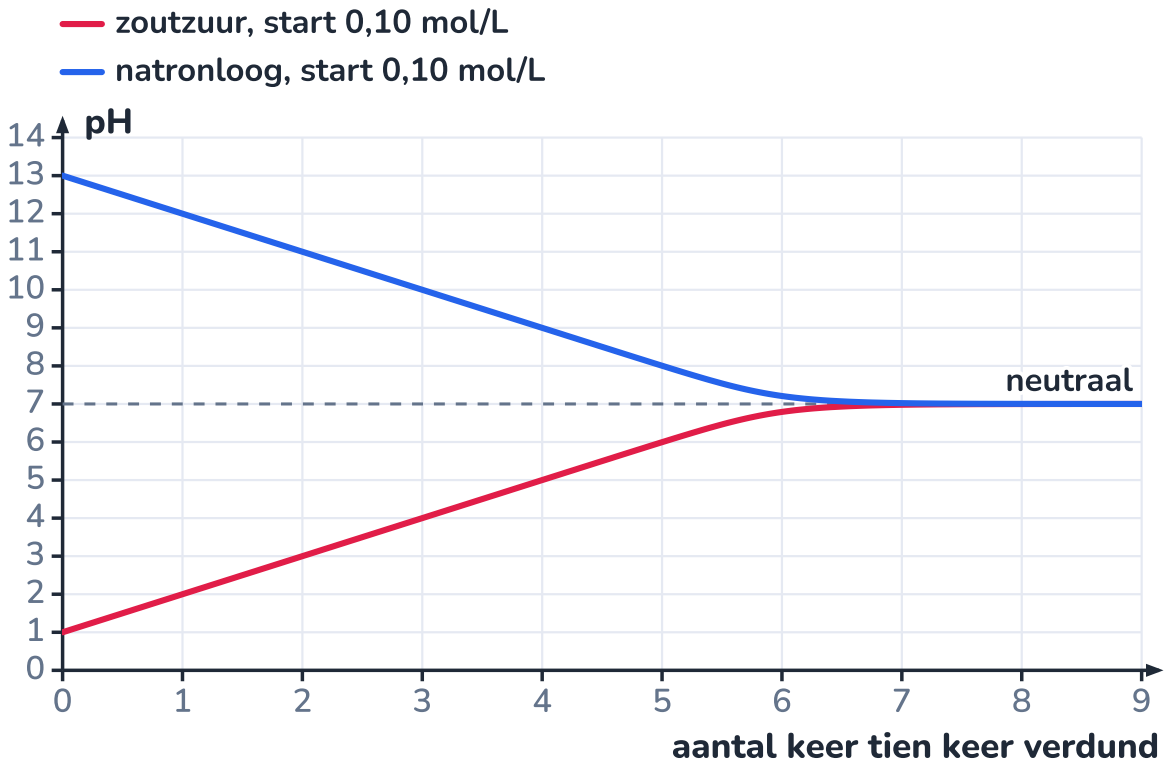
Klopt dit? Schat eerst: 0,020 ligt tussen 0,01 (pH 2) en 0,1 (pH 1), dus de pH ligt tussen 1 en 2. Een base met $[OH^-]$ tussen 10^{-3} en 10^{-2} heeft een pH tussen 11 en 12. Klopt je antwoord niet met zo'n schatting, dan heb je vaak het minteken van de logaritme vergeten.

UITGEWERKT VOORBEELD · 15.7

Verdunnen: tien keer meer water, één pH-eenheid

7/23

(a) Je verdunt 10 mL zoutzuur met pH 1,00 met water tot 1,0 L. Welke pH krijg je? (b) Hoeveel water moet je bij 50 mL zoutzuur met pH 2,0 voegen om pH 4,0 te krijgen? (c) Wat gebeurt er als je natronloog met pH 12,0 honderd keer verdunt? En als je daarna blijft verdunnen?



Exact berekend: elke factor tien is eerst één pH-eenheid, maar rond pH 7 buigen beide lijnen af.

GEGEVEN

GEVRAAGD

- $[H_3O^+] = 10^{-\text{pH}}$; verdunnen met factor f deelt de concentratie door f

- de pH na verdunnen
- het volume water

OPLOSSING

1. (a) Van 10 mL naar 1000 mL is een factor 100. $[H_3O^+]$ gaat van 10^{-1} naar $10^{-3}\,\text{mol/L}$: pH 3,00. Elke factor tien verdunning verhoogt de pH van een sterk zuur met één eenheid.

2. (b) Van pH 2,0 naar pH 4,0 is twee eenheden: een factor $10^2 = 100$. Het eindvolume is $100 \cdot 50\,\text{mL} = 5{,}0\,\text{L}$, dus je voegt $5{,}0 - 0{,}05 = 4{,}95\,\text{L}$ water toe.

3. (c) $[OH^-]$ gaat van 10^{-2} naar $10^{-4}\,\text{mol/L}$: pOH 4, pH 10,0. Een base daalt bij verdunnen naar 7 toe, een zuur stijgt naar 7 toe.

4. Blijf je verdunnen, dan nadert de pH 7, maar ze gaat er nooit voorbij: het water brengt zelf $10^{-7}\,\text{mol/L}$ H_3O^+ en OH^- mee. De grafiek is berekend met de waterconstante erbij.

ANTWOORD

(a) pH 3,00; (b) aanvullen tot 5,0 L, dus bijna 5 L water; (c) pH 10,0, en bij verder verdunnen naar pH 7.

Klopt dit? Zeer verdund zoutzuur van 10^{-8} mol/L heeft geen pH 8 (dat zou basisch zijn!) maar pH 6,98: het zuur is dan een druppel op een gloeiende plaat naast de 10^{-7} mol/L oxoniumionen van het water zelf.

UITLEG · 15.8

Indicatoren: kleurstoffen die de pH verraden

8/23

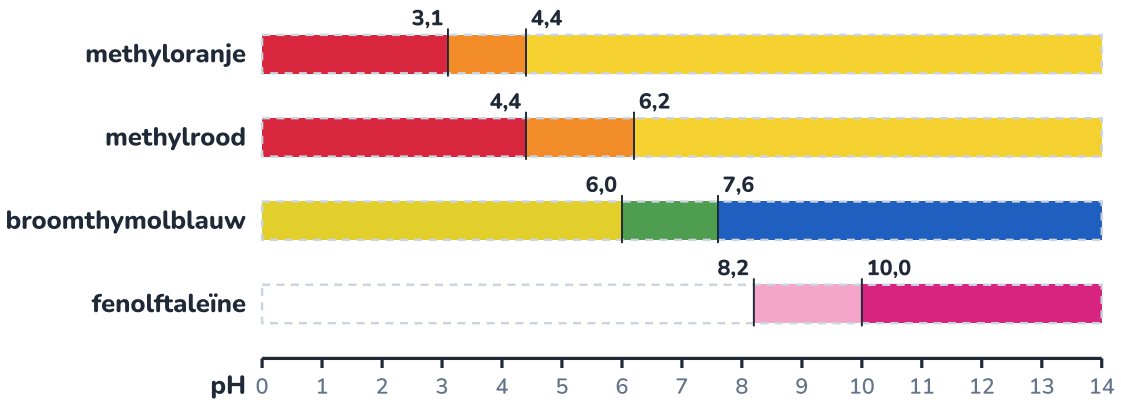
Een **indicator** is een kleurstof die zelf een zwak zuur is. De vorm met het proton heeft een andere kleur dan de vorm zonder. Welke vorm de overhand heeft, hangt af van de pH: onder het **omslaggebied** zie je de zure kleur, erboven de basische, en erin een mengkleur. Elke indicator heeft zijn eigen omslaggebied, meestal zo'n anderhalve tot twee pH-eenheden breed.

indicator	omslaggebied (pH)	zure kant	basische kant
methyloranje	3,1 – 4,4	rood	geel
methylrood	4,4 – 6,2	rood	geel
broomthymolblauw	6,0 – 7,6	geel	blauw
fenolftaleïne	8,2 – 10,0	kleurloos	roze tot paarsrood
lakmoes	4,5 – 8,3	rood	blauw
universele indicator (mengsel)	0 – 14, geleidelijk	rood, oranje, geel, groen, blauw, paars	
rodekoolsap (antocyaan)	geleidelijk	rood, roze, paars	blauw, groen, geel

Eén indicator vertelt je alleen of de pH onder, in of boven zijn omslaggebied ligt. Kleurt broomthymolblauw geel, dan weet je: pH onder 6,0, maar niet of het 5,5 of 1,5 is. Daarom mengt men in **universele indicator** en **pH-papier** meerdere kleurstoffen: elke pH geeft dan een eigen kleur, en je vergelijkt met een kleurenkaart tot op ongeveer één pH-eenheid.

Een indicator combineren

Met twee indicatoren kan je een pH insluiten. Is een oplossing geel met broomthymolblauw (pH < 6,0) en geel met methylrood (pH > 6,2)? Dat kan niet tegelijk: dan heb je slecht gekeken. Geel met broomthymolblauw en oranje met methylrood: pH tussen 4,4 en 6,0.



Echte kleuren, benaderd. Broomthymolblauw slaat om rond pH 7; fenolftaleïne pas in het basische gebied.

PROEF · 15.9

Rodekoolsap als indicator

9/23

ONDERZOEKSVRAAG

Kan je met sap van rode kool zure en basische producten uit de keuken herkennen en rangschikken?

MATERIAAL

een kwart rode kool, heet water (niet kokend), mes, zeef · 8 doorzichtige bekertjes, pH-papier · azijn, citroensap, spuitwater, melk, kraanwater, bakpoeder, soda (natriumcarbonaat), zeepsop · veiligheidsbril

Veiligheid

Bril op. Soda irriteert de ogen ernstig (GHS07). Proef en drink niets in het labo, ook niet wat uit de keuken komt. Laat het hete water eerst wat afkoelen voor je het over de kool giet.

GHS07

soda:

ernstige oogirritatie

Soda is een keukenproduct, maar geen onschuldig: bril op.

WERKWIJZE

1. Snijd de kool in reepjes, overgiet met 500 mL heet water en laat tien minuten trekken. Zeef: je krijgt een paars sap.

2 Giet in elk bekertje 30 mL sap. Voeg in elk bekertje een theelepel (of 10 mL) van één product toe en roer.

3 Noteer de kleur. Meet daarna met pH-papier de pH van elk mengsel.

4 Rangschik de bekertjes van laagste naar hoogste pH. Zet de kleuren naast de pH-waarden.

WAARNEMING

product	kleur met rodekoolsap	pH (papier)
citroensap	rood	2
azijn	rood	3
spuitwater	roze	5
melk	paars	6-7
kraanwater	blauwpaars	7-8
bakpoeder	blauw	8
zeepsop	blauwgroen	9-10
soda	groen	11

Voorbeeldmeting. De tinten verschillen van kool tot kool.

VERKLARING

Rode kool bevat **antocyanen**, de kleurstoffen die ook bosbessen en rode druiven kleuren. Het zijn zwakke zuren met meer dan één proton dat ze kunnen afgeven. Bij elke pH overheerst een andere vorm, met een eigen kleur: rood in zuur, paars rond neutraal, blauw en groen in basisch, en geel in sterk basisch. Daarom geeft rodekoolsap een hele kleurenreeks en geen enkele omslag. Voor een nauwkeurige pH is het te vaag: de kleuren lopen in elkaar over en hangen af van hoeveel sap je gebruikt.

UITLEG · 15.10

Neutralisatie: zuur en base heffen elkaar op

10/23

Giet je zoutzuur bij natronloog, dan reageren de oxoniumionen met de hydroxide-ionen tot water. Dat is de **neutralisatie**:

NEUTRALISATIE, ALS IONENVERGELIJKING

$$\mathrm{H_3O^+\,(aq) + OH^-\,(aq) \longrightarrow 2\,H_2O\,(l)}$$

De andere ionen doen niet mee: ze zijn 'toeschouwers'.

De natrium- en chloride-ionen blijven gewoon opgelost. Damp je de oplossing in, dan blijft er keukenzout over. Met alle stoffen erbij:

$$\mathrm{HCl\,(aq) + NaOH\,(aq) \longrightarrow NaCl\,(aq) + H_2O\,(l)}$$

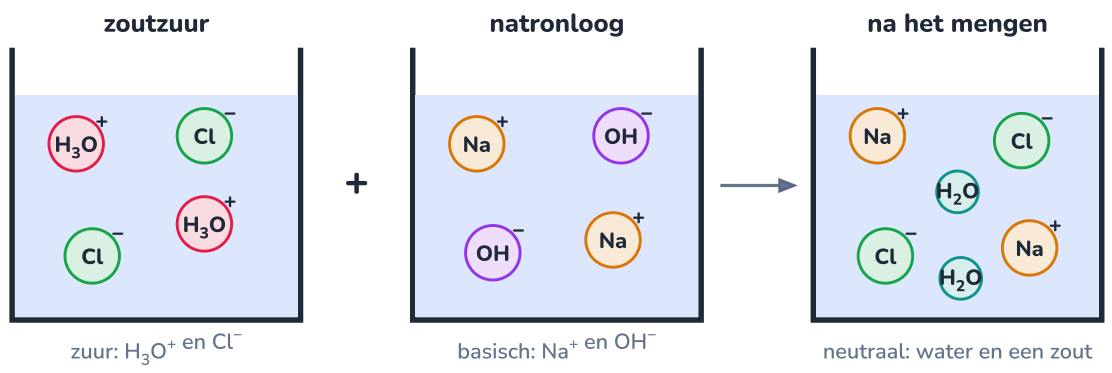
Een zuur plus een base geeft dus **water plus een zout**. Welk zout, hangt af van het zuur en de base: salpeterzuur met kaliloog geeft kaliumnitraat, een van de zouten uit het recept van de voedingsoplossing (hoofdstuk 12): $\mathrm{HNO_3 + KOH \longrightarrow KNO_3 + H_2O}$. Bij zwavelzuur, dat twee protonen kan afgeven, heb je twee keer zoveel base nodig: $\mathrm{H_2SO_4 + 2\,NaOH \longrightarrow Na_2SO_4 + 2\,H_2O}$.

Een neutralisatie is **exotherm** (hoofdstuk 14): per mol gevormd water komt ongeveer 56 kJ vrij. Meng je 50 mL zoutzuur en 50 mL natronloog van elk 1 mol/L, dan warmt het mengsel ongeveer 6,7 °C op.

Neutraliseren doe je vaker dan je denkt: een maagtablet met calciumcarbonaat neutraliseert een teveel aan maagzuur, een tuinier strooit kalk op zure grond, en een gemorst zuur maak je onschadelijk met bakpoeder (het bruist: er ontstaat CO_2).

Na neutralisatie is de pH altijd 7

Alleen als een sterk zuur en een sterke base in precies de juiste hoeveelheid reageren. Blijft er zuur of base over, dan bepaalt die overmaat de pH (volgend voorbeeld). En neutraliseer je een zwak zuur met een sterke base, dan ligt het equivalentiepunt boven 7 (E26).



Oxonium- en hydroxide-ionen worden water; natrium en chloride blijven als toeschouwers achter.

UITGEWERKT VOORBEELD · 15.11

Zuur en base mengen: wie blijft over?

11/23

Je mengt 50,0 mL zoutzuur van 0,100 mol/L met 30,0 mL natronloog van 0,100 mol/L. Welke pH heeft het mengsel?

GEGEVEN

- zoutzuur: $V = 50{,}0\,\text{mL}$, $c = 0{,}100\,\text{mol/L}$
- natronloog: $V = 30{,}0\,\text{mL}$, $c = 0{,}100\,\text{mol/L}$

GEVRAAGD

- pH van het mengsel

OPLOSSING

1. Tel in mol (of millimol: mL maal mol/L geeft mmol). Zuur: $50{,}0 \cdot 0{,}100 = 5{,}00\,\text{mmol}$ H_3O^+. Base: $30{,}0 \cdot 0{,}100 = 3{,}00\,\text{mmol}$ OH^-.
2. Molverhouding 1 : 1: de 3,00 mmol OH^- neutraliseert 3,00 mmol H_3O^+. Er blijft $5{,}00 - 3{,}00 = 2{,}00\,\text{mmol}$ H_3O^+ over.
3. Die overmaat zit in het totale volume: $50{,}0 + 30{,}0 = 80{,}0\,\text{mL}$. $[H_3O^+] = 2{,}00/80{,}0 = 0{,}0250\,\text{mol/L}$.
4. $\text{pH} = -\log(0{,}0250) = 1{,}60$.

ANTWOORD

Het mengsel heeft pH 1,60: het is nog altijd zuur.

Klopt dit? Vergeet het totale volume niet: wie deelt door 50 mL, vindt pH 1,40. Een gevoelscontrole: de begin-pH van het zuur is 1,00; na het neutraliseren van 60 % van het zuur en een verdunning stijgt ze maar 0,6 eenheid. Zo werkt een logaritmische schaal.

Titratie: tot op een druppel meten

Hoeveel zuur zit er in een fles azijn? Met een **titratie** meet je dat. Je doet een nauwkeurig gekend volume van de onbekende oplossing in een erlenmeyer, met een paar druppels indicator. Uit een **buret** laat je er druppel per druppel een oplossing met gekende concentratie bij lopen, tot de indicator net blijvend van kleur verandert.

1. Spoel de buret met de natronloog, vul ze en lees de beginstand af, onderaan de meniscus, tot op 0,05 mL.
2. Breng met een volumepipet precies 25,00 mL van het zuur in de erlenmeyer. Voeg 2 à 3 druppels indicator toe.
3. Laat eerst vlot, en in de buurt van het omslagpunt druppel per druppel natronloog toe, al roerend.
4. Stop bij de eerste druppel die de kleur een halve minuut laat blijven (bij fenolftaleïne: licht roze). Lees de eindstand af.
5. Herhaal tot drie titraties minder dan 0,10 mL van elkaar verschillen, en neem het gemiddelde (hoofdstuk 1).

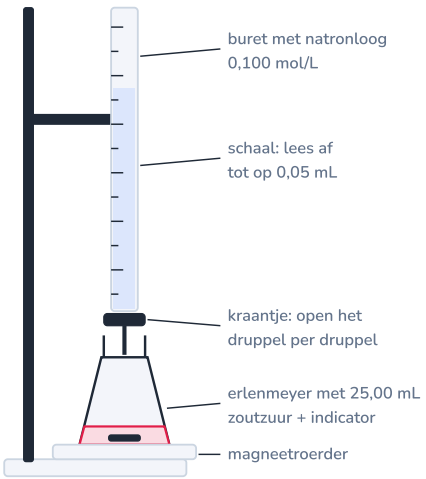
Op het **equivalentiepunt** is er precies evenveel mol base toegevoegd als er mol zuur was. Voor een zuur en een base die 1 : 1 reageren:

EQUIVALENTIEPUNT (MOLVERHOUDING 1 : 1)

$$c_{\text{zuur}} \cdot V_{\text{zuur}} = c_{\text{base}} \cdot V_{\text{base}}$$

Het volume mag in mL, als het aan beide kanten in mL staat. Bij een andere molverhouding reken je via de mol, zoals in hoofdstuk 14.

Het punt waar de indicator omslaat, heet het **eindpunt**. Een goede indicator zorgt dat eindpunt en equivalentiepunt samenvallen, tot op een druppel. Dat de berekening ook voor een zwak zuur als azijnzuur werkt, is geen toeval: je telt alleen hoeveel mol base er nodig is, niet hoeveel protonen er op een bepaald moment vrij zijn.



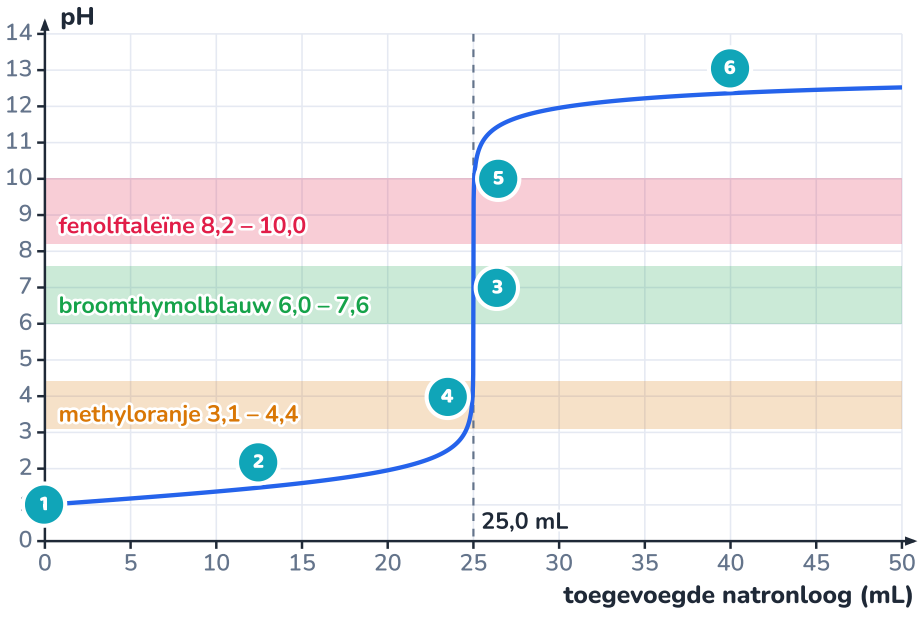
Een witte ondergrond maakt de eerste roze zweem zichtbaar; de roerder mengt elke druppel meteen.

Het eindpunt: één druppel natronloog kleurt de hele oplossing licht roze.

ONTLEED · 15.13

De titratiecurve

Meet je tijdens een titratie na elke toevoeging de pH, dan krijg je een **titratiecurve**. Deze is niet gemeten maar exact berekend (codeblad verderop): 25,0 mL zoutzuur van 0,100 mol/L met natronloog van 0,100 mol/L. De gekleurde banden zijn de omslaggebieden van drie indicatoren.



De sprong bij 25,0 mL gaat door de omslaggebieden van alle drie de indicatoren.

1 Beginpunt
Alleen zoutzuur: pH = –log 0,100 = 1,00.

2 Het vlakke stuk
Na 12,5 mL is de helft van het zuur weg, en toch is de pH maar gestegen tot 1,48: de concentratie is gehalveerd (en verdund), dat is nog geen factor tien.

3 Equivalentiepunt
Na 25,00 mL: evenveel mol OH^- als er H_3O^+ was. Er is alleen nog een oplossing van keukenzout: pH 7,00.

4 Eén druppel ervoor
Bij 24,95 mL (één druppel te weinig) is de pH 4,0: er blijft nog 0,005 mmol zuur over.

5 Eén druppel erna
Bij 25,05 mL is de pH al 10,0. Eén druppel van 0,05 mL doet de pH zes eenheden stijgen: daarom is een titratie zo nauwkeurig.

6 Overmaat base
Nu bepaalt de overmaat natronloog de pH. De curve nadert pH 13 (de pH van de natronloog zelf), maar haalt die niet: de base wordt verdund door het zuur.

UITGEWERKT VOORBEELD · 15.14

Hoeveel azijnzuur zit er in azijn?

Op een fles schoonmaakazijn staat '5 % zuurgraad': 5 g azijnzuur per 100 mL. Je pipetteert 10,00 mL azijn in een maatkolf en vult aan tot 100,0 mL met water. Van die verdunde azijn titreer je telkens 25,00 mL met natronloog van 0,100 mol/L en fenolftaleïne. Je hebt 20,85 mL, 20,80 mL en 20,75 mL nodig. Klopt het etiket?

GEGEVEN

- $CH_3COOH + NaOH \longrightarrow CH_3COONa + H_2O$ (1 : 1)
- $V_{base} = 20{,}80\,\text{mL}$ (gemiddelde), $c_{base} = 0{,}100\,\text{mol/L}$
- $M(CH_3COOH) = 60{,}05\,\text{g/mol}$

GEVRAAGD

- de concentratie azijnzuur in de azijn, in mol/L en in g per 100 mL

OPLOSSING

1. Mol base tot het equivalentiepunt: $n = c \cdot V = 0{,}100\,\text{mol/L} \cdot 0{,}02080\,\text{L} = 2{,}08 \cdot 10^{-3}\,\text{mol}$.

2. Molverhouding 1 : 1: in de 25,00 mL verdunde azijn zat $2{,}08 \cdot 10^{-3}$ mol azijnzuur.

3. In de hele maatkolf (100,0 mL, vier keer zoveel): $8{,}32 \cdot 10^{-3}\,\text{mol}$. Dat kwam allemaal uit de 10,00 mL azijn.

4. Concentratie in de azijn: $c = \dfrac{8{,}32 \cdot 10^{-3}\,\text{mol}}{0{,}01000\,\text{L}} = 0{,}832\,\text{mol/L}$.

5. In gram: $0{,}832 \cdot 60{,}05 = 50{,}0\,\text{g/L}$, of 5,00 g per 100 mL.

ANTWOORD

De azijn bevat 0,832 mol/L azijnzuur, 5,0 g per 100 mL: het etiket klopt.

Klopt dit? Waarom eerst tien keer verdunnen? Onverdund zou je voor 25 mL azijn 208 mL natronloog nodig hebben: meer dan vier buretten. Kies de verdunning zo dat je eindigt tussen 15 en 45 mL. De drie metingen liggen binnen 0,10 mL van elkaar: het gemiddelde is betrouwbaar tot op ongeveer 0,5 %.

CODE · 15.15

Python: de titratiecurve berekenen

15/23

Hoe bereken je de pH *tijdens* een titratie? Tel de overmaat zuur (of base) in mol, deel door het totale volume, en neem de waterconstante mee. Dan wordt het één kwadratische vergelijking: de **ladingsbalans** zegt dat er in de oplossing evenveel positieve als negatieve lading is. Met $h = [\text{H}_3\text{O}^+]$ en d de overmaat zuur in mol/L geldt $h - K_w/h = d$, dus $h^2 - d\,h - K_w = 0$. Rond het equivalentiepunt ($d \approx 0$) geeft dat vanzelf pH 7, en ver ervan gewoon $-\log d$.

Python `titratiecurve.py` 22 regels

```python
# titratiecurve.py - pH tijdens de titratie van een sterk zuur met een sterke base
import math

KW = 1.0e-14            # waterconstante bij 25 graden: [H3O+] * [OH-]


def ph_mengsel(c_zuur, v_zuur, c_base, v_base):
    """pH na het mengen van v_zuur mL sterk zuur (c_zuur mol/L) met v_base mL sterke base (c_base mol/L)."""
    overmaat = (c_zuur * v_zuur - c_base * v_base) / (v_zuur + v_base)   # mol/L zuur (negatief: base)
    # ladingsbalans: [H3O+] - KW / [H3O+] = overmaat  ->  kwadratische vergelijking in [H3O+]
    h = (overmaat + math.sqrt(overmaat ** 2 + 4 * KW)) / 2
    return -math.log10(h)


C_ZUUR, V_ZUUR = 0.100, 25.0     # 25,0 mL zoutzuur van 0,100 mol/L in de erlenmeyer
C_BASE = 0.100                   # natronloog van 0,100 mol/L in de buret

if __name__ == "__main__":
    v_eq = C_ZUUR * V_ZUUR / C_BASE
    print(f"equivalentiepunt bij {v_eq:.2f} mL")
    for v in [0, 10, 20, 24, 24.9, 24.95, 25, 25.05, 25.1, 26, 30, 50]:
        print(f"{v:6.2f} mL   pH {ph_mengsel(C_ZUUR, V_ZUUR, C_BASE, v):5.2f}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
equivalentiepunt bij 25.00 mL
  0.00 mL   pH  1.00
 10.00 mL   pH  1.37
 20.00 mL   pH  1.95
```

```
24.00 mL   pH  2.69
24.90 mL   pH  3.70
24.95 mL   pH  4.00
25.00 mL   pH  7.00
25.05 mL   pH 10.00
25.10 mL   pH 10.30
26.00 mL   pH 11.29
30.00 mL   pH 11.96
50.00 mL   pH 12.52
```

De titratiecurve en de verdunningsgrafiek van dit hoofdstuk komen uit deze functie. In opdracht 15.11 gebruik je ze om te onderzoeken hoe breed de sprong is bij een tien keer meer verdund zuur.

UITLEG · 15.16

De pH-meter en zijn ijking

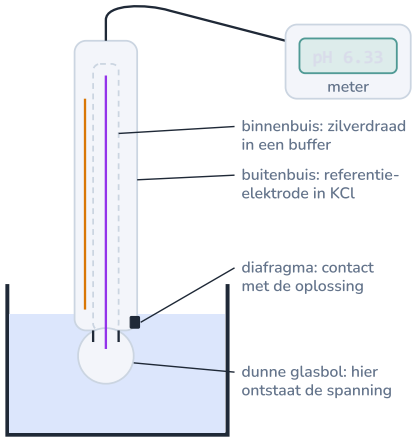
16/23

Een pH-meter meet een **spanning**. Aan het uiteinde van de elektrode zit een heel dunne glasbol. Aan de binnenkant zit een oplossing met een vaste pH, aan de buitenkant de vloeistof die je meet. Hoe groter het verschil in pH, hoe groter de spanning over het glas. Een tweede elektrode, de **referentie**, maakt via een klein poreus **diafragma** contact met de vloeistof en sluit de kring. Een ideale elektrode geeft bij 25 °C 0 mV bij pH 7 en 59,16 mV per pH-eenheid: een lineair verband (waarom precies dat getal, zie je in E27).

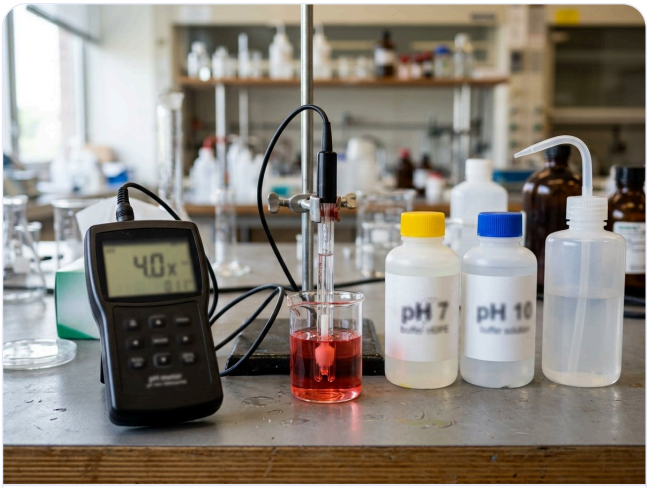
Een echte elektrode wijkt af: het nulpunt verschuift en de helling wordt kleiner naarmate het glas veroudert. Daarom **ijk** (kalibreer) je de meter met **bufferoplossingen**: oplossingen met een nauwkeurig gekende pH die nauwelijks verandert, zelfs als er een beetje zuur of base bij komt.

- **Tweepuntsijking**: eerst buffer pH 7,00 (het nulpunt), dan pH 4,00 als je in het zure gebied meet (hydroponie) of pH 10,00 in het basische. De meter legt een rechte door de twee punten, zoals je in hoofdstuk 3 een sensor ijkte.
- **Driepuntsijking** met 4, 7 en 10 als je over het hele gebied meet.
- Tussen twee buffers spoel je de elektrode met gedemineraliseerd water en dep je ze droog. Niet wrijven.
- De meter berekent de **helling** in procent van 59,16 mV per pH. Veel fabrikanten vragen tussen ongeveer 95 en 105 %, en een nulpunt binnen ongeveer 30 mV. Daarbuiten reinig of vervang je de elektrode.
- De helling hangt af van de temperatuur (bij 20 °C is ze 58,17 mV per pH): goede meters meten de temperatuur mee en compenseren. De pH van een buffer verandert ook een beetje met de temperatuur; de tabel staat op het flesje.

> ⚠ **Een elektrode droogt niet uit**
>
> Bewaar een pH-elektrode altijd met het dopje vol **bewaarvloeistof** (kaliumchloride, 3 mol/L), nooit droog en nooit in gedemineraliseerd water: dan trekken ionen uit het glas en uit de referentie weg, en wordt de meter traag en onbetrouwbaar. Een elektrode gaat een à twee jaar mee.



De spanning ontstaat over de dunne glasbol; de referentie-elektrode sluit de kring via het diafragma.



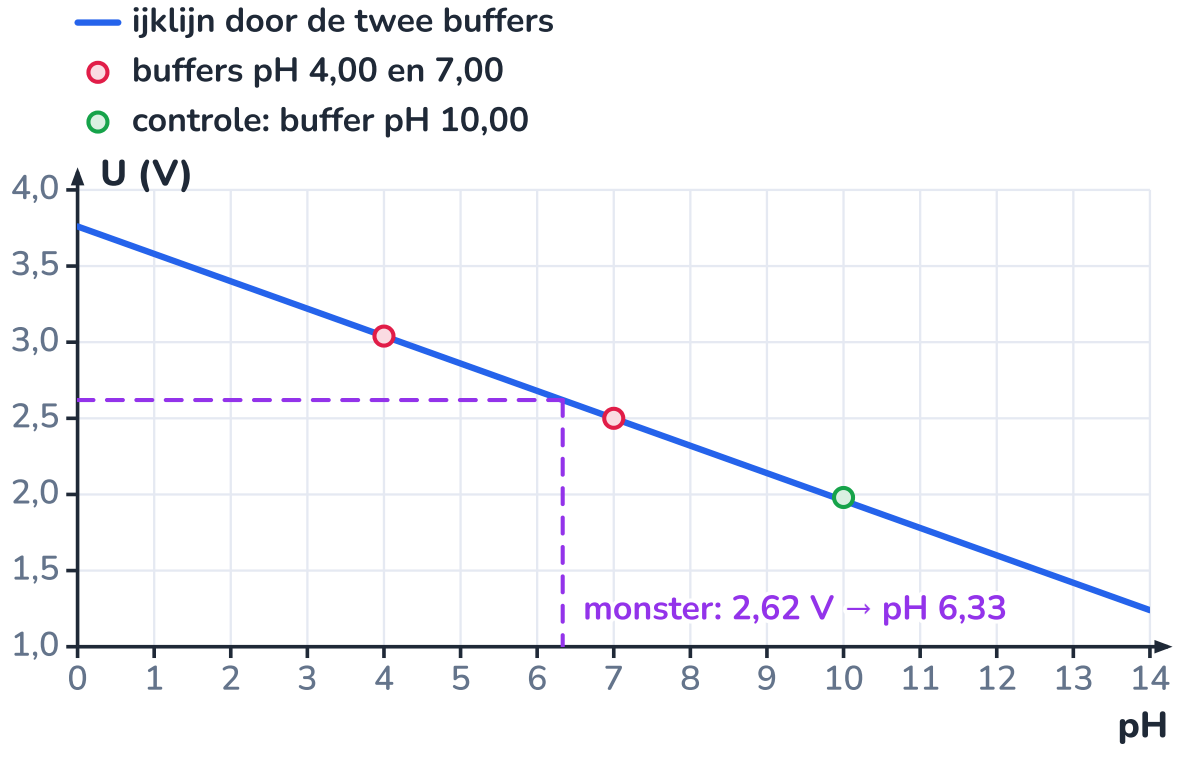
Gekleurde buffers: rood voor pH 4, geel voor pH 7, blauw voor pH 10 (een gangbare afspraak).

UITGEWERKT VOORBEELD · 15.17

Een pH-sensor ijken met twee buffers

17/23

De klimaatmodule krijgt een goedkope pH-module die een spanning tussen 0 en 5 V naar de microcontroller stuurt. In buffer pH 7,00 meet je 2,50 V, in buffer pH 4,00 3,04 V. (a) Stel de ijklijn op. (b) In het reservoir meet je 2,62 V. Welke pH is dat? (c) Ter controle hang je de sensor in buffer pH 10,00: 1,98 V. Hoe groot is de afwijking?



De rechte door de twee buffers; de derde buffer controleert of het verband ook daarbuiten klopt.

GEGEVEN

- $(7{,}00;\ 2{,}50\ \text{V})$ en $(4{,}00;\ 3{,}04\ \text{V})$
- monster: $2{,}62\ \text{V}$; controle: $1{,}98\ \text{V}$ in pH 10,00

GEVRAAGD

- ijklijn
- pH van het monster
- afwijking bij pH 10

OPLOSSING

1. (a) Helling: $a = \dfrac{3{,}04 - 2{,}50}{4{,}00 - 7{,}00} = \dfrac{0{,}54}{-3{,}00} = -0{,}180\ \text{V}$ per pH-eenheid. De spanning daalt als de pH stijgt. Ijklijn: $U = 2{,}50 - 0{,}180 \cdot (\text{pH} - 7{,}00)$.

2. Omgevormd naar wat de microcontroller nodig heeft: $\text{pH} = 7{,}00 + \dfrac{U - 2{,}50}{-0{,}180}$.

3. (b) $\text{pH} = 7{,}00 + \dfrac{2{,}62 - 2{,}50}{-0{,}180} = 7{,}00 - 0{,}67 = 6{,}33$.

4. (c) $\text{pH} = 7{,}00 + \dfrac{1{,}98 - 2{,}50}{-0{,}180} = 7{,}00 + 2{,}89 = 9{,}89$: 0,11 onder 10,00.

ANTWOORD

$\text{pH} = 7{,}00 - (U - 2{,}50)/0{,}180$; **het reservoir staat op pH 6,33; bij pH 10 wijkt de sensor 0,11 af.**

Klopt dit? Een afwijking van 0,1 bij pH 10, ver buiten het gebied tussen de twee buffers, is aanvaardbaar voor de hydroponie, waar je tussen 5,5 en 6,5 werkt: dat ligt tussen de ijkpunten. Wie ook bij pH 10 nauwkeurig wil meten, doet een driepuntsijking. Het monster op pH 6,33 ligt in het werkgebied, maar al boven de streefwaarde 5,8.

UITLEG · 15.18

pH in de bodem en in de hydroponie

18/23

Een plant neemt haar voeding op als opgeloste ionen (hoofdstuk 12). Of een ion opgelost blijft, hangt sterk af van de pH. In de **bodem** doen de meeste groenten het best tussen pH 6 en 7; op zure zandgrond, zoals in de Kempen, strooit men **kalk** $CaCO_3$ om de pH op te trekken (**bekalken**): het carbonaat neutraliseert het zuur. Sommige planten houden net van zuur.

gewas of plant	gunstige bodem-pH
de meeste groenten (sla, kool, bonen), gazon	6,0 – 7,0
aardappel	5,0 – 6,0
blauwe bes, rododendron, heide	4,5 – 5,5

In de **hydroponie** is er geen bodem die schommelingen opvangt: de pH van de voedingsoplossing verandert soms met een halve eenheid in twee dagen. Voor de meeste gewassen ligt het werkgebied tussen **5,5 en 6,5**; sla doet het goed rond 5,8.

pH van de voedingsoplossing	wat er gebeurt
onder 5,0	de wortels raken beschadigd; calcium en magnesium worden slechter opgenomen; zwaar zure oplossing tast ook metalen onderdelen aan
5,5 – 6,5	alle voedingselementen blijven opgelost en opneembaar: het **werkgebied** van de meeste gewassen (sla, kruiden, tomaat)
boven 6,5	ijzer en mangaan slaan neer of worden slecht opneembaar: jonge bladeren worden geel tussen de nerven (ijzergebrek); fosfaat vormt met calcium een neerslag
boven 7,5	kalkaanslag (calciumcarbonaat en -fosfaat) in leidingen en druppelaars

Waarom stijgt de pH vanzelf? Het recept van Habitat levert de stikstof als nitraat NO_3^-. Een plant die meer negatieve ionen opneemt dan positieve, moet haar lading in evenwicht houden: haar wortels geven OH^-- of HCO_3^--ionen af. De oplossing wordt langzaam basischer, zeker als de planten snel groeien.

Met ammonium NH_4^+ als stikstofbron gebeurt het omgekeerde. Daarom heb je in de hydroponie vooral **pH-min** nodig: een verdund zuur, meestal salpeterzuur of fosforzuur, dat meteen ook stikstof of fosfor aanvult.

> ⚠ **EC en pH horen samen**
>
> Zuur toevoegen brengt ook ionen mee: de EC stijgt een beetje. Meet daarom altijd **eerst de pH, dan de EC**, en stuur de pH bij voor je de EC bijstuurt. Laat een pH- en een EC-sonde niet tegelijk meten in hetzelfde reservoir zonder elektrische scheiding: ze kunnen elkaars meting storen.

Eerst de pH, dan de EC: de dagelijkse ronde van een hydroponieteler.

VERHAAL · 15.19

Oceaanverzuring: een logaritme met gevolgen

19/23

Sinds het begin van de industriële revolutie steeg de koolstofdioxide in de lucht van ongeveer 280 naar 425 ppm (hoofdstuk 14). Ongeveer een kwart van de CO_2 die we uitstoten, lost op in de oceaan. Daar vormt het koolzuur, een zwak zuur:

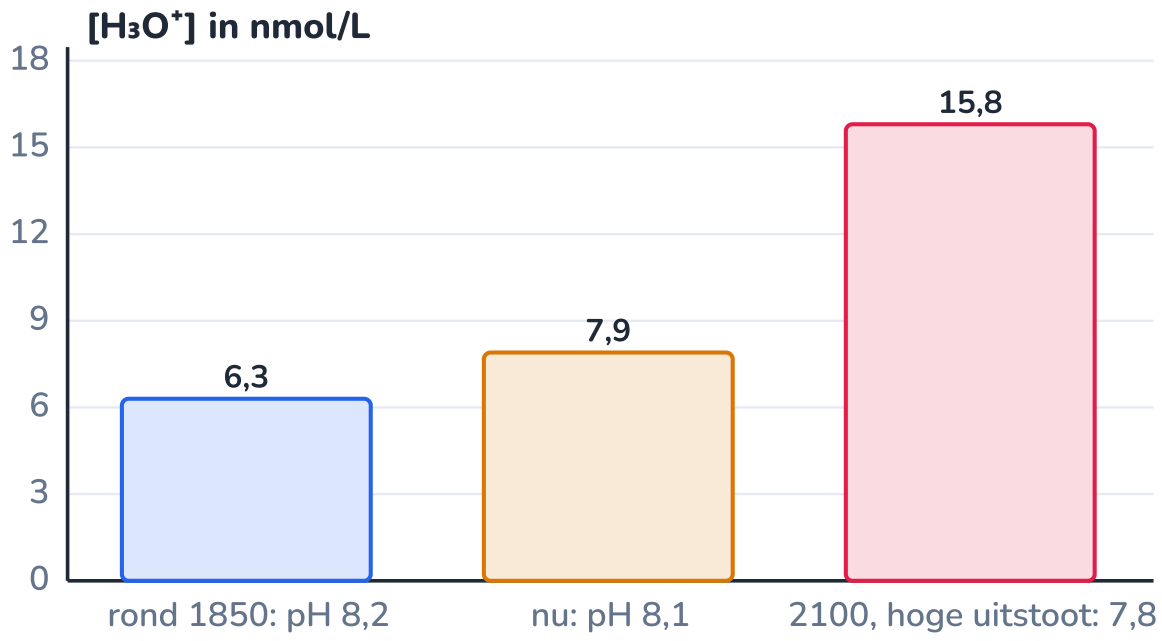
$$\mathrm{CO_2\,(aq) + 2\,H_2O\,(l) \rightleftharpoons H_3O^+\,(aq) + HCO_3^-\,(aq)}$$

Zeewater is licht basisch. Rond 1850 had het oppervlaktewater gemiddeld een pH van ongeveer 8,2; vandaag is dat 8,1. Een daling van 0,1 lijkt niets, maar de schaal is logaritmisch: $10^{0,1} = 1{,}26$, dus ongeveer een kwart meer oxoniumionen. Met nauwkeuriger waarden (van 8,21 naar 8,10) is het bijna 30 %. Bij een hoge uitstoot daalt de pH tegen 2100 nog eens ruim 0,3, tot ongeveer 7,8: dan zijn er $10^{0,4} \approx 2{,}5$ keer zoveel oxoniumionen als in 1850.

Basisch blijft de oceaan, 'verzuren' betekent hier: de pH daalt. Het probleem zit bij de dieren die een skelet of schelp bouwen uit calciumcarbonaat: koralen, oesters, mosselen, zeeslakjes en veel plankton. De extra oxoniumionen pakken carbonaationen weg ($H_3O^+ + CO_3^{2-} \longrightarrow HCO_3^- + H_2O$), precies de bouwstenen die ze nodig hebben. Hun kalk bouwen kost meer energie, en in koud water, waar CO_2 goed oplost, begint dunne kalk al op te lossen.

Meten op zee

Sinds 1988 meet een onderzoeksprogramma ten noorden van Hawaï maandelijks de pH van het oppervlaktewater. De meetreeks daalt gestaag, in het ritme van de stijgende CO_2 in de lucht: ongeveer 0,02 pH-eenheid per tien jaar. Zulke kleine veranderingen zie je alleen met zorgvuldig geijkte meters en een lange meetreeks.



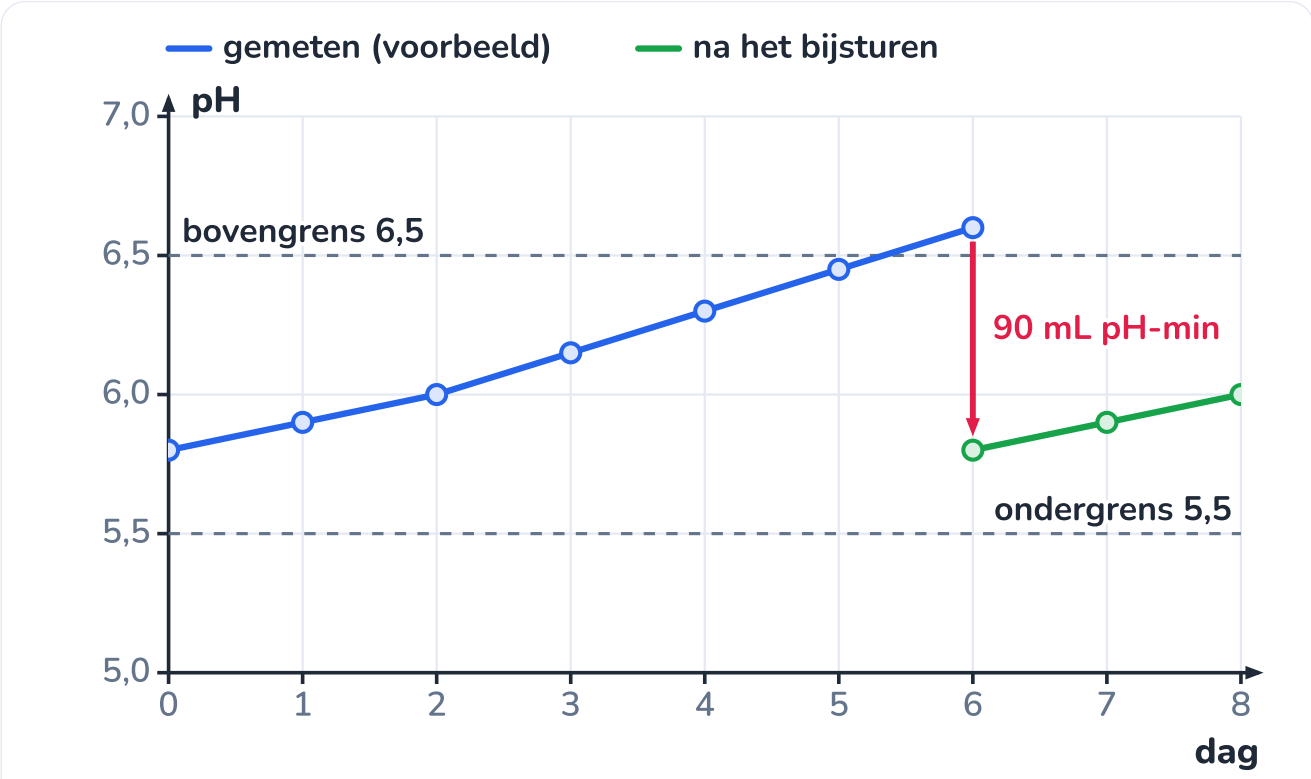
Dezelfde daling in concentraties: van 6,3 naar 7,9 nanomol per liter, en misschien 15,8 in 2100.

UITGEWERKT VOORBEELD · 15.20

Habitat: de pH van het reservoir bijsturen

20/23

De voedingsoplossing van de klimaatmodule start op pH 5,8. Een week lang meet je elke ochtend de pH (grafiek). Op dag 6 staat ze op 6,6 en zit er nog 56 L in het reservoir. De pH-min is een werkoplossing salpeterzuur van 0,50 mol/L. Je giet 1,00 L oplossing uit het reservoir in een maatbeker en druppelt er met een spuitje pH-min bij, al roerend: na 1,6 mL staat de pH op 5,8. (a) Hoeveel pH-min heeft het reservoir nodig? (b) Hoe voeg je het toe? (c) Hoeveel mmol zuur zou zuiver water van 56 L nodig hebben om van 6,6 naar 5,8 te gaan? Wat besluit je?



Voorbeeldmeting: de pH kruipt elke dag omhoog door de nitraatopname; op dag 6 terug naar 5,8 (streefwaarde).

GEGEVEN

- 56 L op pH 6,6; streefwaarde 5,8
- proefje: 1,6 mL pH-min (0,50 mol/L) per 1,00 L
- de pomp zet het reservoir in 12,5 min één keer rond (hoofdstuk 11)

GEVRAAGD

- volume pH-min
- werkwijze
- vergelijking met zuiver water

OPLOSSING

1 (a) Evenredig opschalen: $56 \cdot 1{,}6\,\text{mL} = 89{,}6\,\text{mL} \approx 90\,\text{mL}$ pH-min.

2 (b) Voeg niet alles in één keer toe: giet 45 mL in de stroming bij de pomp, laat de pomp 15 minuten rondpompen (iets meer dan één keer het hele reservoir), meet opnieuw, en voeg de rest toe als de pH nog boven 6,0 staat. Een te grote dosis zet de wortels in één keer in zuur water. Bril en handschoenen: ook verdund salpeterzuur irriteert ogen en huid.

3 (c) Zuiver water: $[\text{H}_3\text{O}^+]$ moet van $10^{-6{,}6} = 2{,}5 \cdot 10^{-7}$ naar $10^{-5{,}8} = 1{,}58 \cdot 10^{-6}\,\text{mol/L}$, een toename van $1{,}33 \cdot 10^{-6}\,\text{mol/L}$. Voor 56 L: $7{,}5 \cdot 10^{-5}\,\text{mol} = 0{,}075\,\text{mmol}$, dat is 0,15 mL pH-min.

4 Het reservoir had $0{,}090\,\text{L} \cdot 0{,}50\,\text{mol/L} = 45\,\text{mmol}$ nodig: ongeveer 600 keer meer. De voedingsoplossing vangt het zuur op: de fosfaat- en waterstofcarbonaationen erin nemen de meeste protonen over. Zo'n oplossing heet een **buffer** (E26). Daarom kan je de dosis niet uitrekenen met $\text{pH} = -\log[\text{H}_3\text{O}^+]$, maar bepaal je ze met een proefje: een kleine titratie.

ANTWOORD

Ongeveer 90 mL pH-min, in twee keer, met een kwartier rondpompen ertussen. Zuiver water zou maar 0,15 mL nodig hebben: de voedingsoplossing buffert.

Klopt dit? Het salpeterzuur brengt ook stikstof mee: 45 mmol N is $45 \cdot 14{,}0 = 630\,\text{mg}$, in 56 L 11 mg/L extra, bovenop de 112 mg/L van het recept: geen probleem. De pH stijgt hier ongeveer 0,13 per dag: reken dus op één correctie per week, en meer als de planten groot zijn.

CODE · 15.21

C++: de pH bewaken op de microcontroller

De microcontroller van de klimaatmodule meet elke minuut. Drie keuzes uit dit hoofdstuk zitten in de code: de **ijklijn** uit het voorbeeld (twee constanten die je na elke ijking aanpast), een **mediaan** van negen metingen tegen uitschieters (hoofdstuk 1), en de **grenzen** van het werkgebied 5,5 tot 6,5. De code doseert zelf niets: bijsturen doe je na een proefje op 1 L.

C++ (Arduino) **ph_reservoir.ino** 54 regels

```cpp
// pH-bewaking van het voedingsreservoir (Arduino Mega 2560 van de klimaatmodule)
// pH-module op A2 (Arduino Mega van de module): de spanning daalt als de pH stijgt. Tweepuntsijking met buffers pH 7,00 en pH 4,00.
const float U_PH7 = 2.50;      // gemeten spanning in buffer pH 7,00 (V)
const float U_PH4 = 3.04;      // gemeten spanning in buffer pH 4,00 (V)
const float HELLING = (U_PH4 - U_PH7) / (4.00 - 7.00);   // V per pH-eenheid (negatief)
const float PH_ONDER = 5.5;    // grenzen van het werkgebied
const float PH_BOVEN = 6.5;
const unsigned long INTERVAL_MS = 60000UL;             // elke minuut meten

unsigned long vorigeMeting = 0;

float leesSpanning() {
  // mediaan van 9 metingen: een uitschieter verandert het resultaat niet
  int w[9];
  for (int i = 0; i < 9; i++) {
    w[i] = analogRead(A2);
    delay(10);
  }
  for (int i = 1; i < 9; i++) {          // sorteren (insertion sort)
    int x = w[i];
    int j = i - 1;
    while (j >= 0 && w[j] > x) {
      w[j + 1] = w[j];
      j--;
    }
    w[j + 1] = x;
  }
  return w[4] * 5.0 / 1023.0;            // de middelste van de negen
}

float naarPH(float u) {
  return 7.00 + (u - U_PH7) / HELLING;
}

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (millis() - vorigeMeting >= INTERVAL_MS) {
    vorigeMeting += INTERVAL_MS;
    float u = leesSpanning();
    float ph = naarPH(u);
    Serial.print("U = ");
    Serial.print(u, 3);
    Serial.print(" V   pH = ");
    Serial.println(ph, 2);
    if (ph > PH_BOVEN) {
      Serial.println("pH te hoog: proefje op 1 L en pH-min doseren");
    } else if (ph < PH_ONDER) {
      Serial.println("pH te laag: dosering nakijken, eventueel pH-plus");
    }
  }
}
```

De helling staat in de code voor 25 °C. Wordt het reservoir 's winters 18 °C, dan is de helling van de elektrode ruim 2 % kleiner; bij pH 6 (één eenheid van het nulpunt) scheelt dat maar 0,02. Wie nauwkeuriger wil zijn, meet de temperatuur mee (de sensor van hoofdstuk 13 zit al in het reservoir) en schaalt de helling met de absolute temperatuur. In opdracht 15.16 ontwerp je een automatische dosering, met de veiligheden die daarbij horen.

NASLAG · 15.22

Overzicht: zuren, basen en pH

22/23

pH

$$\mathrm{pH} = -\log\left[\mathrm{H_3O^+}\right] \qquad \left[\mathrm{H_3O^+}\right] = 10^{-\mathrm{pH}}$$

symbool	betekenis	eenheid
$[\mathrm{H_3O^+}]$	concentratie oxoniumionen	mol/L

waterconstante (25 °C)

$$\left[\mathrm{H_3O^+}\right] \cdot \left[\mathrm{OH^-}\right] = 1{,}0 \cdot 10^{-14} \qquad \mathrm{pH} + \mathrm{pOH} = 14$$

symbool	betekenis	eenheid
$[\mathrm{OH^-}]$	concentratie hydroxide-ionen	mol/L

equivalentiepunt (1 : 1)

$$c_{\text{zuur}} \cdot V_{\text{zuur}} = c_{\text{base}} \cdot V_{\text{base}}$$

symbool	betekenis	eenheid
c	concentratie	mol/L
V	volume	mL of L

tweepuntsijking

$$\mathrm{pH} = 7{,}00 + \frac{U - U_7}{a}$$

symbool	betekenis	eenheid
U_7	spanning in buffer pH 7,00	V
a	helling (U_4 – U_7) / (4,00 – 7,00)	V

indicator	omslaggebied (pH)	zure kant	basische kant
methyloranje	3,1 – 4,4	rood	geel
methylrood	4,4 – 6,2	rood	geel
broomthymolblauw	6,0 – 7,6	geel	blauw
fenolftaleïne	8,2 – 10,0	kleurloos	roze tot paarsrood
lakmoes	4,5 – 8,3	rood	blauw
universele indicator (mengsel)	0 – 14, geleidelijk	rood, oranje, geel, groen, blauw, paars	
rodekoolsap (antocyaan)	geleidelijk	rood, roze, paars	blauw, groen, geel

sterke zuren	zwakke zuren	sterke basen	zwakke basen
zoutzuur HCl, salpeterzuur HNO_3, zwavelzuur H_2SO_4 (eerste proton)	azijnzuur, citroenzuur, koolzuur, fosforzuur	NaOH, KOH, $Ca(OH)_2$	ammoniak, waterstofcarbonaat, carbonaat

Habitat: rekenwaarden uit dit hoofdstuk	
pH voedingsoplossing	streefwaarde 5,8; bijsturen onder 5,5 of boven 6,5
drift	± 0,1 à 0,15 per dag omhoog (nitraatopname)
pH-min	werkoplossing salpeterzuur 0,50 mol/L; dosis via een proefje op 1,0 L (voorbeeld: 1,6 mL/L, 90 mL voor 56 L)
pH-module	2,50 V bij pH 7,00; 3,04 V bij pH 4,00; helling −0,18 V per pH
onderhoud	ijken om de twee weken (buffers 7 en 4); sonde in KCl-bewaarvloeistof

HET GEHEEL · 15.23

Waar zuren en basen terugkomen

De pH verbindt de chemie met het leven: enzymen werken maar in een smal pH-gebied, je bloed houdt zich tussen 7,35 en 7,45, en een plant kan haar voeding maar opnemen als de pH klopt. Wiskundig is de pH een toepassing van de logaritme; technisch is de pH-meter een sensor die je ijkt met een rechte.

F5 Veilig werken in labo en werkplaats
Gevarenpictogrammen; eerst water, dan zuur.

F8 Zuivere stoffen en mengsels
Oplossingen en concentratie in g/L; hier in mol/L.

I3 Verbanden in data
Een sensor ijken met een rechte: de tweepuntsijking van de pH-meter.

I5 Groei en verval: exponentiële modellen
De logaritmische schaal: elke pH-eenheid is een factor tien.

I13 Chemische bindingen en stofeigenschappen
Ionen en polaire moleculen; de EC naast de pH.

I14 Reacties en stoichiometrie
De mol en de molverhouding: rekenen aan neutralisatie en titratie.

I16 Redox en batterijen
Accuzuur, en metalen die in zuur oplossen: redox.

I34 Enzymen en stofwisseling
Het pH-optimum van een enzym.

I36 Hydroponie en voedselproductie
EC en pH samen bewaken in het hydroponiesysteem.

E26 Chemisch evenwicht
Zwakke zuren, Ka en pKa, en buffers: waarom de voedingsoplossing het zuur opvangt.

E27 Elektrochemie: batterijen, brandstofcellen en elektrolyse
Waarom een glaselektrode 59 mV per pH-eenheid geeft.

E33 Het lichaam als regelsysteem
De pH van het bloed als geregelde grootheid.

Habitat
Voor Habitat leggen we vast: de voedingsoplossing heeft een pH-streefwaarde van 5,8; onder 5,5 of boven 6,5 stuur je bij, telkens terug naar 5,8. Met het nitraatrijke recept stijgt de pH ongeveer 0,1 à 0,15 per dag. De pH-min is salpeterzuur van 0,50 mol/L; de dosis bepaal je met een proefje op 1,0 L. De pH-module op de microcontroller wordt om de twee weken geijkt met buffers 7 en 4. In opdracht 15.14 stel je het bewakingsplan op; in hoofdstuk 36 komt alles samen in het hydroponiesysteem.

SAMENVATTING

In het kort

- Zuren vormen in water H_3O^+, basen OH^- (Arrhenius). Ruimer: een zuur geeft een proton af, een base neemt er een op (Brønsted).
- Sterk betekent: (bijna) alle moleculen geven hun proton af. Dat is iets anders dan geconcentreerd.
- In elke waterige oplossing bij 25 °C is $[H_3O^+] \cdot [OH^-] = 1{,}0 \cdot 10^{-14}$. $pH = -\log [H_3O^+]$; pH + pOH = 14.
- De pH-schaal is logaritmisch: één eenheid is een factor tien. Tien keer verdunnen verhoogt de pH van een sterk zuur met één eenheid, maar verdunnen brengt de pH nooit voorbij 7.
- Een indicator verandert van kleur in zijn omslaggebied: methyloranje 3,1-4,4, broomthymolblauw 6,0-7,6, fenolftaleïne 8,2-10,0.
- Neutralisatie: $H_3O^+ + OH^- \rightarrow 2\ H_2O$; zuur + base geeft water + een zout. Blijft er iets over, dan bepaalt die overmaat de pH.
- Bij een titratie voeg je base met gekende concentratie toe tot het equivalentiepunt. Rond dat punt springt de pH: één druppel verandert ze met vele eenheden.

- Een pH-meter meet een spanning; je ijkt hem met buffers (pH 7 en 4, of 7 en 10) en bewaart de elektrode nat.
- De meeste gewassen in de hydroponie willen pH 5,5 tot 6,5. Door de opname van nitraat stijgt de pH; je stuurt bij met pH-min, na een proefje. De oceaan daalde van pH 8,2 naar 8,1: ongeveer 30 % meer H_3O^+.

BEGRIPPEN

Wat je nu kent

zuur (Arrhenius)

Stof die in water H_3O^+-ionen vormt (kortweg H^+). Volgens Brønsted: een stof die een proton afgeeft.

base (Arrhenius)

Stof die in water OH^--ionen vormt. Volgens Brønsted: een stof die een proton opneemt.

oxoniumion

Het ion H_3O^+: een watermolecule met een extra proton. Het maakt een oplossing zuur.

sterk zuur

Zuur waarvan (bijna) alle moleculen in water een proton afgeven, zoals zoutzuur en salpeterzuur.

zwak zuur

Zuur waarvan maar een klein deel van de moleculen een proton afgeeft, zoals azijnzuur en citroenzuur.

pH

Maat voor de zuurgraad: $\mathrm{pH} = -\log[\mathrm{H_3O^+}]$, met de concentratie in mol/L. Onder 7 zuur, 7 neutraal, boven 7 basisch (bij 25 °C).

waterconstante

In water en elke waterige oplossing bij 25 °C geldt $[\mathrm{H_3O^+}] \cdot [\mathrm{OH^-}] = 1{,}0 \cdot 10^{-14}$, dus pH + pOH = 14.

indicator

Kleurstof die van kleur verandert binnen een bepaald pH-gebied, het omslaggebied.

neutralisatie

Reactie van een zuur met een base: H_3O^+ en OH^- vormen water, en er blijft een zout over.

titratie

Methode om een concentratie te bepalen door een oplossing met gekende concentratie toe te voegen tot de reactie precies af is.

equivalentiepunt

Het punt in een titratie waar er precies evenveel mol base is toegevoegd als er mol zuur was.

bufferoplossing

Oplossing waarvan de pH nauwelijks verandert als je er wat zuur of base bij doet; gebruikt om pH-meters te ijken.

oceaanverzuring

De daling van de pH van het zeewater doordat de oceaan koolstofdioxide uit de lucht opneemt.

I INTERMEDIATE · DEEL I4 · CHEMIE EN MATERIALEN

HANDBOEK · HOOFDSTUK 16

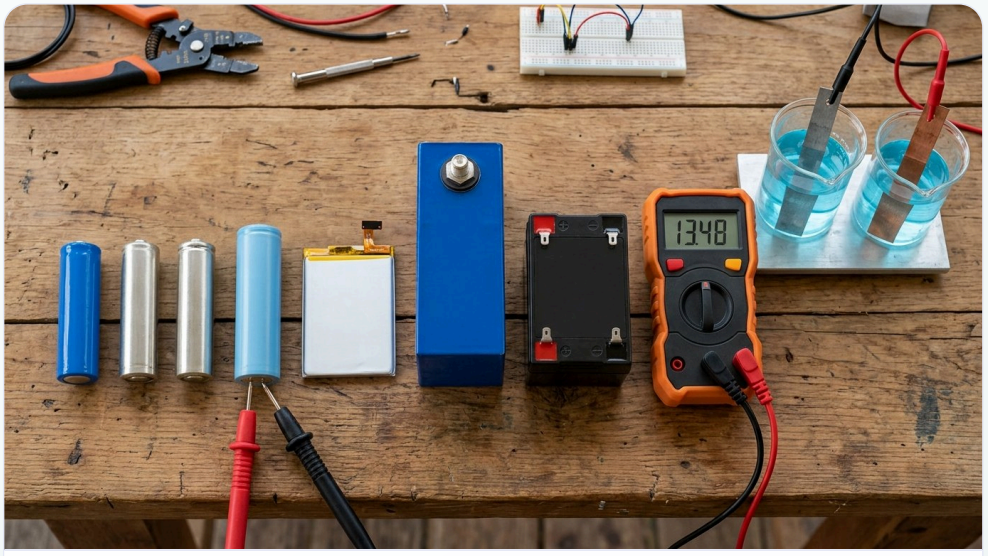
Redox en batterijen

Elektronen die van zink naar koper springen: van de Daniell-cel tot de batterij van de klimaatmodule

Een batterij is een chemische reactie die je in twee helften knipt. De ene stof geeft elektronen af, de andere neemt ze op, en omdat ze elkaar niet mogen raken, moeten de elektronen de lange weg nemen: door je lamp, je telefoon of je pomp. Je leert redoxreacties lezen als elektronenoverdracht, bouwt een Daniell-cel, rekent met ampère-uur en wattuur, vergelijkt batterijtypes, en ziet waarom roest eigenlijk een batterij is die je niet wil. Voor Habitat kies je het batterijtype van de klimaatmodule.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 2× verhaal · 1× het geheel · 2× code · ± 10 lestijden

M Materie en chemie · L Elektriciteit en elektronica · E Energie



Van AA tot LiFePO4: dezelfde natuurkunde, een andere chemie.

Na dit hoofdstuk

- Je kan oxidatie en reductie omschrijven als elektronen afgeven en opnemen, en een redoxreactie opsplitsen in twee kloppende halfreacties.
- Je kan met de spanningsreeks voorspellen of een metaal met de ionen van een ander metaal reageert, en welk metaal de minpool van een cel wordt.
- Je kan de werking van de Daniell-cel uitleggen: anode, kathode, elektronenstroom, ionen in de oplossingen en de rol van de zoutbrug.
- Je kan rekenen met capaciteit (Ah, mAh), energie (Wh) en lading, en met de constante van Faraday de massa berekenen die bij een elektrode verdwijnt of ontstaat.
- Je kan alkaline-, NiMH-, lood-zuur- en lithium-ionbatterijen vergelijken op spanning, energiedichtheid, levensduur en veiligheid.
- Je kan een ontlaadcurve lezen en uitleggen wat ontlaaddiepte, C-rate en bruikbare capaciteit betekenen.
- Je kan de gevaren van batterijen (thermal runaway, zuur, waterstofgas) benoemen en veilige maatregelen nemen.
- Je kan corrosie verklaren als een elektrochemische reactie en uitleggen hoe verf, verzinken en een opofferingsanode beschermen.
- Je kan onderbouwd een batterijtype kiezen voor de klimaatmodule van Habitat.

UITLEG · 16.1

Wat je al weet, en wat nu komt

In F17 zag je dat een batterij energie opslaat, en in F21 hoe je die energie in wattuur berekent. Maar *hoe* een batterij stroom maakt, bleef een zwarte doos. In hoofdstuk 14 leerde je reacties schrijven, in hoofdstuk 15 zag je zink oplossen in zuur. Nu kijk je naar wat er met de elektronen gebeurt.

wat je al weet	waar	wat er nu bij komt
een batterij slaat chemische energie op; energie in Wh en kWh	F17, F21	wat er in een batterij gebeurt, en waarom ze leegloopt
stroom is bewegende lading, $U = R \cdot I$	F19	lading tellen in ampère-uur
roest en de oxidelaag op aluminium	F10	corrosie als elektrochemische reactie
ionen, metaalbinding, vrije elektronen	hoofdstuk 12, 13	elektronen die van het ene deeltje naar het andere springen
reactievergelijkingen, de mol	hoofdstuk 14	halfreacties; rekenen met mol elektronen
metalen lossen op in zuur (waterstofgas)	hoofdstuk 15	waarom het ene metaal wel oplost en het andere niet
de voorlopige batterij van 1,2 kWh	hoofdstuk 9	het echte batterijtype voor Habitat

In dit hoofdstuk

- lees je oxidatie en reductie als elektronen afgeven en opnemen, en splits je reacties in halfreacties;
- voorspel je met de spanningsreeks welke metalen met elkaar reageren;
- bouw je een Daniell-cel en ontleed je hoe ze stroom maakt;
- reken je met ampère-uur, wattuur en de lading van een mol elektronen;
- vergelijk je batterijtypes, lees je ontlaadcurven en leer je veilig laden;
- zie je corrosie als een kleine batterij, en hoe je ze tegenhoudt;
- kies je voor Habitat het batterijtype van de klimaatmodule.

Roest, zink en magnesium: redox gebeurt ook waar je geen batterij ziet.

VERHAAL · 16.2

Kikkerpoten, een zuil van munten en een koperen pot

2/21

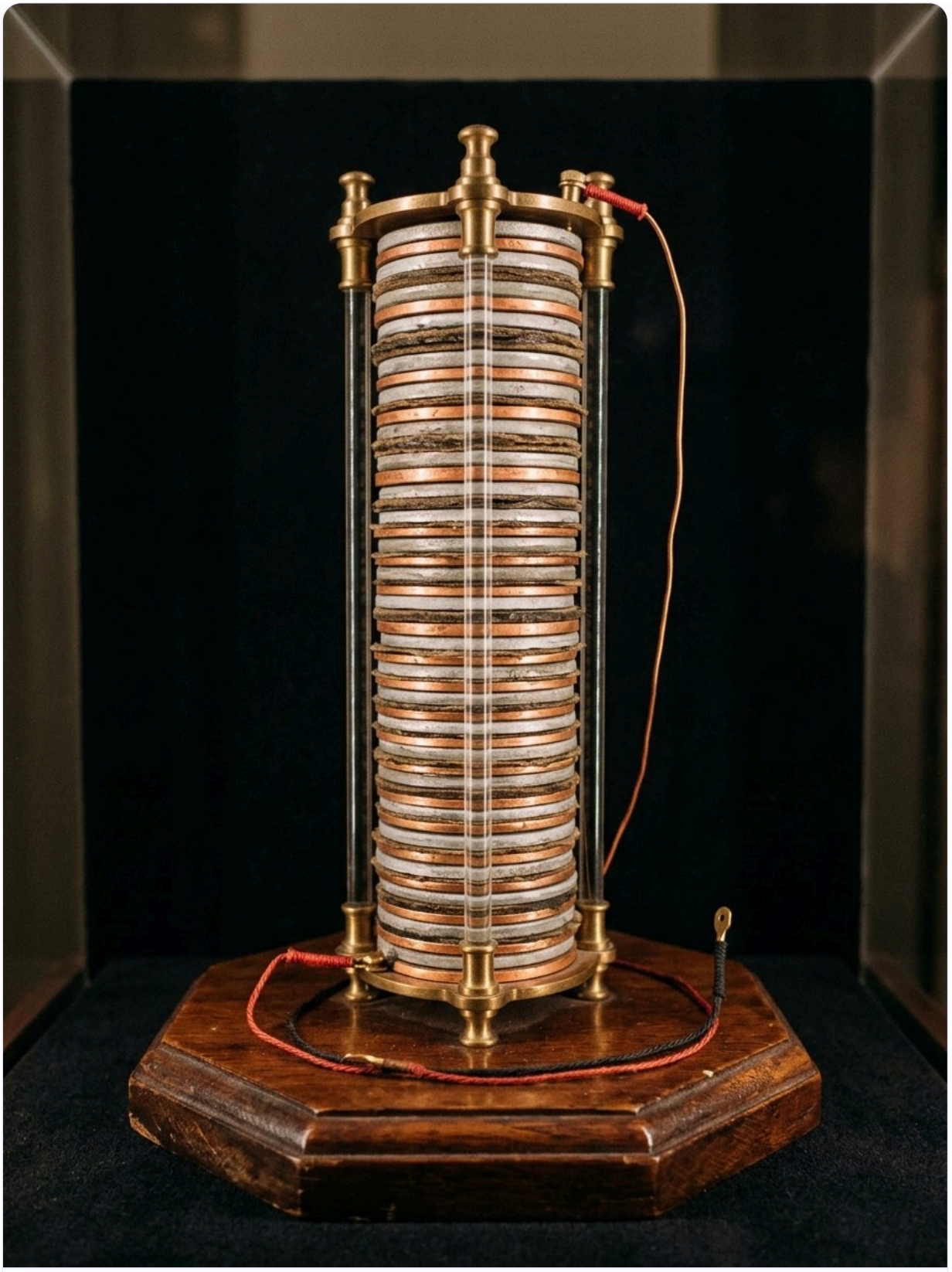
In de jaren 1780 zag de Italiaanse arts Luigi Galvani in Bologna de poten van dode kikkers samentrekken als hij ze met twee verschillende metalen aanraakte. Hij dacht dat de elektriciteit uit het dier kwam: 'dierlijke elektriciteit'. Zijn landgenoot Alessandro Volta, professor in Pavia, twijfelde. Volgens hem kwam de elektriciteit van de twee metalen, en was de kikkerpoot alleen een gevoelige meter.

Om het te bewijzen stapelde Volta schijfjes zink en koper (of zilver) op elkaar, telkens gescheiden door karton dat in zout water gedrenkt was. Hoe hoger de stapel, hoe sterker de schok. In een brief van maart 1800 aan de Royal Society in Londen beschreef hij zijn **zuil**: de eerste bron die een aanhoudende elektrische stroom leverde. De eenheid van spanning, de volt, draagt zijn naam.

Volta's zuil had een zwak punt: de spanning zakte snel in, omdat er waterstofbelletjes op het koper ontstonden. In 1836 loste de Brit John Frederic Daniell dat op met twee *verschillende* oplossingen: zink in zinksulfaat, koper in kopersulfaat, gescheiden door een poreuze wand. Zijn cel gaf urenlang een stabiele spanning van ongeveer 1,1 V en werd de werkpaard van de telegrafie. In 1859 bouwde de Fransman Gaston Planté de eerste herlaadbare batterij: de loodaccu, die nog altijd in elke auto zit.

Waarom dit verhaal ertoe doet

Galvani en Volta hadden allebei half gelijk: zenuwen werken wel degelijk met elektrische signalen, maar de stroom in zijn proef kwam uit de metalen. Het debat dwong Volta tot een proef zonder kikker, en die proef werd de batterij.



De zuil van Volta: tientallen kleine cellen in serie, en de eerste stroom die niet ophield.

UITLEG · 16.3

Oxidatie en reductie: elektronen doorgeven

3/21

Oorspronkelijk betekende **oxidatie**: reageren met zuurstof, zoals bij roesten en verbranden (hoofdstuk 14). Kijk je naar de deeltjes, dan zie je wat die reacties gemeen hebben: het metaal geeft elektronen af. Magnesium dat brandt, wordt magnesiumoxide, een zout met Mg^{2+}- en O^{2-}-ionen: elk

magnesiumatoom gaf twee elektronen aan zuurstof. Maar hetzelfde gebeurt ook zonder zuurstof. Leg je een zinkplaatje in een blauwe oplossing van kopersulfaat, dan geeft zink elektronen aan de koperionen: er slaat koper neer op het zink, en de blauwe kleur verdwijnt.

HALFREACTIES EN REDOXREACTIE

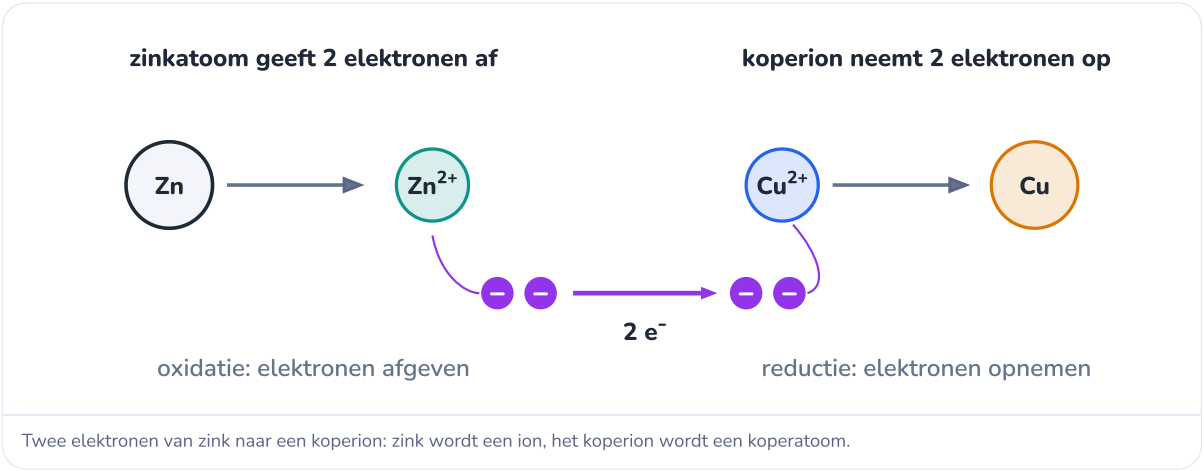
$$\begin{aligned} \text{oxidatie:} &\quad \mathrm{Zn} \longrightarrow \mathrm{Zn^{2+}} + 2\,e^- \\ \text{reductie:} &\quad \mathrm{Cu^{2+}} + 2\,e^- \longrightarrow \mathrm{Cu} \\ \text{samen:} &\quad \mathrm{Zn} + \mathrm{Cu^{2+}} \longrightarrow \mathrm{Zn^{2+}} + \mathrm{Cu} \end{aligned}$$

Het aantal afgegeven elektronen is gelijk aan het aantal opgenomen elektronen: in de totale vergelijking staan ze niet meer.

Oxidatie is elektronen **afgeven**, **reductie** is elektronen **opnemen**. Ze gebeuren altijd samen: een elektron verdwijnt niet, het gaat van het ene deeltje naar het andere. Daarom heet zo'n reactie een **redoxreactie**. De stof die elektronen afgeeft (zink), heet de **reductor**: ze reduceert de andere. De stof die ze opneemt (het koperion), is de **oxidator**.

Reductie: de lading daalt

Een deeltje dat elektronen opneemt, krijgt er negatieve lading bij: zijn lading 'reduceert'. Cu^{2+} wordt Cu (van +2 naar 0), Cl_2 wordt 2 Cl^- (van 0 naar −1). Bij oxidatie stijgt de lading: Zn wordt Zn^{2+}.



Twee elektronen van zink naar een koperion: zink wordt een ion, het koperion wordt een koperatoom.

PROEF · 16.4

Koper uit een blauwe oplossing

4/21

ONDERZOEKSVRAAG

Wat gebeurt er als je een zinkplaatje of een ijzeren spijker in kopersulfaat legt? En een koperplaatje in zinksulfaat?

MATERIAAL

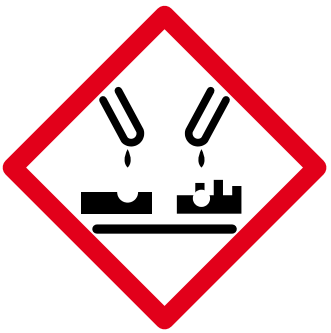
- zinkplaatje, ijzeren spijker, koperplaatje (alle drie geschuurd)
- kopersulfaatoplossing en zinksulfaatoplossing (0,5 mol/L)
- drie reageerbuizen of bekertjes
- thermometer
- veiligheidsbril, handschoenen

Veiligheid

Bril en handschoenen. Kopersulfaat is schadelijk bij inslikken, irriteert de ogen (GHS07) en is giftig voor waterorganismen (GHS09): niet in de gootsteen, maar in het afvalvat voor zware metalen. Zinksulfaat kan ernstig oogletsel geven (GHS05).

kopersulfaat

milieu



zinksulfaat

Kopersulfaat: schadelijk en milieugevaarlijk. Zinksulfaat: gevaar voor de ogen.

WERKWIJZE

1. Giet kopersulfaat in twee buisjes. Meet de temperatuur. Zet in het ene het zinkplaatje, in het andere de spijker.
2. Giet zinksulfaat in het derde buisje en zet er het koperplaatje in.
3. Kijk na 1 minuut, na 10 minuten en na een uur. Meet de temperatuur in het buisje met zink.

WAARNEMING

Het zink en de spijker worden meteen bruinrood bedekt: koper. Na een uur is de blauwe kleur rond het zink veel bleker, en de oplossing is iets opgewarmd. Het koperplaatje in zinksulfaat blijft zoals het was, ook na een dag.

VERKLARING

Zink en ijzer geven elektronen af aan de koperionen: $\mathrm{Zn} + \mathrm{Cu}^{2+} \longrightarrow \mathrm{Zn}^{2+} + \mathrm{Cu}$ en $\mathrm{Fe} + \mathrm{Cu}^{2+} \longrightarrow \mathrm{Fe}^{2+} + \mathrm{Cu}$. Het koper slaat neer, de blauwe koperionen verdwijnen uit de oplossing, en de energie die vrijkomt, wordt warmte. Omgekeerd gebeurt niets: koper geeft zijn elektronen niet af aan zinkionen. Zink en ijzer zijn **onedeler** dan koper: ze geven makkelijker elektronen af. Wat als warmte verloren gaat in het buisje, kan je als elektrische energie oogsten als je oxidatie en reductie uit elkaar haalt: dat is een batterij.

Links zink, rechts een spijker: allebei krijgen ze een koperlaagje.

UITLEG · 16.5

De spanningsreeks: wie geeft zijn elektronen af?

5/21

Door zulke proeven met veel metalen te herhalen, kan je ze rangschikken: van metalen die heel makkelijk elektronen afgeven (**onedel**: lithium, kalium, magnesium, zink) tot metalen die dat bijna niet doen (**edel**: koper, zilver, goud). Dat is de **spanningsreeks**. Waterstof staat erin als ijkpunt.

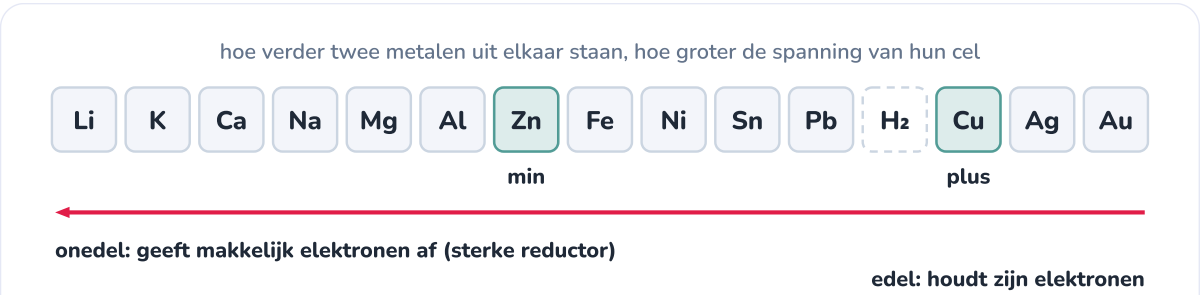
Wat je ermee kan voorspellen:

- Een metaal **reageert met de ionen van een metaal rechts ervan**: zink met koperionen, koper met zilverionen. Omgekeerd niet.
- Metalen links van waterstof lossen op in zuur, met waterstofgas (hoofdstuk 15): $\mathrm{Zn} + 2\,\mathrm{H_3O^+} \longrightarrow \mathrm{Zn^{2+}} + \mathrm{H_2} + 2\,\mathrm{H_2O}$. Koper, zilver en goud niet: daarom blijft goud eeuwenlang blinken.
- In een cel met twee metalen wordt het onedelste de **minpool**. Hoe verder de twee uit elkaar staan, hoe groter de spanning. Hoe groot precies, reken je in E27 met standaardpotentialen.

Lithium staat helemaal links: het geeft zijn elektron nog makkelijker af dan kalium. Daarom haalt een lithiumcel meer dan 3 V, terwijl een cel met zink en koper maar 1,1 V geeft.

Edel betekent: reageert niet

Edel betekent: geeft moeilijk elektronen af. Koper reageert wel, alleen niet met zuren zoals zoutzuur; het wordt groen (kopercarbonaat) op een kerktoren en het reageert met salpeterzuur. Het hangt af van de partner: de reeks vergelijkt telkens twee deeltjes.



De volgorde van de standaardpotentialen. Zink en koper, de metalen van de Daniell-cel, zijn aangeduid.

UITGEWERKT VOORBEELD · 16.6

Redoxreacties opsplitsen in halfreacties

6/21

Splits in halfreacties, duid oxidator en reductor aan, en maak kloppend: (a) aluminiumfolie in een kopersulfaatoplossing; (b) magnesium dat brandt in zuurstof; (c) chloorgas dat door een oplossing van natriumbromide borrelt (de oplossing wordt oranjebruin van broom).

GEGEVEN

- (a) Al en Cu^{2+}
- (b) Mg en O_2, product MgO (Mg^{2+} en O^{2-})
- (c) Cl_2 en Br^-, producten Cl^- en Br_2

GEVRAAGD

- halfreacties, oxidator, reductor, totale vergelijking

OPLOSSING

1 (a) Oxidatie: $\mathrm{Al} \rightarrow \mathrm{Al^{3+}} + 3\,e^-$. Reductie: $\mathrm{Cu^{2+}} + 2\,e^- \rightarrow \mathrm{Cu}$. Evenveel elektronen: het kleinste gemene veelvoud van 3 en 2 is 6. De eerste maal 2, de tweede maal 3: $2\,\mathrm{Al} + 3\,\mathrm{Cu^{2+}} \longrightarrow 2\,\mathrm{Al^{3+}} + 3\,\mathrm{Cu}$. Reductor: Al; oxidator: Cu^{2+}.

2 (b) Oxidatie: $\mathrm{Mg} \rightarrow \mathrm{Mg^{2+}} + 2\,e^-$. Reductie: $\mathrm{O_2} + 4\,e^- \rightarrow 2\,\mathrm{O^{2-}}$. De eerste maal 2: $2\,\mathrm{Mg} + \mathrm{O_2} \longrightarrow 2\,\mathrm{MgO}$. Reductor: Mg; oxidator: O_2.

3 (c) Oxidatie: $2\,\mathrm{Br^-} \rightarrow \mathrm{Br_2} + 2\,e^-$. Reductie: $\mathrm{Cl_2} + 2\,e^- \rightarrow 2\,\mathrm{Cl^-}$. Al evenveel: $\mathrm{Cl_2} + 2\,\mathrm{Br^-} \longrightarrow 2\,\mathrm{Cl^-} + \mathrm{Br_2}$. Reductor: Br^-; oxidator: Cl_2.

ANTWOORD

(a) $2\,\mathrm{Al} + 3\,\mathrm{Cu^{2+}} \longrightarrow 2\,\mathrm{Al^{3+}} + 3\,\mathrm{Cu}$; **(b)** $2\,\mathrm{Mg} + \mathrm{O_2} \longrightarrow 2\,\mathrm{MgO}$; **(c)** $\mathrm{Cl_2} + 2\,\mathrm{Br^-} \longrightarrow 2\,\mathrm{Cl^-} + \mathrm{Br_2}$.

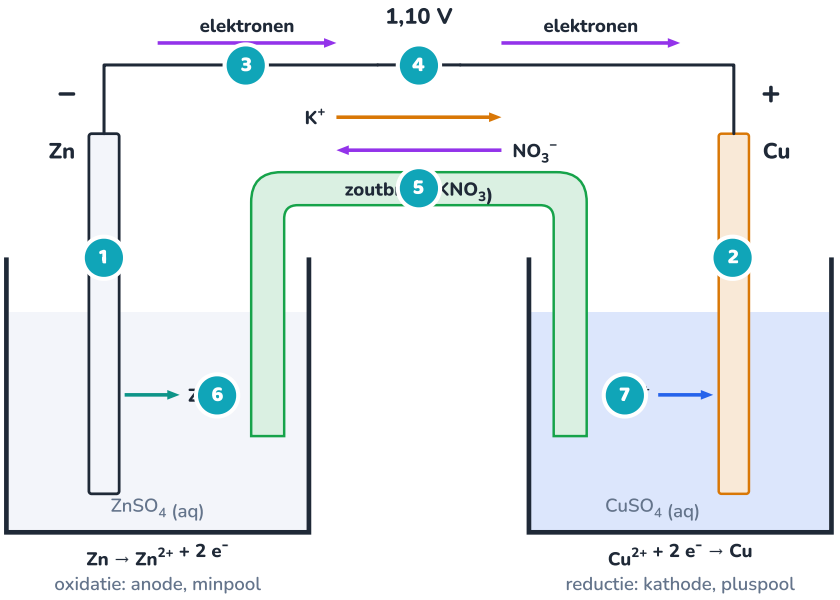
Klopt dit? Controleer twee dingen: de atomen (zoals in hoofdstuk 14) én de lading. Bij (a) is de lading links $3 \cdot (+2) = +6$ en rechts $2 \cdot (+3) = +6$. Klopt de lading niet, dan zijn de elektronen niet in evenwicht. Bij (b) zie je dat verbranden ook redox is; bij (c) dat niet alleen metalen reductor kunnen zijn.

ONTLEED · 16.7

De Daniell-cel

7/21

Zink wil elektronen afgeven aan koperionen. Laat je ze elkaar raken, dan wordt de energie warmte (de proef). Scheid je ze, dan moeten de elektronen door een draad: je hebt een **galvanische cel**. Bij 1 mol/L en 25 °C meet je 1,10 V.



Elektronen door de draad, ionen door de oplossingen en de zoutbrug: samen één gesloten kring.

1 Zinkelektrode: anode (−)
Zinkatomen geven elk twee elektronen af en gaan als Zn^{2+} in oplossing: oxidatie. De elektronen blijven in het metaal achter: het zink wordt de minpool. Het plaatje wordt langzaam dunner.

2 Koperelektrode: kathode (+)
Koperionen uit de oplossing nemen er elk twee elektronen op en slaan neer als koper: reductie. Het plaatje wordt dikker, de oplossing minder blauw.

3 Elektronen door de draad
De elektronen lopen van zink naar koper door de uitwendige kring. De afgesproken stroomzin (F19) is omgekeerd: van + naar −, van koper naar zink.

4 Voltmeter: 1,10 V
Zonder stroom (een voltmeter trekt bijna niets) meet je de volle celspanning. Levert de cel stroom, dan zakt de spanning een beetje door de weerstand in de cel zelf.

5 Zoutbrug
Links komen er positieve zinkionen bij, rechts verdwijnen positieve koperionen. Zonder compensatie zou de lading de reactie meteen stilleggen. Uit de zoutbrug stromen nitraationen naar de zinkkant en kaliumionen naar de koperkant: de kring is gesloten, de oplossingen blijven neutraal.

6 Zinkionen in oplossing
De concentratie Zn^{2+} stijgt tijdens het werken. Zinksulfaat doet zelf niet mee: het zorgt alleen dat de oplossing geleidt.

7 Koperionen naar de elektrode
Als de koperionen op zijn, is de cel leeg. Een Daniell-cel is dus niet herlaadbaar zoals ze hier staat, al kan je met een externe spanning de reactie in principe omkeren.

UITLEG · 16.8

Van cel naar batterij: spanning, capaciteit en energie

8/21

Eén **cel** levert een spanning die vastligt door de chemie: 1,1 V voor zink en koper, 2,0 V voor een loodcel, 3,2 V voor lithiumijzerfosfaat. Een **batterij** is een groep cellen. Zet je cellen **in serie**, dan tellen de spanningen op (F20): zes loodcellen maken een 12 V-autoaccu. Zet je ze **parallel**, dan blijft de spanning gelijk en tellen de capaciteiten op. (In de omgangstaal heet een herlaadbare batterij vaak een **accu**.)

De **capaciteit** Q zegt hoeveel lading een batterij kan leveren. Omdat stroom lading per seconde is ($I = Q/t$), meet je lading ook als stroom maal tijd. De handige eenheid is het **ampère-uur**: 1 Ah is 1 A gedurende een uur, of $1\,\text{A} \cdot 3600\,\text{s} = 3600\,\text{C}$. Kleine batterijen staan in mAh.

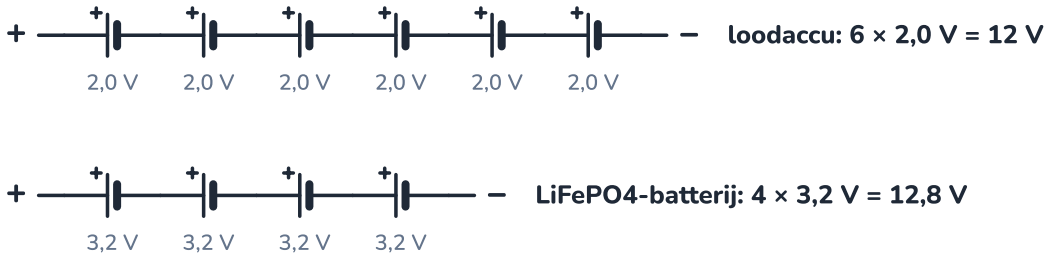
CAPACITEIT EN ENERGIE

$$Q = I \cdot t \qquad E = U \cdot Q = U \cdot I \cdot t \qquad 1\,\text{Ah} = 3600\,\text{C} \qquad 1\,\text{Wh} = 3600\,\text{J}$$

Q in Ah, I in A, t in h, U de (gemiddelde) spanning in V, E in Wh. Omdat de spanning tijdens het ontladen daalt, reken je met de nominale spanning, een gemiddelde.

Twee batterijen met evenveel ampère-uur kunnen dus heel verschillende energie bevatten: 2,0 Ah bij 1,2 V (een NiMH-cel) is 2,4 Wh, 2,0 Ah bij 12,8 V is 25,6 Wh. Vergelijk batterijen daarom in **wattuur**.

De **C-rate** zegt hoe snel je laadt of ontlaadt, ten opzichte van de capaciteit: 1 C is de stroom die de batterij in één uur leeg maakt. Voor een batterij van 120 Ah is 1 C gelijk aan 120 A, en 0,2 C aan 24 A (vijf uur). Hoe groter de C-rate, hoe minder capaciteit je er meestal uithaalt, vooral bij loodaccu's: die worden daarom opgegeven bij een ontlading van 20 uur.



Een '12 V-accu' is zes loodcellen in serie; een LiFePO4-batterij van 12,8 V heeft er vier.

UITGEWERKT VOORBEELD · 16.9

Hoeveel energie zit er in je telefoon?

9/21

Op een smartphonebatterij staat 3,85 V en 4500 mAh. (a) Hoeveel energie bevat ze, in Wh en in kJ? (b) De telefoon verbruikt gemiddeld 1,5 W. Hoe lang gaat hij mee? (c) Hoe lang laadt hij minstens met een lader die 15 W in de batterij stopt?

GEGEVEN

- $U = 3{,}85\,\text{V}$, $Q = 4500\,\text{mAh} = 4{,}5\,\text{Ah}$
- $P = 1{,}5\,\text{W}$ verbruik; 15 W laden

GEVRAAGD

- E
- gebruiksduur
- laadtijd

OPLOSSING

1. (a) $E = U \cdot Q = 3{,}85\,\text{V} \cdot 4{,}5\,\text{Ah} = 17{,}3\,\text{Wh}$. In joule: $17{,}3 \cdot 3600 = 62\,400\,\text{J} \approx 62\,\text{kJ}$.
2. (b) $t = E/P = 17{,}3\,\text{Wh}/1{,}5\,\text{W} = 11{,}5\,\text{h}$.
3. (c) $t = 17{,}3/15 = 1{,}15\,\text{h}$, ongeveer 70 minuten. In werkelijkheid duurt het langer: de laatste 20 % laadt een lithiumbatterij met een steeds kleinere stroom (verderop), en er gaat ook energie verloren als warmte.

ANTWOORD

17,3 Wh (62 kJ); ongeveer 11,5 uur gebruik; minstens 70 minuten laden.

Klopt dit? Ter vergelijking: 62 kJ is ongeveer de energie om een liter water 15 °C op te warmen, of de chemische energie in anderhalve gram benzine. Een batterij slaat dus weinig energie op per kilogram, maar levert ze wel meteen als elektrische energie.

UITGEWERKT VOORBEELD · 16.10

Rekenen met lading: hoeveel zink kost een ampère-uur?

10/21

Een Daniell-cel levert 5,0 uur lang een stroom van 0,10 A. Hoeveel gram zink lost op, en hoeveel gram koper slaat neer? Hoeveel zink kost dus één ampère-uur?

GEGEVEN

- $I = 0{,}10\ \text{A}$, $t = 5{,}0\ \text{h}$
- lading van één elektron: $e = 1{,}602 \cdot 10^{-19}\ \text{C}$; $N_A = 6{,}022 \cdot 10^{23}\ \text{mol}^{-1}$
- $M(\text{Zn}) = 65{,}38\ \text{g/mol}$, $M(\text{Cu}) = 63{,}55\ \text{g/mol}$

GEVRAAGD

- $m(\text{Zn})$, $m(\text{Cu})$, zink per Ah

OPLOSSING

1. Lading: $Q = I \cdot t = 0{,}10\ \text{A} \cdot 18\,000\ \text{s} = 1800\ \text{C}$ (of 0,50 Ah).
2. De lading van één mol elektronen heet de **constante van Faraday**: $F = N_A \cdot e = 6{,}022 \cdot 10^{23} \cdot 1{,}602 \cdot 10^{-19} = 96\,485\ \text{C/mol}$.
3. Mol elektronen: $n(e^-) = Q/F = 1800/96\,485 = 0{,}018\,66\ \text{mol}$.
4. Per zinkatoom gaan er twee elektronen: $n(\text{Zn}) = 0{,}018\,66/2 = 0{,}009\,33\ \text{mol}$, dus $m = 0{,}009\,33 \cdot 65{,}38 = 0{,}61\ \text{g}$ zink. Koper evenveel mol: $0{,}009\,33 \cdot 63{,}55 = 0{,}59\ \text{g}$.
5. Per ampère-uur: $0{,}61/0{,}50 = 1{,}22\ \text{g}$ zink.

ANTWOORD

0,61 g zink lost op, 0,59 g koper slaat neer; één ampère-uur kost 1,22 g zink.

Klopt dit? Zo zie je waarom lithium zo gegeerd is: één mol lithium (6,94 g) levert één mol elektronen, dus 1 Ah kost maar 0,26 g lithium, bijna vijf keer minder dan zink, en bij een drie keer hogere spanning. Een mol elektronen is $96\,485/3600 = 26{,}8\ \text{Ah}$.

UITLEG · 16.11

Batterijtypes vergeleken

11/21

Een batterij die je één keer gebruikt, heet een **primaire** cel; een herlaadbare een **secundaire**. Bij het herladen duw je met een uitwendige spanning de elektronen terug: de reactie loopt in omgekeerde richting. Dat lukt alleen als de producten op de elektroden blijven zitten in een vorm die terug kan.

type	chemie (min / plus)	nominale spanning per cel	energiedichtheid (Wh/kg)	herlaadbaar	typisch gebruik
alkaline	zink / mangaandioxide, kaliloog	1,5 V	100 – 150 (kleine stroom)	nee	afstandsbediening, klok
NiMH	metaalhydride / nikkeloxidehydroxide	1,2 V	60 – 120	ja, ± 500 – 1000 cycli	oplaadbare AA, fototoestel
lood-zuur	lood / looddioxide, zwavelzuur	2,0 V	30 – 40	ja, ± 300 – 800 cycli	startaccu auto, noodstroom

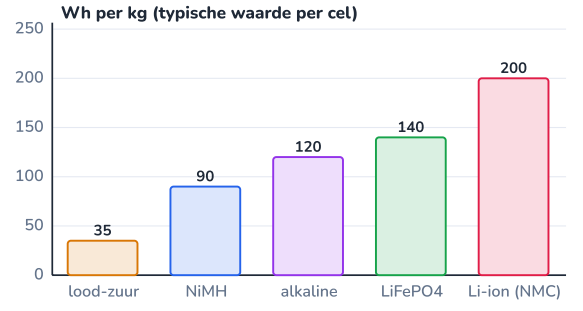
type	chemie (min / plus)	nominale spanning per cel	energiedichtheid (Wh/kg)	herlaadbaar	typisch gebruik
lithium-ion (NMC)	grafiet / nikkel-mangaan-kobaltoxide	3,6 – 3,7 V	150 – 250	ja, ± 500 – 2000 cycli	smartphone, laptop, elektrische auto
LiFePO4 (LFP)	grafiet / lithiumijzerfosfaat	3,2 V	90 – 180	ja, ± 2000 – 6000 cycli	thuisbatterij, zonnesysteem, bus

Orde van grootte per cel. Een batterijpakket, met behuizing, bekabeling en elektronica, haalt minder. Het aantal cycli hangt sterk af van hoe diep je elke keer ontlaadt, van de stroom en van de temperatuur.

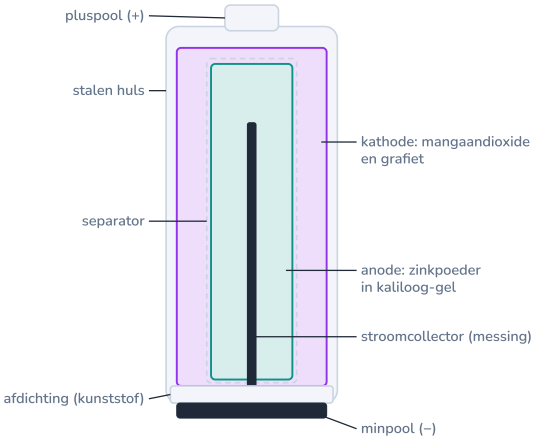
- **Alkaline**: goedkoop, lang houdbaar, maar niet herlaadbaar en zwak bij grote stromen.
- **NiMH**: de herlaadbare AA; 1,2 V in plaats van 1,5 V, maar die spanning blijft lang gelijk.
- **Lood-zuur**: zwaar, maar goedkoop, robuust en goed recycleerbaar (meer dan 95 % van het lood wordt teruggewonnen). Houdt niet van diep ontladen en van lang halfvol staan: dan vormen zich harde loodsulfaatkristallen.
- **Lithium-ion**: de hoogste energiedichtheid; in telefoons, laptops en auto's. De kathode met kobalt en nikkel is duur en kan bij oververhitting zuurstof vrijgeven.
- **LiFePO4**: iets minder energie per kilogram, maar thermisch veel stabieler, met veel meer cycli en zonder kobalt of nikkel. Populair voor vaste opslag.

Waarom geen benzine?

Benzine bevat ongeveer 12 000 Wh chemische energie per kilogram, zo'n vijftig keer meer dan een lithium-ionbatterij. Maar een verbrandingsmotor zet daar maar zo'n 25 à 35 % van om in arbeid, en verbranden is eenrichtingsverkeer. Een elektromotor haalt meer dan 90 % uit de energie van een batterij, en de batterij kan honderden keren opnieuw.



Typische waarden per cel. Hoe hoger, hoe lichter de batterij voor dezelfde energie.



In een AA-batterij zit de pluspool aan de stalen huls en de minpool aan de stroomcollector in het midden.

VERHAAL · 16.12

Een batterij met een Nobelprijs

12/21

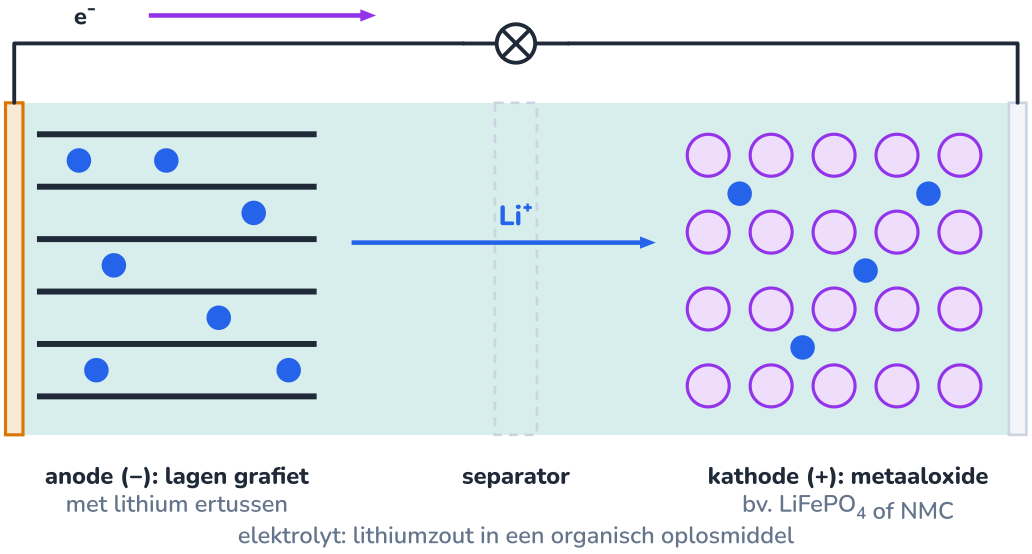
In de jaren 1970 zocht de Brits-Amerikaanse chemicus Stanley Whittingham, in het labo van een oliebedrijf, naar een batterij voor de tijd na de olie. Hij ontdekte dat lithiumionen tussen de lagen van een titaandisulfide kunnen schuiven, als boeken tussen de planken van een kast. John Goodenough, toen in

Oxford, vond in 1980 dat kobaltoxide dat ook kan, en een veel hogere spanning geeft. De Japanse ingenieur Akira Yoshino verving het gevaarlijke lithiummetaal aan de andere kant door koolstof, en in 1991 bracht Sony de eerste lithium-ionbatterij op de markt. In 2019 kregen de drie samen de Nobelprijs voor scheikunde. Goodenough was toen 97, de oudste laureaat ooit.

Zo werkt ze. Aan de minkant zit **grafiet**: lagen koolstofatomen waartussen lithiumatomen passen. Aan de pluskant zit een **metaaloxide** (of een fosfaat) met ruimte voor lithiumionen. Ertussen zit een **elektrolyt**, een lithiumzout opgelost in een organisch oplosmiddel, en een poreuze kunststoffolie, de **separator**, die de twee kanten gescheiden houdt maar ionen doorlaat.

- **Ontladen**: in het grafiet geeft lithium zijn elektron af (oxidatie). Het elektron gaat door de uitwendige kring, het lithiumion door de elektrolyt naar de kathode. Daar komen ze weer samen (reductie). Voor een LiFePO4-cel: $\mathrm{LiC_6} \longrightarrow \mathrm{C_6} + \mathrm{Li^+} + \mathrm{e^-}$ en $\mathrm{FePO_4} + \mathrm{Li^+} + \mathrm{e^-} \longrightarrow \mathrm{LiFePO_4}$.
- **Laden**: de lader duwt elektronen en ionen terug naar het grafiet.

Er wordt dus niets opgebruikt: dezelfde lithiumionen schommelen heen en weer. Daarom noemt men dit soms een schommelstoelbatterij. Hoe de spanning van zo'n cel precies volgt uit de gebruikte stoffen, en waarom ze veroudert, zie je in E27.



Tijdens het ontladen: lithiumionen door de elektrolyt naar rechts, elektronen door de lamp.

UITLEG · 16.13

Laden en ontladen

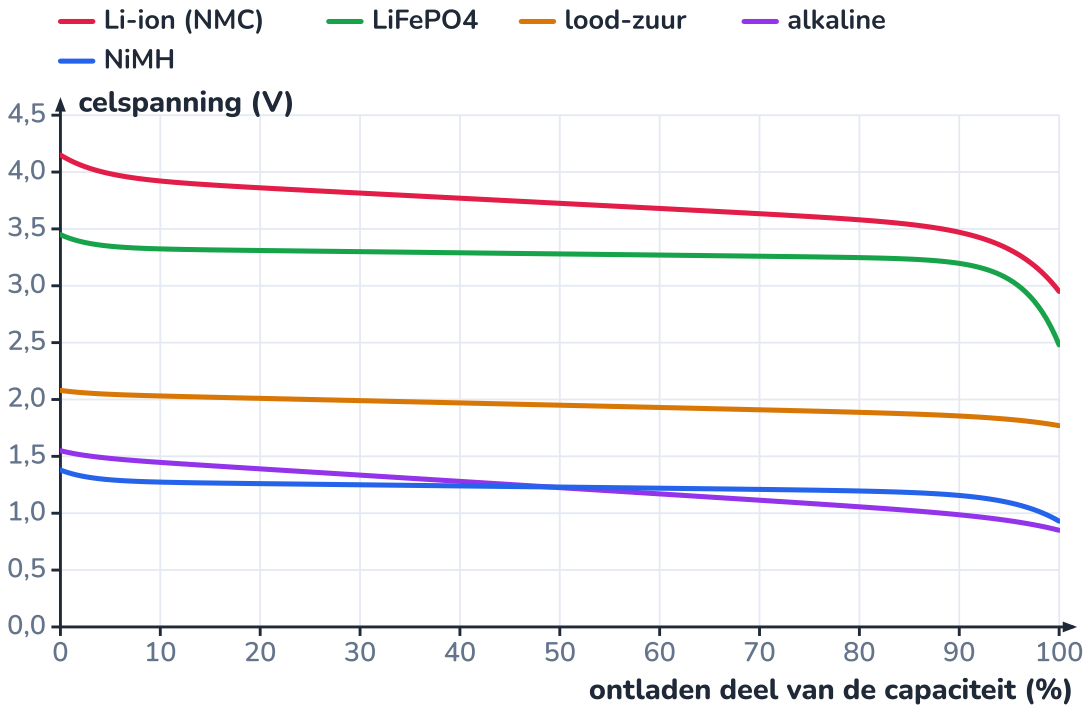
13/21

Tijdens het ontladen daalt de spanning van elke cel. Hoe, hangt af van de chemie: de **ontlaadcurve**. Een LiFePO4-cel blijft het grootste deel van de tijd rond 3,2 à 3,3 V en zakt pas helemaal op het einde; een alkalinebatterij daalt geleidelijk. Onder een bepaalde **eindspanning** mag je niet verder: een lithiumcel die te diep ontladen wordt, raakt beschadigd.

- **Laadtoestand** (SoC, state of charge): hoeveel procent van de capaciteit er nog in zit. Bij een loodaccu lees je ze ruwweg af aan de rustspanning (ongeveer 12,7 V vol, 12,2 V halfvol). Bij LiFePO4 is de curve te vlak: daar tel je de lading die in- en uitgaat, met een **coulombteller**.
- **Ontlaaddiepte** (DoD): welk deel je bij elke cyclus gebruikt. Hoe dieper, hoe minder cycli de batterij meegaat. Een loodaccu houdt het veel langer uit als je hem niet dieper dan 50 % ontlaadt; een LiFePO4-batterij verdraagt 80 % en meer.

- **Laden**: een lithiumcel laad je eerst met een constante stroom tot haar maximumspanning (4,2 V voor NMC, ongeveer 3,6 V voor LiFePO4), en dan met die constante spanning terwijl de stroom daalt. Een loodaccu krijgt na het volladen een lagere *druppelspanning* om vol te blijven.
- **Rendement**: van de energie die je erin stopt, krijg je er bij lithium ongeveer 95 % terug, bij lood ongeveer 80 %. De rest wordt warmte.
- **BMS** (battery management system): de elektronica in een lithiumbatterij die elke cel bewaakt (spanning, stroom, temperatuur), laden en ontladen stopt buiten de grenzen, en de cellen gelijk houdt.

> ⚠ **Koud laden**
>
> Lithiumcellen mogen onder 0 °C niet geladen worden: het lithium zet zich dan als metaal af op het grafiet, wat de cel blijvend beschadigt en kortsluiting kan veroorzaken. Ontladen mag wel in de kou, al levert de cel dan minder. Voor een station in een koud klimaat betekent dat: de batterij binnen, in de verwarmde module.



Vereenvoudigd model met typische waarden per cel; echte curven hangen af van stroom, temperatuur en ouderdom.

CODE · 16.14

Python: een ontlaadtest doorrekenen

14/21

Hoe meet je de capaciteit van een cel? Ontlaad ze met een constante stroom tot de eindspanning, en tel intussen op hoeveel lading en energie eruit kwam: elke minuut $I \cdot \Delta t$ en $U \cdot I \cdot \Delta t$. Zo'n optelling heet **coulombtellen**. Het programma hieronder doet dat voor een NiMH-cel van 2,0 Ah, met het model van de ontlaadcurve in plaats van een echte meting.

Python `ontlaadtest.py` 29 regels

```python
# ontlaadtest.py - een NiMH-cel ontladen met een constante stroom, en tellen wat eruit komt
import math

CAPACITEIT = 2.0      # nominale capaciteit (Ah)
STROOM = 0.50         # ontlaadstroom (A): een kwart van de capaciteit per uur, 0,25 C
EINDSPANNING = 1.00   # hier stopt de test (V)
DT = 60               # een meting per minuut (s)

```

```python

def celspanning(x):
    """Model van de spanning van een NiMH-cel; x = ontladen deel van de capaciteit (0 tot 1)."""
    return 1.28 - 0.10 * x + 0.10 * math.exp(-x / 0.03) - 0.25 * math.exp((x - 1) / 0.05)


lading = 0.0          # geleverde lading (Ah)
energie = 0.0         # geleverde energie (Wh)
t = 0                 # tijd (s)
while True:
    u = celspanning(lading / CAPACITEIT)          # 'meting' van de spanning
    if u < EINDSPANNING:
        break
    lading += STROOM * DT / 3600                  # coulombtellen: I * dt, in Ah
    energie += u * STROOM * DT / 3600             # energie: U * I * dt, in Wh
    t += DT

print(f"duur: {t / 3600:.2f} h")
print(f"geleverde lading: {lading:.3f} Ah")
print(f"geleverde energie: {energie:.3f} Wh")
print(f"gemiddelde spanning: {energie / lading:.3f} V")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
duur: 3.95 h
geleverde lading: 1.975 Ah
geleverde energie: 2.419 Wh
gemiddelde spanning: 1.225 V
```

De cel levert net geen 2,0 Ah: de test stopt bij 1,0 V, iets voor de cel echt leeg is. De gemiddelde spanning, 1,22 V, ligt dicht bij de nominale 1,2 V. Met echte metingen vervang je de functie `celspanning` door de metingen van een multimeter of een microcontroller. In opdracht 16.12 schrijf je een coulombteller voor een hele dag van de klimaatmodule.

UITLEG · 16.15

Veilig met batterijen

15/21

Een batterij bevat veel energie op een kleine plaats, en die energie kan ook in één keer vrijkomen.

- **Kortsluiting**: verbind je de polen rechtstreeks, dan loopt er een enorme stroom. Een autoaccu levert honderden ampère: een ring of een sleutel over de polen wordt gloeiend heet. Leg nooit gereedschap op een accu, en haal bij werk eerst de minpool los.
- **Thermal runaway** bij lithium-ion: door een beschadiging, een productiefout, overladen of oververhitting ontstaat binnenin een kortsluiting. De cel warmt op; boven ongeveer 130 °C smelt de separator, waardoor de kortsluiting groter wordt; de elektrolyt en de kathode ontbinden en geven nog meer warmte, brandbare en giftige gassen, bij NMC ook zuurstof. De reactie versnelt zichzelf en springt over naar de buurcellen. Water helpt maar door af te koelen; het vuur kan uren later opnieuw oplaaien. LiFePO4 is veel stabieler, maar niet onbrandbaar.
- **Loodaccu**: het accuzuur is zwavelzuur van ongeveer 35 %, bijtend (GHS05). Bij het laden, vooral op het einde, ontstaat waterstofgas en zuurstof: explosief. Laad in een verluchte ruimte, zonder vonken.

Veilig gebruik: laad met de juiste lader, op een onbrandbare ondergrond en niet onbewaakt 's nachts naast je bed; gebruik geen batterij die opgezwollen, gedeukt of warm is; bewaar lithiumbatterijen die je lang niet gebruikt halfvol, koel en droog; tape de polen af van batterijen die je inlevert.

Inleveren, niet weggooien

Batterijen horen niet bij het restafval: ze bevatten metalen die je kan terugwinnen, en een lithiumbatterij kan in een vuilniswagen of een sorteercentrum brand veroorzaken. In België zamelt Bebat ze in, in winkels, op scholen en in het recyclagepark.

GHS05

GHS02

apart inleveren

Bijtend accuzuur, ontvlambaar waterstofgas of elektrolyt, en de doorgekruiste container: inleveren.

Een opgezwollen lithiumcel: niet meer laden, niet doorprikken, apart en op een onbrandbare plaats inleveren.

UITLEG · 16.16

Corrosie: een batterij die je niet wil

16/21

IJzer roest alleen als er **water én zuurstof** is: in droge lucht of in zuurstofvrij water nauwelijks. Roesten is een redoxreactie die werkt als een kleine galvanische cel. Onder een waterdruppel wordt het ijzer in het midden, waar weinig zuurstof komt, de **anode**: $\mathrm{Fe} \rightarrow \mathrm{Fe}^{2+} + 2\,e^{-}$. De elektronen lopen door het metaal naar de rand van de druppel, waar veel zuurstof is: de **kathode**, met $\mathrm{O_2} + 2\,\mathrm{H_2O} + 4\,\mathrm{e^-} \longrightarrow 4\,\mathrm{OH^-}$. Waar de ijzer- en hydroxide-ionen elkaar ontmoeten, vormt zich ijzerhydroxide dat verder oxideert tot roest ($\mathrm{Fe_2O_3 \cdot x\,H_2O}$). Zout water geleidt beter en maakt roesten veel sneller: denk aan autos die met strooizout in aanraking komen, of aan de kust.

Contactcorrosie. Raken twee verschillende metalen elkaar in een natte omgeving, dan vormen ze een cel: het onedelste wordt de anode en lost op. Een roestvaststalen schroef in een aluminium profiel laat het aluminium rond de schroef wegvreten. Daarom zitten de RVS-schroeven van het kasframe met **kunststof ringen** in het aluminium (F10): geen contact, geen cel.

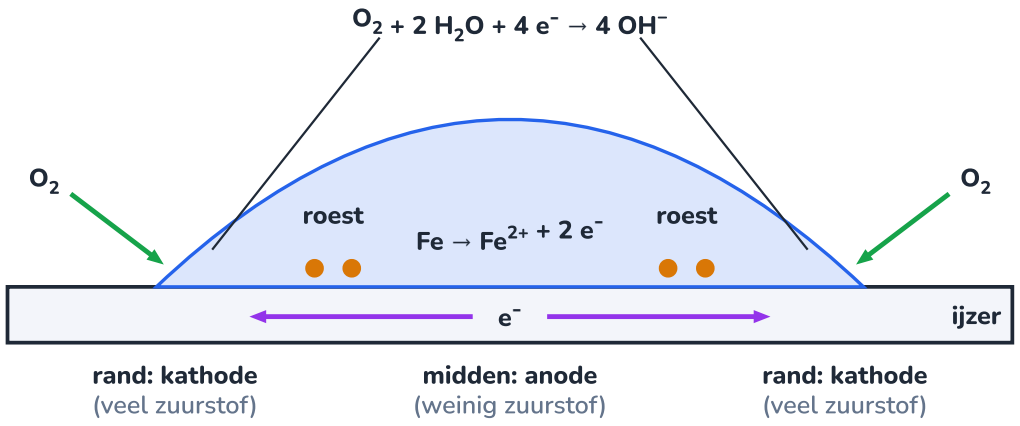
Beschermen kan op drie manieren:

- **Afschermen**: verf, kunststof of olie houdt water en zuurstof weg. Een kras en het roest begint.
- **Verzinken**: staal in een bad van gesmolten zink dompelen, zodat er een laag van enkele tientallen micrometer zink op komt. Zink schermt af, maar beschermt ook bij een kras: zink is onedeler dan ijzer en offert zichzelf op. Het zink oxideert, het ijzer blijft gespaard.
- **Opofferingsanode**: een blok zink of magnesium, elektrisch verbonden met het staal, oxideert in zijn plaats (kathodische bescherming). Zo worden scheepsrompen, havenpalen, ondergrondse leidingen en de binnenkant van boilers beschermd. Het blok moet je af en toe vervangen.

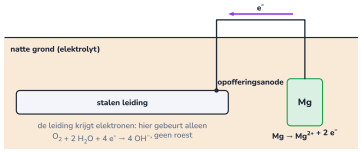
Roestvast staal bevat minstens 10,5 % chroom, dat een dun, zelfherstellend laagje chroomoxide vormt; aluminium beschermt zichzelf met een oxidelaagje (F10).

Een laagje van een edeler metaal is altijd beter

Conservenblik is staal met een laagje tin. Tin is edeler dan ijzer. Zolang het laagje heel is, beschermt het. Maar bij een kras wordt het ijzer de anode van een cel met het tin: het roest dan *sneller* dan zonder laagje. Bij verzinkt staal is het net omgekeerd.



Het ijzer lost op in het midden van de druppel; de roest vormt zich in een ring waar de ionen elkaar ontmoeten.



Het magnesium geeft de elektronen; op het staal gebeurt alleen de reductie van zuurstof.

UITGEWERKT VOORBEELD · 16.17

Een magnesiumanode in een boiler

17/21

In een elektrische boiler van 80 L zit een magnesiumstaaf als opofferingsanode. Gemiddeld loopt er een beschermstroom van 10 mA. Hoeveel magnesium verdwijnt er per jaar? De staaf weegt 450 g; hij moet vervangen worden als er nog een derde over is. Na hoeveel jaar?

GEGEVEN

- $I = 0{,}010\ \text{A}$, $t = 1$ jaar $= 365{,}25 \cdot 24 \cdot 3600\ \text{s}$
- $\text{Mg} \rightarrow \text{Mg}^{2+} + 2\,e^-$; $M(\text{Mg}) = 24{,}31\ \text{g/mol}$; $F = 96\,485\ \text{C/mol}$

GEVRAAGD

- $m(\text{Mg})$ per jaar; levensduur

OPLOSSING

1. Lading in een jaar: $Q = 0{,}010 \cdot 3{,}156 \cdot 10^7 = 3{,}16 \cdot 10^5\ \text{C}$.
2. Mol elektronen: $3{,}16 \cdot 10^5 / 96\,485 = 3{,}27\ \text{mol}$; mol magnesium: $3{,}27/2 = 1{,}64\ \text{mol}$.
3. Massa: $1{,}64 \cdot 24{,}31 = 40\ \text{g}$ per jaar.
4. Er mag $\frac{2}{3} \cdot 450 = 300\ \text{g}$ verdwijnen: $300/40 = 7{,}5$ jaar.

ANTWOORD

Ongeveer 40 g magnesium per jaar; na zo'n 7 à 8 jaar moet de staaf vervangen worden.

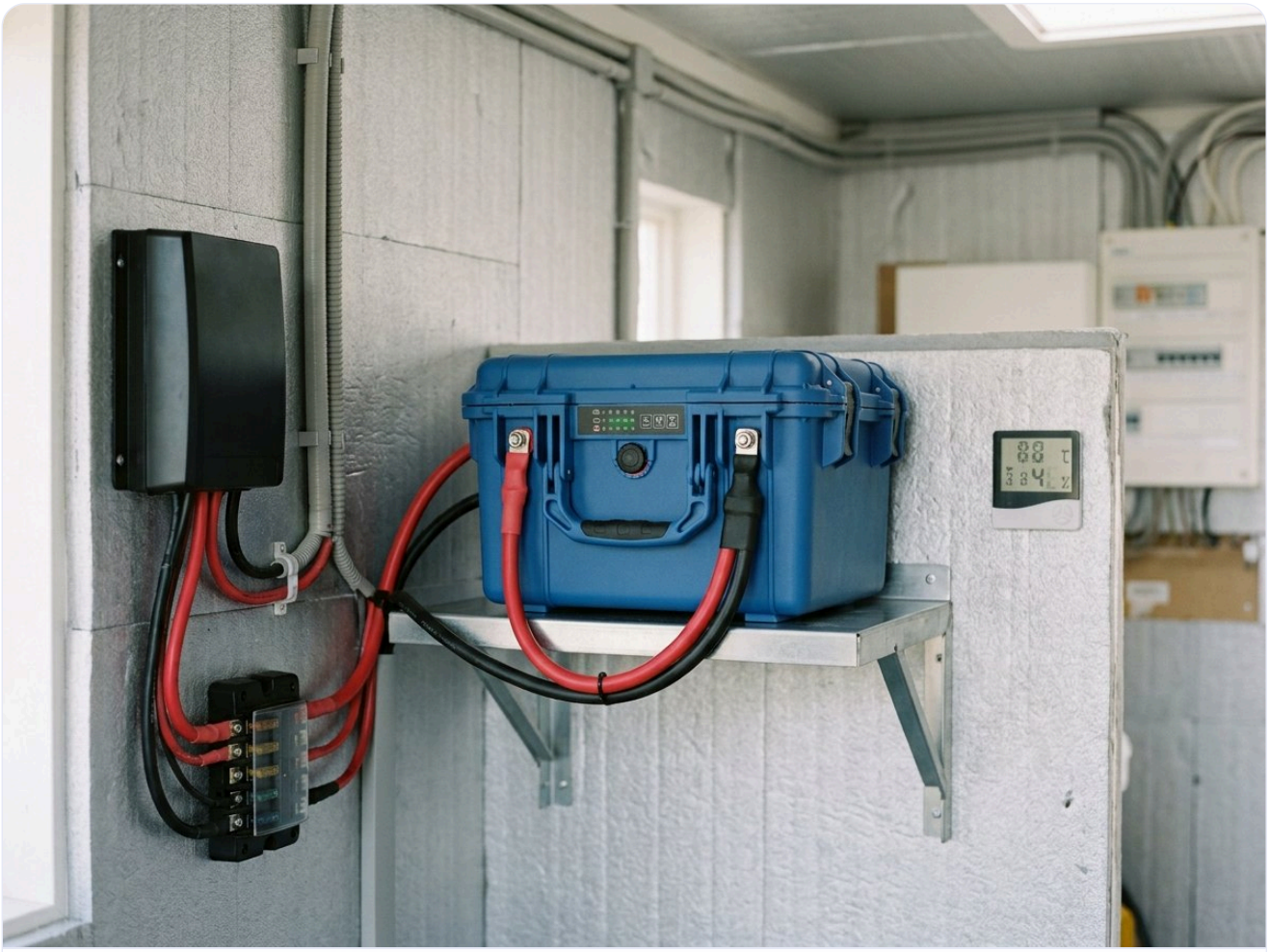
Klopt dit? Dat is een ideale berekening: in werkelijkheid corrodeert magnesium ook gedeeltelijk 'voor niets', en kalkrijk of zacht water verandert de stroom. Fabrikanten raden daarom aan de anode om de paar jaar na te kijken. Met zink in plaats van magnesium zou het per jaar 107 g zijn: zink is zwaarder per mol elektronen.

UITGEWERKT VOORBEELD · 16.18

Habitat: loodaccu of LiFePO4 voor de klimaatmodule?

18/21

De klimaatmodule heeft 's nachts en op donkere dagen **1,2 kWh bruikbare energie** nodig (hoofdstuk 9). Vergelijk een loodaccu (AGM, 12 V) en een LiFePO4-batterij (12,8 V) op nominale capaciteit, massa, laadverliezen en levensduur. Neem één cyclus per dag. Rekenwaarden: lood 50 % ontlaaddiepte, 35 Wh/kg, 80 % rendement, 500 cycli; LiFePO4 80 % ontlaaddiepte, 110 Wh/kg (pakket), 95 % rendement, 3000 cycli.



De batterij van de klimaatmodule: binnen, boven 0 °C, met zekering en BMS.

GEGEVEN

- $E_{\text{bruikbaar}} = 1{,}2\,\text{kWh}$
- lood: DoD 50 %, 35 Wh/kg, η = 80 %, 500 cycli; LiFePO4: DoD 80 %, 110 Wh/kg, η = 95 %, 3000 cycli

GEVRAAGD

- nominale energie en capaciteit
- massa
- energie om elke dag te laden
- levensduur

OPLOSSING

1. Nominaal nodig: lood $1{,}2/0{,}50 = 2{,}4\,\text{kWh}$, dus $2400\,\text{Wh}/12\,\text{V} = 200\,\text{Ah}$. LiFePO4: $1{,}2/0{,}80 = 1{,}5\,\text{kWh}$, dus $1500/12{,}8 = 117\,\text{Ah}$: kies 120 Ah (1,54 kWh).

2. Massa: lood $2400/35 = 69\,\text{kg}$; LiFePO4 $1536/110 = 14\,\text{kg}$.

3. Laden: om 1,2 kWh terug te krijgen, moet je er bij lood $1{,}2/0{,}80 = 1{,}50\,\text{kWh}$ in stoppen, bij LiFePO4 $1{,}2/0{,}95 = 1{,}26\,\text{kWh}$. Elke dag 0,24 kWh meer zonne-energie voor lood.

4. Levensduur bij een cyclus per dag: lood $500/365 = 1{,}4$ jaar; LiFePO4 $3000/365 = 8{,}2$ jaar.

5. Temperatuur: LiFePO4 mag niet onder 0 °C geladen worden: de batterij komt binnen, in de verwarmde module. Een BMS beschermt de cellen. Lood verdraagt kou beter bij het laden, maar levert in de kou minder.

ANTWOORD

Habitat kiest LiFePO4, 12,8 V en 120 Ah (1,54 kWh nominaal, ongeveer 1,2 kWh bruikbaar): vijf keer lichter dan een loodaccu met dezelfde bruikbare energie, 16 % minder laadverlies, en ongeveer zes keer zo lang bruikbaar. De aankoop is duurder, maar per geleverde kWh over de levensduur meestal goedkoper.

Klopt dit? De batterijkar van hoofdstuk 6 en 8 woog met batterijen 80 kg: ongeveer wat een loodaccu van 2,4 kWh met een kar erbij weegt. Met LiFePO4 draag je de batterij onder de arm naar binnen. Of 1,2 kWh genoeg is voor meerdere donkere dagen na elkaar, beslis je in hoofdstuk 20: dan kan de capaciteit nog groeien, maar het type ligt vast.

CODE · 16.19

C++: de laadtoestand bijhouden

19/21

Omdat de spanning van LiFePO4 zo vlak loopt, schat de microcontroller de laadtoestand door lading te tellen, net als de ontlaadtest. Een stroomsensor (hier een ACS712 voor ±20 A) geeft 2,50 V bij 0 A en 0,100 V per ampère. Dat bereik volstaat zolang de laadregelaar hoogstens 20 A levert, zoals hier; met de grotere zonnepanelen van hoofdstuk 20 loopt de laadstroom op een heldere middag op tot ±74 A, en in hoofdstuk 30 neemt een shunt met een INA226 het meten over. Een spanningsdeler (F20, hoofdstuk 22) brengt de batterijspanning onder 5 V. Elke seconde trekt de code $I \cdot \Delta t$ af van de lading. Omdat elke meting een kleine fout heeft, loopt zo'n teller weg: daarom zet de code de laadtoestand terug op 100 % zodra de batterij echt vol is (hoge spanning, bijna geen laadstroom meer).

C++ (Arduino) **laadtoestand.ino** 46 regels

```cpp
// Laadtoestand (SoC) van de LiFePO4-batterij van de klimaatmodule schatten door lading te tellen (Arduino Mega 2560)
// Stroomsensor ACS712-20A op A4: 2,50 V bij 0 A, 0,100 V per ampere (positief = ontladen).
// Spanningsdeler 100 kOhm / 33 kOhm op A3 voor de batterijspanning (tot 20 V). Bord: Arduino Mega 2560.
const float CAPACITEIT_AH = 120.0;
const float NUL_V = 2.50;
const float V_PER_A = 0.100;
const float DELER = (100.0 + 33.0) / 33.0;
const unsigned long DT_MS = 1000;         // elke seconde

float soc = 100.0;                        // laadtoestand in procent (start: vol)
unsigned long vorige = 0;

float spanningOp(int pin) {
  long som = 0;
  for (int i = 0; i < 16; i++) {          // 16 metingen middelen tegen ruis
    som += analogRead(pin);
  }
  return som / 16.0 * 5.0 / 1023.0;
}

void setup() {
  Serial.begin(9600);
```

```
}

void loop() {
  if (millis() - vorige >= DT_MS) {
    vorige += DT_MS;
    float stroom = (spanningOp(A4) - NUL_V) / V_PER_A;        // A, positief bij ontladen
    float uBat = spanningOp(A3) * DELER;                      // V
    float dtUur = (DT_MS / 1000.0) / 3600.0;
    soc -= stroom * dtUur / CAPACITEIT_AH * 100.0;            // coulombtellen
    if (uBat >= 14.2 && stroom < 0.0 && stroom > -2.0) {      // bijna vol: opnieuw ijken op 100 %
      soc = 100.0;
    }
    soc = constrain(soc, 0.0, 100.0);
    Serial.print(uBat, 2);
    Serial.print(" V  ");
    Serial.print(stroom, 2);
    Serial.print(" A  SoC ");
    Serial.print(soc, 1);
    Serial.println(" %");
    if (soc < 20.0) {
      Serial.println("Batterij bijna op de ondergrens van 20 %: niet-dringende verbruikers uit");
    }
  }
}
```

Een stap van de ADC is $5/1023 = 4{,}9\,\text{mV}$, dus 0,05 A: bij een kleine stroom is de meting grof. Door 16 metingen te middelen en elke dag opnieuw te ijken op 'vol' blijft de fout beperkt. De grens van 20 % komt uit de ontlaaddiepte van 80 %: daaronder schakelt de module wat niet dringend is uit. In het integratieproject (hoofdstuk 40) wordt dit de energiebewaking van de klimaatmodule.

NASLAG · 16.20

Overzicht: redox en batterijen

20/21

oxidatie (afgeven) en reductie (opnemen)

$\text{Zn} \rightarrow \text{Zn}^{2+} + 2\,e^-$ $\qquad$ $\text{Cu}^{2+} + 2\,e^- \rightarrow \text{Cu}$

capaciteit en energie

$$Q = I \cdot t \qquad E = U \cdot Q$$

symbool	betekenis	eenheid
Q	lading, capaciteit	Ah (1 Ah = 3600 C)
E	energie	Wh (1 Wh = 3600 J)
U	nominale spanning	V

lading en mol elektronen

$$n(e^-) = \frac{Q}{F} \qquad F = 96\,485\,\text{C/mol} = 26{,}8\,\text{Ah}$$

symbool	betekenis	eenheid
F	constante van Faraday	C/mol

bruikbare energie en ontlaaddiepte

$$E_{\text{nominaal}} = \frac{E_{\text{bruikbaar}}}{\text{DoD}}$$

symbool	betekenis	eenheid
DoD	ontlaaddiepte (als decimaal getal)	–

type	chemie (min / plus)	nominale spanning per cel	energiedichtheid (Wh/kg)	herlaadbaar	typisch gebruik
alkaline	zink / mangaandioxide, kaliloog	1,5 V	100 – 150 (kleine stroom)	nee	afstandsbediening, klok
NiMH	metaalhydride / nikkeloxidehydroxide	1,2 V	60 – 120	ja, ± 500 – 1000 cycli	oplaadbare AA, fototoestel

type	chemie (min / plus)	nominale spanning per cel	energiedichtheid (Wh/kg)	herlaadbaar	typisch gebruik
lood-zuur	lood / looddioxide, zwavelzuur	2,0 V	30 – 40	ja, ± 300 – 800 cycli	startaccu auto, noodstroom
lithium-ion (NMC)	grafiet / nikkel-mangaan-kobaltoxide	3,6 – 3,7 V	150 – 250	ja, ± 500 – 2000 cycli	smartphone, laptop, elektrische auto
LiFePO4 (LFP)	grafiet / lithiumijzerfosfaat	3,2 V	90 – 180	ja, ± 2000 – 6000 cycli	thuisbatterij, zonnesysteem, bus

Orde van grootte per cel. Een batterijpakket, met behuizing, bekabeling en elektronica, haalt minder. Het aantal cycli hangt sterk af van hoe diep je elke keer ontlaadt, van de stroom en van de temperatuur.

Habitat: rekenwaarden uit dit hoofdstuk	
batterij klimaatmodule	LiFePO4 12,8 V (4 cellen), 120 Ah = 1,54 kWh nominaal; 80 % ontlaaddiepte → 1,2 kWh bruikbaar; ± 14 kg
rendement, levensduur	± 95 % laden en ontladen; ± 3000 cycli bij 80 %: ± 8 jaar
plaats en bewaking	in de verwarmde module (laden alleen tussen 0 en 45 °C), met BMS; laadtoestand met een coulombteller
alternatief lood-zuur	12 V, 200 Ah (2,4 kWh), ± 69 kg, ± 80 %, ± 500 cycli bij 50 %
kasframe	aluminium met RVS-schroeven en kunststof ringen: geen contactcorrosie

HET GEHEEL · 16.21

Waar redox en batterijen terugkomen

21/21

Redox is overal: in batterijen en brandstofcellen, in roest, in verbranding, in de fotosynthese en in je cellen, waar suiker stap voor stap zijn elektronen afgeeft. De batterij verbindt chemie (M) met elektriciteit (L) en energie (E): hoeveel lading, bij welke spanning, met welk rendement.

F17 Energie uit zon, wind en water
Energie opslaan: batterij, water omhoog, warmtebuffer.

F19 Spanning, stroom en weerstand
Stroom als bewegende lading en de stroomzin van + naar –.

F21 Elektrisch vermogen en energie
Energie in Wh en kWh, vermogen $P = U \cdot I$.

F10 Materialen en hun eigenschappen
Roest, de oxidelaag van aluminium, en de kunststof ringen van het kasframe.

I9 Arbeid, energie en vermogen
De voorlopige batterij van 1,2 kWh en het energiebudget.

I14 Reacties en stoichiometrie
De mol: hier per mol elektronen, met de constante van Faraday.

I15 Zuren, basen en pH
Metalen die in zuur oplossen, en het zwavelzuur van de loodaccu.

I20 Zonne-energie dimensioneren
Hoeveel dagen autonomie, en dus hoe groot de batterij definitief wordt.

I29 Microcontrollers programmeren in C++
De coulombteller op de microcontroller met millis().

I38 Duurzaam ontwerpen
Grondstoffen, levensduur en recyclage van batterijen.

E27 Elektrochemie: batterijen, brandstofcellen en elektrolyse
Standaardpotentialen, de wet van Nernst, lithium-ion in detail, elektrolyse en brandstofcellen.

E24 Vermogenselektronica en elektrische machines
De MPPT-lader tussen zonnepaneel en batterij.

Habitat

Voor Habitat leggen we vast: de klimaatmodule krijgt een LiFePO4-batterij van 12,8 V en 120 Ah (1,54 kWh nominaal, 1,2 kWh bruikbaar bij 80 % ontlaaddiepte, ongeveer 14 kg), met BMS, in de verwarmde module. De laadtoestand volgt uit een coulombteller. In opdracht 16.13 toets je de keuze aan een koude week; in hoofdstuk 20 bepaal je hoeveel dagen autonomie de module nodig heeft en dus de definitieve capaciteit.

SAMENVATTING

In het kort

- Oxidatie is elektronen afgeven, reductie is elektronen opnemen; ze gebeuren altijd samen in een redoxreactie. De reductor geeft, de oxidator neemt.
- Je schrijft een redoxreactie als twee halfreacties met evenveel elektronen; atomen én lading moeten kloppen.
- De spanningsreeks rangschikt metalen van onedel (lithium, zink) tot edel (koper, goud). Een metaal reageert met de ionen van een edeler metaal.
- In een galvanische cel verlopen oxidatie (anode, minpool) en reductie (kathode, pluspool) gescheiden: elektronen gaan door de draad, ionen door de oplossing en de zoutbrug. De Daniell-cel (zink/koper) geeft 1,10 V.
- Capaciteit in Ah (1 Ah = 3600 C), energie $E = U \cdot Q$ in Wh. Eén mol elektronen is $F = 96\,485$ C = 26,8 Ah.
- Batterijtypes verschillen in celspanning (1,2 tot 3,7 V), energiedichtheid (30 tot 250 Wh/kg) en levensduur. Cellen in serie tellen hun spanningen op.
- De ontlaaddiepte bepaalt hoeveel van de nominale capaciteit bruikbaar is en hoe lang de batterij meegaat.
- Lithiumbatterijen kunnen door thermal runaway in brand vliegen; loodaccu's geven bij laden waterstof. Batterijen lever je in.
- Roesten is een kleine galvanische cel. Verf schermt af; zink en magnesium offeren zichzelf op voor staal.
- Voor Habitat: LiFePO4, 12,8 V, 120 Ah, binnen in de module.

BEGRIPPEN

Wat je nu kent

oxidatie
Het afgeven van elektronen door een deeltje, bv. $Zn \rightarrow Zn^{2+} + 2\ e^-$.

reductie
Het opnemen van elektronen door een deeltje, bv. $Cu^{2+} + 2\ e^- \rightarrow Cu$.

redoxreactie
Reactie waarbij elektronen van de ene stof (de reductor) naar de andere (de oxidator) gaan; oxidatie en reductie gebeuren altijd samen.

halfreactie
De helft van een redoxreactie: alleen de oxidatie of alleen de reductie, met de elektronen erbij geschreven.

spanningsreeks
Rangschikking van metalen (en waterstof) volgens hoe makkelijk ze elektronen afgeven; van onedel naar edel.

galvanische cel
Opstelling waarin een redoxreactie in twee gescheiden helften verloopt, zodat de elektronen door een uitwendige kring moeten.

anode
De elektrode waar oxidatie gebeurt; in een galvanische cel de minpool.

kathode
De elektrode waar reductie gebeurt; in een galvanische cel de pluspool.

zoutbrug
Verbinding met een zoutoplossing tussen de twee helften van een cel; de ionen erin sluiten de kring en houden de oplossingen neutraal.

capaciteit
Hoeveel lading een batterij kan leveren, in ampère-uur (Ah): 1 Ah = 3600 C.

energiedichtheid
Energie per kilogram (Wh/kg) of per liter (Wh/L) die een batterij kan opslaan.

constante van Faraday
De lading van één mol elektronen: $F = 96\,485\ \mathrm{C/mol}$, of 26,8 Ah.

ontlaaddiepte
Welk deel van de capaciteit je bij een cyclus gebruikt (Engels: depth of discharge, DoD).

thermal runaway
Zichzelf versnellende opwarming in een batterij: warmte doet reacties sneller gaan, die nog meer warmte maken, tot brand.

corrosie
Aantasting van een metaal door een redoxreactie met zijn omgeving, zoals roesten van ijzer.

opofferingsanode
Blok van een onedeler metaal (zink, magnesium) dat in plaats van het te beschermen metaal oxideert.

HANDBOEK · HOOFDSTUK 17

Polymeren en composieten

Kettingen van duizenden schakels: waarom een PLA-print in de kas doorzakt, en een koolstofvezelfiets niet

Een plastic fles, een 3D-print, een fietsband en de wand van de meetkas zijn allemaal gemaakt van **polymeren**: moleculen van duizenden gelijke schakels. Hoe die ketens aan elkaar hangen, verklaart waarom de ene kunststof smelt en de andere niet, waarom een print bij 60 °C slap wordt, en waarom vezels in een kunststof een fietsframe opleveren dat lichter is dan aluminium. Je leert ook wat 'bioplastic' en 'afbreekbaar' echt betekenen, en wat de cijfers in het recyclagedriehoekje zeggen. Voor Habitat kies je het filament voor onderdelen die in de warme, vochtige kas moeten overleven.

9× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 9 lestijden

M Materie en chemie · D Ontwerpen en maken

Een sensorhouder voor de kas, laag per laag. Maar welk filament houdt het uit in de zon?

Na dit hoofdstuk

- Je kan uitleggen wat monomeren en polymeren zijn, en uit de molaire massa de ketenlengte berekenen.
- Je kan additie- en condensatiepolymerisatie onderscheiden en een stuk van de polymeerketen tekenen uit het monomeer.
- Je kan thermoplasten, thermoharders en elastomeren herkennen en hun gedrag bij verwarmen verklaren met hun structuur.
- Je kan de glasovergangstemperatuur uitleggen en gebruiken om een kunststof te kiezen voor een bepaalde temperatuur.
- Je kan PLA, PETG, ABS en ASA vergelijken voor 3D-printen, en uitleggen waarom een print zwakker is tussen de lagen.
- Je kan de werking van vezelcomposieten uitleggen en met de mengregel hun stijfheid en dichtheid schatten.
- Je kan biogebaseerd en biologisch afbreekbaar uit elkaar houden en uitleggen onder welke voorwaarden PLA composteert.
- Je kan recyclagecodes 1 tot 7 lezen en de kunststofkringloop beschrijven, met de grenzen van recyclage.
- Je kan met een eenvoudig warmtemodel schatten hoe warm een onderdeel in de kas wordt, en een filament kiezen.

UITLEG · 17.1

Wat je al weet, en wat nu komt

In F10 deelde je materialen in families in, en leerde je dat kunststoffen licht, isolerend en vormbaar zijn. In F32 printte je een sensorhouder in PLA. In hoofdstuk 13 en 14 zag je hoe atomen binden en hoe je met formules rekent. Nu kijk je in een kunststof naar binnen: het zijn kettingen, en hun gedrag volgt uit hoe die kettingen liggen en aan elkaar hangen.

wat je al weet	waar	wat er nu bij komt
kunststoffen en composieten als materiaalfamilie; recyclagecodes	F10	waarom kunststoffen zich zo gedragen: ketens
PLA-filament, laaghoogte en vulling	F32	welk filament voor welk onderdeel, en waarom een print zwakke richtingen heeft
atoombindingen, van der Waalskrachten	hoofdstuk 13	reuzenmoleculen en de krachten tussen hun ketens
reactievergelijkingen, molaire massa	hoofdstuk 14	polymerisatie in formules; ketenlengte uit de molaire massa
de kas haalt 32 °C op een zonnige aprildag	F4	hoe warm een onderdeel in de zon wordt, en welk filament dat uithoudt

In dit hoofdstuk

- zie je hoe monomeren tot polymeren koppelen, door additie of condensatie;
- verklaar je waarom thermoplasten smelten en thermoharders niet, en wat de glasovergang is;
- vergelijk je PLA, PETG, ABS en ASA, en ontdek je waarom een print een zwakke richting heeft;
- reken je met de mengregel aan vezelcomposieten van glas, koolstof en vlas;
- zoek je uit wat 'bioplastic', 'afbreekbaar' en de recyclagecodes echt betekenen;
- kies je voor Habitat het filament voor onderdelen in de kas.

Twee klemmen, één zomer in de kas: het zwarte PLA-exemplaar is doorgezakt.

VERHAAL · 17.2

Van Gent naar bakeliet

2/19

Leo Baekeland werd in 1863 in Gent geboren, als zoon van een schoenmaker. Hij studeerde scheikunde aan de Gentse universiteit, promoveerde op zijn eenentwintigste en gaf er les, tot hij in 1889 naar de Verenigde Staten vertrok. Daar vond hij Velox uit, fotopapier dat je bij kunstlicht kon ontwikkelen. In 1899 verkocht hij het aan Eastman Kodak, naar verluidt voor 750 000 dollar: genoeg om nooit meer te moeten werken.

Baekeland richtte in plaats daarvan een labo in bij zijn huis, en zocht een vervanger voor schellak, een natuurlijke hars van schildluizen die gebruikt werd om elektrische draden te isoleren. Hij liet fenol reageren met formaldehyde, twee goedkope grondstoffen (fenol kwam uit koolteer). Anderen hadden dat al geprobeerd, maar kregen een onbruikbare brij. Baekeland leerde de reactie te sturen met druk en temperatuur in een verwarmd vat, en kreeg in 1907 een hard, glanzend materiaal dat elektriciteit isoleerde, niet smolt en nauwelijks brandde. Hij noemde het **bakeliet**: de eerste volledig synthetische kunststof.

Bakeliet werd telefoontoestellen, radiokasten, stopcontacten, schakelaars en pijpen. Het is een **thermoharder**: eens uitgehard, kan je het niet meer smelten. Dat maakte het ideaal voor alles wat warm wordt, en onmogelijk om te recycleren door omsmelten. Beide eigenschappen kom je in dit hoofdstuk tegen.

Waarom dit verhaal ertoe doet

Baekeland was geen toevallige uitvinder: hij zocht systematisch naar de juiste omstandigheden voor een reactie die anderen hadden opgegeven. Dat is ook wat een 3D-printer doet met temperatuur en snelheid: dezelfde kunststof geeft een sterke of een zwakke print.

Bakeliet: glanzend, hard, hittebestendig, en ontworpen door een Gentenaar.

UITLEG · 17.3

Monomeren en polymeren

Een **polymeer** (Grieks: 'veel delen') is een reuzenmolecuul uit duizenden kleine, gelijke eenheden: de **monomeren** ('één deel'). Polyetheen, de kunststof van plastic zakjes en flacons, is een ketting van CH_2-groepen; elke twee ervan kwamen uit één molecule etheen C_2H_4.

De ketens zijn niet allemaal even lang. De **polymerisatiegraad** n, het aantal monomeren per keten, ligt meestal tussen enkele duizenden en honderdduizenden. Je berekent hem uit de molaire massa (hoofdstuk 14):

POLYMERISATIEGRAAD

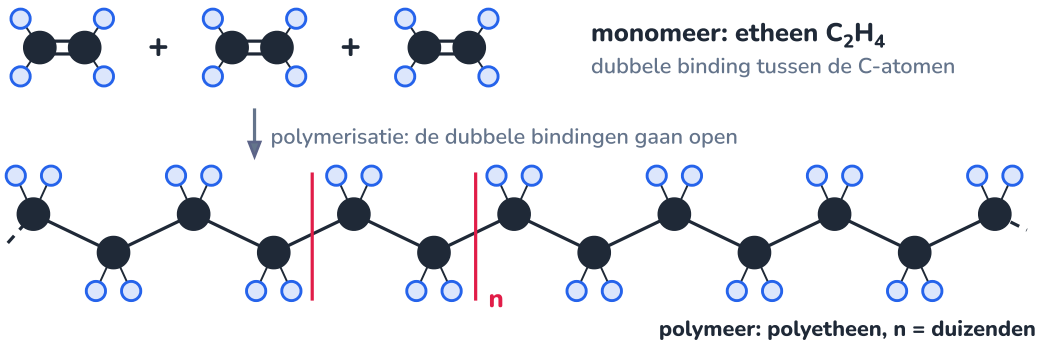
$$n = \frac{M_{\text{polymeer}}}{M_{\text{eenheid}}}$$

M_{eenheid} is de molaire massa van de herhalende eenheid, bij additiepolymeren dezelfde als die van het monomeer.

Omdat de moleculen zo lang zijn, raken ze in elkaar verstrengeld als spaghetti. Tussen de ketens werken alleen de zwakke van der Waalskrachten (hoofdstuk 13), maar over duizenden atomen opgeteld zijn die samen sterk. Hoe langer de ketens, hoe taaier de kunststof. Soms liggen stukken van de ketens netjes geordend naast elkaar, zoals in een kristal: de kunststof is dan **semikristallijn**. Liggen ze helemaal door elkaar, dan is ze **amorf**, en vaak doorzichtig (polycarbonaat, PETG).

Natuurlijke polymeren

Polymeren zijn geen uitvinding van de mens. Cellulose (hout, katoen, vlas) is een keten van glucose-eenheden, zetmeel ook, eiwitten zijn ketens van aminozuren, en DNA is een keten van nucleotiden (E29). Rubber komt uit het melksap van een boom.



Zwart: koolstof, blauw: waterstof. Tussen de rode haken staat één herhalende eenheid, n keer herhaald.

UITGEWERKT VOORBEELD · 17.4

Hoe lang is een polymeerketen?

4/19

Een molecule polyetheen uit een melkflacon heeft een molaire massa van ongeveer 200 000 g/mol. (a) Hoeveel etheeneenheden bevat ze? (b) Elke eenheid maakt de keten, helemaal gestrekt, 0,255 nm langer. Hoe lang is de keten? (c) De keten is ongeveer 0,5 nm dik. Vergelijk met een spaghettisliert van 1 mm dik.

GEGEVEN

- $M = 200\,000\ \mathrm{g/mol}$
- $M(\mathrm{C_2H_4}) = 2 \cdot 12{,}011 + 4 \cdot 1{,}008 = 28{,}05\ \mathrm{g/mol}$
- 0,255 nm per eenheid; dikte 0,5 nm

GEVRAAGD

- n
- lengte
- vergelijking

OPLOSSING

1 (a) $n = \dfrac{200\,000}{28{,}05} = 7129$ eenheden, afgerond 7100.

2 (b) $L = 7130 \cdot 0{,}255\ \mathrm{nm} = 1820\ \mathrm{nm} = 1{,}8\ \mu\mathrm{m}$.

3 (c) Lengte gedeeld door dikte: $1820/0{,}5 \approx 3600$. Een sliert van 1 mm dik met dezelfde verhouding is 3,6 m lang.

ANTWOORD

Ongeveer 7100 eenheden; gestrekt 1,8 µm lang, 3600 keer langer dan dik: een spaghetti van 3,6 m.

Klopt dit? In werkelijkheid is de keten niet gestrekt maar een wirwar: in een melkflacon liggen miljarden zulke slierten in elkaar verward. Daardoor scheurt de flacon niet makkelijk: een scheur moet ketens lostrekken uit die kluwen.

UITLEG · 17.5

Polymerisatie: additie en condensatie

5/19

Er zijn twee manieren om monomeren aan elkaar te rijgen.

Additiepolymerisatie. Het monomeer heeft een dubbele binding tussen twee koolstofatomen. Die gaat open, en elk koolstofatoom gebruikt het vrijgekomen elektron om aan het volgende monomeer te binden. Er valt niets af: de keten weegt precies evenveel als de monomeren samen.

ADDITIE: ETHEEN WORDT POLYETHEEN

$$n\ \mathrm{CH_2{=}CH_2} \longrightarrow [-\mathrm{CH_2{-}CH_2}-]_n$$

Verander je één waterstofatoom van etheen, dan krijg je een andere kunststof: een methylgroep geeft polypropeen (PP), een chlooratoom polyvinylchloride (PVC), een benzeenring polystyreen (PS). ABS is een mengpolymeer van drie monomeren: acrylonitril, butadieen en styreen.

Condensatiepolymerisatie. Het monomeer heeft aan beide uiteinden een reactieve groep, bijvoorbeeld een zuurgroep –COOH en een alcoholgroep –OH. Bij elke koppeling reageren ze met elkaar, en er komt een klein molecule vrij, meestal water. Zo maakt men PLA uit melkzuur, PET uit tereftaalzuur en ethaandiol, en nylon uit een zuur en een amine. Bakeliet is ook een condensatiepolymeer.

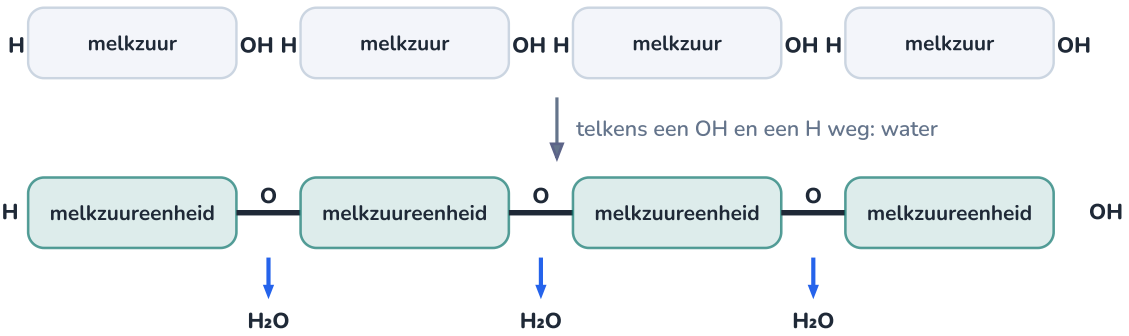
CONDENSATIE: MELKZUUR WORDT PLA

$$n\ \mathrm{HO{-}CH(CH_3){-}COOH} \longrightarrow \mathrm{H{-}[O{-}CH(CH_3){-}CO]_n{-}OH} + (n-1)\ \mathrm{H_2O}$$

In de fabriek maakt men PLA via een tussenstof (lactide) die zonder water koppelt; het resultaat is hetzelfde polymeer.

Herken ze aan de keten

Bestaat de keten alleen uit koolstofatomen, dan is het meestal een additiepolymeer (PE, PP, PVC, PS, ABS). Zitten er zuurstof- of stikstofatomen in de ketting zelf (–O– of –NH–), dan is het een condensatiepolymeer (PET, PLA, nylon, PC). Die bindingen kan water, traag, weer openbreken: daarom kan PLA afgebroken worden.



Bij elke koppeling vertrekt een watermolecule; de melkzuureenheden blijven via een zuurstofatoom aan elkaar vast.

UITGEWERKT VOORBEELD · 17.6

Van suiker naar PLA: een massabalans

6/19

PLA wordt gemaakt uit suiker (glucose): bacteriën zetten die om in melkzuur, $\mathrm{C_6H_{12}O_6} \rightarrow 2\,\mathrm{C_3H_6O_3}$, en melkzuur wordt gepolymeriseerd. Hoeveel melkzuur en hoeveel glucose zijn er minstens nodig voor een rol van 1,00 kg PLA-filament? Hoeveel water komt er bij de polymerisatie vrij?

GEGEVEN

- $M(\mathrm{C_3H_6O_3}) = 90{,}08\,\mathrm{g/mol}$ (melkzuur)
- herhalende eenheid van PLA: $\mathrm{C_3H_4O_2}$, $M = 72{,}06\,\mathrm{g/mol}$
- $M(\mathrm{C_6H_{12}O_6}) = 180{,}16\,\mathrm{g/mol}$

GEVRAAGD

- massa melkzuur, glucose en water

OPLOSSING

1. Eén melkzuurmolecule levert één eenheid in de keten en één molecule water (bij een lange keten telt die ene '-1' niet mee).
2. Mol eenheden in 1,00 kg PLA: $1000/72{,}06 = 13{,}88\,\mathrm{mol}$. Evenveel mol melkzuur: $13{,}88 \cdot 90{,}08 = 1250\,\mathrm{g}$.
3. Water: $13{,}88 \cdot 18{,}02 = 250\,\mathrm{g}$. Controle met massabehoud: $1250 - 1000 = 250\,\mathrm{g}$. Klopt.
4. Glucose: één molecule geeft twee melkzuur, en de massa verandert niet ($180{,}16 = 2 \cdot 90{,}08$). Dus ook 1,25 kg glucose.

ANTWOORD

Minstens 1,25 kg melkzuur, dus 1,25 kg suiker, voor 1,00 kg PLA; er komt 0,25 kg water vrij.

Klopt dit? In de praktijk is er meer suiker nodig: de bacteriën gebruiken zelf ook suiker om te leven, en niet elke stap haalt 100 %. En die suiker komt van een akker: 'biogebaseerd' kost ook grond, water en energie.

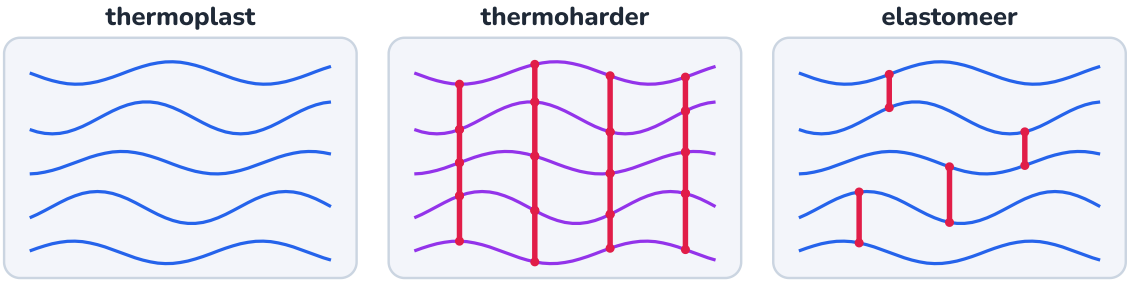
UITLEG · 17.7

Thermoplasten, thermoharders en elastomeren

7/19

Wat een kunststof doet als je hem verwarmt, hangt af van de bindingen *tussen* de ketens.

- **Thermoplasten** hebben losse ketens die alleen door zwakke krachten en verstrengeling samenhangen. Warm je ze op, dan gaan de ketens bewegen en langs elkaar schuiven: de kunststof wordt zacht en smelt. Koelt hij af, dan wordt hij weer vast. Je kan hem dus **omsmelten**: spuitgieten, extruderen, 3D-printen en recycleren. PE, PP, PVC, PS, PET, PLA, PETG, ABS en PC zijn thermoplasten.
- **Thermoharders** vormen tijdens het uitharden een driedimensionaal **netwerk**: de ketens zijn op veel plaatsen met atoombindingen aan elkaar vastgemaakt. Ze kunnen niet meer schuiven, dus smelten ze niet. Te warm, en ze verkolen. Bakeliet, epoxy (twee-componentenlijm, de hars in composieten), polyesterhars en melamine zijn thermoharders. Ze zijn hard, stijf en hittebestendig, maar moeilijk te recycleren.
- **Elastomeren** hebben weinig bruggen tussen lange, zeer beweeglijke ketens. Trek je eraan, dan strekken de ketens zich; laat je los, dan trekken de bruggen alles terug. Natuurrubber werd pas bruikbaar toen Charles Goodyear in 1839 ontdekte dat verhitten met zwavel zulke bruggen legt: **vulkaniseren**. Autobanden, elastiekjes en afdichtingen zijn elastomeren. Flexibel filament (TPU) is een thermoplastisch elastomeer: de bruggen zijn harde stukjes keten die bij het printen smelten.



Rood: bindingen tussen de ketens. Geen bruggen: smelt. Veel bruggen: stijf netwerk. Enkele bruggen: rubber.

UITLEG · 17.8

De glasovergang: waarom een print zacht wordt

8/19

Een thermoplast smelt niet plots. Lang daarvoor gebeurt er al iets: bij de **glasovergangstemperatuur** T_g krijgen de ketens in het amorfe deel genoeg energie om te gaan draaien en schuiven. Onder T_g is de kunststof **glasachtig**: hard, stijf, soms bros. Boven T_g wordt hij **rubberachtig**: zacht, buigzaam, en een belast onderdeel gaat langzaam vervormen (**kruip**). Een semikristallijne kunststof houdt boven T_g nog samen door zijn kristallen, tot die bij de **smelttemperatuur** T_m smelten.

kunststof	soort	Tg (°C)	Tm (°C)	bij kamertemperatuur
polyetheen (HDPE)	semikristallijn	± −120	± 130	taai, buigzaam
polypropeen (PP)	semikristallijn	± −10	± 165	taai; bros in de vrieskou
PLA	meestal amorf geprint	55 – 60	± 150 – 175	stijf, wat bros

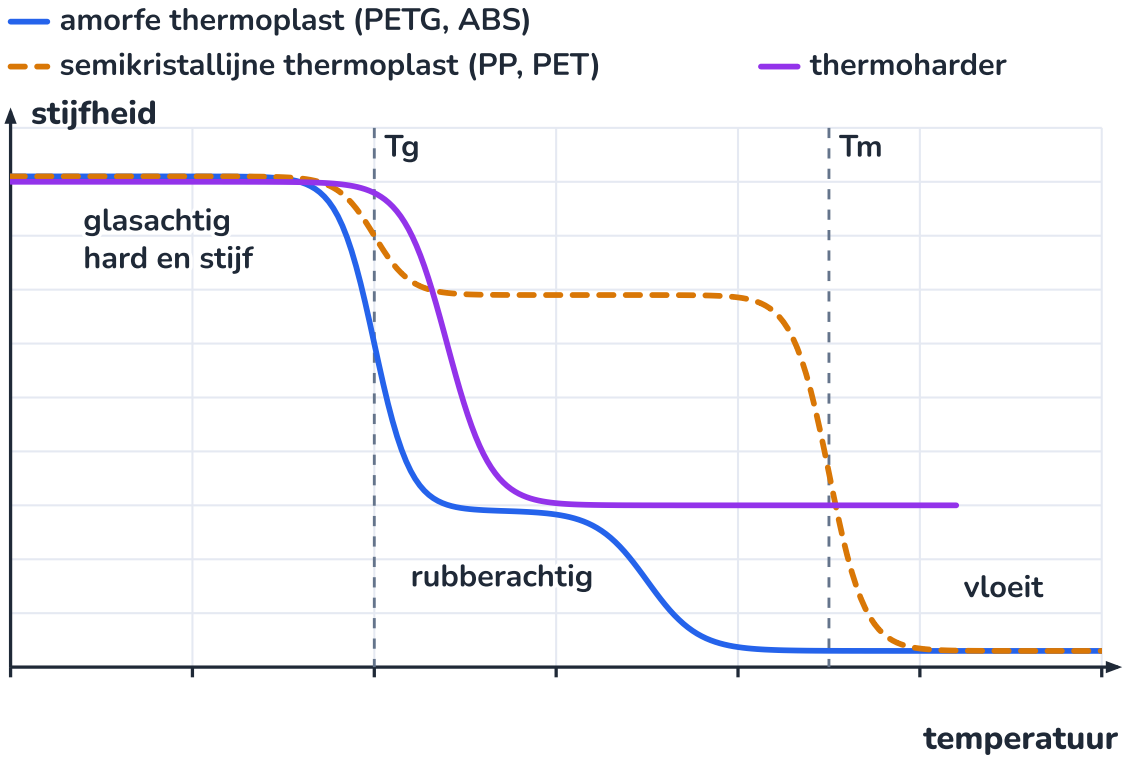
kunststof	soort	Tg (°C)	Tm (°C)	bij kamertemperatuur
PET (flessen)	semikristallijn	± 75	± 255	stijf en taai
PETG	amorf	± 80	–	stijf en taai
PVC (hard)	amorf	± 80	–	stijf
ASA, ABS	amorf	± 100 – 105	–	stijf en slagvast
polystyreen (PS)	amorf	± 100	–	stijf, bros
polycarbonaat (PC, kaswand)	amorf	± 147	–	stijf en zeer slagvast

Afgeronde waarden; ze hangen af van de precieze samenstelling en van de meetmethode.

Twee gevolgen die je meteen herkent. Een PLA-print in een auto in de zomerzon zakt door: het dashboard wordt gemakkelijk warmer dan 60 °C. En een emmer uit PP kan bij strenge vorst breken als je hem laat vallen: onder zijn T_g wordt PP bros. Polyetheen heeft dat probleem niet: zijn T_g ligt ver onder elk winterweer.

Een vuistregel voor het ontwerp

Kies voor een onderdeel dat onder belasting vorm moet houden een kunststof met een T_g die **minstens 15 °C** boven de hoogste temperatuur van het onderdeel ligt. Voor semikristallijne kunststoffen zoals PP en PE is de smelttemperatuur de grens, met een ruimere marge.



Kwalitatieve schets: bij Tg valt de stijfheid van een amorfe thermoplast honderden keren terug; een thermoharder blijft stijver.

PROEF · 17.9

Wanneer wordt een print zacht?

9/19

ONDERZOEKSVRAAG

Bij welke temperatuur verliezen staafjes uit PLA, PETG en ABS (of ASA) hun stijfheid? Klopt dat met hun Tg?

MATERIAAL

- drie geprinte staafjes van elk materiaal, 80 × 10 × 2 mm, liggend geprint, 100 % vulling
- waterkoker, koud water, bekerglazen van 600 mL, thermometer
- tang, hittebestendige handschoenen, veiligheidsbril
- een potlood om rond te buigen

Veiligheid

Water boven 60 °C geeft in enkele seconden brandwonden. Zet de bekers stabiel op een bakplaat, werk met een tang en handschoenen, en giet nooit kokend water boven een ander. Laat de staafjes eerst even afkoelen voor je ze met de hand aanraakt.

PLA uit het bad van 60 °C buigt als was; PETG en ABS blijven recht.

WERKWIJZE

1. Maak waterbaden van 40, 50, 60, 70, 80 en 90 °C door kokend en koud water te mengen. Controleer met de thermometer.
2. Leg een staafje van elk materiaal twee minuten in het bad van 40 °C.
3. Haal ze er met de tang uit en probeer ze meteen, met handschoenen, rond het potlood te buigen. Noteer: stijf, buigt licht, zacht.
4. Herhaal met nieuwe staafjes in de warmere baden. Laat gebogen staafjes afkoelen en bekijk of ze gebogen blijven.

WAARNEMING

bad (°C)	PLA	PETG	ABS
40	stijf	stijf	stijf
50	stijf	stijf	stijf
60	zacht: buigt rond het potlood	stijf	stijf
70	slap	buigt licht	stijf
80	slap	zacht	stijf
90	slap	slap	buigt licht

Voorbeeldwaarneming. Afgekoeld blijven de gebogen staafjes in hun nieuwe vorm.

VERKLARING

Elk materiaal wordt zacht in de buurt van zijn glasovergangstemperatuur: PLA net boven 55-60 °C, PETG rond 80 °C, ABS pas rond 100 °C (en kokend water mag je niet gebruiken). Dat de gebogen vorm blijft na het afkoelen, toont dat het thermoplasten zijn: boven Tg schuiven de ketens, onder Tg vriezen ze in hun nieuwe positie vast. Zo buigen installateurs ook PVC-buizen met een föhn. Een staafje uit een thermoharder zou nergens in dit bereik zacht worden.

UITLEG · 17.10

PLA, PETG, ABS en ASA: filament kiezen

10/19

Alle gangbare filamenten zijn thermoplasten: de printer smelt ze in de spuitmond en legt ze laag per laag neer. Maar hun eigenschappen verschillen sterk.

	PLA	PETG	ABS	ASA
grondstof	suiker uit maïs, suikerriet of bieten	aardolie	aardolie	aardolie
glasovergang Tg	55 – 60 °C	± 80 °C	± 105 °C	± 100 °C
spuitmond	190 – 220 °C	220 – 250 °C	230 – 260 °C	240 – 260 °C
printbed	50 – 60 °C (of koud)	70 – 85 °C	90 – 110 °C	90 – 110 °C
dichtheid	1,24 g/cm³	1,27 g/cm³	1,04 g/cm³	1,07 g/cm³
printen	zeer makkelijk	makkelijk, wat draadjes	krimpt en trekt krom: gesloten printer	zoals ABS
dampen	weinig	weinig	styreen: goed verluchten	goed verluchten
warm en vochtig	wordt zacht vanaf ± 55 °C; valt langzaam uiteen (hydrolyse)	goed bestand	goed bestand	goed bestand
zonlicht (UV)	matig	redelijk	slecht: vergeelt, wordt bros	zeer goed
afbreekbaar	industrieel composteerbaar	nee	nee	nee

Richtwaarden: kijk voor het printen altijd naar de gegevens van de fabrikant van je filament.

- **PLA** is het makkelijkste filament: weinig krimp, geen verwarmd bed nodig, weinig geur. Maar het wordt al zacht rond 55 °C, en in warme, vochtige lucht breken water en warmte de esterbindingen in de keten langzaam open (hydrolyse). Prima voor prototypes en onderdelen binnen.

- **PETG** is PET waarin een extra glycolbouwsteen de kristallisatie tegenhoudt: amorf, taai, doorzichtig mogelijk, goed bestand tegen water en veel chemicaliën, en zacht pas rond 80 °C.
- **ABS** houdt van warmte (Tg rond 105 °C) en is slagvast, maar krimpt bij het afkoelen, trekt krom en geeft styreendampen: print in een gesloten, verluchte printer. In zonlicht vergeelt het en wordt het bros.
- **ASA** is een verwant van ABS waarin het rubberdeel vervangen is door een acrylaat dat UV-licht verdraagt: het buitenfilament bij uitstek.

> ⚠ **Vocht in het filament**
>
> PETG, PLA en vooral nylon trekken vocht uit de lucht. Nat filament sist en borrelt in de hete spuitmond, en de print wordt zwakker. Bewaar rollen dicht met droogzakjes, en droog ze bij twijfel enkele uren op lage temperatuur (volg de fabrikant).

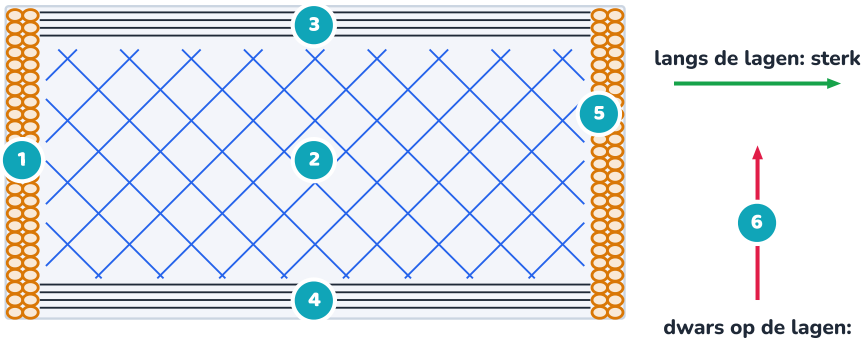
Zelfde vorm, andere kunststof: PLA, PETG, ABS en ASA.

ONTLEED · 17.11

Een 3D-print van binnen

11/19

Een geprint onderdeel is geen massief blok. Het is opgebouwd uit rupsen gesmolten kunststof: een huid van wanden, onder- en bovenlagen, en binnenin een vulling. Die opbouw maakt het **anisotroop**: zijn sterkte hangt af van de richting, net als bij hout of een vezelcomposiet.



doorsnede loodrecht op de printlagen (schematisch, niet op schaal)

Oranje: de rupsen van de wanden, blauw: de vulling, dunne lijnen: de volle onder- en bovenlagen.

1 Wanden
Twee tot vier rupsen dik. Ze dragen het meeste van de belasting: meer wanden maakt een onderdeel sterker dan meer vulling.

2 Vulling
Hier 20 % in een raster. Ze steunt de bovenlagen en de wanden, en spaart materiaal en printtijd. Voor een onderdeel dat veel kracht moet opnemen: 40 à 100 %.

3 Bovenlagen
Enkele volle lagen sluiten het onderdeel af. Te weinig, en je ziet de vulling erdoor, of er komt water binnen.

4 Onderlagen
De eerste laag hecht aan het printbed. Bij ABS trekt het afkoelen de hoeken hier omhoog (kromtrekken).

5 Laaghechting
Elke rups is gesmolten op de vorige, die al half afgekoeld was: de ketens raken maar gedeeltelijk in elkaar verstrengeld. Een print is daardoor tot twee keer zwakker tussen de lagen dan erlangs.

6 Zwakke richting
Kies de printrichting zo dat de grootste trekkracht langs de lagen loopt, niet ertussen. Een haak print je liggend, niet rechtop.

UITLEG · 17.12

Vezelcomposieten: glas, koolstof en vlas

12/19

Een **composiet** combineert twee materialen tot iets beters (F10: gewapend beton). Bij een **vezelcomposiet** liggen dunne, sterke en stijve vezels in een kunststof, de **matrix**, meestal epoxy- of polyesterhars. De vezels dragen de trekkracht; de matrix houdt ze op hun plaats, verdeelt de kracht over de vezels en beschermt ze. Een losse glasvezel breekt makkelijk bij een kras; in hars samen met miljoenen andere niet.

materiaal	E (GPa)	dichtheid (g/cm³)	treksterkte (MPa)	waar
glasvezel (E-glas)	± 72	2,55	2000 – 3500	boten, windturbinebladen, printplaten
koolstofvezel	± 230	1,80	3500 – 5000	fietsen, vliegtuigen, sportmateriaal
vlasvezel	50 – 70	1,45	500 – 1500	autopanelen, fietsen, surfplanken
epoxyhars (matrix)	± 3	1,20	60 – 80	lijmt de vezels samen
staal (constructie)	210	7,85	± 400 – 500	ter vergelijking
aluminium	70	2,70	± 100 – 300	ter vergelijking

Vezelwaarden gelden voor een losse vezel, als orde van grootte.

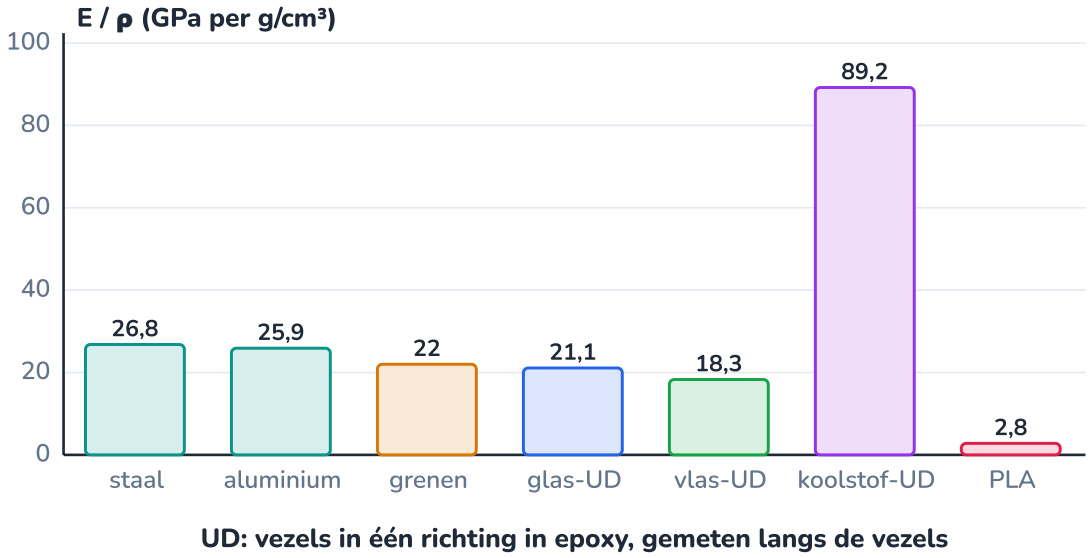
Langs de vezels schat je de stijfheid en de dichtheid met de **mengregel**: een gewogen gemiddelde naar het volumedeel V_v van de vezels.

MENGREGEL (LANGS DE VEZELS)

$$E_c = V_v\,E_v + (1 - V_v)\,E_m \qquad \rho_c = V_v\,\rho_v + (1 - V_v)\,\rho_m$$

v: vezel, m: matrix, c: composiet. Dwars op de vezels is het composiet veel slapper: daar geldt deze regel niet. Daarom legt men vezels in lagen in verschillende richtingen.

- **Glasvezel**: goedkoop, sterk, roest niet: boten, windturbinebladen, printplaten (de groene plaat onder je microcontroller).
- **Koolstofvezel**: veel stijver per kilogram dan staal of aluminium, maar duur en energie-intensief om te maken. Bij de Boeing 787 is ongeveer de helft van de massa composiet.
- **Vlasvezel**: uit de stengels van vlas, een plant die in West-Europa (Frankrijk, België, Nederland) meer geteeld wordt dan waar ook. De Leie heette vroeger de 'Gouden Rivier', omdat het vlas er te roten lag. Vlascomposiet is lichter dan glasvezelcomposiet, dempt trillingen, en kost minder energie om te maken.



Per kilogram is alleen koolstofvezelcomposiet echt stijver dan staal en aluminium; glas en vlas winnen vooral op sterkte, prijs of corrosie.

Vlasweefsel in epoxy: een Vlaamse vezel in een hightech composiet.

UITGEWERKT VOORBEELD · 17.13

De mengregel: vlas, glas of koolstof?

13/19

Een fietsenmaker vergelijkt drie composietbuizen met de vezels in de lengte. (a) Bereken voor elk de stijfheid E en de dichtheid langs de vezels: glasvezel met 55 % vezels, koolstofvezel met 60 %, vlas met 40 %, telkens in epoxy. (b) Bereken de specifieke stijfheid E/ρ en vergelijk met aluminium (70 GPa, 2,70 g/cm³).

GEGEVEN

- epoxy: $E_m = 3\,\text{GPa}$, $\rho_m = 1{,}20\,\text{g/cm}^3$
- glas: 72 GPa, 2,55 g/cm³; koolstof: 230 GPa, 1,80 g/cm³; vlas: 55 GPa, 1,45 g/cm³

GEVRAAGD

- E_c, ρ_c, E_c/ρ_c

OPLOSSING

1. Glas: $E_c = 0{,}55 \cdot 72 + 0{,}45 \cdot 3 = 39{,}6 + 1{,}35 = 41\,\text{GPa}$; $\rho_c = 0{,}55 \cdot 2{,}55 + 0{,}45 \cdot 1{,}20 = 1{,}94\,\text{g/cm}^3$; $E/\rho = 21$.
2. Koolstof: $E_c = 0{,}60 \cdot 230 + 0{,}40 \cdot 3 = 139\,\text{GPa}$; $\rho_c = 0{,}60 \cdot 1{,}80 + 0{,}40 \cdot 1{,}20 = 1{,}56\,\text{g/cm}^3$; $E/\rho = 89$.
3. Vlas: $E_c = 0{,}40 \cdot 55 + 0{,}60 \cdot 3 = 23{,}8\,\text{GPa}$; $\rho_c = 0{,}40 \cdot 1{,}45 + 0{,}60 \cdot 1{,}20 = 1{,}30\,\text{g/cm}^3$; $E/\rho = 18$.
4. Aluminium: $70/2{,}70 = 26$.

ANTWOORD

Koolstof: 139 GPa bij 1,56 g/cm³ (89 per g/cm³), meer dan drie keer zo stijf per kilogram als aluminium. Glas (41 GPa, 21) en vlas (24 GPa, 18) zijn per kilogram iets *minder* stijf dan aluminium.

Klopt dit? Waarom dan toch glas of vlas? Ze zijn veel sterker per kilogram dan aluminium (een glasvezelcomposiet haalt zo'n 1000 MPa), ze roesten niet, laten zich in elke vorm gieten, en zijn goedkoper dan koolstof. En de mengregel geldt maar langs de vezels: in een echte buis liggen lagen ook schuin, en dan is het composiet minder stijf dan deze getallen.

UITLEG · 17.14

Bioplastics en afbreekbaarheid

14/19

Het woord **bioplastic** wordt voor twee heel verschillende dingen gebruikt.

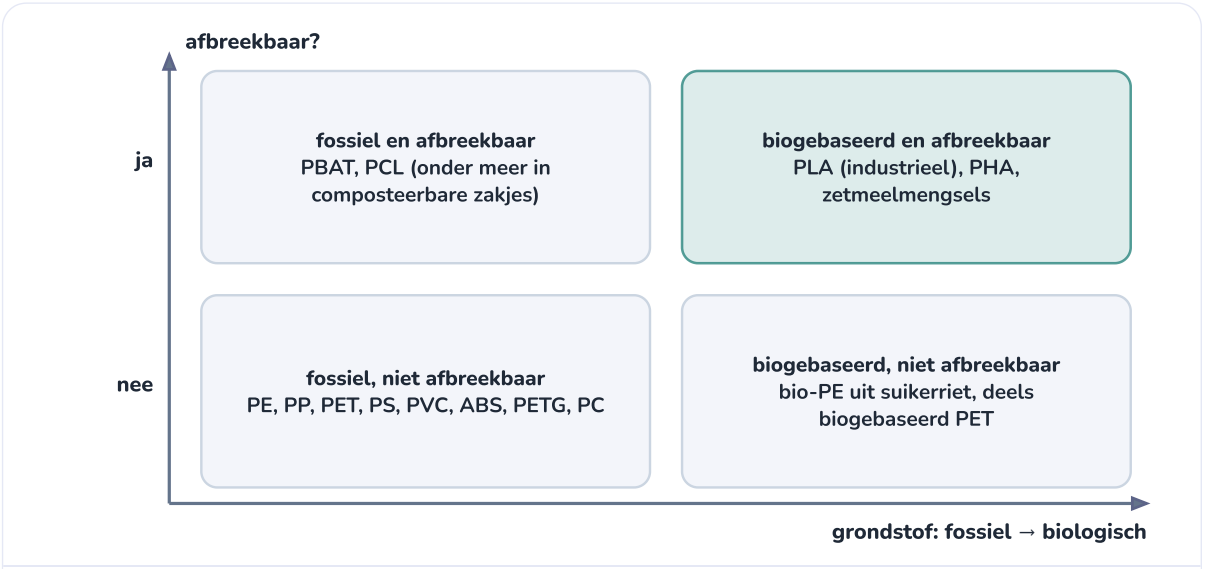
- **Biogebaseerd**: gemaakt uit planten in plaats van aardolie. Bio-PE uit suikerriet is chemisch precies hetzelfde als gewoon PE: het breekt even slecht af.
- **Biologisch afbreekbaar**: micro-organismen kunnen het omzetten in water, CO_2 en biomassa. Dat hangt niet af van waar de grondstof vandaan komt, maar van de bindingen in de keten én van de omstandigheden: temperatuur, vocht, zuurstof, de juiste micro-organismen. Er bestaan ook afbreekbare kunststoffen uit aardolie.

PLA is biogebaseerd en **industrieel composteerbaar**. Een verpakking mag dat alleen zeggen als ze de Europese norm EN 13432 haalt: in een industriële composteerinstallatie binnen 12 weken uiteengevallen (hoogstens 10 % stukjes groter dan 2 mm) en binnen 6 maanden voor minstens 90 % omgezet, zonder schadelijke resten. In zo'n installatie wordt het binnenin ongeveer 55 à 70 °C, en het is er vochtig. Rond 58 °C wordt PLA zacht, neemt het water op en breken de esterbindingen: dan kunnen micro-organismen het afbreken.

In een **tuincompostvat**, dat zelden zo warm wordt, gebeurt er maanden of jaren bijna niets; in de bodem of in de zee nog minder. PLA is dus **niet** tuincomposteerbaar. Wat wel thuis composteert, draagt een apart label, zoals 'OK compost HOME' (van het certificatiebureau TÜV Austria, dat de labels overnam van het Belgische Vinçotte). Of een composteerbaar product bij het gft mag, bepaalt je gemeente: kijk het na.

Afbreekbaar plastic mag in de natuur

Nee. 'Afbreekbaar' zegt alleen dat het onder bepaalde omstandigheden kan. Een PLA-bekertje in het bos of in de zee ligt er jaren, en valt ondertussen uiteen in kleine stukjes. Zwerfvuil blijft zwerfvuil, welk plastic het ook is.



Twee vragen, vier vakjes: waar komt het vandaan, en kan het (onder de juiste omstandigheden) afgebroken worden?

UITLEG · 17.15

Recyclagecodes en de kunststofkringloop

15/19

In het driehoekje op een kunststof voorwerp staat een getal van 1 tot 7: de **recyclagecode**, ingevoerd door de Amerikaanse kunststofindustrie in 1988. Het zegt *welke* kunststof het is, niet dat het gerecycleerd wordt.

code	kunststof	voorbeelden	blauwe zak (verpakking)?
1	PET	drankflessen, fruitschaaltjes	ja
2	HDPE (PE-HD)	flacons van wasmiddel en shampoo, doppen	ja
3	PVC	buizen, raamprofielen, kabelmantels	zelden verpakking; bouwafval naar het recyclagepark
4	LDPE (PE-LD)	folie, zakjes, krimpfolie	ja
5	PP	potjes, doppen, plantenbakjes van de meetkas	ja (als verpakking)
6	PS	yoghurtpotjes, piepschuim	potjes ja; groot piepschuim naar het recyclagepark
7	overige: PC, PLA, ABS, mengsels	kasplaten, 3D-prints, elektronica	alleen als het verpakking is

Sinds 2021 mag in heel België alle plastic *verpakking* in de blauwe zak. Een 3D-print, een emmer of speelgoed is geen verpakking.

De kringloop. Na het gebruik zijn er vier wegen:

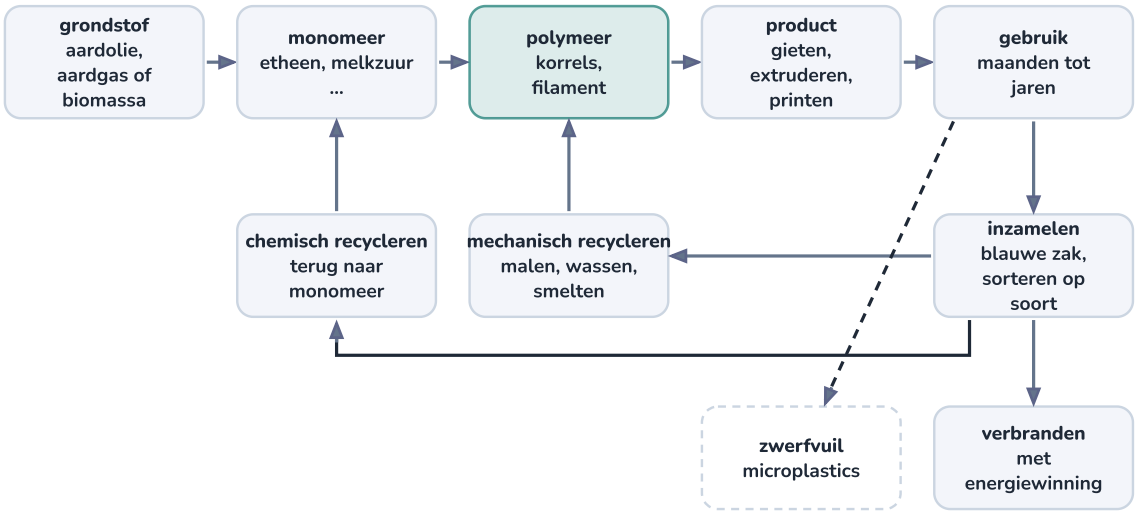
- **Mechanisch recycleren**: sorteren per soort (in de sorteercentra met infraroodcamera's die elke kunststof herkennen), malen, wassen, smelten tot nieuwe korrels. Werkt alleen voor thermoplasten, en alleen goed voor zuivere stromen: PET-flessen worden opnieuw flessen. Bij elke smeltbeurt breken er ketens; gerecycleerde kunststof wordt daarom vaak een minder veeleisend product.
- **Chemisch recycleren**: de ketens terug afbreken tot monomeren, en opnieuw polymeriseren. Kan ook voor vuile of gemengde stromen, maar kost veel energie. PLA en PET, met hun esterbindingen, lenen

zich er goed toe.

- **Verbranden met energieterugwinning**: de energie in de ketens wordt warmte en stroom, maar de koolstof wordt CO_2 (opdracht 17.14).
- **Lekken naar het milieu**: zwerfvuil valt uiteen tot **microplastics**, deeltjes kleiner dan 5 mm, die inmiddels in zee, in de bodem en in voedsel teruggevonden worden.

3D-printafval

Mislukte prints, steunmateriaal en restjes filament zijn geen verpakking: ze horen niet in de blauwe zak. Houd ze per soort apart (PLA bij PLA): sommige makerspaces malen en extruderen ze tot nieuw filament. Gemengd zijn ze bijna niet te recycleren.



Van grondstof naar product en terug: hoe dichter de lus bij het product sluit, hoe minder energie ze kost.

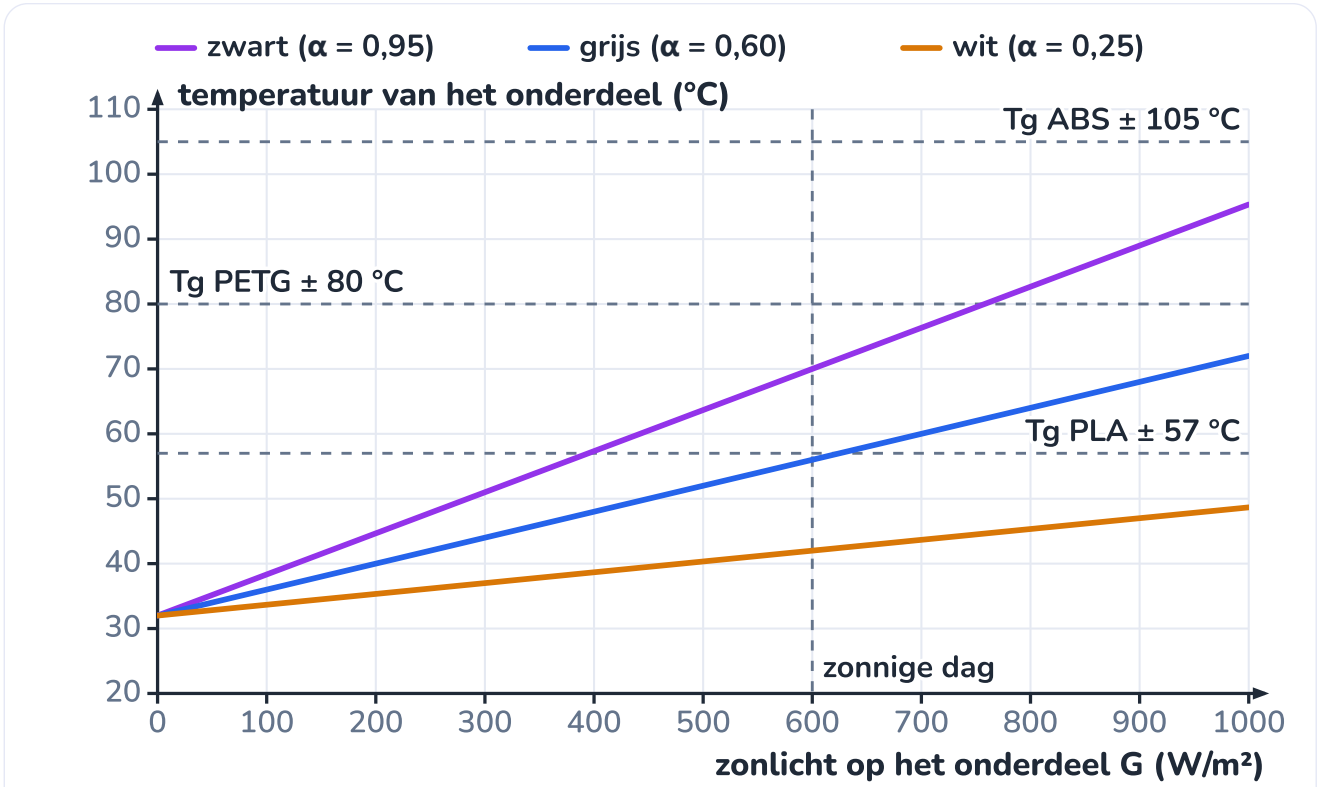
Infraroodcamera's herkennen elke kunststof; luchtstoten blazen ze naar de juiste stroom.

UITGEWERKT VOORBEELD · 17.16

Habitat: hoe warm wordt een onderdeel in de kas?

16/19

Op een zonnige aprildag wordt de lucht in de meetkas 32,0 °C (F4); in de zomer, met het dak dicht, nog warmer. Een geprinte klem voor de druppelslang hangt in de zon achter de polycarbonaatwand. Hoe warm wordt ze, in zwart, grijs en wit? Welk filament kies je?



Model T = T_lucht + α·G/h met h = 15 W/(m²·K): een zwart onderdeel komt in de zon boven de Tg van PLA én van PETG.

GEGEVEN

- zonlicht op de klem achter de kaswand: $G \approx 600\,\mathrm{W/m^2}$ (de wand laat ± 80 % door)
- deel van het zonlicht dat de kleur opneemt: zwart $\alpha = 0{,}95$, grijs $0{,}60$, wit $0{,}25$
- warmteafgifte aan de lucht (stilstaande lucht, ook straling): $h \approx 15\,\mathrm{W/(m^2 \cdot K)}$
- vuistregel: $T_g \geq T_{\text{onderdeel}} + 15\,^\circ\mathrm{C}$

GEVRAAGD

- $T_{\text{onderdeel}}$ per kleur
- filamentkeuze

OPLOSSING

1. Model: in evenwicht geeft het onderdeel evenveel warmte af als het zonlicht opneemt, per m² oppervlak: $\alpha\, G = h\,(T - T_{\text{lucht}})$, dus $T = T_{\text{lucht}} + \dfrac{\alpha\, G}{h}$. (Warmteoverdracht kwantitatief volgt in hoofdstuk 19 en E17; hier is h een geschatte waarde.)
2. Zwart: $T = 32 + \dfrac{0{,}95 \cdot 600}{15} = 32 + 38 = 70\,^\circ\mathrm{C}$. Grijs: $32 + 24 = 56\,^\circ\mathrm{C}$. Wit: $32 + 10 = 42\,^\circ\mathrm{C}$.
3. Nodige T_g: zwart 85 °C: alleen ASA of ABS. Grijs 71 °C: PETG, ASA, ABS. Wit 57 °C: PLA net, PETG ruim.
4. Maar de kas is ook **vochtig** (± 70 % luchtvochtigheid, condens): PLA veroudert er door hydrolyse, en bij 42 °C kruipt het al merkbaar onder belasting. En in de zomer ligt de kaslucht hoger: bij 40 °C wordt wit 50 °C en zwart 78 °C.

ANTWOORD

Habitat print de onderdelen in de kas in PETG, wit of lichtgrijs (een wit onderdeel blijft rond 42 à 50 °C, ruim onder 80 °C). Wat zwart moet zijn, in de volle zon hangt of buiten komt, wordt ASA: UV-bestendig en pas zacht rond 100 °C. PLA blijft voor prototypes en binnen.

Klopt dit? Een zwart dashboard in de zon wordt 70 à 90 °C: het model zit in de juiste orde van grootte. Wind of ventilatie maakt h groter (bij $h = 25$ wordt zwart 55 °C in plaats van 70 °C): het model met stilstaande lucht is dus aan de veilige kant. Meet het na met een temperatuursensor op een geprinte klem (opdracht 17.13).

CODE · 17.17

Python: het filament laten kiezen

17/19

Het model en de vuistregel passen in een paar regels code. Het programma berekent per kleur de temperatuur van het onderdeel en zoekt welke filamenten een T_g hebben die minstens 15 °C hoger ligt. De grafiek van het vorige blad is met dezelfde functie getekend.

Python **filamentkeuze.py** 20 regels

```python
# filamentkeuze.py - hoe warm wordt een 3D-print in de kas, en welk filament houdt dat uit?
T_LUCHT = 32.0        # hoogste luchttemperatuur in de kas (graden Celsius), zonnige aprildag
G = 600.0             # zonlicht op het onderdeel, achter de kaswand (W/m2)
H = 15.0              # warmteoverdracht naar de lucht bij stilstaande lucht (W/(m2 K))
MARGE = 15.0          # vuistregel: Tg minstens 15 graden boven de temperatuur van het onderdeel

KLEUREN = {"zwart": 0.95, "grijs": 0.60, "wit": 0.25}      # welk deel van het zonlicht de kleur opneemt
TG = {"PLA": 57, "PETG": 80, "ASA": 100, "ABS": 105}       # glasovergangstemperatuur (graden Celsius)


def temperatuur(absorptie, t_lucht=T_LUCHT, g=G, h=H):
    """Evenwicht: opgenomen zonlicht = afgegeven warmte, dus absorptie * g = h * (T - t_lucht)."""
    return t_lucht + absorptie * g / h


if __name__ == "__main__":
    for kleur, a in KLEUREN.items():
        t = temperatuur(a)
        goed = [naam for naam, tg in TG.items() if tg >= t + MARGE]
        print(f"{kleur:5s}: {t:5.1f} graden -> geschikt: {', '.join(goed) or 'geen'}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
zwart:  70.0 graden -> geschikt: ASA, ABS
grijs:  56.0 graden -> geschikt: PETG, ASA, ABS
wit  :  42.0 graden -> geschikt: PLA, PETG, ASA, ABS
```

Het programma zegt dat PLA voor een wit onderdeel net kan: het kijkt alleen naar de temperatuur. Vocht, UV-licht en kruip staan niet in het model; daarvoor moet je zelf nadenken (of het model uitbreiden, opdracht 17.10). Een model beslist niet: het rekent, en jij kiest.

NASLAG · 17.18

Overzicht: polymeren en composieten

18/19

polymerisatiegraad

$$n = \frac{M_{\text{polymeer}}}{M_{\text{eenheid}}}$$

symbool	betekenis	eenheid
n	aantal eenheden per keten	–
M	molaire massa	g/mol

mengregel (langs de vezels)

$$E_c = V_v\,E_v + (1 - V_v)\,E_m$$

symbool	betekenis	eenheid
V_v	volumedeel vezels	–
E	elasticiteitsmodulus	GPa

onderdeel in de zon (model)

$$T = T_{\text{lucht}} + \frac{\alpha\,G}{h}$$

symbool	betekenis	eenheid
α	deel van het zonlicht dat opgenomen wordt	–
G	zonlicht	W/m²
h	warmteoverdracht naar de lucht	W/(m²·K)

	PLA	PETG	ABS	ASA
grondstof	suiker uit maïs, suikerriet of bieten	aardolie	aardolie	aardolie

	PLA	PETG	ABS	ASA
glasovergang Tg	55 – 60 °C	± 80 °C	± 105 °C	± 100 °C
spuitmond	190 – 220 °C	220 – 250 °C	230 – 260 °C	240 – 260 °C
printbed	50 – 60 °C (of koud)	70 – 85 °C	90 – 110 °C	90 – 110 °C
dichtheid	1,24 g/cm³	1,27 g/cm³	1,04 g/cm³	1,07 g/cm³
printen	zeer makkelijk	makkelijk, wat draadjes	krimpt en trekt krom: gesloten printer	zoals ABS
dampen	weinig	weinig	styreen: goed verluchten	goed verluchten
warm en vochtig	wordt zacht vanaf ± 55 °C; valt langzaam uiteen (hydrolyse)	goed bestand	goed bestand	goed bestand
zonlicht (UV)	matig	redelijk	slecht: vergeelt, wordt bros	zeer goed
afbreekbaar	industrieel composteerbaar	nee	nee	nee

Richtwaarden: kijk voor het printen altijd naar de gegevens van de fabrikant van je filament.

code	kunststof	voorbeelden	blauwe zak (verpakking)?
1	PET	drankflessen, fruitschaaltjes	ja
2	HDPE (PE-HD)	flacons van wasmiddel en shampoo, doppen	ja
3	PVC	buizen, raamprofielen, kabelmantels	zelden verpakking; bouwafval naar het recyclagepark
4	LDPE (PE-LD)	folie, zakjes, krimpfolie	ja
5	PP	potjes, doppen, plantenbakjes van de meetkas	ja (als verpakking)
6	PS	yoghurtpotjes, piepschuim	potjes ja; groot piepschuim naar het recyclagepark
7	overige: PC, PLA, ABS, mengsels	kasplaten, 3D-prints, elektronica	alleen als het verpakking is

Sinds 2021 mag in heel België alle plastic *verpakking* in de blauwe zak. Een 3D-print, een emmer of speelgoed is geen verpakking.

Habitat: rekenwaarden uit dit hoofdstuk	
filament in de kas	PETG, wit of lichtgrijs (Tg ± 80 °C)
donker, volle zon, buiten	ASA (Tg ± 100 °C, UV-bestendig)
PLA	alleen prototypes en binnen
vuistregel	Tg minstens 15 °C boven de hoogste temperatuur van het onderdeel
model	G ≈ 600 W/m² achter de kaswand, h ≈ 15 W/(m²·K): bij 32 °C zwart 70 °C, grijs 56 °C, wit 42 °C

HET GEHEEL · 17.19

Waar polymeren en composieten terugkomen

19/19

Polymeren zijn de brug tussen chemie (M) en ontwerpen en maken (D): wie de ketens begrijpt, kiest beter materiaal, print sterker en ontwerpt voor recyclage. Ze raken ook aan leven (cellulose, eiwitten, DNA) en aan energie en klimaat (grondstoffen, verbranding, recyclage).

F10 Materialen en hun eigenschappen

Materiaalfamilies, composieten en recyclagecodes: hier van binnenuit verklaard.

F32 Van idee tot object: CAD en 3D-printen

Laaghoogte, vulling en supports; hier: welk filament en welke printrichting.

I13 Chemische bindingen en stofeigenschappen

Atoombindingen in de keten, van der Waalskrachten ertussen.

I14 Reacties en stoichiometrie

Molaire massa en massabalans van een polymerisatie.

I16 Redox en batterijen

De separator in een lithiumcel is een poreuze kunststoffolie die bij ± 130 °C smelt.

I19 Warmtetransport en isolatie

Het warmtemodel van een onderdeel in de zon, kwantitatief.

I37 Parametrisch ontwerpen

Ontwerpen voor 3D-printen: wanddikte, printrichting, snap-fits.

I38 Duurzaam ontwerpen

Levenscyclus, circulair ontwerpen en materiaalkeuze met gewichten.

E12 Sterkteleer

Spanning, rek en de elasticiteitsmodulus: rekenen aan stijfheid.

E28 Materiaalkunde

Polymeerstructuur en glasovergang, composieten en Ashby-diagrammen in detail.

E47 Duurzaamheid en levenscyclusanalyse

Is een PLA-onderdeel echt duurzamer dan een PETG-onderdeel? Een LCA.

Habitat

Voor Habitat leggen we vast: geprinte onderdelen in de kas zijn uit PETG in een lichte kleur; donkere onderdelen in de volle zon en onderdelen buiten uit ASA; PLA alleen voor prototypes. De vuistregel is een Tg van minstens 15 °C boven de hoogste temperatuur van het onderdeel. In opdracht 17.13 toets je de keuze aan een zomerdag en aan een meting; in hoofdstuk 37 ontwerp je de behuizing van de elektronica parametrisch.

SAMENVATTING

In het kort

- Een polymeer is een reuzenmolecuul uit duizenden monomeren; de polymerisatiegraad is M(polymeer) / M(eenheid).
- Additiepolymerisatie opent dubbele bindingen zonder bijproduct (PE, PP, PVC, PS); condensatiepolymerisatie koppelt reactieve groepen en splitst water af (PLA, PET, nylon, bakeliet).
- Thermoplasten (losse ketens) smelten en zijn omsmeltbaar; thermoharders (netwerk) smelten niet; elastomeren (enkele bruggen) veren terug.
- Boven de glasovergangstemperatuur Tg wordt een amorfe kunststof zacht en gaat een belast onderdeel kruipen: PLA rond 55-60 °C, PETG rond 80 °C, ABS rond 105 °C.
- Een 3D-print is anisotroop: sterk langs de lagen, zwakker ertussen. Wanden dragen meer dan vulling.
- In een vezelcomposiet dragen de vezels, houdt de matrix ze samen. Langs de vezels: $E_c = V_v E_v + (1 - V_v) E_m$. Koolstofvezel is per kilogram veel stijver dan metaal.
- Biogebaseerd en biologisch afbreekbaar zijn twee verschillende dingen. PLA is industrieel composteerbaar (EN 13432, rond 58 °C en vochtig), niet in de tuin en niet in de natuur.
- De recyclagecode zegt welke kunststof het is, niet of ze gerecycleerd wordt. Mechanisch recycleren vraagt zuivere stromen thermoplast.
- Voor Habitat: PETG in een lichte kleur in de kas, ASA in de zon en buiten, PLA voor prototypes.

BEGRIPPEN

Wat je nu kent

monomeer
Klein molecuul dat zich met vele andere tot een polymeer kan binden, zoals etheen of melkzuur.

polymeer
Reuzenmolecuul uit duizenden aan elkaar gebonden eenheden, zoals polyetheen of PLA.

additiepolymerisatie
Polymerisatie waarbij monomeren met een dubbele binding aan elkaar koppelen zonder bijproduct.

condensatiepolymerisatie
Polymerisatie waarbij bij elke koppeling een klein molecuul, meestal water, vrijkomt.

thermoplast
Kunststof met losse ketens die bij verwarmen zacht wordt en smelt, en opnieuw gevormd kan worden.

thermoharder
Kunststof met een dicht netwerk van bindingen tussen de ketens; smelt niet, ontbindt bij te hoge temperatuur.

elastomeer
Kunststof met enkele bruggen tussen beweeglijke ketens: rekt ver uit en veert terug (rubber).

glasovergangstemperatuur
Temperatuur Tg waarboven de ketens in het amorfe deel beweeglijk worden: de kunststof gaat van hard en glasachtig naar zacht en rubberachtig.

composiet
Materiaal uit twee of meer materialen die samen betere eigenschappen hebben, zoals vezels in een kunststof.

mengregel
Schatting van een eigenschap van een composiet als gewogen gemiddelde naar het volumedeel: $E_c = V_v E_v + (1 - V_v) E_m$.

biogebaseerd
Gemaakt uit plantaardige grondstoffen in plaats van aardolie; zegt niets over afbreekbaarheid.

biologisch afbreekbaar
Kan door micro-organismen omgezet worden in water, CO_2 en biomassa, onder bepaalde omstandigheden.

recyclagecode

Nummer 1 tot 7 in een driehoekje dat zegt uit welke kunststof een voorwerp bestaat; geen belofte dat het gerecycleerd wordt.

HANDBOEK · HOOFDSTUK 18

Warmte berekenen

Hoeveel warmte een stof opslaat, waarom ijs smelten zoveel kost, en hoe groot een warmtebuffer moet zijn

In Fundamental rekende je met een vuistregel: 4,2 kJ om één liter water één graad warmer te maken. Maar hoeveel warmte slaat een aluminium frame op, of de lucht in een kas, of een blok ijs dat smelt? In dit hoofdstuk leer je warmte berekenen voor élke stof met $Q = m \cdot c \cdot \Delta T$, reken je met smelt- en verdampingswarmte, stel je een **warmtebalans** op als je stoffen mengt, en meet je in het labo de soortelijke warmtecapaciteit van een metaal, met een correctie voor het warmteverlies. Voor Habitat bereken je hoe groot een watervat moet zijn om de klimaatmodule 's nachts warm te houden zonder één wattuur uit de batterij.

7× uitleg · 8× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 2× verhaal · 1× het geheel · 1× code · ± 8 lestijden

Energie · Wiskundig modelleren · Materie en chemie

Twee vaten water als warmtebuffer: overdag laden ze op, 's nachts geven ze hun warmte terug.

Na dit hoofdstuk

- Je kan $Q = m \cdot c \cdot \Delta T$ in alle richtingen gebruiken en uitleggen wat de soortelijke warmtecapaciteit zegt over een stof.
- Je kan de warmtecapaciteit $C = m \cdot c$ van een voorwerp of een systeem berekenen en bepalen welk onderdeel de meeste warmte opslaat.
- Je kan rekenen met smelt- en verdampingswarmte ($Q = m \cdot L$) en een opwarmcurve met plateaus tekenen en verklaren.
- Je kan een warmtebalans opstellen bij het mengen van stoffen, ook met ijs, en de eindtemperatuur berekenen.
- Je kan de soortelijke warmtecapaciteit van een metaal meten met een calorimeter en corrigeren voor de warmtecapaciteit van de calorimeter en voor warmteverlies.
- Je kan een warmtebuffer dimensioneren en water vergelijken met andere materialen, met een faseovergang en met elektrisch verwarmen.
- Je kan een warmtebalans programmeren in Python en de uitkomst controleren.

UITLEG · 18.1

Wat je al weet, en wat nu komt

1/22

In hoofdstuk F16 leerde je het verschil tussen **temperatuur** (hoe snel de deeltjes gemiddeld bewegen) en **warmte** (energie die van warm naar koud stroomt). Je rekende met een vuistregel voor water: om 1 L water 1 °C warmer te maken, is ongeveer 4,2 kJ nodig. Met die regel vond je dat het reservoir van de meetkas, 24 L die 's nachts 5 °C afkoelt, ongeveer 504 kJ afgeeft.

uit Fundamental	in dit hoofdstuk
4,2 kJ per liter per graad, alleen voor water (F16)	$Q = m \cdot c \cdot \Delta T$ voor elke stof, met c uit een tabel
temperatuur tegenover warmte (F16)	de warmtecapaciteit $C = m \cdot c$ van een voorwerp: wat slaat het meeste op?
smelten, stollen, verdampen, condenseren (F6)	smelt- en verdampingswarmte: warmte zonder temperatuurstijging
de 504 kJ van het reservoir van de meetkas (F16)	een warmtebuffer voor de klimaatmodule berekenen
joule, wattuur, kilowattuur (F15, F21, hoofdstuk 9)	warmte omrekenen en vergelijken met de batterij (hoofdstuk 16)

Die vuistregel roept vragen op. Waarom zou 1 kg aluminium ook 4,2 kJ nodig hebben? (Het is maar 0,9 kJ.) En wat gebeurt er als een blok ijs smelt: waar gaat de warmte heen, als de temperatuur op 0 °C blijft staan? In dit hoofdstuk

- leer je warmte berekenen voor elke stof, en voor een heel voorwerp;
- reken je met de warmte die smelten en verdampen kosten;
- voorspel je de eindtemperatuur als je warm en koud mengt, en meet je in het labo hoe goed die voorspelling klopt;
- bereken je voor Habitat hoe groot een watervat moet zijn om de klimaatmodule 's nachts warm te houden.

VERHAAL · 18.2

Een professor, een fles ijs en 139 verborgen graden

2/22

In de winter van 1761 liet **Joseph Black**, professor in Glasgow, twee gelijke flessen hangen in een grote zaal waar het 47 °F was (ongeveer 8 °C). In de ene zat ijs van 32 °F, in de andere evenveel ijskoud water van 33 °F. Beide kregen van de lucht in de zaal even snel warmte, want ze hingen er op dezelfde manier en verschilden bijna niet in temperatuur.

Het water warmde gewoon op, tot 40 °F: het had 7 'graden warmte' gekregen, zoals Black het noemde. Het ijs had **21 keer zo lang** nodig om te smelten en tot dezelfde 40 °F te komen. Het had dus $21 \times 7 = 147$ graden warmte gekregen, waarvan maar 8 zichtbaar waren op de thermometer. De overige 139 graden waren 'verdwenen' in het smelten. Black noemde die warmte **latent**: verborgen.

Omgerekend vond Black dat het smelten van ijs evenveel warmte kost als dezelfde massa water 77 °C warmer maken: ongeveer 320 kJ per kilogram. De waarde van vandaag is 334 kJ per kilogram, 80 °C water of 143 °F. Met twee flessen, een thermometer en een klok zat hij er minder dan 4 % naast.

Black maakte als een van de eersten een scherp onderscheid tussen de *hoeveelheid* warmte en de *temperatuur*, en merkte op dat gelijke massa's van verschillende stoffen niet evenveel warmte nodig

hebben om even warm te worden. Aan dezelfde universiteit werkte een jonge instrumentmaker met wie hij bevriend raakte: James Watt, die in 1765 de stoommachine veel zuiniger maakte met een afzonderlijke condensor.

Waarom dit verhaal ertoe doet

Black mat warmte zonder één joule te kennen: hij vergeleek tijden en temperaturen onder dezelfde omstandigheden. Dat is precies wat een calorimeter doet (verder in dit hoofdstuk). En de verborgen warmte van smelten en verdampen is geen curiositeit: ze beschermt bloesems tegen de vorst en koelt de kweekruimte van Habitat.

Twee flessen in dezelfde zaal: het ijs had 21 keer zo lang nodig om 40 °F te bereiken als het water.

UITLEG · 18.3

De soortelijke warmtecapaciteit

3/22

Hoeveel warmte Q nodig is om een stof op te warmen, hangt af van drie dingen: **hoeveel** stof (de massa m), **hoeveel graden** warmer (de temperatuurstijging ΔT) en **welke** stof. Twee keer zoveel water vraagt twee keer zoveel warmte; twee keer zoveel graden ook. Het derde, de stofeigenschap, heet de **soortelijke warmtecapaciteit** c:

WARMTE OM EEN STOF OP TE WARMEN (OF WARMTE DIE VRIJKOMT BIJ AFKOELEN)

$$Q = m \cdot c \cdot \Delta T$$

Q in J, m in kg, $\Delta T = T_{\text{eind}} - T_{\text{begin}}$ in K (of °C: een verschil is in beide hetzelfde getal), c in J per kg per K, geschreven $\mathrm{J/(kg \cdot K)}$. Is ΔT negatief, dan is ook Q negatief: de stof *geeft* warmte af.

De soortelijke warmtecapaciteit is dus de warmte die 1 kg van de stof 1 K warmer maakt. Voor water is dat 4186 J: de 4,2 kJ van hoofdstuk F16, want 1 L water is 1 kg. Aluminium heeft maar 897 J nodig, koper 385 J. Een koperen pan van 1 kg wordt dus elf keer sneller heet dan 1 kg water op hetzelfde vuur.

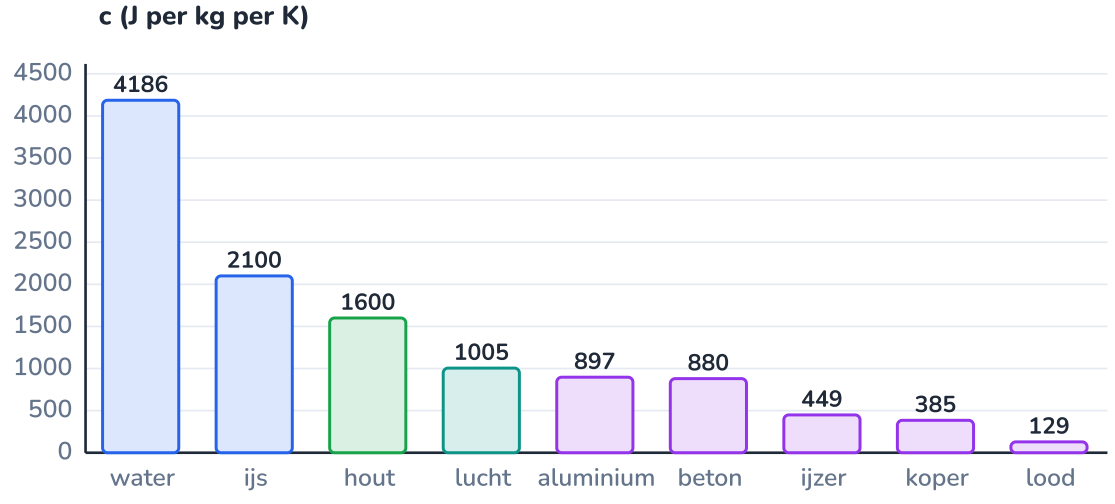
stof	c (J per kg per K)	stof	c (J per kg per K)
water (vloeibaar)	4186	lucht (droog, constante druk)	1005
menselijk lichaam (gemiddeld)	± 3500	aluminium	897
ethanol	2440	beton	± 880 (800 à 1000)
ijs (−10 à 0 °C)	± 2100	glas	± 840
waterdamp (100 °C)	± 2000	zand (droog)	± 830
paraffine (vast)	± 2000	ijzer	449
polypropeen (PP)	± 1900	koper	385
hout (vuren)	± 1600	lood	129
polycarbonaat (PC)	± 1200		

Waarden rond kamertemperatuur. Voor water verandert c tussen 0 en 100 °C minder dan 1 %; de ronde waarde 4,2 kJ per kg per K is die van hoofdstuk F16.

Waarom heeft water zo'n grote c? Warmte die je toevoert, gaat naar de beweging van de deeltjes. In water trekken de moleculen elkaar sterk aan met waterstofbruggen (hoofdstuk 13): een deel van de toegevoerde energie gaat naar het rekken en breken van die bruggen, niet naar snellere beweging. Daardoor stijgt de temperatuur trager. Metalen hebben zware atomen: per kilogram zijn het er weinig, en dus is er per kilogram weinig energie nodig.

Een stof met een grote c wordt warmer

Het omgekeerde is waar. Een grote c betekent dat de stof *veel* warmte nodig heeft voor één graad: ze warmt traag op en koelt traag af. Daarom is het strandzand 's middags gloeiend heet en de zee nog fris, terwijl ze dezelfde zon kregen. En daarom is water de beste warmtebuffer die je makkelijk kan krijgen.



Water springt eruit: per kilogram vier keer zoveel als lucht en bijna elf keer zoveel als koper.

UITGEWERKT VOORBEELD · 18.4

Water, aluminium en lucht opwarmen

(a) Hoeveel warmte is nodig om 1,5 L water van 15 °C aan de kook te brengen (100 °C)? (b) En voor de aluminium pan van 0,80 kg waarin het water zit? (c) De lucht in de kweekruimte van Habitat (binnen 2,40 m × 2,00 m × 2,40 m, dichtheid 1,20 kg/m³) wordt 10 °C warmer. Hoeveel warmte is dat, en hoeveel liter water kan je daarmee 10 °C opwarmen?

GEGEVEN

- water: $m = 1{,}5\,\text{kg}$, $c = 4186\,\text{J/(kg}\cdot\text{K)}$, $\Delta T = 85\,\text{K}$
- pan: $m = 0{,}80\,\text{kg}$, $c = 897\,\text{J/(kg}\cdot\text{K)}$, $\Delta T = 85\,\text{K}$
- lucht: $V = 11{,}52\,\text{m}^3$, $\rho = 1{,}20\,\text{kg/m}^3$, $c = 1005\,\text{J/(kg}\cdot\text{K)}$, $\Delta T = 10\,\text{K}$

GEVRAAGD

- Q_{water}, Q_{pan}, Q_{lucht}
- gelijkwaardige hoeveelheid water

OPLOSSING

1. (a) $Q = m \cdot c \cdot \Delta T = 1{,}5 \cdot 4186 \cdot 85 = 533\,700\,\text{J} \approx 534\,\text{kJ}$.
2. (b) $Q = 0{,}80 \cdot 897 \cdot 85 = 61\,000\,\text{J} = 61\,\text{kJ}$: ongeveer een negende van wat het water vraagt, hoewel de pan meer dan de helft zo zwaar is.
3. (c) Massa van de lucht: $m = \rho \cdot V = 1{,}20 \cdot 11{,}52 = 13{,}8\,\text{kg}$. $Q = 13{,}8 \cdot 1005 \cdot 10 = 139\,000\,\text{J} = 139\,\text{kJ}$.
4. Dezelfde 139 kJ warmen $m = \dfrac{Q}{c \cdot \Delta T} = \dfrac{139\,000}{4186 \cdot 10} = 3{,}3\,\text{kg}$ water 10 °C op: een emmertje van 3,3 L.

ANTWOORD

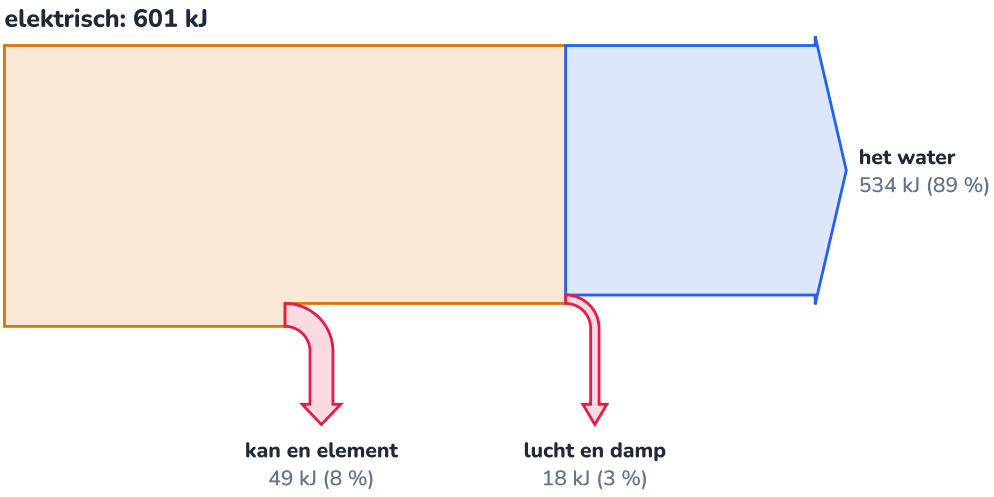
(a) 534 kJ. (b) 61 kJ. (c) 139 kJ, evenveel als nodig is voor 3,3 L water. Alle lucht van een ruimte van 11,5 m³ slaat evenveel warmte op als een emmertje water.

Klopt dit? Per kubieke meter: water $1000 \cdot 4186 = 4{,}19\,\text{MJ}$ per K, lucht $1{,}20 \cdot 1005 = 1{,}2\,\text{kJ}$ per K: water slaat per m³ ongeveer 3500 keer meer warmte op. Daarom koelt een kas zo snel af als de zon weg is, en daarom zet je er water in als buffer, geen extra lucht.

UITGEWERKT VOORBEELD · 18.5

Het rendement van een waterkoker

Een waterkoker van 2200 W brengt 1,5 L water van 15 °C aan de kook in 4 min 33 s. (a) Hoeveel elektrische energie nam hij op? (b) Hoeveel warmte ging naar het water? (c) Wat is het rendement, en waar bleef de rest? De kan is van polypropeen (0,25 kg, $c \approx 1900$), de bodem met het element van roestvast staal (0,20 kg, $c \approx 500$); ze warmen ongeveer even veel op als het water.



De breedte van elke band is evenredig met de energie; bijna negen tiende komt in het water terecht.

GEGEVEN

- $P = 2200\,\mathrm{W}$, $t = 273\,\mathrm{s}$
- water $1{,}5\,\mathrm{kg}$, van 15 naar 100 °C
- kan $0{,}25\,\mathrm{kg}$ PP, bodem $0{,}20\,\mathrm{kg}$ RVS

GEVRAAGD

- E_{el}, Q_{water}, rendement, verliezen

OPLOSSING

1. (a) $E = P \cdot t = 2200 \cdot 273 = 600\,600\,\mathrm{J} \approx 601\,\mathrm{kJ}$ (hoofdstuk 9 en F21).
2. (b) $Q = 1{,}5 \cdot 4186 \cdot 85 = 533{,}7\,\mathrm{kJ}$.
3. (c) Rendement: $\eta = 533{,}7 / 600{,}6 = 89\,\%$.
4. De kan zelf: $C = 0{,}25 \cdot 1900 + 0{,}20 \cdot 500 = 575\,\mathrm{J/K}$, dus $575 \cdot 85 = 48{,}9\,\mathrm{kJ}$. De rest, $600{,}6 - 533{,}7 - 48{,}9 = 18{,}0\,\mathrm{kJ}$, ging naar de lucht rond de kan en met de eerste damp mee.

ANTWOORD

(a) 601 kJ. (b) 534 kJ. (c) Een rendement van 89 %: 49 kJ warmt de kan zelf op, 18 kJ verdwijnt naar de lucht.

Klopt dit? Een waterkoker is een van de efficiëntste manieren om water te verwarmen: het element zit in het water, en de opwarming duurt kort, zodat er weinig tijd is om warmte te verliezen. Een pan op een gaskookplaat haalt veel minder: de hete gassen stromen langs de pan omhoog. Kook alleen het water dat je nodig hebt: de warmte in de kan en in overtollig water is verloren.

UITLEG · 18.6

De warmtecapaciteit van een voorwerp

6/22

Voor een heel voorwerp of een heel systeem, zoals een kas, een pan of een huis, is het handig om $m \cdot c$ samen te nemen. Dat product heet de **warmtecapaciteit** C van het voorwerp:

WARMTECAPACITEIT

$$C = m \cdot c \qquad Q = C \cdot \Delta T \qquad C_{\mathrm{totaal}} = m_1 c_1 + m_2 c_2 + \ldots$$

C in J/K: hoeveel warmte het hele voorwerp nodig heeft voor één kelvin. Bestaat het uit verschillende stoffen die even warm zijn, dan tel je hun warmtecapaciteiten op.

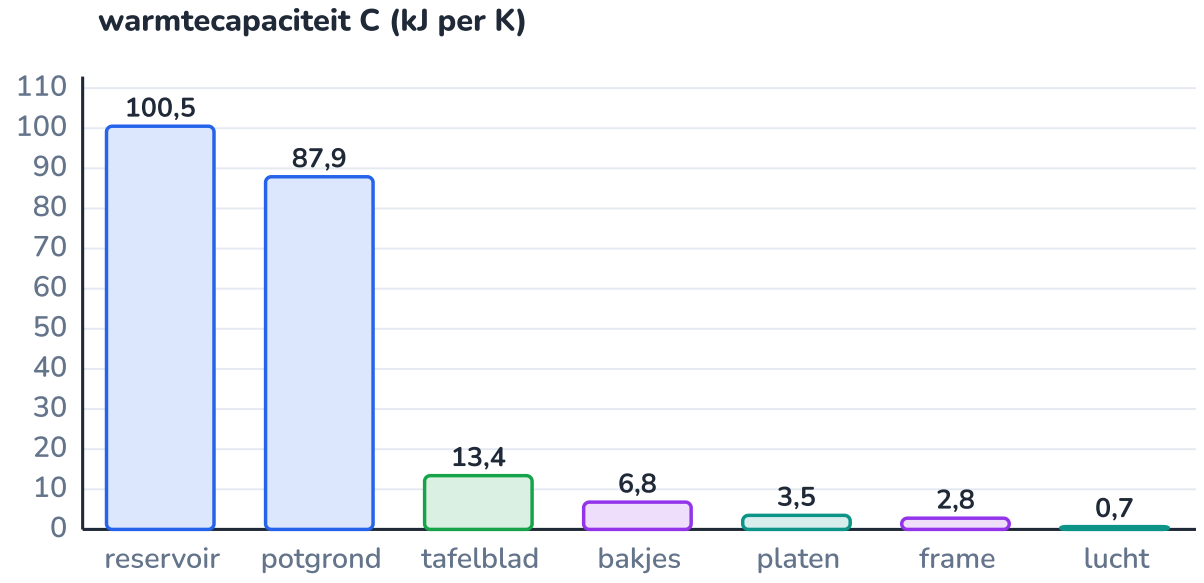
Let op de twee woorden: de soortelijke warmtecapaciteit c is een eigenschap van een **stof** (per kilogram), de warmtecapaciteit C van een **voorwerp** (dit ene voorwerp). Een grote pan en een klein pannetje van hetzelfde staal hebben dezelfde c, maar een verschillende C.

Met C zie je meteen welk onderdeel van een systeem de meeste warmte opslaat. Voor de meetkas van Fundamental ziet dat er zo uit (massa's uit hoofdstuk F7 en F10; van de vochtige potgrond telt hier alleen het water, 21 kg, want dat weegt het zwaarst door):

onderdeel	m (kg)	c (J per kg per K)	$C = m \cdot c$ (kJ per K)	aandeel
water in het reservoir	24,0	4186	100,5	46,6 %
water in de potgrond	21,0	4186	87,9	40,8 %
tafelblad (multiplex)	8,4	1600	13,4	6,2 %
bakjes (PP)	3,6	1900	6,8	3,2 %
platen (PC)	2,9	1200	3,5	1,6 %
frame (aluminium)	3,1	900	2,8	1,3 %
lucht	0,69	1005	0,7	0,3 %
samen			**216**	100 %

Water als maatstaf

Een warmtecapaciteit kan je ook uitdrukken als een hoeveelheid water: het **waterequivalent**. Het tafelblad van 13,4 kJ/K gedraagt zich als $13\,440 / 4186 = 3{,}2$ kg water. Bij een calorimeter heet dat de **waterwaarde** (verder in dit hoofdstuk).



Het water (in het reservoir en in de potgrond) slaat bijna 90 % van de warmte op; de lucht nog geen half procent.

UITGEWERKT VOORBEELD · 18.7

Habitat: de 504 kJ van de meetkas, nu exact

7/22

In hoofdstuk F16 gaf het reservoir van de meetkas (24 L die 's nachts 5 °C afkoelt) met de vuistregel 504 kJ. (a) Reken het exact met $Q = m \cdot c \cdot \Delta T$. (b) Hoe groot is de warmtecapaciteit van de hele meetkas volgens de tabel? Hoeveel warmte geeft ze af als ze 's nachts gemiddeld 5 °C afkoelt? (c) De platen verliezen 14 W per °C verschil met buiten (hoofdstuk F16). Stel dat er alleen lucht in de kas zat. Hoe snel zou ze afkoelen bij 5 °C verschil?

GEGEVEN

- $m = 24\,\text{kg}$, $c = 4186\,\text{J/(kg}\cdot\text{K)}$, $\Delta T = -5\,\text{K}$
- $C_{\text{meetkas}} = 216\,\text{kJ/K}$ (tabel)
- verlies $14\,\text{W}$ per °C; lucht: $0{,}69\,\text{kg}$, $c = 1005\,\text{J/(kg}\cdot\text{K)}$

GEVRAAGD

- $Q_{\text{reservoir}}$
- Q_{meetkas}
- afkoelsnelheid van alleen de lucht

OPLOSSING

1. (a) $Q = 24 \cdot 4186 \cdot (-5) = -502\,000\,\text{J} = -502\,\text{kJ}$. Het minteken zegt: het water geeft 502 kJ af. De vuistregel gaf 504 kJ, want 4,2 is 4186 naar boven afgerond (0,3 %).
2. (b) $Q = C \cdot \Delta T = 216 \cdot 5 \approx 1078\,\text{kJ}$ (plus nog wat uit de droge stof van de grond). Twee keer zoveel als het reservoir alleen: het water in de potgrond is een tweede buffer.
3. (c) Bij 5 °C verschil verliest de kas $14 \cdot 5 = 70\,\text{W}$: 70 J per seconde. De lucht alleen heeft $C = 0{,}69 \cdot 1005 = 693\,\text{J/K}$, dus ze zou $70/693 = 0{,}10\,\text{K}$ per seconde afkoelen: 6 °C per minuut.

ANTWOORD

(a) 502 kJ. (b) Ongeveer 1078 kJ voor de hele meetkas. (c) Alleen lucht zou 6 °C per minuut afkoelen; dankzij het water gaat het 300 keer trager.

Klopt dit? Het verschil tussen (c) en de werkelijkheid is de kern van dit hoofdstuk: wat een ruimte 's nachts warm houdt, is niet de lucht maar de massa water erin. De warmte stroomt van het water naar de lucht en van de lucht door de platen naar buiten. Hoe snel dat laatste gaat, bereken je in hoofdstuk 19.

UITLEG · 18.8

Smelten en verdampen: warmte zonder temperatuurstijging

8/22

Verwarm een blok ijs van −20 °C met een constant vermogen en meet de temperatuur. Eerst stijgt ze, tot 0 °C. Dan blijft ze staan, hoewel je blijft verwarmen: het ijs smelt. Pas als al het ijs gesmolten is, stijgt de temperatuur weer, tot 100 °C. Daar blijft ze opnieuw staan, veel langer: het water kookt weg. Zo'n grafiek heet een **opwarmcurve**; de horizontale stukken zijn de **plateaus**.

Waar gaat de warmte op een plateau heen? Niet naar snellere beweging (de temperatuur stijgt niet), maar naar het **losmaken van de deeltjes** (hoofdstuk F6). In ijs zitten de watermoleculen vast in een rooster; om ze los te krijgen, moeten veel waterstofbruggen breken. Bij verdampen moeten de moleculen helemaal van elkaar los, en dat kost nog veel meer:

SMELT- EN VERDAMPINGSWARMTE

$$Q = m \cdot L_s \qquad Q = m \cdot L_v$$

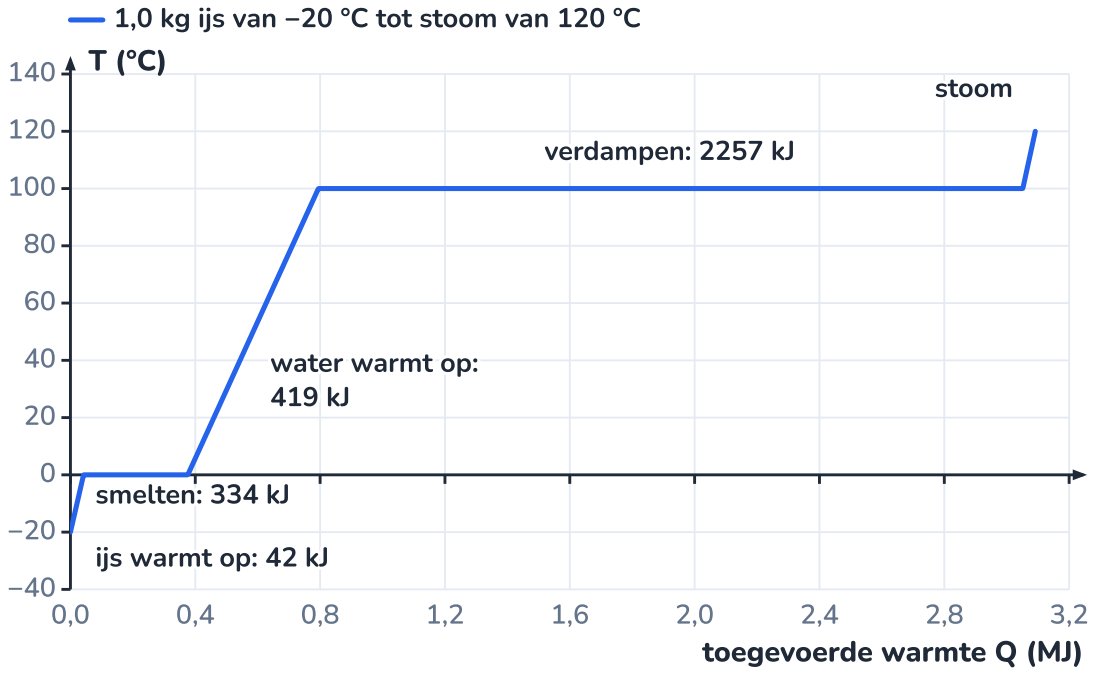
L_s: smeltwarmte, L_v: verdampingswarmte, in J/kg (of kJ/kg). Voor water: $L_s = 334\,\text{kJ/kg}$, $L_v = 2257\,\text{kJ/kg}$ bij 100 °C. Bij stollen en condenseren komt precies dezelfde warmte weer vrij.

smeltwarmte L_s	kJ/kg	verdampingswarmte L_v	kJ/kg
ijs (0 °C)	334	water bij 100 °C	2257
aluminium (660 °C)	397	water bij 20 °C	2454
ijzer (1538 °C)	247	water bij 0 °C	2501
koper (1085 °C)	205	ethanol bij 78 °C	± 850
paraffine (± 20 à 60 °C, naargelang de soort)	150 à 250		
lood (327 °C)	23		

Tussen haakjes het smeltpunt. Stollen en condenseren geven dezelfde hoeveelheid warmte terug.

Vergelijk de getallen voor water. Smelten kost evenveel als hetzelfde water 80 °C warmer maken. Verdampen kost bijna zeven keer zoveel als smelten, en meer dan vijf keer zoveel als het water van 0 tot 100 °C opwarmen. In de opwarmcurve is het verdampingsplateau het langste stuk: 73 % van alle warmte gaat daarnaar.

> ⚠ **Verdampen gebeurt ook onder 100 °C**
>
> Water verdampt bij elke temperatuur, alleen trager (was die droogt, zweet dat opdroogt). Bij lagere temperatuur kost het zelfs iets *meer* warmte per kilogram: 2454 kJ bij 20 °C, 2501 kJ bij 0 °C. Bij 100 °C kookt het water: dan vormen er zich ook dampbellen in de vloeistof zelf.



Bij constante toevoer is de horizontale as ook een tijdas: het kookplateau duurt bijna zeven keer zo lang als het smeltplateau.

UITGEWERKT VOORBEELD · 18.9

Van ijsblok tot stoom

9/22

Een blok ijs van 1,0 kg en −20 °C wordt helemaal omgezet in stoom van 120 °C. (a) Bereken de warmte voor elke fase en de totale warmte. (b) Hoe lang duurt dat op een kookplaat die 1500 W aan de pan afgeeft? (c) Welke fase duurt het langst?

GEGEVEN

GEVRAAGD

- $m = 1{,}0\,\text{kg}$; $c_{\text{ijs}} = 2100$, $c_{\text{water}} = 4186$, $c_{\text{stoom}} \approx 2000\,\text{J/(kg}\cdot\text{K)}$
- $L_s = 334\,\text{kJ/kg}$, $L_v = 2257\,\text{kJ/kg}$
- $P = 1500\,\text{W}$

- warmte per fase
- tijd
- langste fase

OPLOSSING

1. Ijs opwarmen van −20 tot 0 °C: $Q_1 = 1{,}0 \cdot 2100 \cdot 20 = 42\,\text{kJ}$.
2. Smelten bij 0 °C: $Q_2 = 1{,}0 \cdot 334 = 334\,\text{kJ}$.
3. Water opwarmen van 0 tot 100 °C: $Q_3 = 1{,}0 \cdot 4186 \cdot 100 = 419\,\text{kJ}$.
4. Verdampen bij 100 °C: $Q_4 = 1{,}0 \cdot 2257 = 2257\,\text{kJ}$.
5. Stoom opwarmen van 100 tot 120 °C: $Q_5 = 1{,}0 \cdot 2000 \cdot 20 = 40\,\text{kJ}$.
6. Samen: $42 + 334 + 419 + 2257 + 40 = 3092\,\text{kJ} \approx 3{,}1\,\text{MJ}$. Tijd: $t = Q/P = 3\,092\,000/1500 = 2061\,\text{s} \approx 34\,\text{min}$, waarvan $2\,257\,000/1500 = 1505\,\text{s}$ (25 min) voor het verdampen.

ANTWOORD

(a) 42, 334, 419, 2257 en 40 kJ: samen 3,1 MJ (0,86 kWh). (b) Ongeveer 34 minuten. (c) Het verdampen: 25 van de 34 minuten.

Klopt dit? Wie ooit een pan water liet droogkoken, weet het: aan de kook brengen gaat snel, helemaal wegkoken duurt lang. De uitkomst in kWh is handig om te onthouden: 1 kg water verdampen kost ongeveer 0,63 kWh. Daarom is een droogkast of een wasdroger een grote verbruiker.

VERHAAL · 18.10

IJs tegen de vorst: bloesems in Haspengouw

10/22

In april staan de fruitbomen van Haspengouw in bloei. Een heldere nacht met vorst kan dan een hele oogst vernietigen: bij enkele graden onder nul bevriest het vocht in de bloem, en ijskristallen scheuren de cellen. Sommige fruittelers doen daarom iets wat tegen elk gevoel ingaat: ze zetten in de vriesnacht de **beregening** aan en spuiten water over de bomen.

Het water bevriest op de takken en de bloesems. Bij het **stollen** komt de smeltwarmte vrij: 334 kJ per kilogram water. Zolang er vloeibaar water bij blijft komen dat bevriest, blijft het mengsel van water en ijs op 0 °C, en daarmee ook de bloem erin. Zo'n bloesem kan rond 0 °C blijven, ook als het in de boomgaard −5 à −6 °C wordt. Telers rekenen met ongeveer **30 000 liter water per hectare per uur**: 3 mm per uur.

Reken na wat dat oplevert. Per uur bevriest op elke vierkante meter 3 kg water, dus komt er $3 \cdot 334 = 1002\,\text{kJ}$ vrij: $1\,002\,000\,\text{J}/3600\,\text{s} = 278\,\text{W}$ per vierkante meter. Evenveel als een straalkacheltje boven elke vierkante meter boomgaard, maar dan zonder brandstof.

Nooit halverwege stoppen

De beregening moet blijven draaien tot de zon het ijs heeft doen smelten. Stopt ze te vroeg, dan verdampt en sublimeert er water uit het ijs, en dat *kost* warmte: per kilogram ongeveer zeven keer meer dan het bevriezen opleverde. De bloesem wordt dan kouder dan zonder beregening. Om dezelfde reden werkt de techniek slecht als het hard waait (opdracht 18.13).

Bloesems in een laagje ijs: zolang er water bij komt dat bevriest, blijft de bloem rond 0 °C.

UITLEG · 18.11

Verdampen koelt, condenseren verwarmt

11/22

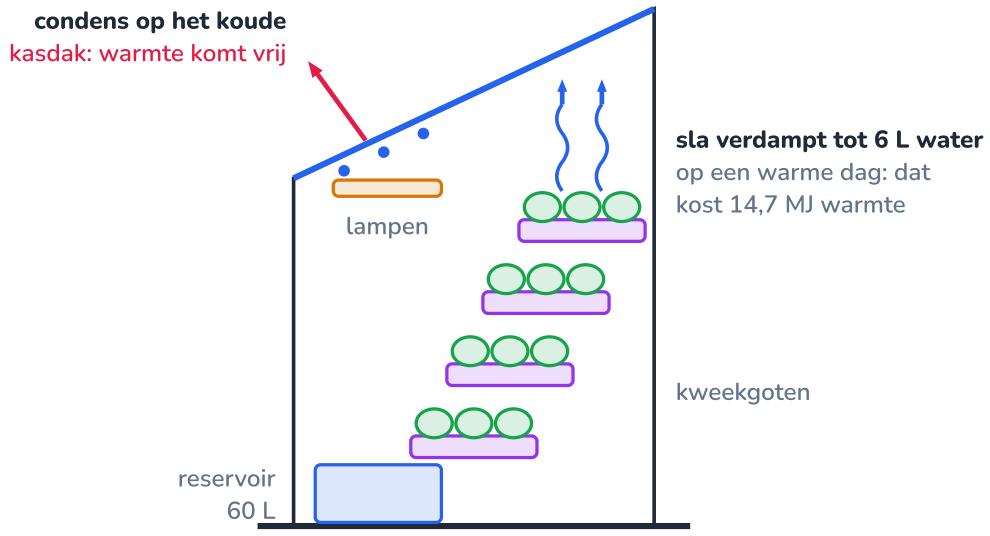
Als water verdampt, neemt het de verdampingswarmte mee uit zijn omgeving. Zo koelt je lichaam af als zweet opdroogt: per liter zweet 2,4 MJ, genoeg om 70 kg lichaam (± 3500 J per kg per K) bijna 10 °C af te koelen. En zo koelen **planten** hun bladeren: via de huidmondjes verdampt water (**transpiratie**, hoofdstuk F27).

In de kweekruimte van Habitat verdampen de slaplanten op een warme dag tot 6 L water (hoofdstuk 13). Bij ongeveer 20 °C kost dat $6 \cdot 2454\,\text{kJ} = 14{,}7\,\text{MJ}$, of 4,1 kWh: meer dan de lampen per dag aan elektrische energie opnemen (3,0 kWh, hoofdstuk 20). Die warmte komt uit het zonlicht op de bladeren en uit de lucht: verdampen is een airco.

Maar de energie is niet weg. Ze zit als latente warmte in de waterdamp, en komt vrij waar de damp **condenseert**. 's Nachts is dat vooral het koude kasdak: de druppels die je 's ochtends onder het dak ziet, hebben daar hun warmte afgegeven, en die lekt door het dak naar buiten. Wie de waterdamp met de ventilatie naar buiten blaast, blaast ook de warmte naar buiten. Die kringloop van water en warmte moet de klimaatregeling van de module beheren (hoofdstuk 31).

Koelen met verdamping

In droge, warme streken koelt men gebouwen en kassen met natte matten en een ventilator: de lucht wordt koeler doordat ze water verdampt. In een vochtige kas werkt dat slecht: lucht die al bijna verzadigd is, kan weinig water meer opnemen (hoofdstuk F6, dauwpunt).



Doorsnede van de kweekruimte (op schaal): de sla verdampt water, de damp condenseert tegen het koude kasdak en geeft daar zijn warmte af.

UITLEG · 18.12

De warmtebalans: wat de ene afgeeft, neemt de andere op

12/22

Giet warm water bij koud water, of laat een heet blokje metaal in koud water zakken. Warmte stroomt van warm naar koud, tot alles dezelfde **eindtemperatuur** T_e heeft. Als er geen warmte naar buiten lekt (een goed geïsoleerd vat, of een snelle menging), dan gaat er ook geen warmte verloren: behoud van energie (hoofdstuk 9). Wat de warme stof afgeeft, neemt de koude op:

WARMTEBALANS BIJ MENGEN (1 IS WARM, 2 IS KOUD)

$$Q_{\text{af}} = Q_{\text{op}} \quad \Longleftrightarrow \quad m_1 c_1 (T_1 - T_e) = m_2 c_2 (T_e - T_2)$$

Elke term is positief: de warme stof koelt af van T_1 tot T_e, de koude warmt op van T_2 tot T_e.

Los op naar T_e en je krijgt een **gewogen gemiddelde**, waarin elke stof meetelt met haar warmtecapaciteit:

EINDTEMPERATUUR (ZONDER FASEOVERGANG)

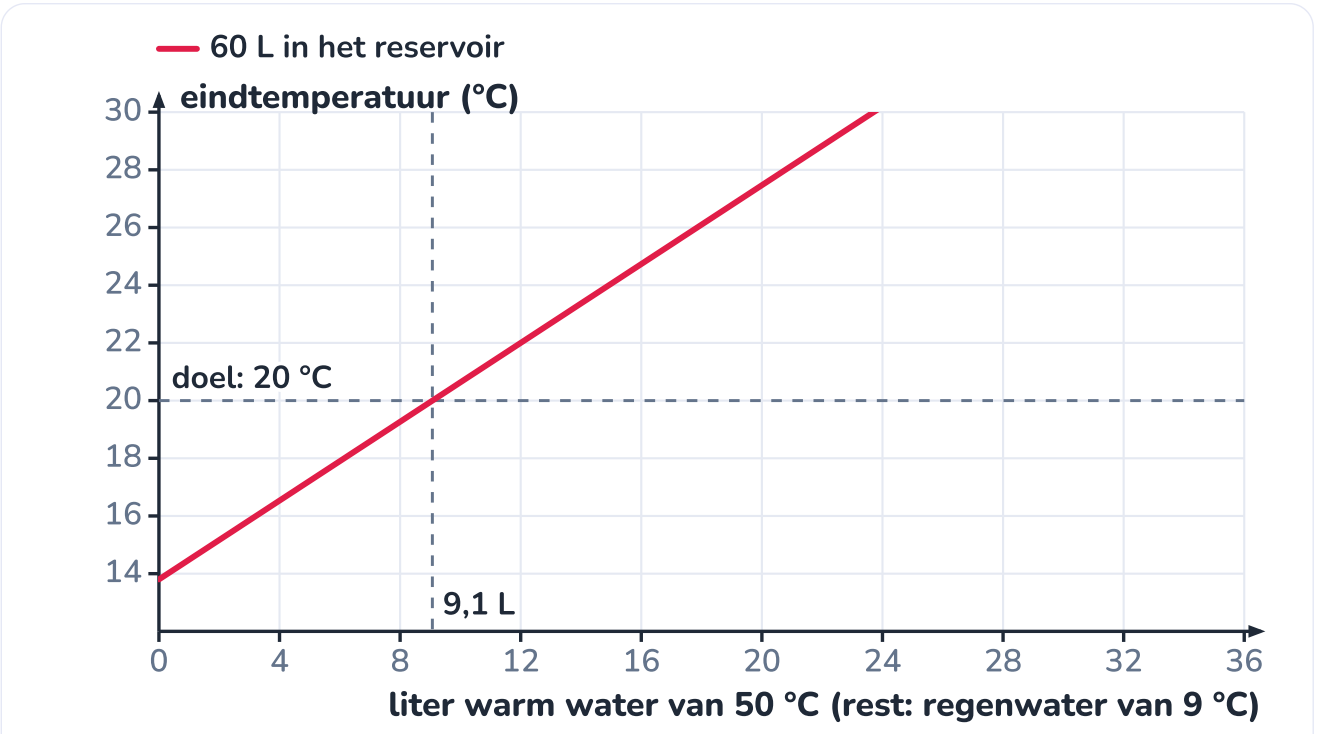
$$T_e = \frac{m_1 c_1 T_1 + m_2 c_2 T_2 + \dots}{m_1 c_1 + m_2 c_2 + \dots}$$

Zelfde stof (c valt weg): $T_e = \dfrac{m_1 T_1 + m_2 T_2}{m_1 + m_2}$. Evenveel warm als koud water geeft precies het gemiddelde.

Twee gevolgen. De eindtemperatuur ligt altijd **tussen** de begintemperaturen, en dichter bij die van de stof met de grootste $m \cdot c$. En de formule werkt voor meer dan twee stoffen: tel gewoon alle termen op, ook de calorimeter zelf.

Warm plus koud is altijd het gemiddelde

Alleen bij gelijke warmtecapaciteit. Een aluminium blokje van 250 g en 100 °C in 200 g water van 18 °C geeft geen 59 °C, maar ongeveer 35 °C: het water heeft vier keer de warmtecapaciteit van het blokje en 'trekt' de eindtemperatuur naar zich toe.



Bijvullen van het reservoir (voorbeeld hierna): elke liter warm water maakt het geheel ongeveer 0,7 °C warmer.

UITGEWERKT VOORBEELD · 18.13

Habitat: het reservoir bijvullen op 20 °C

13/22

In het voedingsreservoir van de klimaatmodule staat nog 24 L oplossing van 21 °C. Het moet worden bijgevuld tot 60 L met regenwater van 9 °C uit de opvangtank. Wortels houden niet van een koude douche: de oplossing moet rond 20 °C blijven. (a) Welke temperatuur krijg je met alleen regenwater? (b) Hoeveel liter van de 36 L moet je vervangen door warm water van 50 °C om op 20 °C uit te komen? (c) Hoeveel energie kost het om dat warme water op te warmen van 9 tot 50 °C?

GEGEVEN

- 24 L van 21 °C; aanvullen met 36 L
- regenwater 9 °C, warm water 50 °C
- 1 L oplossing $\approx$ 1 kg, $c = 4186\,\mathrm{J/(kg\cdot K)}$

GEVRAAGD

- T_e met alleen regenwater
- liter warm water
- energie voor het warme water

OPLOSSING

1. (a) Eén stof, dus c valt weg: $T_e = \dfrac{24\cdot 21 + 36\cdot 9}{60} = \dfrac{504+324}{60} = 13{,}8\,^\circ\mathrm{C}$. Te koud.
2. (b) Noem x het aantal liter warm water. Dan komt er $36 - x$ liter regenwater bij: $24\cdot 21 + 50\,x + 9\,(36 - x) = 60\cdot 20$.
3. Uitwerken: $504 + 50\,x + 324 - 9\,x = 1200$, dus $41\,x = 372$ en $x = 9{,}1\,\mathrm{L}$ warm water en 26,9 L regenwater.
4. (c) $Q = m\cdot c\cdot \Delta T = 9{,}1\cdot 4186\cdot (50-9) = 1{,}56\,\mathrm{MJ}$, of $1\,560\,000/3\,600\,000 = 0{,}43\,\mathrm{kWh}$.

ANTWOORD

(a) 13,8 °C. (b) 9,1 L warm water van 50 °C en 26,9 L regenwater. (c) 1,56 MJ = 0,43 kWh, bijna een derde van de bruikbare batterij (1,2 kWh, hoofdstuk 16).

Klopt dit? Het resultaat past in de grafiek van het vorige blad (20 °C bij 9,1 L). En de energie: om 60 L van 13,8 naar 20 °C te brengen, is $60\cdot 4186\cdot 6{,}2 = 1{,}56\,\mathrm{MJ}$ nodig, precies hetzelfde. Of je het water vooraf of achteraf opwarmt, de energie blijft dezelfde. Slimmer:

laat het regenwater eerst een dag in de warme module op temperatuur komen.

UITGEWERKT VOORBEELD · 18.14

IJsblokjes in de limonade

14/22

Je doet twee ijsblokjes van samen 50 g, rechtstreeks uit de diepvries (−18 °C), in 0,30 kg limonade van 22 °C. Limonade is bijna water. (a) Smelt al het ijs? (b) Wat is de eindtemperatuur? (c) En als je er 200 g ijs in doet?

GEGEVEN

- limonade: $0{,}30\,\text{kg}$, 22 °C, $c = 4186\,\text{J}/(\text{kg}\cdot\text{K})$
- ijs: $0{,}050\,\text{kg}$ (of $0{,}200\,\text{kg}$), −18 °C, $c_{\text{ijs}} = 2100\,\text{J}/(\text{kg}\cdot\text{K})$, $L_s = 334\,\text{kJ}/\text{kg}$

GEVRAAGD

- smelt alles?
- T_e

OPLOSSING

1. Werk met een boekhouding. Hoeveel warmte kan de limonade afgeven tot ze 0 °C is? $0{,}30 \cdot 4186 \cdot 22 = 27\,630\,\text{J}$.
2. Hoeveel vraagt het ijs om op 0 °C te komen en te smelten? $0{,}050 \cdot 2100 \cdot 18 + 0{,}050 \cdot 334\,000 = 1890 + 16\,700 = 18\,590\,\text{J}$. Dat is minder dan 27 630 J: **al het ijs smelt**.
3. De rest, $27\,630 - 18\,590 = 9040\,\text{J}$, warmt al het water (nu 0,35 kg) op vanaf 0 °C: $T_e = \dfrac{9040}{0{,}35 \cdot 4186} = 6{,}2\,^\circ\text{C}$.
4. (c) Met 200 g ijs vraagt het ijs $0{,}200 \cdot 2100 \cdot 18 + 0{,}200 \cdot 334\,000 = 74\,360\,\text{J}$: veel meer dan de 27 630 J. Niet alles smelt, en het mengsel blijft op 0 °C. Na het opwarmen van het ijs ($7560\,\text{J}$) blijft $27\,630 - 7560 = 20\,070\,\text{J}$ over om te smelten: $20\,070/334\,000 = 0{,}060\,\text{kg}$. Er blijft 140 g ijs drijven.

ANTWOORD

(a) Ja. (b) 6,2 °C. (c) Met 200 g ijs: 0 °C, en 140 g ijs blijft over.

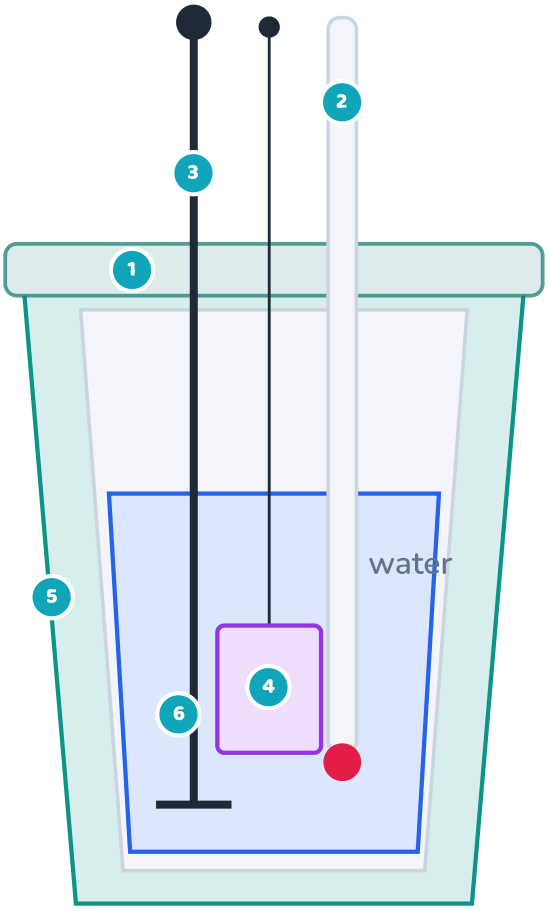
Klopt dit? Een faseovergang maakt een warmtebalans in stukken: controleer eerst wat de warmte 'kan kopen' tot 0 °C, en pas dan de formule toe. Wie meteen de formule voor T_e invult voor (c), vindt een temperatuur onder 0 °C met gesmolten ijs: onmogelijk. Het programma op het codeblad doet dezelfde controle.

ONTLEED · 18.15

De calorimeter

15/22

Een **calorimeter** is een vat waarin je warmte-uitwisseling zo goed mogelijk afschermt van de omgeving, zodat de warmtebalans klopt. In het labo volstaan twee piepschuimbekers die in elkaar passen, met een deksel. Elk onderdeel heeft een taak.



Twee bekers met een luchtlaag, een deksel met gaatjes voor de thermometer, de roerder en de draad.

1 Deksel
Houdt de warme lucht en de waterdamp boven het water binnen. Zonder deksel verlies je vooral warmte door verdampen en stroming.

2 Thermometer
Een temperatuursensor met 0,1 °C resolutie, halverwege het water. Lees om de 30 s af, ook vóór het mengen: die meting heb je nodig voor de correctie.

3 Roerder
Een ring die je op en neer beweegt. Zonder roeren is het water bovenaan warmer dan onderaan, en meet de thermometer maar één plek.

4 Het blokje aan een draad
Van tevoren minstens tien minuten in kokend water: dan heeft het de temperatuur van dat water. Snel overbrengen, zonder water mee te nemen.

5 Twee bekers, een luchtlaag
Piepschuim geleidt slecht, en de stilstaande lucht tussen de bekers nog slechter (hoofdstuk 19). De binnenbeker neemt zelf toch wat warmte op: de waterwaarde.

6 Het water
Gewogen op de balans (0,1 g), niet afgemeten met een maatbeker: de massa telt in de balans, en een maatbeker is minder nauwkeurig.

UITGEWERKT VOORBEELD · 18.16

De waterwaarde van de calorimeter

16/22

Een calorimeter neemt zelf ook warmte op: de binnenbeker, de roerder, de sensor. Om die warmtecapaciteit C_{cal} te meten, giet je 150 g water van 52,0 °C in de calorimeter waarin al 150 g water van 18,0 °C staat (en die dus ook 18,0 °C is). Na roeren lees je 34,6 °C. Bereken C_{cal} en de waterwaarde.

GEGEVEN

GEVRAAGD

- koud: $m_2 = 0{,}150\,\text{kg}$ water en de calorimeter, $T_2 = 18{,}0\,^\circ\text{C}$
- warm: $m_1 = 0{,}150\,\text{kg}$, $T_1 = 52{,}0\,^\circ\text{C}$
- $T_e = 34{,}6\,^\circ\text{C}$

- C_{cal} in J/K en als waterwaarde

OPLOSSING

1. Zonder calorimeter zou T_e het gemiddelde zijn: 35,0 °C. Het is 0,4 °C minder: de calorimeter heeft warmte opgenomen.
2. Warmtebalans: $m_1 c\,(T_1 - T_e) = m_2 c\,(T_e - T_2) + C_{\text{cal}}\,(T_e - T_2)$.
3. Afgegeven: $0{,}150 \cdot 4186 \cdot 17{,}4 = 10\,926\,\text{J}$. Opgenomen door het koude water: $0{,}150 \cdot 4186 \cdot 16{,}6 = 10\,423\,\text{J}$.
4. De calorimeter nam het verschil op: $10\,926 - 10\,423 = 503\,\text{J}$, voor een stijging van 16,6 °C: $C_{\text{cal}} = 503/16{,}6 = 30\,\text{J/K}$.
5. Waterwaarde: $30/4186 = 0{,}0072\,\text{kg}$: de calorimeter gedraagt zich als 7 g extra water.

ANTWOORD

$C_{\text{cal}} \approx 30\,\text{J/K}$**, een waterwaarde van ongeveer 7 g.**

Klopt dit? Klein, maar niet verwaarloosbaar: 7 g naast 150 g is 5 %. Eén meting heeft wel een grote onzekerheid, want 0,1 °C fout op T_e verandert C_{cal} met ongeveer 8 J/K (hoofdstuk 1). Herhaal dus drie keer en neem het gemiddelde. In de proef hierna rekenen we met $C_{\text{cal}} = 30\,\text{J/K}$.

PROEF · 18.17

De soortelijke warmtecapaciteit van aluminium meten

17/22

ONDERZOEKSVRAAG

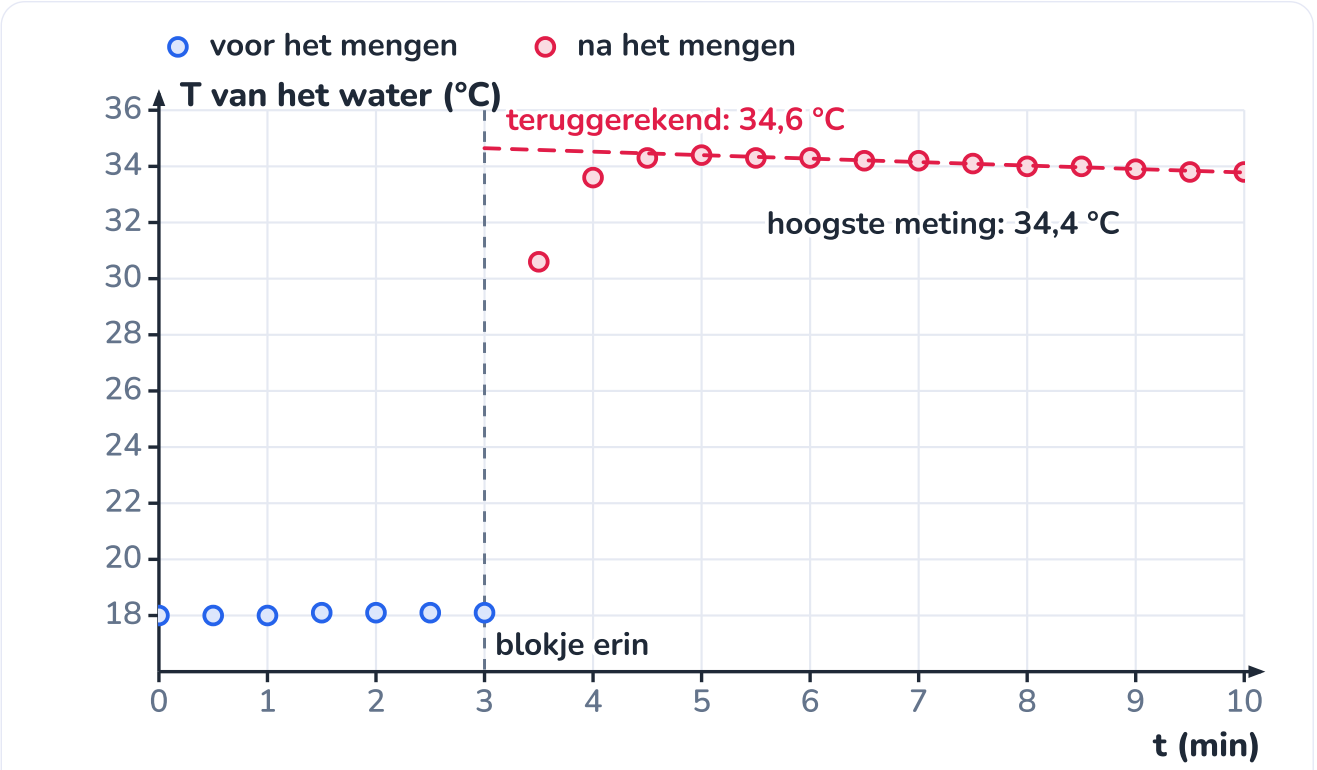
Hoe groot is c voor aluminium, en hoeveel scheelt het als je corrigeert voor de calorimeter en voor warmteverlies?

MATERIAAL

calorimeter van het vorige blad ($C_{\text{cal}} = 30\,\text{J/K}$), roerder, temperatuursensor · aluminium blokje van 250,0 g aan een draad · kookpot met water op een kookplaat · balans (0,1 g), stopwatch, keukenpapier

Veiligheid

Kokend water en een heet blokje: werk met de draad, niet met je vingers, en zet de kookpot achteraan op de kookplaat.



De stippellijn trekt de afkoeling na het mengen terug tot het moment van mengen: dat geeft een betere eindtemperatuur dan de hoogste meting.

WERKWIJZE

1. Hang het blokje minstens tien minuten in kokend water. Noteer de temperatuur van het kokende water: dat is de begintemperatuur van het blokje.
2. Weeg 200,0 g water van ongeveer 18 °C in de binnenbeker, zet het deksel erop en lees drie minuten lang om de 30 s de temperatuur af.
3. Breng het blokje op $t = 3{,}0$ min snel over (even afdeppen op keukenpapier) en sluit het deksel. Roer.
4. Lees tot $t = 10$ min om de 30 s af. Zet alle metingen in een grafiek.
5. Trek door de punten van na het mengen (vanaf 5 min, als alles gelijk verdeeld is) een rechte en verleng ze terug tot $t = 3{,}0$ min.

WAARNEMING

Het kokende water was 99,5 °C (de luchtdruk was die dag iets lager dan normaal). Het water in de calorimeter was vlak voor het mengen 18,1 °C. De grafiek toont de metingen. De hoogste meting is 34,4 °C, bij 5 minuten; daarna daalt de temperatuur langzaam, 0,12 °C per minuut. De teruggetrokken rechte snijdt $t = 3{,}0$ min bij **34,6 °C**.

berekening met	T_e (°C)	C_{cal}	c_{Al} (J per kg per K)	afwijking van 897
alleen het water, hoogste meting	34,4	0	838	−6,5 %
met de calorimeter, hoogste meting	34,4	30 J/K	869	−3,2 %
met de calorimeter, gecorrigeerd	34,6	30 J/K	882	−1,7 %

VERKLARING

De warmtebalans: het blokje geeft af wat het water en de calorimeter opnemen,

$$c_{\text{Al}} = \frac{(m_w\,c_w + C_{\text{cal}})\,(T_e - T_w)}{m_{\text{Al}}\,(T_{\text{Al}} - T_e)} = \frac{(0{,}200 \cdot 4186 + 30) \cdot (34{,}6 - 1}{0{,}250 \cdot (99{,}5 - 34{,}6)}$$

Waarom teruggerekend? Het mengen duurt een minuut of twee. In die tijd lekt er al warmte naar buiten, zodat de hoogste meting te laag is. Na het mengen daalt de temperatuur ongeveer lineair: dat is het lek. Door die rechte terug te trekken tot het moment van mengen, schat je de temperatuur die het water gehad zou hebben als het mengen ogenblikkelijk was gebeurd.

Waarom nog steeds te laag? Er blijft een systematische fout van ongeveer 2 %. Het blokje koelt af in de seconden tussen de kookpot en de calorimeter, zodat het met minder dan 99,5 °C in het water komt. Die fout kan je niet uit de grafiek halen: je verkleint ze door sneller over te brengen, of door een zwaarder blokje te nemen (dat relatief minder afkoelt).

CODE · 18.18

Een warmtebalans in Python

18/22

Een warmtebalans is een optelsom, en met een faseovergang een optelsom met een keuze. Dat is werk voor een computer. De functie `mengen` berekent het gewogen gemiddelde voor zoveel stoffen als je wil; `met_ijs` maakt dezelfde boekhouding als in het voorbeeld van de limonade: eerst kijken wat de warmte kan 'kopen', dan pas rekenen.

Python mengen.py 32 regels

```python
# mengen.py - de eindtemperatuur van een mengsel, ook met ijs dat (gedeeltelijk) smelt
C_WATER = 4186.0     # J/(kg K)
C_IJS = 2100.0       # J/(kg K)
L_SMELT = 334e3      # J/kg


def mengen(delen):
    """delen: lijst van (massa in kg, c in J/(kg K), temperatuur in °C), zonder faseovergang."""
    teller = sum(m * c * T for m, c, T in delen)
    noemer = sum(m * c for m, c, T in delen)
    return teller / noemer                              # een gewogen gemiddelde


def met_ijs(m_water, T_water, m_ijs, T_ijs):
    """Water boven 0 °C met ijs van T_ijs (<= 0 °C). Geeft (eindtemperatuur, overgebleven ijs in kg)."""
    q_beschikbaar = m_water * C_WATER * T_water         # het water afkoelen tot 0 °C
    q_opwarmen = m_ijs * C_IJS * (0 - T_ijs)            # het ijs opwarmen tot 0 °C
    q_smelten = m_ijs * L_SMELT                         # het ijs smelten
    if q_beschikbaar < q_opwarmen:
        return None, m_ijs                              # er bevriest water: valt buiten dit programma
    if q_beschikbaar < q_opwarmen + q_smelten:          # niet alles smelt: het mengsel blijft op 0 °C
        gesmolten = (q_beschikbaar - q_opwarmen) / L_SMELT
        return 0.0, m_ijs - gesmolten
    rest = q_beschikbaar - q_opwarmen - q_smelten       # alles gesmolten: de rest warmt al het water op
    return rest / ((m_water + m_ijs) * C_WATER), 0.0


```

```python
print("reservoir bijvullen:", round(mengen([(24, C_WATER, 21), (9.1, C_WATER, 50), (26.9, C_WATER, 9)]), 1), "°C")
print("aluminium in de calorimeter:", round(mengen([(0.250, 897, 99.5), (0.200, C_WATER, 18.1), (1, 30, 18.1)]), 1), "°C")
for m_ijs in (0.050, 0.200):
    T, rest = met_ijs(0.30, 22.0, m_ijs, -18.0)
    print(f"limonade + {m_ijs * 1000:.0f} g ijs: {T:.1f} °C, {rest * 1000:.0f} g ijs over")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
reservoir bijvullen: 20.0 °C
aluminium in de calorimeter: 34.8 °C
limonade + 50 g ijs: 6.2 °C, 0 g ijs over
limonade + 200 g ijs: 0.0 °C, 140 g ijs over
```

Drie controles tegelijk: het reservoir komt op 20 °C uit (voorbeeld hiervoor), de calorimeter met het aluminium blokje geeft de eindtemperatuur die je zou meten zonder warmteverlies en met een blokje dat echt 99,5 °C is, en de limonade klopt met het uitgewerkte voorbeeld. Merk op dat de calorimeter in `mengen` als een 'stof' van 1 kg met $c = 30$ meetelt: alleen het product $m \cdot c$ doet ertoe. In opdracht 18.11 breid je het programma uit.

UITLEG · 18.19

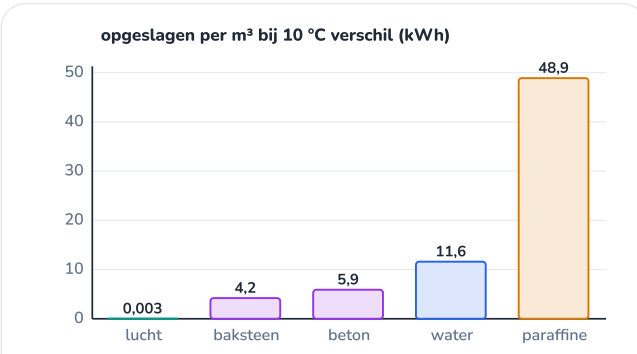
Water als warmtebuffer

Een **warmtebuffer** slaat warmte op wanneer er te veel is en geeft ze terug wanneer er te weinig is. De zonneboiler bewaart de zon van de middag in een vat voor de douche van 's avonds (hoofdstuk F17). Grote glastuinbouwbedrijven hebben naast hun kassen een geïsoleerde watertank van honderden kubieke meter: overdag slaan ze er de warmte in op die vrijkomt bij hun eigen stroomproductie, 's nachts verwarmen ze er de kas mee.

Hoeveel een buffer opslaat, volgt uit $Q = m \cdot c \cdot \Delta T$: je wil een grote $m \cdot c$, en een temperatuurzwaai ΔT die je mag toelaten. Water wint op bijna alle punten: de grootste c van alle gewone stoffen, goedkoop, niet giftig, en het kan stromen, zodat je de warmte met een pomp verplaatst. Per kubieke meter wint beton het niet, ook al is het 2,4 keer zo zwaar.

Er is één manier om per kilogram meer op te slaan: een **faseovergang**. Een faseovergangsmateriaal (PCM, *phase change material*), zoals een paraffine of een zouthydraat dat smelt rond 20 à 30 °C, neemt bij het smelten 150 à 250 kJ per kg op, bij een bijna vaste temperatuur. Dat is zoveel als water dat 40 à 60 °C opwarmt. PCM is wel duurder, en het moet in dunne lagen zitten, want vast paraffine geleidt warmte slecht.

> ⚠ **Een warmtebuffer is geen batterij**
>
> De warmte in een vat van 30 °C kan je gebruiken om iets van 16 °C warm te houden, maar je kan er geen lamp of pomp mee laten draaien. Warmte bij zo'n klein temperatuurverschil terug omzetten in elektrische energie gaat nauwelijks (hoofdstuk E15). Een warmtebuffer is dus goed voor warmte, een batterij voor elektriciteit.



Per m³ en bij 10 °C temperatuurzwaai; bij paraffine komt de smeltwarmte erbij (200 kJ/kg, 800 kg/m³). Lucht slaat bijna niets op.

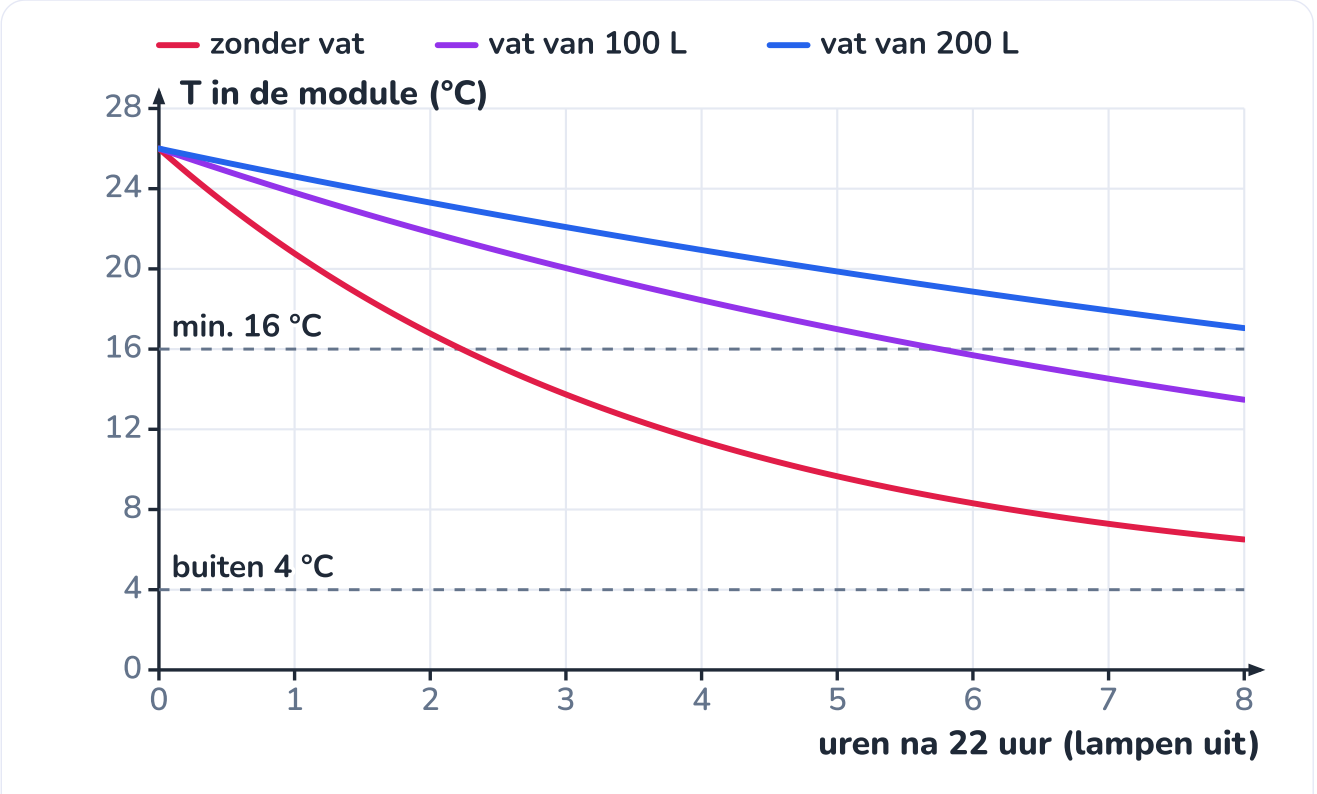
Een warmtebuffer naast een glastuinbouwbedrijf: overdag laden, 's nachts de kas verwarmen.

UITGEWERKT VOORBEELD · 18.20

Habitat: een watervat als nachtbuffer voor de klimaatmodule

20/22

Op een zonnige lentedag warmen de zon (door het kasdak) en de lampen de klimaatmodule op tot 26 °C. Om 22 uur gaan de lampen uit. Buiten is het die nacht gemiddeld 4 °C, en de module verliest ongeveer **20 W per °C** verschil met buiten (hoofdstuk 19 rekent het na). De sla mag 's nachts niet onder 16 °C zakken. In de module staan al het voedingsreservoir (60 L) en 11,5 m³ lucht. (a) Hoeveel warmte verliest de module tussen 22 en 6 uur? (b) Hoe groot moet een extra watervat zijn om dat op te vangen? (c) Hoeveel elektrische energie zou een verwarming nodig hebben om zonder vat 16 °C te houden?



Lucht, reservoir en vat koelen samen af, zoals een kop thee (hoofdstuk 5): hoe groter de warmtecapaciteit, hoe trager.

GEGEVEN

- verlies $20\,\mathrm{W}$ per °C, nacht $8\,\mathrm{h}$, buiten $4\,^\circ\mathrm{C}$
- begin $26\,^\circ\mathrm{C}$, ondergrens $16\,^\circ\mathrm{C}$
- reservoir $60\,\mathrm{kg}$ water; lucht $13{,}8\,\mathrm{kg}$, $c = 1005\,\mathrm{J/(kg\cdot K)}$

GEVRAAGD

- warmteverlies per nacht
- grootte van het vat
- energie voor elektrisch verwarmen

OPLOSSING

1. (a) De temperatuur daalt van 26 naar 16 °C; reken met het gemiddelde, 21 °C: gemiddeld $21 - 4 = 17$ °C verschil, dus $20 \cdot 17 = 340\,\mathrm{W}$. Per nacht: $Q = 340\,\mathrm{W} \cdot 8 \cdot 3600\,\mathrm{s} = 9{,}79\,\mathrm{MJ}$.

2. (b) Die warmte moet komen uit alles wat 10 °C afkoelt: $C = Q/\Delta T = 9{,}79\,\mathrm{MJ}/10\,\mathrm{K} = 979\,\mathrm{kJ/K}$. Reservoir en lucht leveren al $60 \cdot 4186 + 13{,}8 \cdot 1005 = 265\,\mathrm{kJ/K}$.

3. Het vat moet de rest leveren: $979 - 265 = 714\,\mathrm{kJ/K}$, dus $m = 714\,000/4186 = 171\,\mathrm{kg}$. Kies een vat van **200 L**.

4. Controle met het afkoelmodel van hoofdstuk 5 (grafiek): met 200 L is de module om 6 uur nog 17,0 °C, met 100 L 13,5 °C, zonder vat 6,5 °C.

5. (c) Zonder vat moet een verwarming 16 °C houden: $20 \cdot (16 - 4) = 240\,\mathrm{W}$, acht uur lang: $240 \cdot 8 = 1920\,\mathrm{Wh} = 1{,}92\,\mathrm{kWh}$.

ANTWOORD

(a) Ongeveer 9,8 MJ per nacht. (b) Een vat van 200 L water houdt de module boven 16 °C. (c) 1,92 kWh elektrisch per nacht: meer dan de hele bruikbare batterij van 1,2 kWh (hoofdstuk 16).

Klopt dit? Het vat moet overdag weer 'opladen': $200 \cdot 4186 \cdot 10 = 8{,}4\,\mathrm{MJ}$ (2,3 kWh). Dat doen de zon door het kasdak en de lampen, die hun 190 W volledig als warmte in de module achterlaten (hoofdstuk 20). Met een faseovergangsmateriaal dat bij 22 °C stolt (200 kJ/kg) zou ongeveer 36 kg volstaan in plaats van 171 kg water, maar het kost meer. Een donker vat in het zonlicht onder het kasdak laadt het snelst op.

NASLAG · 18.21

Overzicht: warmte berekenen

21/22

warmte en temperatuurverandering

$$Q = m \cdot c \cdot \Delta T$$

symbool	betekenis	eenheid
Q	warmte (negatief: afgegeven)	J
m	massa	kg
c	soortelijke warmtecapaciteit	J/(kg·K)
ΔT	temperatuurverandering	K of °C

warmtecapaciteit van een voorwerp

$$C = m \cdot c \qquad Q = C \cdot \Delta T$$

symbool	betekenis	eenheid
C	warmtecapaciteit	J/K

smelten en verdampen (stollen en condenseren: zelfde warmte vrij)

$$Q = m \cdot L_s \qquad Q = m \cdot L_v$$

symbool	betekenis	eenheid
L_s	smeltwarmte (ijs: 334)	kJ/kg
L_v	verdampingswarmte (water, 100 °C: 2257)	kJ/kg

eindtemperatuur bij mengen (zonder faseovergang)

$$T_e = \frac{m_1 c_1 T_1 + m_2 c_2 T_2 + \ldots}{m_1 c_1 + m_2 c_2 + \ldots}$$

calorimetrie (mengmethode)

$$c_{\text{metaal}} = \frac{(m_w c_w + C_{\text{cal}})(T_e - T_w)}{m_{\text{m}}\,(T_{\text{m}} - T_e)}$$

symbool	betekenis	eenheid
C_{cal}	warmtecapaciteit van de calorimeter	J/K

stof	c (J per kg per K)	stof	c (J per kg per K)
water (vloeibaar)	4186	lucht (droog, constante druk)	1005
menselijk lichaam (gemiddeld)	± 3500	aluminium	897
ethanol	2440	beton	± 880 (800 à 1000)

stof	c (J per kg per K)	stof	c (J per kg per K)
ijs (−10 à 0 °C)	± 2100	glas	± 840
waterdamp (100 °C)	± 2000	zand (droog)	± 830
paraffine (vast)	± 2000	ijzer	449
polypropeen (PP)	± 1900	koper	385
hout (vuren)	± 1600	lood	129
polycarbonaat (PC)	± 1200		

Waarden rond kamertemperatuur. Voor water verandert c tussen 0 en 100 °C minder dan 1 %; de ronde waarde 4,2 kJ per kg per K is die van hoofdstuk F16.

smeltwarmte L_s	kJ/kg	verdampingswarmte L_v	kJ/kg
ijs (0 °C)	334	water bij 100 °C	2257
aluminium (660 °C)	397	water bij 20 °C	2454
ijzer (1538 °C)	247	water bij 0 °C	2501
koper (1085 °C)	205	ethanol bij 78 °C	± 850
paraffine (± 20 à 60 °C, naargelang de soort)	150 à 250		
lood (327 °C)	23		

Tussen haakjes het smeltpunt. Stollen en condenseren geven dezelfde hoeveelheid warmte terug.

omrekenen	
1 kWh	3,6 MJ
1 kcal (voeding)	4,184 kJ
1 L water 1 °C warmer	4,19 kJ = 1,16 Wh
1 kg water verdampen (100 °C)	2,26 MJ = 0,63 kWh
1 kg ijs smelten	334 kJ = 0,093 kWh

Werkwijze warmtebalans met ijs: (1) warmte die het warme deel kan afgeven tot 0 °C; (2) warmte die het ijs nodig heeft tot 0 °C en om te smelten; (3) vergelijken: is (1) groter, dan smelt alles en warmt de rest het water op; is (1) kleiner, dan blijft het mengsel op 0 °C.

HET GEHEEL · 18.22

Waar warmte berekenen naartoe leidt

22/22

Warmte berekenen is de brug tussen **Energie** en **Materie en chemie**: c, L_s en L_v zijn stofeigenschappen die je met het deeltjesmodel en de bindingen verklaart. Met de calorimeter en de correctie voor warmteverlies zit je in **Onderzoeken en meten**, met de gewogen gemiddelden en de afkoelcurve in **Wiskundig modelleren**. En de warmtebuffer is een ontwerpbeslissing voor Habitat.

F6 Het deeltjesmodel
Faseovergangen en temperatuur als beweging van deeltjes: waar de latente warmte heen gaat.

F16 Warmte en temperatuur
De vuistregel 4,2 kJ per liter per graad en de 504 kJ van het reservoir van de meetkas.

F17 Energie uit zon, wind en water
De zonneboiler en de warmtebuffer als vorm van energieopslag.

I1 Meetfouten en nauwkeurigheid

I5 Groei en verval: exponentiële modellen

I9 Arbeid, energie en vermogen

Systematische fouten (het blokje dat afkoelt) tegenover toevallige, en de onzekerheid op C_{cal}.

Afkoelen als exponentieel verval: het nachtmodel van de module.

Behoud van energie en rendement: in een warmtebalans gaat geen energie verloren.

I13 Chemische bindingen en stofeigenschappen

Waterstofbruggen verklaren de grote c en de grote verdampingswarmte van water.

I14 Reacties en stoichiometrie

Exotherme en endotherme reacties: warmte die bij een reactie vrijkomt of opgenomen wordt.

I19 Warmtetransport en isolatie

Hoe snel de warmte uit de module weglekt: het getal 20 W per °C nagerekend.

I20 Zonne-energie dimensioneren

Waarom de verwarming niet op de batterij kan, en de lampen die ook de module verwarmen.

E15 De hoofdwetten van de thermodynamica

Waarom je warmte van 30 °C nauwelijks terug omzet in arbeid.

E17 Warmteoverdracht kwantitatief

Het dynamische warmtemodel: warmtecapaciteit en warmteverlies samen geven een tijdconstante.

Habitat

Voor Habitat leggen we vast: in de klimaatmodule komt naast het voedingsreservoir een donker watervat van 200 L als nachtbuffer, in het zonlicht onder het kasdak. Met het reservoir en de lucht erbij is de warmtecapaciteit van de module dan ongeveer 1,1 MJ per °C, en blijft ze in een lentenacht (4 °C buiten) boven 16 °C zonder verwarming. Elektrisch verwarmen zou ongeveer 1,9 kWh per nacht kosten, meer dan de bruikbare batterij. In opdracht 18.12 dimensioneer je de buffer voor een koudere nacht.

SAMENVATTING

In het kort

- De warmte om een stof op te warmen is $Q = m \cdot c \cdot \Delta T$. De soortelijke warmtecapaciteit c is een stofeigenschap: water 4186, aluminium 897, lucht 1005 J per kg per K.
- De warmtecapaciteit van een voorwerp is $C = m \cdot c$; bij een systeem tel je de delen op. In een kas slaat het water bijna alle warmte op, de lucht bijna niets.
- Bij smelten en verdampen stijgt de temperatuur niet: de warmte maakt de deeltjes los. $Q = m \cdot L$, met $L_s = 334\,\text{kJ/kg}$ en $L_v = 2257\,\text{kJ/kg}$ (bij 100 °C) voor water. Stollen en condenseren geven dezelfde warmte terug.
- Verdampen koelt (zweet, transpiratie), condenseren verwarmt (druppels op het kasdak); bevriezend water kan bloesems tegen vorst beschermen.
- Bij mengen is de afgegeven warmte gelijk aan de opgenomen warmte; de eindtemperatuur is een gewogen gemiddelde met gewichten $m \cdot c$. Met ijs: eerst de boekhouding tot 0 °C.
- In een calorimeter tel je de waterwaarde mee en corrigeer je voor warmteverlies door de afkoelrechte terug te trekken tot het moment van mengen.
- Water is de beste eenvoudige warmtebuffer; een faseovergangsmateriaal slaat per kg meer op. Voor Habitat houdt een vat van 200 L de module een lentenacht lang boven 16 °C.

BEGRIPPEN

Wat je nu kent

soortelijke warmtecapaciteit

De warmte die nodig is om 1 kg van een stof 1 K (of 1 °C) warmer te maken; symbool c, in J per kg per K.

warmtecapaciteit

De warmte die nodig is om een heel voorwerp 1 K warmer te maken: $C = m \cdot c$, in J per K.

smeltwarmte
De warmte die nodig is om 1 kg van een vaste stof op haar smeltpunt te laten smelten, zonder dat de temperatuur stijgt; bij stollen komt ze weer vrij.

verdampingswarmte
De warmte die nodig is om 1 kg vloeistof te laten verdampen; bij condenseren komt ze weer vrij. Voor water 2257 kJ/kg bij 100 °C.

latente warmte
Verzamelnaam voor smelt- en verdampingswarmte: warmte die een faseovergang veroorzaakt in plaats van een temperatuurstijging.

warmtebalans
In een geïsoleerd geheel is de warmte die de warmere delen afgeven gelijk aan de warmte die de koudere opnemen.

calorimeter
Een goed geïsoleerd vat waarin je warmte-uitwisseling meet, bijvoorbeeld twee piepschuimbekers met een deksel.

waterwaarde
De massa water die evenveel warmte opneemt als de calorimeter zelf; een andere manier om C_{cal} uit te drukken.

warmtebuffer
Een massa (meestal water) die warmte opslaat wanneer er te veel is en ze afgeeft wanneer er te weinig is.

HANDBOEK · HOOFDSTUK 19

Warmtetransport en isolatie

Van lambda naar U: hoeveel warmte door een wand lekt, waar ze ontsnapt, en hoe je ze binnenhoudt

Een wand van 20 cm houtskelet met wol laat per vierkante meter ruim vijfentwintig keer minder warmte door dan een ruit enkel glas. Waarom, en hoeveel precies? In dit hoofdstuk reken je warmtegeleiding uit met de **lambda-waarde** van een materiaal, tel je lagen op tot een **R-waarde**, en vat je een hele wand samen in één getal: de **U-waarde**. Daarmee bereken je het warmteverlies van een raam, een kas of een huis met $P = U \cdot A \cdot \Delta T$. Je ontdekt waarom houten stijlen en aluminium profielen **koudebruggen** zijn, waarom HR-glas zoveel beter is dan dubbel glas, en je legt voor Habitat de wandopbouw en het warmteverlies van de klimaatmodule vast.

7× uitleg · 6× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 8 lestijden

E Energie · D Ontwerpen en maken · W Wiskundig modelleren

Door een warmtecamera: het doorzichtige dak lekt het meeste warmte, en zelfs de houten stijlen in de wand zijn te zien.

Na dit hoofdstuk

- Je kan warmtegeleiding door een plaat berekenen met $P = \lambda \cdot A \cdot \Delta T / d$ en materialen vergelijken met hun lambda-waarde.
- Je kan de warmteweerstand $R = d/\lambda$ van lagen berekenen, lagen in serie optellen met de overgangsweerstanden, en er de U-waarde uit halen.
- Je kan het warmteverlies door een wand, een raam of een dak berekenen met $P = U \cdot A \cdot \Delta T$, en een warmteverliescoëfficiënt H opbouwen uit alle delen, ventilatie inbegrepen.
- Je kan het temperatuurverloop in een wand tekenen, de binnenoppervlaktetemperatuur berekenen en het risico op condens en schimmel beoordelen.
- Je kan koudebruggen herkennen, hun effect berekenen met parallelle wegen of met een ψ-waarde, en maatregelen voorstellen.
- Je kan uitleggen waarom dubbel, HR- en driedubbel glas beter isoleren dan enkel glas, en hun U-waarde berekenen uit de lagen.
- Je kan de U-waarde van een doos meten en de meting vergelijken met een berekening.
- Je kan het warmteverlies van een kleine kas en van de klimaatmodule berekenen en verbeteren.

UITLEG · 19.1

Wat je al weet, en wat nu komt

1/19

In hoofdstuk F16 volgde je de warmte op haar drie wegen: geleiding, stroming en straling. Je zag dat stilstaande lucht goed isoleert, dat een aluminium frame een koudebrug is, en je kreeg een getal om platen te vergelijken: de U-waarde. Met die U-waarde vond je dat de platen van de meetkas ongeveer 50 kJ per uur per graad verschil laten ontsnappen. In hoofdstuk 18 zag je hoeveel warmte een ruimte opslaat; nu bereken je hoe snel ze weglekt.

uit Fundamental en hoofdstuk 18	in dit hoofdstuk
geleiding, stroming, straling (F16)	geleiding in een formule: $P = \lambda \cdot A \cdot \Delta T / d$
een tabel met 'warmtegeleiding' van materialen (F16)	de lambda-waarde, de R-waarde van een laag en van een hele wand
U-waarde als getal: enkel glas 5,8, kanaalplaat 3,9 (F16)	de U-waarde berekenen, en $P = U \cdot A \cdot \Delta T$
koudebrug: het aluminium frame (F16)	koudebruggen berekenen en oplossen; condens en schimmel
de meetkas verliest ± 50 kJ per uur per °C (F16)	nagerekend, en het warmteverlies van de klimaatmodule
weerstanden in serie en parallel (F20)	lagen in serie, parallelle wegen: dezelfde rekenregels

In dit hoofdstuk

- reken je warmtegeleiding uit met een formule, en vergelijk je materialen met hun lambda-waarde;
- tel je lagen op tot de U-waarde van een wand, precies zoals weerstanden in serie;
- volg je het temperatuurverloop door een wand, en zie je waar condens en schimmel ontstaan;
- reken je koudebruggen en glas door, en meet je de U-waarde van een doos;
- leg je voor Habitat de wand van de klimaatmodule vast en bereken je haar warmteverlies.

VERHAAL · 19.2

Een huis dat zichzelf warm houdt

2/19

In 1991 trokken vier gezinnen in een rij van vier huizen in Kranichstein, een wijk van de Duitse stad Darmstadt. De huizen zagen er gewoon uit, maar ze hadden geen klassieke verwarming nodig. De natuurkundige **Wolfgang Feist** had ze samen met architecten ontworpen als proef: wat als je zo goed isoleert dat de warmte van de zon door de ramen, van de bewoners en van de toestellen bijna volstaat?

De recepten waren eenvoudig, maar streng: dikke isolatie rondom, zonder onderbrekingen; drievoudige beglazing; geen koudebruggen; een gebouw dat luchtdicht is, met een ventilatiesysteem dat de warmte uit de afgevoerde lucht terugwint. Jarenlange metingen toonden dat die huizen minder dan 10 kWh warmte per vierkante meter per jaar nodig hadden. Daaruit groeide de **passiefhuisnorm**: hoogstens 15 kWh per m² per jaar.

Wat toen experimenteel was, zit vandaag in de bouwregels. In Vlaanderen legt de EPB-regelgeving (energieprestatie en binnenklimaat) voor nieuwbouw een maximale U-waarde op: 0,24 W/(m²·K) voor een muur of een dak, 0,30 voor een vloer, 1,1 voor beglazing. Wat die getallen betekenen, en hoe je ze uitrekent voor een wand die je zelf ontwerpt, leer je in dit hoofdstuk.

Waarom dit verhaal ertoe doet

Isoleren is de goedkoopste kilowattuur: warmte die niet weglekt, hoef je niet op te wekken. Voor Habitat geldt dat dubbel. De klimaatmodule heeft een doorzichtig dak voor de planten, en dat dak is de zwakke plek. Het brein van het station (hoofdstuk E39) rekent met het warmteverlies dat jij hier uitrekent.

Rijhuizen die met de warmte van de zon, de bewoners en de toestellen bijna warm blijven.

UITLEG · 19.3

Warmtegeleiding in een formule

3/19

Neem een plaat met dikte d en oppervlakte A, aan de ene kant warm (T_1), aan de andere kant koud (T_2). Hoeveel warmte gaat er per seconde door? Vier dingen tellen, en je kent ze allemaal uit ervaring: een twee keer zo groot raam laat twee keer zoveel door; twee keer zoveel temperatuurverschil ook; een twee keer zo dikke plaat de helft; en het materiaal. Samen:

WARMTEGELEIDING DOOR EEN PLAAT (WET VAN FOURIER, EENVOUDIGE VORM)

$$P = \frac{\lambda \cdot A \cdot \Delta T}{d}$$

P: warmtestroom in W (joule per seconde); λ: warmtegeleidingscoëfficiënt in W/(m·K); A in m²; $\Delta T = T_1 - T_2$ in K; d in m.

De **lambda-waarde** λ is de stofeigenschap. Ze zegt hoeveel watt er door een plaat van 1 m² en 1 m dik gaat bij 1 K verschil. De waarden lopen enorm uiteen, van 400 voor koper tot 0,026 voor stilstaande lucht: vijftienduizend keer minder.

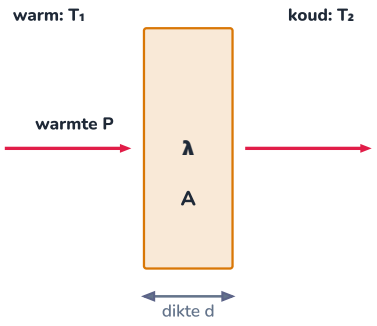
materiaal	λ (W/(m·K))	materiaal	λ (W/(m·K))
koper	400	gipskarton	0,25
aluminium	± 200	polycarbonaat	0,20
staal	± 50	vuren hout, OSB, multiplex	0,13
beton	± 1,5	houtvezelisolatieplaat	0,045
glas	± 1,0	minerale wol, EPS (piepschuim)	0,035
baksteen	± 0,9	PUR, PIR	0,022 à 0,026
water	0,6	stilstaande lucht	0,026

Afgeronde rekenwaarden. Isolatieplaten hebben een gecertificeerde λ op de verpakking (de 'lambda-waarde'); reken met die waarde.

Waarom isoleren wol en piepschuim zo goed, hoewel wol (glas- of steenvezels) en polystyreen zelf niet zo slecht geleiden? Omdat ze vooral uit **stilstaande lucht** bestaan, opgesloten in kleine cellen of tussen vezels, zodat ze niet kan stromen (hoofdstuk F16). Hun λ komt daardoor dicht bij die van lucht. PUR en PIR doen nog beter: hun cellen zijn gevuld met een gas dat warmte nog slechter geleidt dan lucht.

⚠ Nat isoleert niet

Water geleidt warmte meer dan twintig keer beter dan stilstaande lucht (0,6 tegenover 0,026). Isolatie die nat wordt, verliest dus een groot deel van haar werking. Daarom komt aan de warme binnenkant van een wand een laag die de waterdamp tegenhoudt (verder in dit hoofdstuk), en moet een wand aan de buitenkant kunnen uitdrogen.



Hoe groter λ, A en ΔT, hoe meer warmte; hoe dikker de plaat, hoe minder.

UITGEWERKT VOORBEELD · 19.4

Piepschuim tegen beton

4/19

(a) Hoeveel warmte gaat er door 1,0 m² piepschuim (EPS) van 10 cm dik bij 20 °C verschil? (b) En door 1,0 m² beton van 10 cm? (c) Hoe dik zou de betonnen wand moeten zijn om even goed te isoleren als 10 cm EPS? (We laten de overgangsweerstanden aan het oppervlak nog even weg.)

GEGEVEN

- $A = 1{,}0\,\mathrm{m}^2$, $d = 0{,}10\,\mathrm{m}$, $\Delta T = 20\,\mathrm{K}$
- $\lambda_{\mathrm{EPS}} = 0{,}035\,\mathrm{W/(m\cdot K)}$, $\lambda_{\mathrm{beton}} = 1{,}5\,\mathrm{W/(m\cdot K)}$

GEVRAAGD

- P_{EPS}, P_{beton}
- d_{beton} voor hetzelfde verlies

OPLOSSING

1. (a) $P = \dfrac{0{,}035 \cdot 1{,}0 \cdot 20}{0{,}10} = 7{,}0\ \mathrm{W}$.

2. (b) $P = \dfrac{1{,}5 \cdot 1{,}0 \cdot 20}{0{,}10} = 300\ \mathrm{W}$: 43 keer zoveel.

3. (c) Zelfde verlies als λ/d gelijk is: $d_{\text{beton}} = d_{\text{EPS}} \cdot \dfrac{\lambda_{\text{beton}}}{\lambda_{\text{EPS}}} = 0{,}10 \cdot \dfrac{1{,}5}{0{,}035} = 4{,}3\ \mathrm{m}$.

ANTWOORD

(a) 7,0 W. (b) 300 W. (c) Een betonnen wand van 4,3 m dik isoleert even goed als 10 cm piepschuim.

Klopt dit? Eenheden: $\mathrm{W/(m \cdot K) \cdot m^2 \cdot K/m = W}$. En de orde van grootte: 7 W per m² is het vermogen van een klein ledlampje. Beton draagt, isolatie isoleert: daarom heeft een moderne muur beide lagen.

UITLEG · 19.5

Warmteweerstand: lagen tellen op

5/19

Een echte wand bestaat uit lagen. Om ze samen te nemen, schrijf je de formule van het vorige blad per vierkante meter en met de **warmteweerstand** $R = d/\lambda$ van elke laag:

WARMTEWEERSTAND VAN EEN LAAG

$$R = \frac{d}{\lambda} \qquad \frac{P}{A} = \frac{\Delta T}{R}$$

R in m²·K/W: hoeveel kelvin verschil nodig is om 1 W door 1 m² te duwen. Dikker of beter isolerend: grotere R.

Dezelfde warmte moet door alle lagen na elkaar, zoals dezelfde stroom door weerstanden in serie (hoofdstuk F20). Dus tellen de warmteweerstanden op. Er komen nog twee weerstanden bij die je niet ziet: het dunne laagje lucht tegen het binnenoppervlak en tegen het buitenoppervlak. Warmte moet ook daar door, door geleiding, stroming en straling samen. Die **overgangsweerstanden** zijn afgesproken waarden (norm EN ISO 6946):

overgangsweerstand	m²·K/W
R_{si} binnen, wand (warmte horizontaal)	0,13
R_{si} binnen, dak (warmte naar boven)	0,10
R_{si} binnen, vloer (warmte naar beneden)	0,17
R_{se} buiten (wind)	0,04

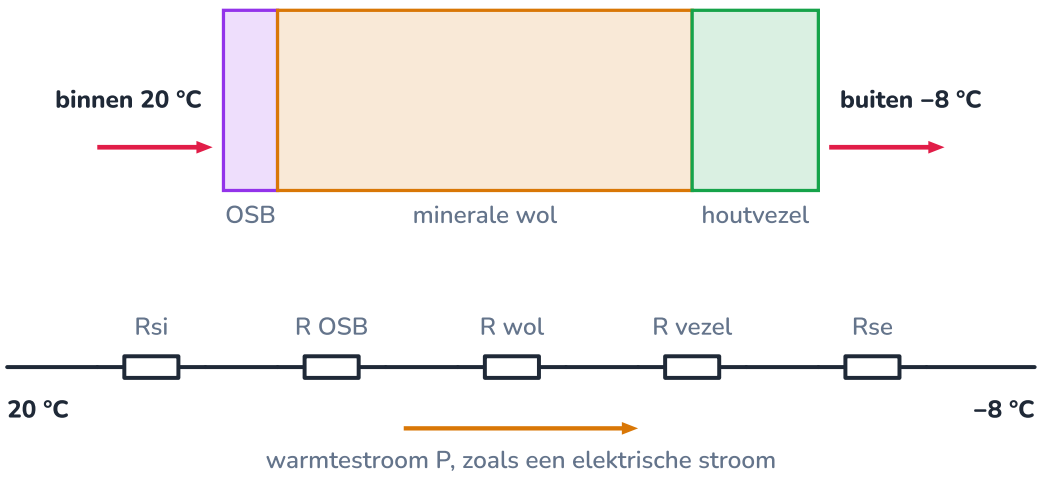
TOTALE WARMTEWEERSTAND VAN EEN WAND

$$R_T = R_{si} + \frac{d_1}{\lambda_1} + \frac{d_2}{\lambda_2} + \cdots + R_{se}$$

Waarom verschillen die waarden? Warme lucht stijgt: tegen een plafond gaat de warmte door stroming makkelijk naar boven (kleine R_{si}), tegen een vloer blijft de koude lucht liggen (grote R_{si}). Buiten blaast de wind de luchtlaag weg: R_{se} is klein.

Het elektrische model

Temperatuurverschil ↔ spanning, warmtestroom ↔ stroom, warmteweerstand ↔ weerstand. Dan is $\Delta T = R \cdot P$ de wet van Ohm (hoofdstuk F19). En net zoals de spanning over weerstanden in serie verdeelt, verdeelt het temperatuurverschil zich over de lagen: over de laag met de grootste R valt het grootste temperatuurverschil (hoofdstuk 22, de spanningsdeler).



Dezelfde warmte gaat door alle lagen na elkaar: de weerstanden tellen op, zoals weerstanden in serie.

UITLEG · 19.6

De U-waarde en P = U · A · ΔT

6/19

Bouwers vatten een hele wand samen in één getal, het omgekeerde van de totale warmteweerstand: de **U-waarde** of warmtedoorgangscoëfficiënt.

U-WAARDE EN WARMTEVERLIES

$$U = \frac{1}{R_T} \qquad P = U \cdot A \cdot \Delta T$$

U in W/(m²·K): hoeveel watt door 1 m² gaat bij 1 K verschil tussen de lucht binnen en buiten. Hoe kleiner, hoe beter.

Voor een heel gebouw tel je alle delen op tot de **warmteverliescoëfficiënt** H, in W/K. Daar komt nog iets bij: de lucht die je ververst. Elke kubieke meter verse lucht moet je opwarmen, en dat volgt uit hoofdstuk 18:

WARMTEVERLIES VAN EEN GEBOUW

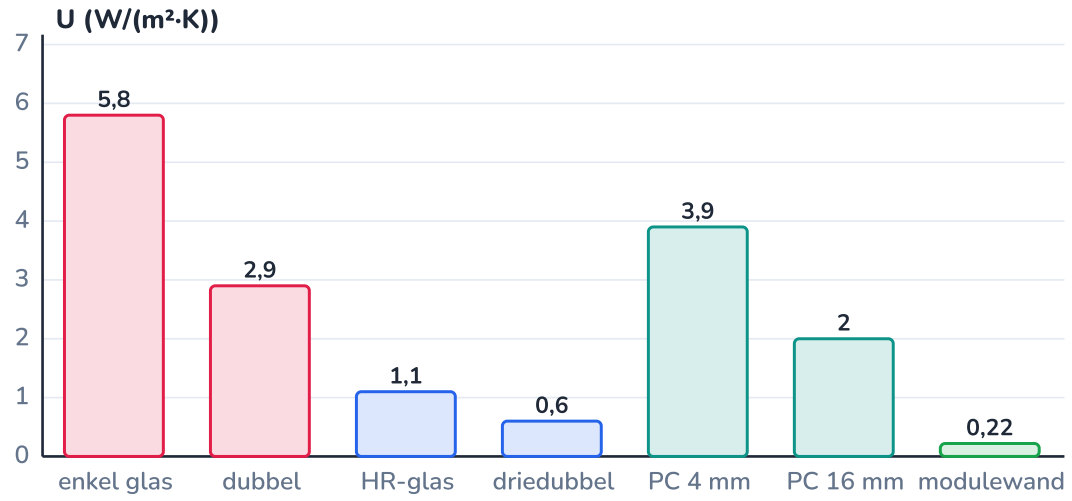
$$H = \sum U \cdot A + H_{\text{kb}} + \rho \cdot c \cdot \dot{V} \qquad P = H \cdot \Delta T$$

$\dot{V}$: ventilatiedebiet in m³/s; voor lucht is $\rho \cdot c = 1{,}20 \cdot 1005 \approx 1200\,\text{J}/(\text{m}^3 \cdot \text{K})$. H_{kb}: de koudebruggen (verder in dit hoofdstuk).

onderdeel	U (W/(m²·K)), orde van grootte
enkel glas	5,8
volle bakstenen muur (jaren 1930)	± 2
spouwmuur zonder isolatie	± 1,5
EPB-eis muur en dak (Vlaanderen, nieuwbouw)	0,24
passiefhuiswand	± 0,1 à 0,15

Warmte gaat naar boven, dus isoleer alleen het dak

Warme *lucht* stijgt, maar *warmte* stroomt in elke richting van warm naar koud: door de muren, de vloer en de ramen. Het dak verliest wel iets meer per m² (kleinere R_{si}) en is vaak het slechtst geïsoleerd in oude huizen: daarom begin je er vaak mee. Maar tel voor een eerlijk beeld altijd $U \cdot A$ van élk deel op.



U-waarden vergeleken: een ruit enkel glas laat per m² 26 keer zoveel warmte door als de wand van de klimaatmodule.

UITGEWERKT VOORBEELD · 19.7

Habitat: de U-waarde van de modulewand

7/19

De wand van de klimaatmodule is een houtskeletwand. Van binnen naar buiten: een OSB-plaat van 15 mm (die ook de waterdamp tegenhoudt), 145 mm minerale wol tussen vuren stijlen, een doorlopende houtvezelplaat van 40 mm en een dunne pleisterlaag (die je verwaarloost). Bereken de U-waarde door de wol (zonder de stijlen). Voldoet de wand aan de EPB-eis van 0,24?

GEGEVEN

- OSB $d = 0{,}015\,\text{m}$, $\lambda = 0{,}13$; wol $d = 0{,}145\,\text{m}$, $\lambda = 0{,}035$; houtvezel $d = 0{,}040\,\text{m}$, $\lambda = 0{,}045$ (alle λ in W/(m·K))
- $R_{si} = 0{,}13$, $R_{se} = 0{,}04\,\text{m}^2 \cdot \text{K/W}$

GEVRAAGD

- R per laag, R_T, U

OPLOSSING

1

LAAG	D (M)	Λ	$R = D/\Lambda$ (M²·K/W)
R_{si}			0,13
OSB	0,015	0,13	0,12
minerale wol	0,145	0,035	4,14
houtvezelplaat	0,040	0,045	0,89
R_{se}			0,04
R_T			**5,32**

2 $U = 1/R_T = 1/5{,}32 = 0{,}188\,\text{W}/(\text{m}^2 \cdot \text{K})$.

3 De wol levert 78 % van de totale weerstand; de overgangsweerstanden samen maar 3 %.

ANTWOORD

Door de wol is $U = 0,19\,\mathrm{W/(m^2 \cdot K)}$: ruim onder 0,24. Maar een houtskeletwand bestaat niet alleen uit wol: de stijlen zijn een koudebrug (verder in dit hoofdstuk).

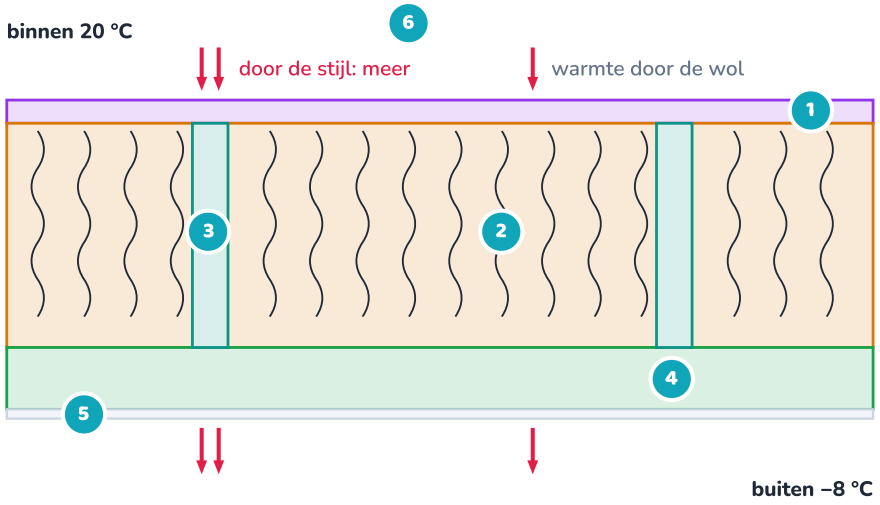
Klopt dit? Bij 20 °C binnen en −8 °C buiten verliest 1 m² van deze wand $0,188 \cdot 1 \cdot 28 = 5,3\,\mathrm{W}$. Een ruit enkel glas van 1 m²: $5,8 \cdot 28 = 162\,\mathrm{W}$.

ONTLEED · 19.8

De wand van de klimaatmodule

8/19

Een horizontale doorsnede, van bovenaf bekeken: binnen boven, buiten onder. De dikte is op schaal (1 px per mm), de breedte twee keer ingekrompen: de stijlen staan in werkelijkheid 60 cm uit elkaar. Elke laag heeft een eigen taak.



Doorsnede van de wand: OSB, minerale wol tussen stijlen, doorlopende houtvezelplaat, pleister.

1 OSB-plaat (15 mm)
Maakt de wand stijf en luchtdicht, en remt de waterdamp van binnen af. Warme binnenlucht bevat veel waterdamp: in de koude wol zou die condenseren (verder op dit blad: het temperatuurverloop).

2 Minerale wol (145 mm)
De eigenlijke isolatie: $R = 4,14$ van de totale $5,32$ m²·K/W. De wol vult het vak tussen de stijlen helemaal, zonder spleten: een kier van enkele millimeter laat lucht stromen.

3 Vuren stijl (45 × 145 mm)
Draagt de wand, maar geleidt warmte bijna vier keer beter dan wol (0,13 tegenover 0,035): een koudebrug. Stijlen, regels en de randen van de openingen samen: ongeveer 12 % van de oppervlakte.

4 Houtvezelplaat (40 mm)
Loopt ononderbroken over de stijlen heen: zo isoleert ze ook de koudebrug van het hout. Ze laat waterdamp door, zodat de wand naar buiten kan uitdrogen.

5 Pleister
Beschermt tegen regen en uv. Te dun om mee te tellen in de warmteweerstand.

6 Binnen- en buitenlucht
De luchtlaagjes tegen de oppervlakken tellen mee als $R_{si} = 0,13$ en $R_{se} = 0,04$: samen maar 3 % van de weerstand van deze wand, maar bij een ruit enkel glas bijna alles.

Het temperatuurverloop in een wand

Door elke laag gaat dezelfde warmtestroom $q = P/A = U \cdot \Delta T$. Over een laag valt dan een temperatuurverschil $q \cdot R$: hoe groter de weerstand van de laag, hoe groter de temperatuurval. Bij 20 °C binnen en −8 °C buiten gaat er door de wol van de modulewand $q = 0{,}188 \cdot 28 = 5{,}27\ \mathrm{W/m^2}$, en de temperatuur verloopt zo:

laag	R	temperatuurval $q \cdot R$	temperatuur erna
binnenlucht → oppervlak	0,13	0,68 °C	19,3 °C
OSB	0,12	0,61 °C	18,7 °C
minerale wol	4,14	21,82 °C	−3,1 °C
houtvezelplaat	0,89	4,68 °C	−7,8 °C
oppervlak → buitenlucht	0,04	0,21 °C	−8,0 °C

BINNENOPPERVLAKTETEMPERATUUR EN TEMPERATUURFACTOR

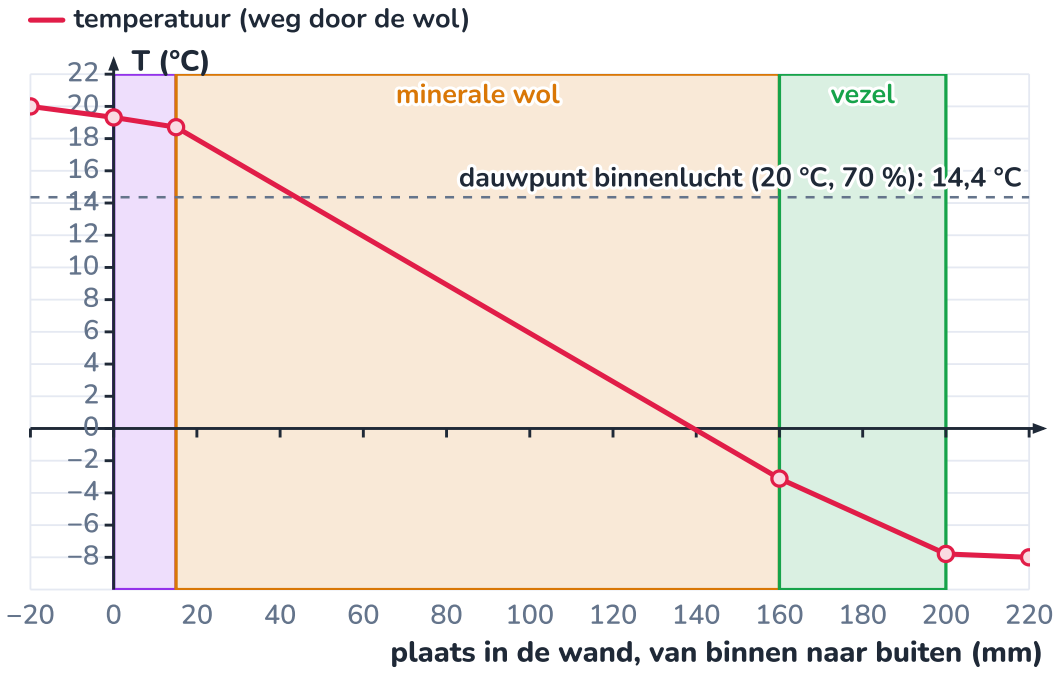
$$T_{si} = T_i - U \cdot R_{si} \cdot (T_i - T_e) \qquad f = \frac{T_{si} - T_e}{T_i - T_e} = 1 - U \cdot R_{si}$$

T_i, T_e: lucht binnen en buiten. Hoe kleiner U, hoe dichter T_{si} bij de binnentemperatuur en hoe dichter f bij 1.

Twee dingen vallen op. Het **binnenoppervlak** is maar 0,7 °C kouder dan de lucht: de wand voelt niet koud aan, en er condenseert niets tegen. Bij een ruit enkel glas is dat anders: $T_{si} = 20 - 5{,}8 \cdot 0{,}13 \cdot 28 = -1{,}1\,^\circ\mathrm{C}$, dus ijsbloemen. En **in de wol** zakt de temperatuur onder het dauwpunt van de binnenlucht. Zou die vochtige lucht tot daar kunnen doordringen, dan condenseert ze in de isolatie (hoofdstuk F6). Daarom zit de dampremmende laag aan de **warme** kant, en is de buitenkant dampopen: wat er toch in komt, kan naar buiten weg.

⚠ Schimmel begint aan het koudste plekje

Schimmel groeit al als het oppervlak lang vochtig blijft, bij een relatieve vochtigheid tegen het oppervlak van ongeveer 80 %, nog voor er druppels zijn. In België gebruikt men daarom als vuistregel dat de temperatuurfactor overal minstens 0,7 moet zijn, ook in hoeken en bij koudebruggen. Enkel glas haalt 0,25; HR-glas 0,86; de modulewand 0,98.



De grootste temperatuurval zit in de wol, de laag met de grootste R. Waar de lijn onder het dauwpunt zakt, mag geen vochtige binnenlucht komen.

UITLEG · 19.10

Koudebruggen

10/19

Een **koudebrug** (ook bouwknoop genoemd) is een plaats waar warmte makkelijker naar buiten lekt dan in de rest van de wand. Drie soorten:

- **een beter geleidend materiaal** dat door de isolatie heen loopt: houten stijlen, een stalen balk, een aluminium profiel, een betonnen balkonplaat die van binnen naar buiten doorloopt;
- **een onderbroken isolatielaag**: waar de isolatie van muur en dak niet op elkaar aansluiten, rond een raam, bij een kier;
- **de vorm**: in een buitenhoek staat een klein binnenoppervlak tegenover een groot buitenoppervlak.

Koudebruggen kosten warmte, maar vooral: het binnenoppervlak is er kouder, en daar condenseert vocht en groeit schimmel. Je rekent ze op twee manieren:

PARALLELLE WEGEN (BV. STIJLEN IN EEN WAND)

$$U_{\text{gem}} = (1 - f) \cdot U_{\text{isolatie}} + f \cdot U_{\text{hout}}$$

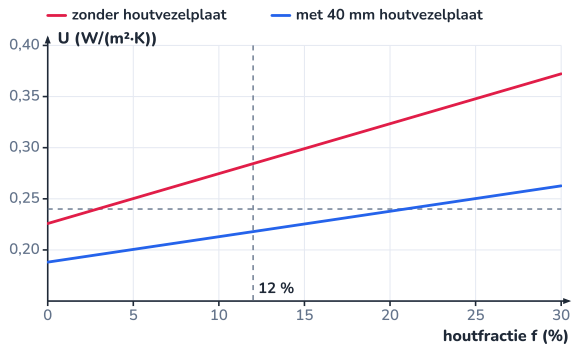
f: het deel van de oppervlakte dat door het hout gaat. Twee wegen naast elkaar, zoals parallelle weerstanden: de U-waarden (geleidingen) tellen op, gewogen met hun oppervlakte.

LINEAIRE KOUDEBRUG (BV. EEN ALUMINIUM KADER, EEN BALKONPLAAT, EEN VLOERRAND)

$$P_{\text{kb}} = \psi \cdot L \cdot \Delta T$$

ψ (psi) in W/(m·K): extra verlies per meter lengte per kelvin; L in m.

De remedie is bijna altijd dezelfde: **zorg dat de isolatie ononderbroken doorloopt**. Een doorlopende isolatieplaat over de stijlen, een kunststof onderbreking in aluminium raamprofielen, isolatie die van de muur over de vloerrand tot in het dak loopt. Wie een isolatieplan tekent, trekt één lijn rond het hele gebouw zonder de pen op te heffen.



Hoe meer hout, hoe groter U; de horizontale stippellijn is de EPB-eis van 0,24. De doorlopende houtvezelplaat halveert bijna het effect van de stijlen.

Op een warmtebeeld van een houtskeletwand verraden de stijlen zich als warmere strepen.

UITGEWERKT VOORBEELD · 19.11

Habitat: de stijlen als koudebrug, en de remedie

11/19

In de modulewand staan de stijlen 60 cm uit elkaar; met regels en de randen van de openingen gaat **12 %** van de oppervlakte door hout (vuren, $\lambda = 0{,}13$) in plaats van door wol. (a) Bereken de gemiddelde U-waarde van de wand zonder de houtvezelplaat (alleen OSB en wol tussen de stijlen). (b) Met de doorlopende houtvezelplaat van 40 mm. (c) Hoeveel procent verhoogt het hout de U-waarde in beide gevallen?

GEGEVEN

- $f = 0{,}12$
- lagen en λ zoals in het vorige voorbeeld; stijl: 145 mm vuren in plaats van wol

GEVRAAGD

- U_{gem} zonder en met houtvezelplaat
- toename door het hout

OPLOSSING

1. (a) Door de wol: $R_T = 0{,}13 + 0{,}12 + 4{,}14 + 0{,}04 = 4{,}43$, $U = 0{,}226$. Door de stijl: $R_T = 0{,}13 + 0{,}12 + \dfrac{0{,}145}{0{,}13} + 0{,}04 = 1{,}40$, $U = 0{,}714$.
2. Gemiddeld: $U = 0{,}88 \cdot 0{,}226 + 0{,}12 \cdot 0{,}714 = 0{,}284\ \mathrm{W/(m^2 \cdot K)}$: boven de EPB-eis.
3. (b) Met 40 mm houtvezel ($R = 0{,}89$) erbij, óók voor de stijl: wol $U = 0{,}188$, stijl $U = 0{,}437$, gemiddeld $0{,}88 \cdot 0{,}188 + 0{,}12 \cdot 0{,}437 = 0{,}218\ \mathrm{W/(m^2 \cdot K)}$.
4. (c) Zonder houtvezel: $0{,}284/0{,}226 = 1{,}26$, dus 26 % meer. Met houtvezel: $0{,}218/0{,}188 = 1{,}16$, dus 16 % meer.

ANTWOORD

Zonder houtvezelplaat $U = 0,28$ (te hoog), met $U = 0,22\ \mathrm{W/(m^2 \cdot K)}$ (voldoet). De stijlen verhogen de U-waarde met 26 %, of met 16 % als een doorlopende laag ze afschermt. Voor Habitat: $U_{\text{wand}} = 0,22$.

Klopt dit? De houtvezelplaat verlaagt de U-waarde van de stijl van 0,71 naar 0,44: meer dan de helft van het effect. Dat is de betekenis van 'ononderbroken isoleren'. De binnenoppervlaktetemperatuur ter hoogte van een stijl is bij −8 °C buiten $20 - 0{,}44 \cdot 0{,}13 \cdot 28 = 18{,}4\ ^\circ\mathrm{C}$: geen schimmelrisico.

UITLEG · 19.12

12/19

Enkel, dubbel en driedubbel glas

Glas zelf isoleert slecht ($\lambda \approx 1$) en een ruit is maar 4 mm dik: de glaslaag heeft een weerstand van 0,004 m²·K/W. Bij enkel glas zit bijna alle weerstand in de twee luchtlaagjes tegen het oppervlak: $R_T = 0{,}13 + 0{,}004 + 0{,}04 = 0{,}174$, dus $U = 5{,}8$.

Dubbel glas sluit een laag lucht op tussen twee ruiten. Die spouw is een extra weerstand, maar minder dan je zou denken: over de spouw gaat warmte door geleiding, een beetje stroming, en vooral door **straling** van de warme ruit naar de koude (hoofdstuk F16). In gewoon dubbel glas is dat ongeveer twee derde van de warmte. Een luchtspouw van 12 mm heeft daardoor maar $R \approx 0{,}17$, en de ruit $U \approx 2{,}8$.

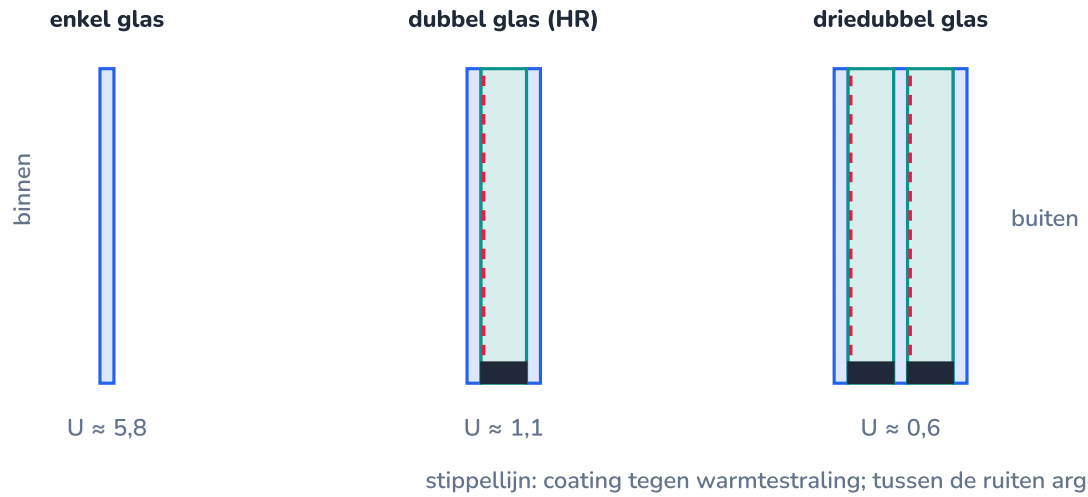
Hoogrendementsglas (HR-glas) pakt de straling aan met een onzichtbare **coating**: een metaallaagje van enkele nanometer dat licht doorlaat maar warmtestraling terugkaatst, zoals de zilverlaag van een thermosfles. De spouw wordt gevuld met **argon**, een edelgas dat slechter geleidt dan lucht. Een spouw van 15 mm krijgt zo $R \approx 0{,}75$, en de ruit $U \approx 1{,}1$. **Driedubbel glas** heeft twee zulke spouwen: $U \approx 0{,}6$.

beglazing	U (W/(m²·K))	binnenoppervlak bij −8 °C	lichtdoorlaat (orde van grootte)
enkel glas	5,8	−1 °C	± 90 %
dubbel glas met lucht	± 2,8	10 °C	± 80 %
HR-glas (coating, argon)	± 1,1	16 °C	± 80 %
driedubbel glas	± 0,6	18 °C	± 70 %
polycarbonaat, kanaalplaat 4 mm (meetkas)	3,9	6 °C	± 80 %
polycarbonaat, 16 mm vijf wanden (kasdak)	± 2,0	14 °C	± 60 à 65 %

Binnenoppervlak bij 20 °C binnen en −8 °C buiten (voor het dak met $R_{si} = 0{,}10$). De EPB legt voor nieuwe ramen $U_g \leq 1{,}1$ op voor het glas en $U_w \leq 1{,}5$ voor het hele raam, kader inbegrepen.

Hoe breder de spouw, hoe beter

Tot ongeveer 15 à 16 mm wel. In een bredere spouw begint het gas rond te stromen (warm langs de binnenruit omhoog, koud langs de buitenruit omlaag) en neemt het verlies weer toe. Daarom is een oude 'dubbele raam' met 10 cm lucht ertussen niet beter dan modern dubbel glas.



Elke extra spouw met coating en argon halveert ongeveer het verlies; het glas zelf telt nauwelijks mee.

Stalen van enkel, dubbel en driedubbel glas: de afstandhouders aan de rand zijn zelf een kleine koudebrug.

UITGEWERKT VOORBEELD · 19.13

Glas doorgerekend, en een oud raam vervangen

13/19

(a) Bereken de U-waarde van dubbel glas met een luchtspouw ($R_{\text{spouw}} = 0{,}17$) en van HR-glas ($R_{\text{spouw}} = 0{,}75$), elk met twee ruiten van 4 mm ($\lambda = 1{,}0$). (b) Een raam van 1,5 m² met enkel glas wordt vervangen door HR-glas. Hoeveel watt minder verlies bij 20 °C binnen en 0 °C buiten? (c) Een stookseizoen komt overeen met ongeveer 5000 uur bij gemiddeld 10 °C verschil. Hoeveel kWh bespaart het raam per jaar?

GEGEVEN

- $R_{si} = 0{,}13$, $R_{se} = 0{,}04$, ruit $R = 0{,}004$
- $A = 1{,}5\,\text{m}^2$; enkel glas $U = 5{,}8$

GEVRAAGD

- U_{dubbel}, U_{HR}
- vermogen en energie die je bespaart

OPLOSSING

1. (a) Dubbel: $R_T = 0{,}13 + 0{,}004 + 0{,}17 + 0{,}004 + 0{,}04 = 0{,}348$, $U = 2{,}9$. HR: $R_T = 0{,}13 + 0{,}004 + 0{,}75 + 0{,}004 + 0{,}04 = 0{,}928$, $U = 1{,}08$.
2. (b) Enkel: $5{,}8 \cdot 1{,}5 \cdot 20 = 174\,\text{W}$; HR: $1{,}08 \cdot 1{,}5 \cdot 20 = 32\,\text{W}$. Verschil 142 W.
3. (c) $(5{,}8 - 1{,}08) \cdot 1{,}5 \cdot 10 \cdot 5000 = 354\,000\,\text{Wh} = 354\,\text{kWh}$ per jaar.

ANTWOORD

(a) Dubbel glas $U = 2{,}9$, HR-glas $U = 1{,}1$. (b) 142 W minder. (c) Ongeveer 350 kWh per jaar.

Klopt dit? De berekende waarden liggen dicht bij de waarden van fabrikanten (2,8 à 2,9 en 1,0 à 1,1): het model met lagen in serie werkt. 350 kWh warmte is ongeveer 35 m³ aardgas. Het kader is in dit voorbeeld niet meegerekend: een oud houten of aluminium kader kan zelf een koudebrug zijn.

PROEF · 19.14

De U-waarde van een isolatiedoos meten

14/19

ONDERZOEKSVRAAG

Klopt de berekende U-waarde van een piepschuimplaat met wat je meet?

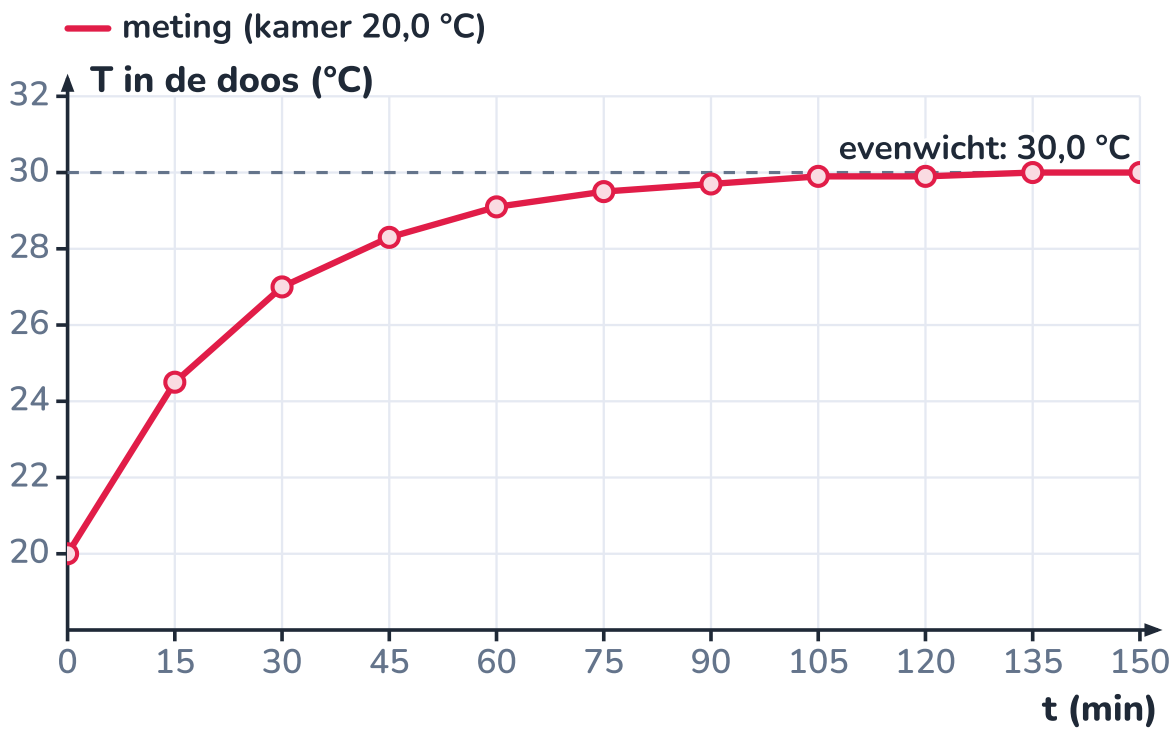
MATERIAAL

- doos van EPS, 30 mm dik, binnen 30 × 30 × 30 cm, met deksel
- vermogensweerstand 22 Ω / 10 W op een 12 V-voeding
- twee multimeters (spanning over en stroom door de weerstand)
- twee temperatuursensoren: in de doos (in het midden) en in de klas

stopwatch of datalogger

Veiligheid

Een vermogensweerstand wordt heet: leg hem op een metalen plaatje, niet rechtstreeks op het piepschuim, en zet de voeding nooit hoger dan 12 V.



De temperatuur in de doos stijgt tot er evenveel warmte door de wanden weglekt als de weerstand levert.

WERKWIJZE

1. Leg de weerstand in de doos, voer de draden en de sensor door een klein gaatje in het deksel en dicht het gaatje met plakband.
2. Zet de voeding aan. Meet de spanning over en de stroom door de weerstand: dat geeft het vermogen $P = U \cdot I$.
3. Lees elke 15 minuten de temperatuur binnen en buiten af, tot de binnentemperatuur niet meer stijgt (± 2 uur).
4. In evenwicht gaat evenveel warmte door de wanden naar buiten als de weerstand levert: $H = P/\Delta T$ en $U = H/A$.
5. Bereken de U-waarde ook uit de lagen, en vergelijk.

WAARNEMING

De weerstand kreeg 12,0 V en 0,545 A: $P = 6{,}54\,\mathrm{W}$. De klas bleef 20,0 °C. De grafiek toont de temperatuur in de doos: na ongeveer twee uur stabiel op 30,0 °C, dus $\Delta T = 10{,}0\,\mathrm{K}$.

	gemeten	berekend
H (W/K)	$6{,}54/10{,}0 = 0{,}654$	$0{,}895 \cdot 0{,}653 = 0{,}585$
U (W/(m²·K))	$0{,}654/0{,}653 = 1{,}00$	$1/(0{,}13 + 0{,}030/0{,}035 + 0{,}13) = 0{,}895$

VERKLARING

Voor de oppervlakte neem je het gemiddelde van binnen- en buitenmaat: zes vlakken van 0,33 m × 0,33 m, samen $0{,}653\,\mathrm{m}^2$. Binnenshuis is de lucht aan de buitenkant van de doos ook stilstaand: reken aan beide kanten met 0,13.

De meting geeft een U-waarde die ongeveer 12 % groter is dan de berekening. Dat is geen meetfout maar een les: de ribben en hoeken van de doos zijn koudebruggen door hun vorm, de naad van het deksel laat wat lucht door, en de kabels geleiden warmte naar buiten. Precies die dingen maken dat een echt gebouw meer verliest dan de som van zijn U-waarden: daarom rekenen ontwerpers de koudebruggen apart mee.

De opwarmkromme zelf lijkt op het afkoelen uit hoofdstuk 5, omgekeerd: hoe dichter bij het evenwicht, hoe trager. Wie niet lang genoeg wacht, meet een te klein ΔT en dus een te grote U.

UITGEWERKT VOORBEELD · 19.15

De meetkas nagerekend

15/19

In hoofdstuk F16 kreeg je voor de meetkas: 3,60 m² kanaalplaat met $U = 3{,}9$ geeft 14 W per °C, 'zonder tafel en kieren'. Reken nu volledieger. De kas staat op het tafelblad (1,20 m × 0,60 m, multiplex 18 mm, $\lambda = 0{,}13$) dat de bodem vormt, met onder het tafelblad buitenlucht. Door de kieren wordt de lucht (576 L) ongeveer drie keer per uur ververst (schatting). Bereken H, ook in kJ per uur per °C.

GEGEVEN

- platen: $A = 3{,}60\,\mathrm{m}^2$, $U = 3{,}9$
- tafelblad: $A = 0{,}72\,\mathrm{m}^2$, $d = 0{,}018\,\mathrm{m}$, $\lambda = 0{,}13$, $R_{si} = 0{,}17$ (warmte naar beneden), $R_{se} = 0{,}04$
- ventilatie: $3 \cdot 0{,}576 = 1{,}73\,\mathrm{m}^3/\mathrm{h}$

GEVRAAGD

- H van de meetkas

OPLOSSING

1. Platen: $3{,}60 \cdot 3{,}9 = 14{,}0\,\mathrm{W/K}$.
2. Tafelblad: $R_T = 0{,}17 + \dfrac{0{,}018}{0{,}13} + 0{,}04 = 0{,}348$, $U = 2{,}87$, dus $0{,}72 \cdot 2{,}87 = 2{,}1\,\mathrm{W/K}$.
3. Kieren: $\rho \cdot c \cdot \dot{V} = 1{,}20 \cdot 1005 \cdot \dfrac{1{,}73}{3600} = 0{,}58\,\mathrm{W/K}$.
4. Samen: $H = 14{,}0 + 2{,}1 + 0{,}6 = 16{,}7\,\mathrm{W/K}$, of $16{,}7 \cdot 3{,}6 = 60\,\mathrm{kJ}$ per uur per °C.

ANTWOORD

$H \approx 16{,}7\,\mathrm{W/K}$, ongeveer 60 kJ per uur per °C in plaats van 50. De platen blijven de grootste post (84 %), het dunne tafelblad telt ook mee.

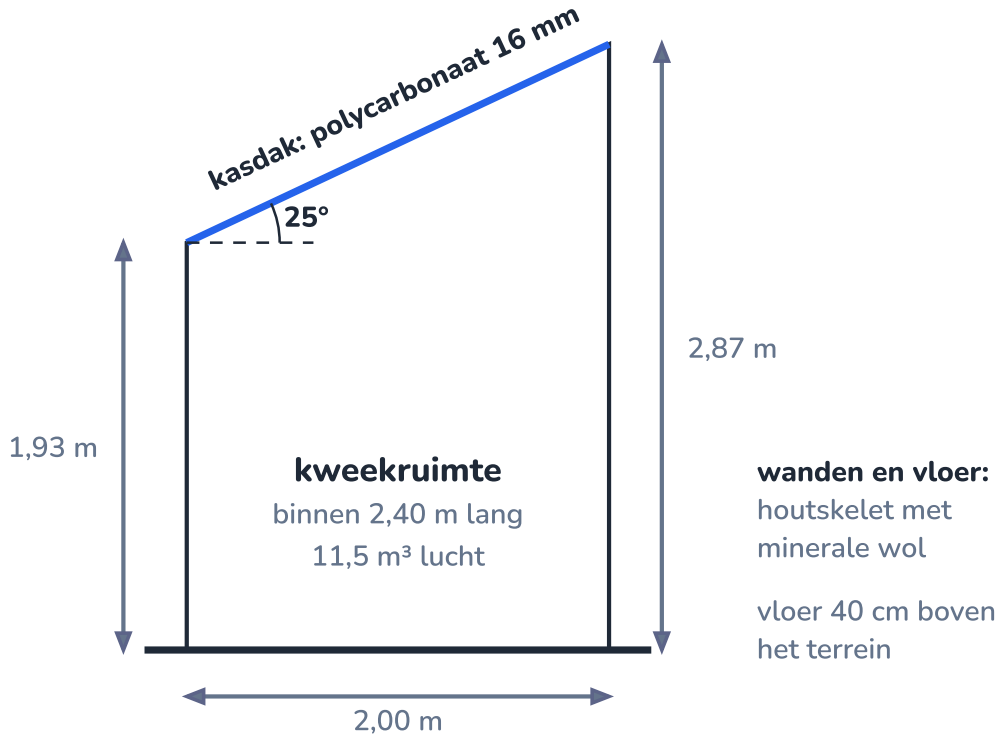
Klopt dit? Het aluminium frame is niet meegerekend: het zit aan de binnenkant van de platen en verhoogt vooral het oppervlak dat warmte afgeeft. Wie de nachtbuffer van hoofdstuk F16 opnieuw doorrekent met 60 in plaats van 50 kJ per uur per °C, vindt 0,7 in plaats van 0,8 °C: de besluiten blijven dezelfde.

UITGEWERKT VOORBEELD · 19.16

Habitat: het warmteverlies van de klimaatmodule

16/19

De kweekruimte van de klimaatmodule is binnen 2,40 m lang en 2,00 m diep, met een kasdak onder 25° (figuur): de lage wand is 1,93 m, de hoge 2,87 m, gemiddeld 2,40 m. Eén zijwand grenst aan het verwarmde station en telt niet mee. Het kasdak is meerwandig polycarbonaat van 16 mm ($U = 2{,}0$) in een aluminium kader ($\psi \approx 0{,}15\,\mathrm{W/(m \cdot K)}$, schatting). Wanden: $U = 0{,}22$; vloer (40 cm boven het terrein, dezelfde opbouw tussen de vloerbalken van hoofdstuk 10): $U = 0{,}21$. De ventilatie ververst 11 m³ per uur (hoofdstuk 14). (a) Bereken H. (b) Welk vermogen is nodig bij −8 °C buiten (20 °C binnen)? (c) Wat levert een nachtscherm onder het dak op, dat een extra weerstand van 0,30 m²·K/W geeft?



Doorsnede van de kweekruimte op schaal: het kasdak (blauw) is de grootste warmtelek.

GEGEVEN

- dak $5{,}30\,\mathrm{m}^2$ ($2{,}40 \cdot 2{,}00 / \cos 25°$), kader $9{,}2\,\mathrm{m}$
- buitenwanden: $2{,}40 \cdot 1{,}93 + 2{,}40 \cdot 2{,}87 + 2{,}00 \cdot 2{,}40 = 16{,}32\,\mathrm{m}^2$
- vloer $4{,}80\,\mathrm{m}^2$; ventilatie $11\,\mathrm{m}^3/\mathrm{h}$

GEVRAAGD

- H
- P bij −8 °C
- effect van het nachtscherm

OPLOSSING

1

DEEL	OPPERVLAKTE OF LENGTE	U OF Ψ	H (W/K)	AANDEEL
kasdak (polycarbonaat 16 mm)	5,30 m²	2,0	10,6	52 %
drie buitenwanden	16,32 m²	0,22	3,6	18 %
vloer (boven buitenlucht)	4,80 m²	0,21	1,0	5 %
aluminium kader van het kasdak	9,2 m	ψ ≈ 0,15 W/(m·K)	1,4	7 %
ventilatie (11 m³/h)		$\rho \cdot c \cdot \dot{V}$	3,7	18 %
totaal			**20,2**	100 %

2 (b) $P = H \cdot \Delta T = 20{,}2 \cdot 28 = 567\,\mathrm{W}$.

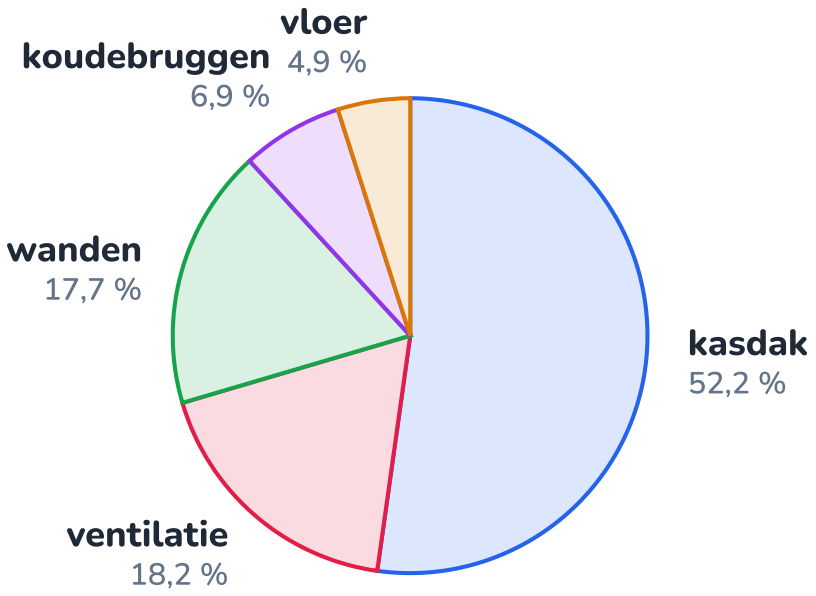
3 (c) Dak met scherm: $U = \dfrac{1}{1/2{,}0 + 0{,}30} = 1{,}25$, dus $H = 16{,}3\,\mathrm{W/K}$ (20 % minder), en bij −8 °C $455\,\mathrm{W}$.

ANTWOORD

(a) $H \approx 20{,}2\,\mathrm{W/K}$: de 20 W per °C waarmee hoofdstuk 18 rekende. Het kasdak is goed voor meer dan de helft. (b) Ongeveer 567 W bij −8 °C. (c) Met een nachtscherm 16,3 W/K.

Klopt dit? Het brein van het station (hoofdstuk E39) rekent met 50 W per °C voor het hele station: de klimaatmodule neemt daar ongeveer 40 % van voor haar rekening, vooral door haar doorzichtige dak. Op een winterdag (gemiddeld 4 °C buiten) verliest ze $20{,}2 \cdot 16 \cdot 24 = 7{,}8\,\mathrm{kWh}$ warmte. De lampen leveren daarvan 3,0 kWh (hoofdstuk 20); de rest, ongeveer 4,7 kWh, komt van de verwarming

van het station, en dat kan niet uit de kleine batterij van de module. Met een warmtepomp (hoofdstuk E16) kost het ongeveer een derde daarvan aan elektriciteit.



Aandeel van elk deel in het warmteverlies van de module (zonder nachtscherm).

CODE · 19.17

Een wandrekenaar in Python

17/19

Wie varianten wil vergelijken (dikkere wol, meer hout, een ander dak), laat de computer rekenen. Het programma telt de lagen op tot een R_T, combineert de wegen door de wol en door het hout, en bouwt het warmteverlies van de module op uit oppervlaktes, U-waarden, de koudebrug van het kader en de ventilatie.

Python `wandrekenaar.py` 33 regels

```python
# wandrekenaar.py - U-waarden met lagen en houtstijlen, en het warmteverlies van de klimaatmodule
RSI, RSI_OMLAAG, RSE = 0.13, 0.17, 0.04       # overgangsweerstanden (m2 K/W)


def R_totaal(lagen, rsi=RSI, rse=RSE):
    """lagen: lijst van (dikte in m, lambda in W/(m K))."""
    return rsi + sum(d / lam for d, lam in lagen) + rse


def U_met_hout(lagen_iso, lagen_hout, f_hout, rsi=RSI):
    """Twee parallelle wegen: door de isolatie en door het hout (aandeel f_hout)."""
    return (1 - f_hout) / R_totaal(lagen_iso, rsi) + f_hout / R_totaal(lagen_hout, rsi)


OSB, WOL, VUREN, VEZEL = 0.13, 0.035, 0.13, 0.045
U_wand = U_met_hout([(0.015, OSB), (0.145, WOL), (0.040, VEZEL)],
                    [(0.015, OSB), (0.145, VUREN), (0.040, VEZEL)], 0.12)
U_vloer = U_met_hout([(0.018, OSB), (0.145, WOL), (0.040, VEZEL)],
                     [(0.018, OSB), (0.145, VUREN), (0.040, VEZEL)], 45 / 400, RSI_OMLAAG)
U_dak = 2.0                                   # meerwandig polycarbonaat 16 mm (fabrikant)

delen = [("kasdak", 5.30, U_dak), ("wanden", 16.32, U_wand), ("vloer", 4.80, U_vloer)]
H = 0.0
```

```python
for naam, A, U in delen:
    print(f"{naam:7s} A = {A:5.2f} m2   U = {U:.3f}   U*A = {U * A:5.2f} W/K")
    H += U * A
H_kb = 0.15 * 9.21                          # aluminium kader rond het kasdak: psi maal lengte
H_vent = 1.20 * 1005 * 11 / 3600            # 11 m3 lucht per uur verversen
H += H_kb + H_vent
print(f"koudebruggen {H_kb:.2f} W/K, ventilatie {H_vent:.2f} W/K")
print(f"totaal H = {H:.1f} W/K")
for T_buiten in (-8, 0, 4):
    print(f"buiten {T_buiten:3d} C: {H * (20 - T_buiten):4.0f} W")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
kasdak  A =  5.30 m2   U = 2.000   U*A = 10.60 W/K
wanden  A = 16.32 m2   U = 0.218   U*A =  3.56 W/K
vloer   A =  4.80 m2   U = 0.213   U*A =  1.02 W/K
koudebruggen 1.38 W/K, ventilatie 3.69 W/K
totaal H = 20.2 W/K
buiten  -8 C:  567 W
buiten   0 C:  405 W
buiten   4 C:  324 W
```

Twee dingen om zelf te proberen (opdracht 19.10): vervang het dak door 16 mm plaat met drie wanden ($U = 2{,}4$) en kijk hoeveel H stijgt; of zoek hoe dik de wol moet zijn om zonder houtvezelplaat aan 0,24 te komen. De functie `U_met_hout` is de formule voor parallelle wegen; `R_totaal` die voor lagen in serie.

NASLAG · 19.18

Overzicht: warmtetransport en isolatie

18/19

warmtegeleiding door een plaat

$$P = \frac{\lambda \cdot A \cdot \Delta T}{d}$$

symbool	betekenis	eenheid
λ	warmtegeleidingscoëfficiënt	W/(m·K)
d	dikte	m
A	oppervlakte	m²

warmteweerstand en U-waarde

$$R = \frac{d}{\lambda} \qquad R_T = R_{si} + \sum \frac{d}{\lambda} + R_{se} \qquad U = \frac{1}{R_T}$$

symbool	betekenis	eenheid
R	warmteweerstand	m²·K/W
U	warmtedoorgangscoëfficiënt	W/(m²·K)

warmteverlies

$$P = U \cdot A \cdot \Delta T \qquad H = \sum UA + \sum \psi L + \dot{V}$$

symbool	betekenis	eenheid
H	warmteverliescoëfficiënt	W/K
ψ	lineaire koudebrug	W/(m·K)
$\dot{V}$	ventilatiedebiet	m³/s

parallelle wegen (f: aandeel van weg 2)

$$U_{\text{gem}} = (1 - f)\,U_1 + f\,U_2$$

binnenoppervlaktetemperatuur en temperatuurfactor

$$T_{si} = T_i - U R_{si}(T_i - T_e) \qquad f_{Rsi} = 1 - U R_{si}$$

materiaal	λ (W/(m·K))	materiaal	λ (W/(m·K))
koper	400	gipskarton	0,25
aluminium	± 200	polycarbonaat	0,20
staal	± 50	vuren hout, OSB, multiplex	0,13
beton	± 1,5	houtvezelisolatieplaat	0,045
glas	± 1,0	minerale wol, EPS (piepschuim)	0,035
baksteen	± 0,9	PUR, PIR	0,022 à 0,026
water	0,6	stilstaande lucht	0,026

Afgeronde rekenwaarden. Isolatieplaten hebben een gecertificeerde λ op de verpakking (de 'lambda-waarde'); reken met die waarde.

overgangsweerstand	m²·K/W
R_{si} wand / dak / vloer	0,13 / 0,10 / 0,17
R_{se} (buiten)	0,04
binnenshuis, aan de 'buitenkant' van een doos	0,13

Habitat: rekenwaarden uit dit hoofdstuk	
wand klimaatmodule	OSB 15 mm, minerale wol 145 mm tussen vuren stijlen (12 % hout), houtvezelplaat 40 mm, pleister: U = 0,22
vloer	OSB 18 mm, wol 145 mm tussen de balken (h.o.h. 40 cm), houtvezelplaat 40 mm: U = 0,21
kasdak	polycarbonaat 16 mm, vijf wanden: U = 2,0; 5,30 m²; aluminium kader $\psi \approx 0{,}15$ over 9,2 m
warmteverlies module	H = 20,2 W/K (met nachtscherm 16,3); 567 W bij −8 °C buiten
meetkas	H ≈ 16,7 W/K (platen 14,0, tafelblad 2,1, kieren 0,6)

HET GEHEEL · 19.19

Waar isolatie naartoe leidt

19/19

Warmtetransport verbindt **Energie** met **Ontwerpen en maken**: een wandopbouw kiezen is een ontwerpbeslissing met afwegingen (licht tegenover isolatie, dikte tegenover ruimte, kosten tegenover verbruik). De rekenregels zijn die van de **Elektriciteit**: lagen in serie en wegen in parallel. Met de isolatiedoos zit je in **Onderzoeken en meten**, met condens en schimmel in **Materie** (het dauwpunt).

F6 Het deeltjesmodel
Condens en dauwpunt: waarom vocht op koude oppervlakken neerslaat.

F16 Warmte en temperatuur
Geleiding, stroming en straling; de eerste U-waarden en de koudebrug van het frame.

F20 Serie- en parallelschakeling
Weerstanden in serie en parallel: dezelfde rekenregels als lagen en parallelle wegen.

I5 Groei en verval: exponentiële modellen
Opwarmen en afkoelen van de isolatiedoos: exponentieel naar een evenwicht.

I10 Moment en evenwicht
De vloerbalken van de module, waartussen de vloerisolatie zit.

I14 Reacties en stoichiometrie
De ventilatie van 11 m³ per uur voor CO_2, die ook warmte kost.

I18 Warmte berekenen
Warmte opslaan en warmte verliezen: samen bepalen ze hoe snel de module afkoelt.

I20 Zonne-energie dimensioneren
Waarom de verwarming niet in het energiebudget van de module zit.

I22 De spanningsdeler en sensoren
De temperatuur verdeelt zich over de lagen zoals de spanning over een spanningsdeler.

I31 Regelsystemen
De verwarming, het nachtscherm en de ventilatie van de module regelen met hysterese.

I38 Duurzaam ontwerpen
Isolatiematerialen kiezen op hun milieu-impact: houtvezel, wol of PIR.

E16 Warmtepompen en koelen
Het warmteverlies van de module dekken met een warmtepomp.

E17 Warmteoverdracht kwantitatief
Convectie en straling in formules, thermische weerstanden en een dynamisch warmtemodel.

E39 PID-regeling
Het warmtemodel van het station (50 W per °C) waarin de module zit.

Habitat

Voor Habitat leggen we vast: de wanden en de vloer van de klimaatmodule zijn houtskelet met 145 mm minerale wol en een doorlopende houtvezelplaat van 40 mm (U = 0,22 en 0,21); het kasdak is meerwandig polycarbonaat van 16 mm (U = 2,0) met een nachtscherm. Samen met de ventilatie verliest de module 20,2 W per °C verschil met buiten (16,3 met het nachtscherm dicht): ongeveer 570 W bij −8 °C. De verwarming komt van het station. In opdracht 19.12 zoek je hoe je dat verlies nog verkleint.

SAMENVATTING

In het kort

- Warmtegeleiding door een plaat: $P = \lambda \cdot A \cdot \Delta T / d$. De lambda-waarde gaat van 400 (koper) tot 0,026 W/(m·K) (stilstaande lucht); isolatiematerialen sluiten lucht op.
- De warmteweerstand van een laag is $R = d/\lambda$. Lagen in serie: de weerstanden tellen op, samen met R_{si} en R_{se}. De U-waarde is $1/R_T$.
- Warmteverlies: $P = U \cdot A \cdot \Delta T$ per deel; voor een gebouw $H = \sum UA + \sum \psi L + \rho c \dot{V}$ en $P = H \cdot \Delta T$.
- Het temperatuurverschil verdeelt zich over de lagen naar hun weerstand. Een koud binnenoppervlak (kleine temperatuurfactor) geeft condens en schimmel; de dampremmende laag hoort aan de warme kant.
- Koudebruggen: stijlen, metalen profielen, onderbroken isolatie. Reken met parallelle wegen of met ψ; de remedie is een ononderbroken isolatielaag.
- Dubbel glas wint weinig door de lucht zelf; de coating (tegen straling) en het argon maken HR-glas (U ≈ 1,1) en driedubbel glas (U ≈ 0,6) zoveel beter.
- Een gemeten U-waarde is groter dan de berekende: randen, naden en kabels zijn koudebruggen.
- De klimaatmodule verliest 20,2 W per °C, meer dan de helft door het kasdak; met een nachtscherm 16,3 W per °C.

BEGRIPPEN

Wat je nu kent

lambda-waarde
De warmtegeleidingscoëfficiënt λ van een materiaal in W/(m·K): hoeveel watt door een plaat van 1 m² en 1 m dik gaat bij 1 K verschil. Hoe kleiner, hoe beter het isoleert.

R-waarde
De warmteweerstand van een laag, $R = d/\lambda$, in m²·K/W. Lagen na elkaar: de R-waarden tellen op.

overgangsweerstand
De warmteweerstand van de dunne luchtlaag tegen een oppervlak: R_{si} binnen (0,13 voor een wand), R_{se} buiten (0,04).

U-waarde
De warmtedoorgangscoëfficiënt van een wand, dak of raam: $U = 1/R_{\text{totaal}}$ in W/(m²·K); het verlies is $P = U \cdot A \cdot \Delta T$.

warmteverliescoëfficiënt
De som H van alle $U \cdot A$ (plus koudebruggen en ventilatie) van een gebouw, in W/K: het verlies per graad verschil met buiten.

koudebrug
Een plaats waar warmte makkelijker naar buiten lekt dan in de rest van de wand, door een beter geleidend materiaal, een onderbroken isolatielaag of een hoek.

ψ-waarde
Het extra warmteverlies van een lineaire koudebrug, in W per meter lengte per kelvin.

temperatuurfactor
$f = (T_{si} - T_e)/(T_i - T_e)$: hoe warm het binnenoppervlak blijft; in België gebruikt men $f \geq 0{,}7$ als vuistregel tegen schimmel.

hoogrendementsglas

Dubbel glas met een coating die warmtestraling terugkaatst en een edelgas (argon) in de spouw: U ongeveer 1,1 W/(m²·K).

HANDBOEK · HOOFDSTUK 20

Zonne-energie dimensioneren

Hoeveel zon, hoeveel panelen, hoeveel batterij: de klimaatmodule op eigen zonnestroom

Op een heldere zomermiddag valt er ongeveer 1000 W zonlicht op elke vierkante meter. Een zonnepaneel maakt daar een vijfde van elektriciteit van, maar alleen overdag, en in december maar een derde van wat het in juni levert. In dit hoofdstuk leer je rekenen met **zonnestraling** in W/m² en met de hoek waaronder ze invalt, met het **rendement** en de **wattpiek** van een paneel, en met de opbrengst per dag en per seizoen uit PVGIS. Je stelt een **energiebudget** op, dimensioneert een **batterij** voor de nacht en voor grijze dagen, en kiest de panelen en de batterij van de klimaatmodule van Habitat. Het antwoord is eerlijker dan je zou hopen: de jaarbalans klopt, de winter niet.

10× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 9 lestijden

E Energie · L Elektriciteit en elektronica · W Wiskundig modelleren

Vier panelen onder 60° voor de klimaatmodule: steil, om de lage winterzon zo goed mogelijk te vangen.

Na dit hoofdstuk

- Je kan zonnestraling uitdrukken in W/m² en in kWh/m² per dag (volle zonuren), en de standaardtestcondities uitleggen.
- Je kan met $G = G_n \cdot \cos\theta$ berekenen hoeveel zonlicht een paneel vangt bij een gegeven zonnehoogte en hellingshoek.
- Je kan het rendement en het piekvermogen (Wp) van een paneel berekenen, en het effect van de celtemperatuur inschatten.
- Je kan een I-U-kromme meten en er het punt van het grootste vermogen uit halen.
- Je kan met PVGIS de opbrengst van een installatie per dag, per maand en per jaar berekenen, en een kritieke maand kiezen.
- Je kan een energiebudget opstellen met vermogen en gebruiksduur per verbruiker, en de grootste posten herkennen.
- Je kan een batterij dimensioneren voor de nacht en voor een aantal dagen autonomie, met ontlaaddiepte en rendement.
- Je kan een energiebudget per maand en per uur doorrekenen in Python en de uitkomst beoordelen.
- Je kan het zonnepaneel en de batterij van de klimaatmodule kiezen en die keuze met getallen verantwoorden.

UITLEG · 20.1

Wat je al weet, en wat nu komt

1/21

In hoofdstuk F17 koos je voor de meetkas een paneel van 50 Wp en een batterij van ongeveer 250 Wh. Je leerde wat een wattpiek is, rekende met volle zonuren en met de tabel van PVGIS, en zag dat een paneel in december maar een derde levert van wat het in juni levert. In hoofdstuk 6 koos je de hellingshoek van het paneel van de klimaatmodule: 60°, voor de winter. In hoofdstuk 16 viel de keuze op een LiFePO4-batterij van 1,2 kWh bruikbaar, met de vraag of die volstaat.

uit vorige hoofdstukken	in dit hoofdstuk
watt en wattuur, wattpiek, volle zonuren, PVGIS (F17)	zonnestraling in W/m², rendement en Wp berekenen, PVGIS per maand gebruiken
$P = U \cdot I$, energie in kWh (F21)	de I-U-kromme en het punt van het grootste vermogen
de zonnehoogte en de invalshoek, 60° voor de winter (hoofdstuk 6)	$G = G_n \cdot \cos\theta$: hoeveel licht het paneel vangt
de pomp: 12 W, plan B 48 Wh per dag (hoofdstuk 9)	het volledige energiebudget van de klimaatmodule
LiFePO4 12,8 V, 120 Ah, 1,2 kWh bruikbaar (hoofdstuk 16)	volstaat die batterij? nachtbuffer en dagen autonomie
de meetkas: 50 Wp en 250 Wh (F17)	de klimaatmodule: panelen en batterij definitief kiezen

Dat is de vraag van dit hoofdstuk. Je

- rekent met zonnestraling in W/m², met de hoek waaronder ze invalt en met het rendement van een paneel;
- meet de I-U-kromme van een paneel en vindt het punt van het grootste vermogen;
- berekent de opbrengst per dag en per maand, en stelt het energiebudget van de klimaatmodule op;
- dimensioneert de batterij voor de nacht en voor grijze dagen, met een programma dat elke maand uur per uur doorrekent;
- kiest de panelen en de batterij van de klimaatmodule, en zegt eerlijk wat niet lukt.

VERHAAL · 20.2

Van 6 % in een laboratorium naar 21 % op een dak

2/21

In april 1954 stelden drie onderzoekers van Bell Labs in de Verenigde Staten, Daryl Chapin, Calvin Fuller en Gerald Pearson, de eerste bruikbare siliciumzonnecel voor. Ze zette ongeveer **6 %** van het zonlicht om in elektrische energie: genoeg om een speelgoedmolentje en een kleine radiozender te laten werken. Kranten schreven over een 'zonnebatterij'. Maar de cellen waren peperduur: stroom uit de zon kostte honderden keren meer dan stroom uit het net.

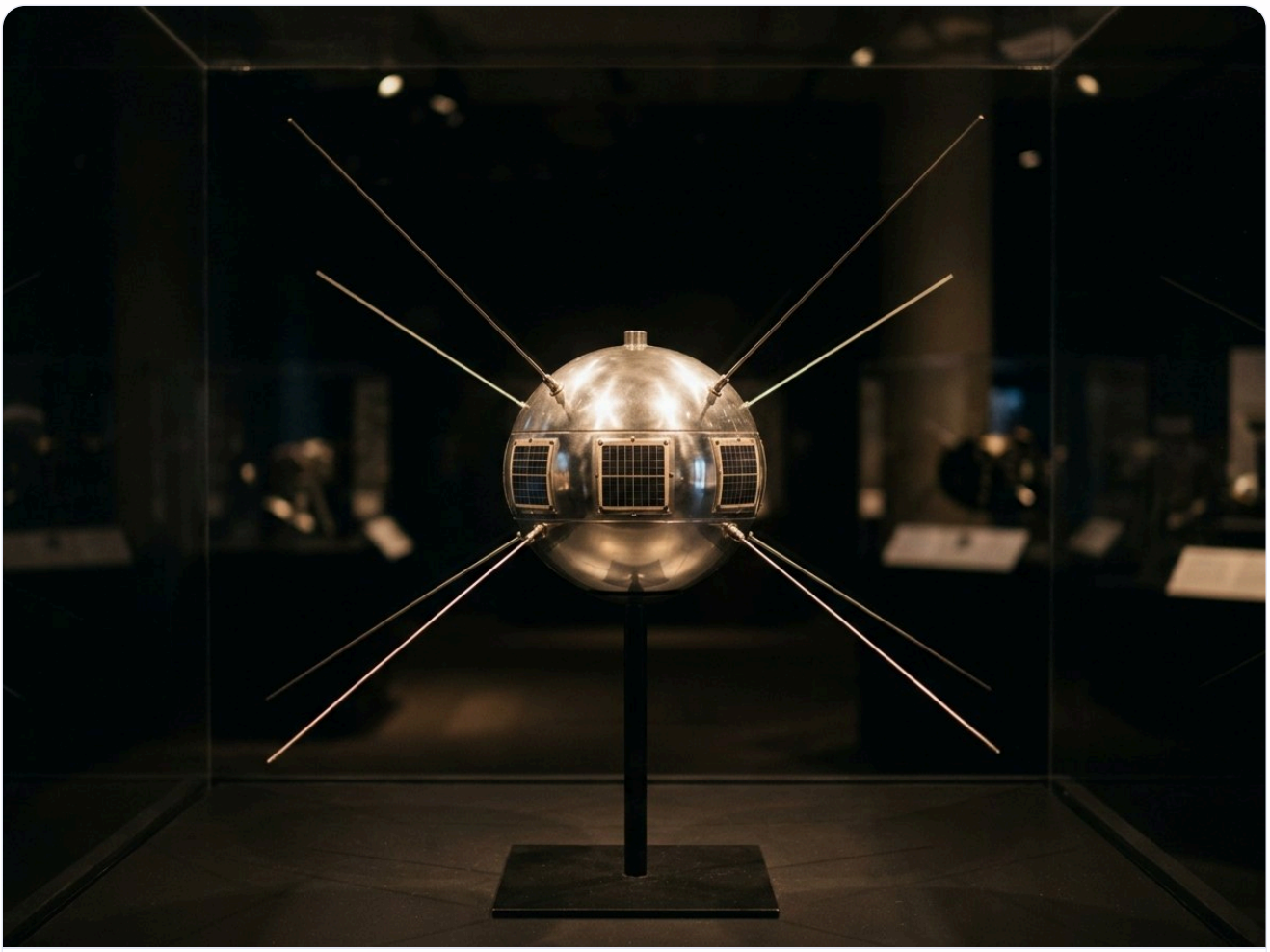
De eerste klant die de prijs niet erg vond, was de ruimtevaart. In maart 1958 lanceerden de Verenigde Staten de satelliet **Vanguard 1**, een bolletje van 16 cm met zes kleine zonnepanelen. Een van zijn zenders werkte op een gewone batterij en viel na enkele maanden stil; de zender op zonnecellen bleef jaren werken. Vanguard 1 draait nog altijd rond de aarde: het oudste door mensen gemaakte voorwerp in een baan om de aarde.

Zeventig jaar later zetten gewone panelen op een dak ongeveer 20 à 22 % van het zonlicht om, en zijn ze zo goedkoop geworden dat België in 2025 al 15 % van zijn elektriciteit uit de zon haalde (hoofdstuk F17). Wie een systeem ontwerpt dat alleen op de zon draait, zoals een satelliet of de klimaatmodule van

Habitat, stelt nog altijd dezelfde vragen als de ingenieurs van Vanguard: hoeveel licht valt er, hoeveel maakt het paneel ervan, en hoe overbrug je de uren zonder zon?

Waarom dit verhaal ertoe doet

Een net aansluiten is makkelijk: tekort en overschot lost het net op. Een eiland op zonnestroom (een satelliet, een poolstation, een berghut, de klimaatmodule) moet elke wattuur zelf verantwoorden. Dat is dimensioneren.



Een replica van Vanguard 1 (1958): zes kleine zonnepanelen hielden een zender jarenlang in leven.

UITLEG · 20.3

Zonnestraling: watt per vierkante meter

3/21

Buiten de dampkring valt er op elke vierkante meter die loodrecht op de zonnestralen staat **1361 W** zonlicht: de **zonneconstante**. Onderweg door de lucht wordt een deel weerkaatst, verstrooid en opgenomen (door waterdamp, stof, ozon). Aan de grond blijft er op een heldere middag ongeveer **1000 W/m²** over. Die grootheid heet de **instraling** G, in W/m².

Zonlicht komt op twee manieren aan. Het **directe** licht komt rechtstreeks van de zonneschijf en werpt scherpe schaduwen. Het **diffuse** licht is door de lucht en de wolken verstrooid en komt uit de hele hemel. Op een grijze dag is er alleen diffuus licht, vaak maar 50 à 200 W/m². In België is ongeveer de helft van al het zonlicht over een jaar diffuus.

Wat je voor een paneel wil weten, is niet het vermogen op één ogenblik, maar de **energie** per dag: de oppervlakte onder de kromme van de instraling. In kWh per m² per dag is dat meteen het aantal **volle zonuren**: evenveel uren aan 1000 W/m² als die dag oplevert (hoofdstuk F17).

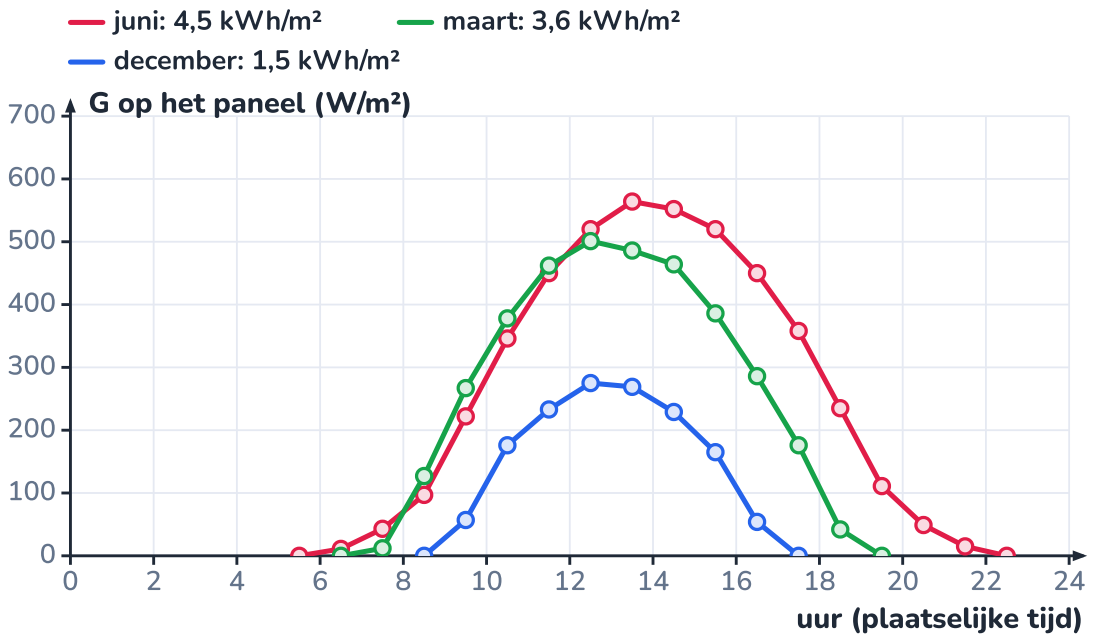
ENERGIE VAN HET ZONLICHT PER DAG

$$H_{\text{dag}} = \sum G \cdot \Delta t \qquad \text{volle zonuren} = \frac{H_{\text{dag}}}{1000\,\text{W/m}^2}$$

H_{dag} in Wh/m² (of kWh/m²) per dag; met uurwaarden van G is $\Delta t = 1\,\text{h}$.

Standaardtestcondities (STC)

Om panelen te kunnen vergelijken, meet men ze allemaal onder dezelfde afgesproken omstandigheden: 1000 W/m² loodrecht op het paneel, een celtemperatuur van 25 °C en een spectrum zoals zonlicht na 1,5 keer de dikte van de dampkring (AM1,5). Het vermogen onder STC is de **wattpiek**. In België komt een paneel zelden precies in die omstandigheden: als er 1000 W/m² valt, is het paneel meestal veel warmer dan 25 °C.



PVGIS-gemiddelden: in december valt er maar een derde van het zonlicht van juni, in minder dan acht uur.

Een pyranometer meet de instraling in W/m²: één horizontaal, één in het vlak van de panelen.

UITLEG · 20.4

De hoek van de zon, terug naar hoofdstuk 6

4/21

Een paneel vangt het meeste licht als de stralen er loodrecht op vallen. Maken ze een hoek θ met de **normaal** (de lijn loodrecht op het paneel), dan wordt dezelfde bundel over een grotere oppervlakte uitgesmeerd, en valt er per m² minder (hoofdstuk 6):

DIRECT ZONLICHT OP EEN PANEEL DAT NAAR HET ZUIDEN KIJKT, OP HET MIDDAGUUR

$$G_{\text{paneel}} = G_n \cdot \cos\theta \qquad \theta = |\,(90^\circ - \beta) - h\,|$$

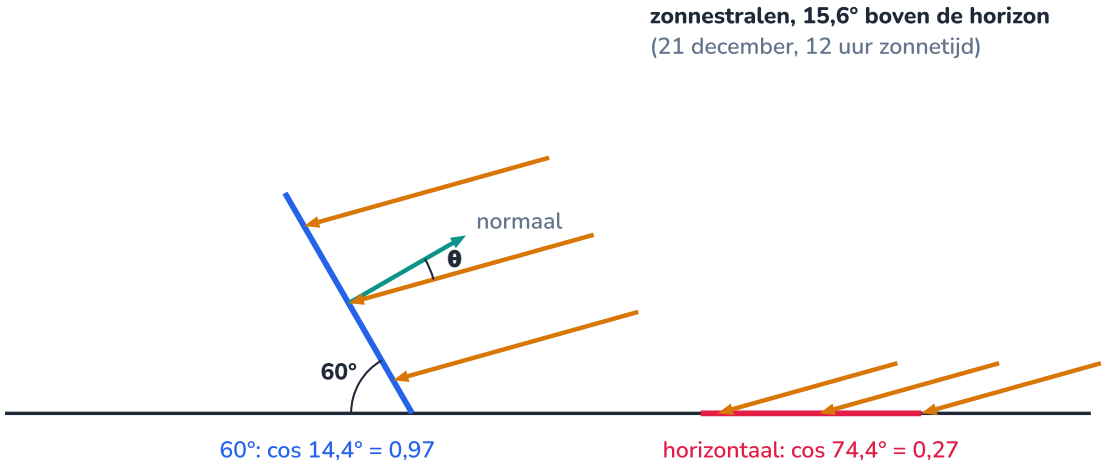
G_n: instraling loodrecht op de stralen; β: hellingshoek van het paneel; h: zonnehoogte. De formule voor θ geldt alleen op het middaguur, voor een paneel op het zuiden.

In Brussel staat de middagzon op 21 december maar 15,6° hoog, op 21 juni 62,4° (hoofdstuk 6). Een horizontaal paneel vangt in december dus maar $\cos(90^\circ - 15{,}6^\circ) = 0{,}27$ van het directe licht, een paneel onder 60° bijna alles: $\cos 14{,}4^\circ = 0{,}97$. In juni is het omgekeerd: dan vangt een paneel onder 39° meer ($\cos 11{,}4^\circ = 0{,}98$) dan een onder 60° ($\cos 32{,}4^\circ = 0{,}84$).

Het diffuse licht komt uit de hele hemel en hangt veel minder af van de richting van het paneel. Een steil paneel 'ziet' wel een kleiner stuk hemel en wat meer grond. PVGIS rekent dat allemaal mee: daarom gebruiken we voor de opbrengst per maand de tabel van PVGIS, en de cosinus om te begrijpen waarom de hellingshoek zo'n verschil maakt.

⚠ Schaduw telt dubbel

Een schaduw op één cel van een paneel kan het hele paneel lamleggen, want de cellen staan in serie (hoofdstuk F17). Daarom staan de rijen panelen van Habitat minstens 3,61 m uit elkaar (hoofdstuk 6): zelfs op 21 december valt de schaduw van de ene rij dan net niet op de volgende.



Bij een lage decemberzon vangt een paneel onder 60° bijna al het directe licht, een horizontaal paneel een kwart.

UITGEWERKT VOORBEELD · 20.5

Een heldere decembermiddag: plat, 39°, 60° of 90°?

5/21

Op 21 december om 12 uur zonnetijd staat de zon in Brussel 15,6° hoog. Het is helder: loodrecht op de stralen valt $G_n = 600\ \text{W/m}^2$ direct licht (de lage zon gaat door een dikke laag lucht). Hoeveel direct licht valt er per m² op een paneel op het zuiden dat horizontaal ligt, dat onder 39° staat, onder 60°, en verticaal tegen een gevel?

GEGEVEN

- $h = 15{,}6^\circ$, $G_n = 600\ \text{W/m}^2$
- $\beta = 0^\circ, 39^\circ, 60^\circ, 90^\circ$

GEVRAAGD

- θ en G_{paneel} voor elke hoek

OPLOSSING

1. Horizontaal: $\theta = |90° - 0° - 15{,}6°| = 74{,}4°$; $G = 600 \cdot \cos 74{,}4° = 600 \cdot 0{,}269 = 161\,\mathrm{W/m^2}$.
2. 39°: $\theta = |51° - 15{,}6°| = 35{,}4°$; $G = 600 \cdot 0{,}815 = 489\,\mathrm{W/m^2}$.
3. 60°: $\theta = |30° - 15{,}6°| = 14{,}4°$; $G = 600 \cdot 0{,}969 = 581\,\mathrm{W/m^2}$.
4. 90°: $\theta = |0° - 15{,}6°| = 15{,}6°$; $G = 600 \cdot 0{,}963 = 578\,\mathrm{W/m^2}$.

ANTWOORD

Horizontaal 161, onder 39° 489, onder 60° 581 en verticaal 578 W/m². In december vangt een steil paneel 3,6 keer zoveel direct licht als een plat paneel.

Klopt dit? Een verticaal paneel doet het in december bijna even goed als een paneel onder 60°, maar in de zomer veel slechter: in juni is θ dan 62,4° en vangt het maar de helft. Het diffuse licht zit niet in deze berekening; op een heldere decemberdag komt er nog ongeveer 50 à 100 W/m² bij.

UITLEG · 20.6

Rendement en wattpiek

Het **rendement** van een paneel is het deel van het zonlicht dat elektrische energie wordt:

RENDEMENT EN PIEKVERMOGEN

$$\eta = \frac{P}{G \cdot A} \qquad P_{\text{piek}} = \eta_{\text{STC}} \cdot 1000\,\mathrm{W/m^2} \cdot A$$

P: elektrisch vermogen (W); G: instraling op het paneel (W/m²); A: oppervlakte van het paneel (m²). Het piekvermogen in Wp is het vermogen onder standaardtestcondities.

De panelen van de klimaatmodule meten 1,65 m × 1,00 m (hoofdstuk 10) en hebben een piekvermogen van **350 Wp**. Hun rendement is $\eta = 350/(1000 \cdot 1{,}65) = 21{,}2\,\%$. De rest van het zonlicht wordt weerkaatst of **warmte**.

En die warmte kost vermogen. Een paneel in de zon wordt veel warmer dan de lucht, en hoe warmer de cel, hoe kleiner haar spanning. Twee vuistregels voor kristallijn silicium:

CELTEMPERATUUR EN VERMOGEN (VUISTREGELS)

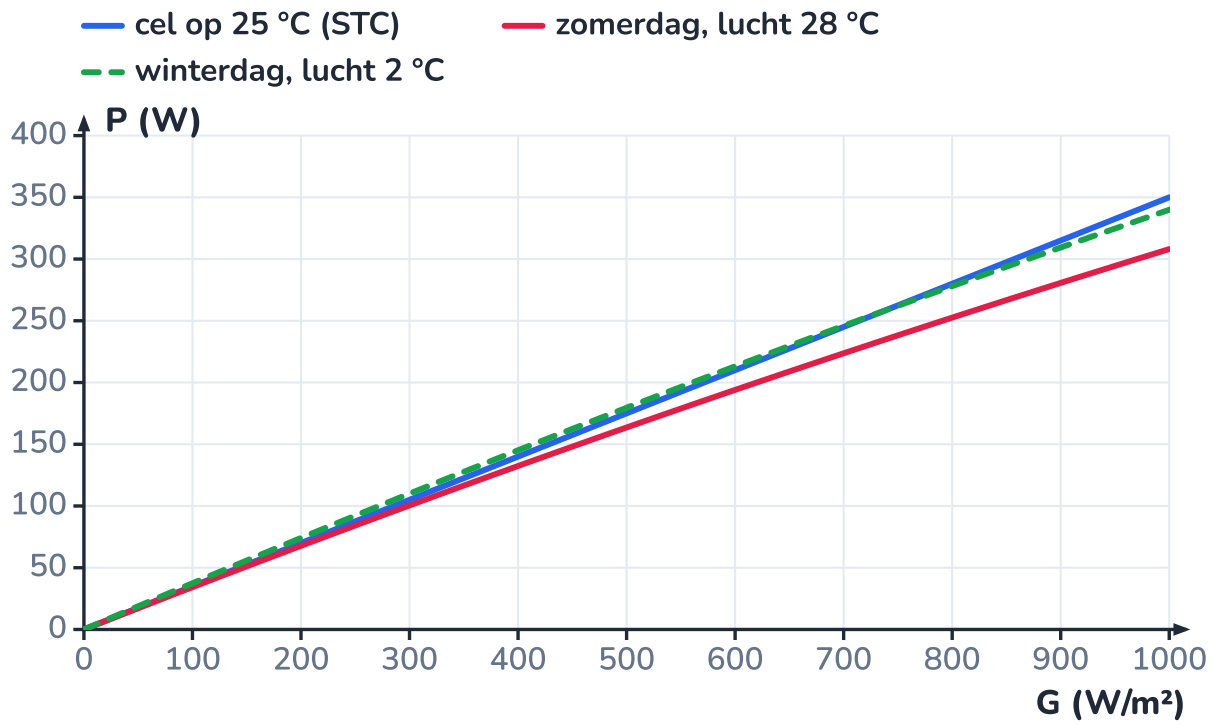
$$T_{\text{cel}} \approx T_{\text{lucht}} + \frac{\text{NOCT} - 20}{800} \cdot G \qquad P \approx P_{\text{piek}} \cdot \frac{G}{1000} \cdot \left(1 + \gamma\,(T_{\text{cel}} - 25)\right)$$

NOCT: de celtemperatuur bij 800 W/m² en 20 °C lucht, typisch ongeveer 45 °C; γ: temperatuurcoëfficiënt, typisch ongeveer $-0{,}35\,\%$ per K ($-0{,}0035$ per K).

Op een zomermiddag (900 W/m², 28 °C) wordt de cel ongeveer 56 °C en levert het paneel ongeveer 281 W in plaats van 315 W. Op een heldere wintermiddag (600 W/m², 2 °C) is de cel maar 21 °C en levert het 213 W: méér dan 60 % van 350. Koud en helder is ideaal voor een zonnepaneel.

Een paneel van 350 Wp levert 350 W

Alleen onder standaardtestcondities, en die zijn in België zeldzaam. Over een heel jaar levert één wattpiek in Brussel ongeveer 1 kWh: een paneel van 350 Wp dus ongeveer 350 kWh per jaar, alsof het 1000 uur op vol vermogen draaide, van de 8760 uur in een jaar.



Het vermogen stijgt recht evenredig met de instraling; een warm paneel levert bij dezelfde instraling minder.

UITGEWERKT VOORBEELD · 20.7

Het paneel van de klimaatmodule

7/21

Een paneel van 1,65 m × 1,00 m heeft op het typeplaatje: 350 Wp, NOCT 45 °C, temperatuurcoëfficiënt −0,35 % per K. (a) Bereken het rendement onder STC. (b) Wat levert het bij 800 W/m² en 20 °C lucht? (c) Hoeveel zonlicht valt er dan op het paneel, en wat is het rendement nu?

GEGEVEN

- $A = 1{,}65\,\mathrm{m}^2$, $P_{\text{piek}} = 350\,\mathrm{Wp}$
- NOCT $= 45\,^\circ\mathrm{C}$, $\gamma = -0{,}0035$ per K
- $G = 800\,\mathrm{W/m}^2$, $T_{\text{lucht}} = 20\,^\circ\mathrm{C}$

GEVRAAGD

- η_{STC}
- P bij 800 W/m²
- η bij 800 W/m²

OPLOSSING

1. (a) $\eta = \dfrac{350}{1000 \cdot 1{,}65} = 0{,}212 = 21{,}2\,\%$.

2. (b) Celtemperatuur: $T_{\text{cel}} = 20 + \dfrac{45 - 20}{800} \cdot 800 = 45\,^\circ\mathrm{C}$ (precies de NOCT). Vermogen: $P = 350 \cdot \dfrac{800}{1000} \cdot (1 - 0{,}0035 \cdot 20) = 280 \cdot 0{,}93 = 260\,\mathrm{W}$.

3. (c) Zonlicht op het paneel: $800 \cdot 1{,}65 = 1320\,\mathrm{W}$. Rendement: $260/1320 = 0{,}197 = 19{,}7\,\%$.

ANTWOORD

(a) 21,2 %. (b) Ongeveer 260 W. (c) Er valt 1320 W op het paneel; het rendement is nu 19,7 %.

Klopt dit? Het rendement daalt met 7 % van zijn waarde omdat de cel 20 K warmer is dan 25 °C. Daarom hangen panelen liefst met wat ruimte erachter, zodat de wind ze koelt. De vier panelen van de module (6,6 m²) leveren onder deze omstandigheden samen ongeveer 1,04 kW.

PROEF · 20.8

De I-U-kromme van een zonnepaneel

8/21

ONDERZOEKSVRAAG

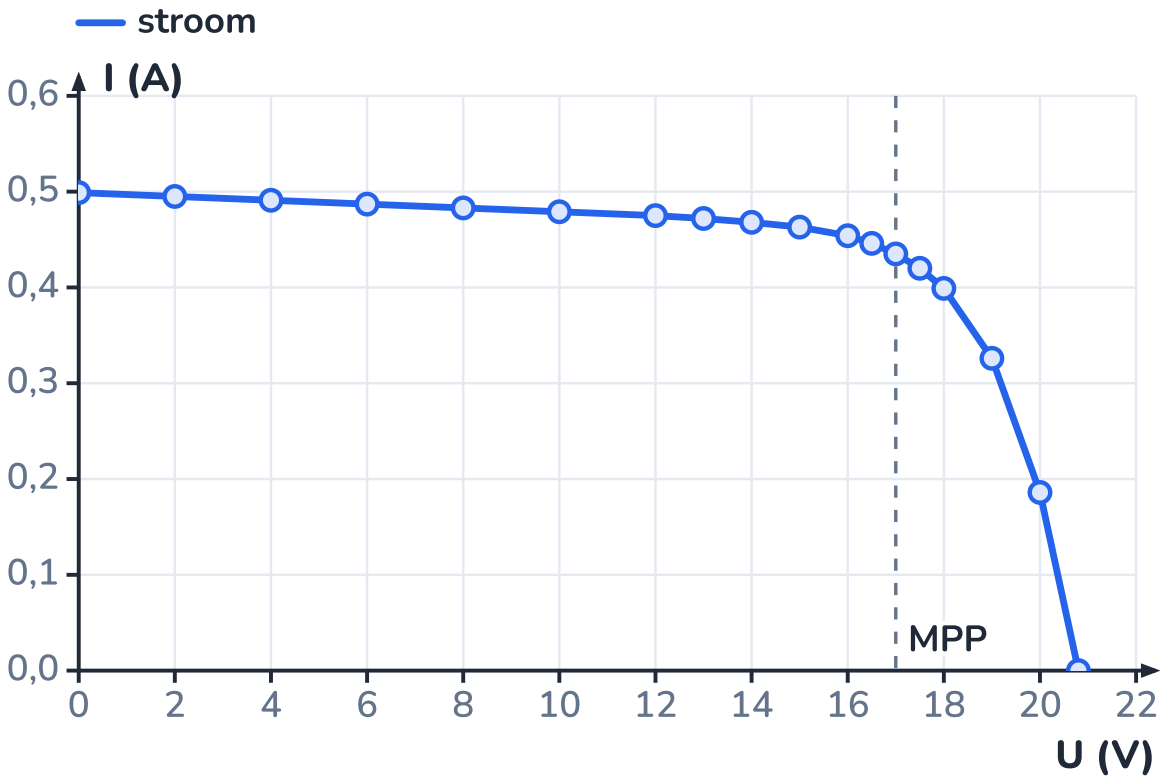
Bij welke spanning levert een zonnepaneel het grootste vermogen?

MATERIAAL

- zonnepaneeltje van 10 Wp (36 cellen, 0,34 m × 0,24 m)
- potentiometer van 1 kΩ (5 W) of een reeks vermogensweerstanden
- twee multimeters: een ampèremeter in serie en een voltmeter over het paneel
- zonlicht, gradenboog, eventueel een stralingsmeter

Veiligheid

Een paneel van 10 Wp levert weinig stroom en is veilig. Grote panelen in serie geven honderden volt gelijkspanning: daar werk je nooit aan.



Tot ongeveer 16 V levert het paneel een bijna constante stroom; daarna zakt ze snel.

WERKWIJZE

1. Richt het paneel loodrecht op de zon en laat het een paar minuten opwarmen tot de metingen stabiel zijn.
2. Meet de kortsluitstroom (alleen de ampèremeter, zonder belasting) en de openklemspanning (alleen de voltmeter).
3. Sluit de potentiometer aan en draai hem in stappen van kortsluiting naar open. Noteer telkens U en I.
4. Bereken bij elke meting het vermogen $P = U \cdot I$ en zet I en P uit tegen U.
5. Meet of schat de instraling en bereken het rendement bij het grootste vermogen.

WAARNEMING

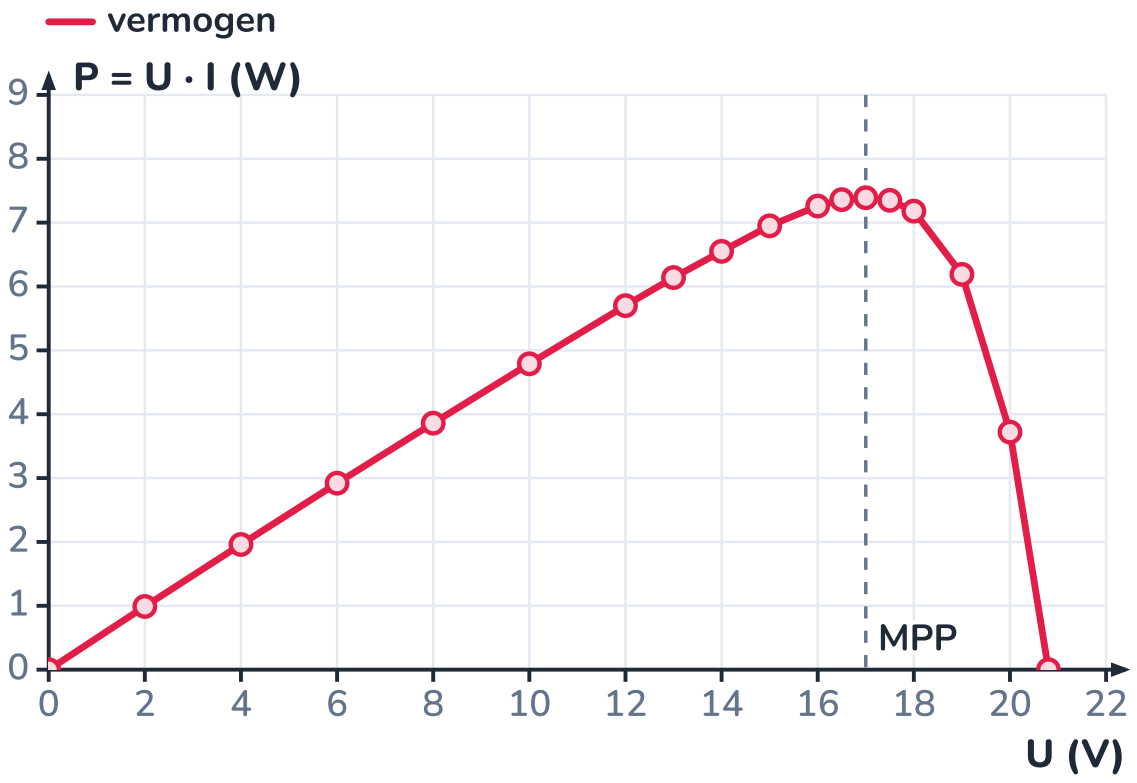
U (V)	0,0	4,0	8,0	12,0	14,0	16,0	17,0	18,0	20,0
I (mA)	499	491	483	475	468	454	435	399	
P (W)	0,00	1,96	3,86	5,70	6,55	7,26	7,39	7,18	

Voorbeeldmeting in de zon (± 800 W/m², paneel ± 40 °C), om de andere meting. Kortsluitstroom 499 mA, openklemspanning 20,8 V. Het grootste vermogen is 7,4 W bij 17 V en 435 mA.

VERKLARING

Bij kleine spanning levert het paneel een bijna constante stroom: het gedraagt zich als een **stroombron**, want de stroom hangt af van hoeveel licht er valt. Boven ongeveer 16 V zakt de stroom snel, tot nul bij de openklemspanning. Het vermogen is nul aan beide uiteinden en heeft een maximum daartussen: het **maximum power point** (MPP), hier bij ongeveer 17 V.

Rendement: het paneel vangt $800 \cdot 0{,}34 \cdot 0{,}24 = 65\ \text{W}$ zonlicht en levert 7,4 W, dus $\eta \approx 11\ \%$: kleine panelen hebben veel rand zonder cellen. Wie een 12 V-batterij rechtstreeks aan dit paneel hangt, dwingt het op ongeveer 13 V: dan levert het maar ongeveer 6,1 W. Een **MPPT-laadregelaar** houdt het paneel op zijn MPP en zet die spanning om naar de spanning van de batterij. Hoe hij dat doet, leer je in hoofdstuk E24.



Het vermogen U · I heeft een top: het MPP, bij ongeveer 17 V en 7,4 W.

UITLEG · 20.9

Opbrengst per dag en per seizoen

9/21

De energie die een installatie per dag levert, volgt uit het piekvermogen en het aantal volle zonuren, met een factor voor alle verliezen samen (warmte, zwak licht, vuil, kabels, laadregelaar): de **prestatieverhouding** PR, in België ongeveer 0,8.

DAGOPBRENGST

$$E_{\text{dag}} = P_{\text{piek}} \cdot \text{volle zonuren} \cdot PR$$

P_{piek} in kWp, volle zonuren in h, E_{dag} in kWh. PVGIS geeft meteen de opbrengst per Wp per dag, verliezen inbegrepen.

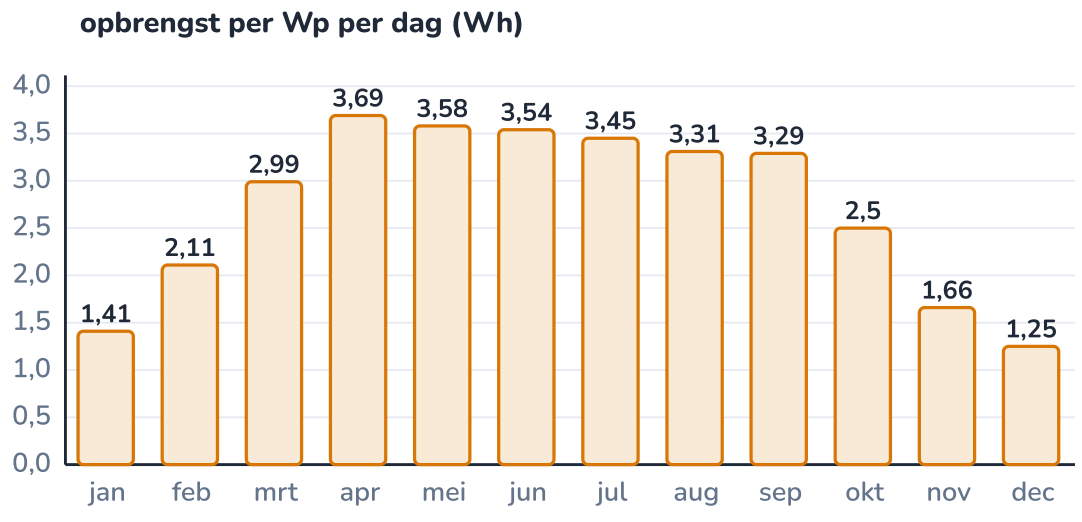
De tabel van PVGIS voor Brussel, 60° zuid, geeft voor elke maand de opbrengst per wattpiek per dag. Over een jaar levert één wattpiek 998 Wh, dus ongeveer 1 kWh. Maar dat jaar is scheef verdeeld: in april levert een paneel bijna drie keer zoveel per dag als in december.

	jan	feb	mrt	apr	mei	jun	jul	aug	sep	okt	nov	dec
Wh per Wp per dag	1,41	2,11	2,99	3,69	3,58	3,54	3,45	3,31	3,29	2,50	1,66	1,25

Wie een installatie ontwerpt die het hele jaar moet werken, kiest daarom een **kritieke maand**: de maand waarin de verhouding tussen verbruik en opbrengst het ongunstigst is. Dimensioneer je voor december, dan heb je in juni bijna drie keer te veel. Dimensioneer je voor juni, dan zit je in december zonder. Er is geen gratis oplossing: kiezen is afwegen.

⚠ Gemiddelden verbergen grijze weken

De getallen van PVGIS zijn gemiddelden over bijna twintig jaar. Een heldere junidag levert anderhalve keer het gemiddelde, een grijze decemberdag vaak minder dan een vijfde. Een installatie die op het gemiddelde precies rondkomt, komt de helft van de dagen tekort. Daarvoor dient de batterij, en een energiebeheer dat weet wat eerst mag uitvallen.



PVGIS, Brussel, 60° zuid: van 1,25 Wh per Wp in december tot 3,69 in april.

UITGEWERKT VOORBEELD · 20.10

Wat levert één paneel van 350 Wp?

10/21

(a) Hoeveel energie levert één paneel van 350 Wp op een gemiddelde dag in juni en in december, en in die hele maand? (b) Hoeveel per jaar? (c) De vier panelen van de klimaatmodule leveren samen 1,40 kWp. Vergelijk hun jaaropbrengst met het jaarverbruik van de module (3,27 kWh per dag, verder in dit hoofdstuk).

GEGEVEN

GEVRAAGD

- $P_{\text{piek}} = 350\,\text{Wp}$; juni 3,54 en december 1,25 Wh per Wp per dag
- jaar: 998 Wh per Wp

- dag- en maandopbrengst
- jaaropbrengst
- vergelijking met het verbruik

OPLOSSING

1. (a) Juni: $350 \cdot 3{,}54 = 1239\,\text{Wh} = 1{,}24\,\text{kWh}$ per dag, $1{,}24 \cdot 30 = 37\,\text{kWh}$ in de maand. December: $350 \cdot 1{,}25 = 438\,\text{Wh}$ per dag, $0{,}438 \cdot 31 = 13{,}6\,\text{kWh}$ in de maand.
2. (b) $350 \cdot 998 = 349\,184\,\text{Wh} \approx 349\,\text{kWh}$ per jaar.
3. (c) Vier panelen: $4 \cdot 349 = 1397\,\text{kWh}$ per jaar. Verbruik: $3{,}27 \cdot 365 = 1193\,\text{kWh}$ per jaar.

ANTWOORD

(a) 1,24 kWh per dag (37 kWh) in juni, 0,44 kWh per dag (13,6 kWh) in december. (b) Ongeveer 349 kWh per jaar. (c) De panelen leveren over het jaar 17 % meer dan de module verbruikt.

Klopt dit? De jaarbalans klopt dus. Maar de module verbruikt elke dag evenveel, en de panelen niet: in december leveren ze 1,75 kWh per dag voor een verbruik van 3,27. Een batterij kan een nacht overbruggen, geen seizoen: 200 kWh van de zomer bewaren voor de winter zou een batterij van meer dan een ton vragen (hoofdstuk E27 onderzoekt waterstof als seizoensopslag).

UITLEG · 20.11

Het energiebudget van de klimaatmodule

11/21

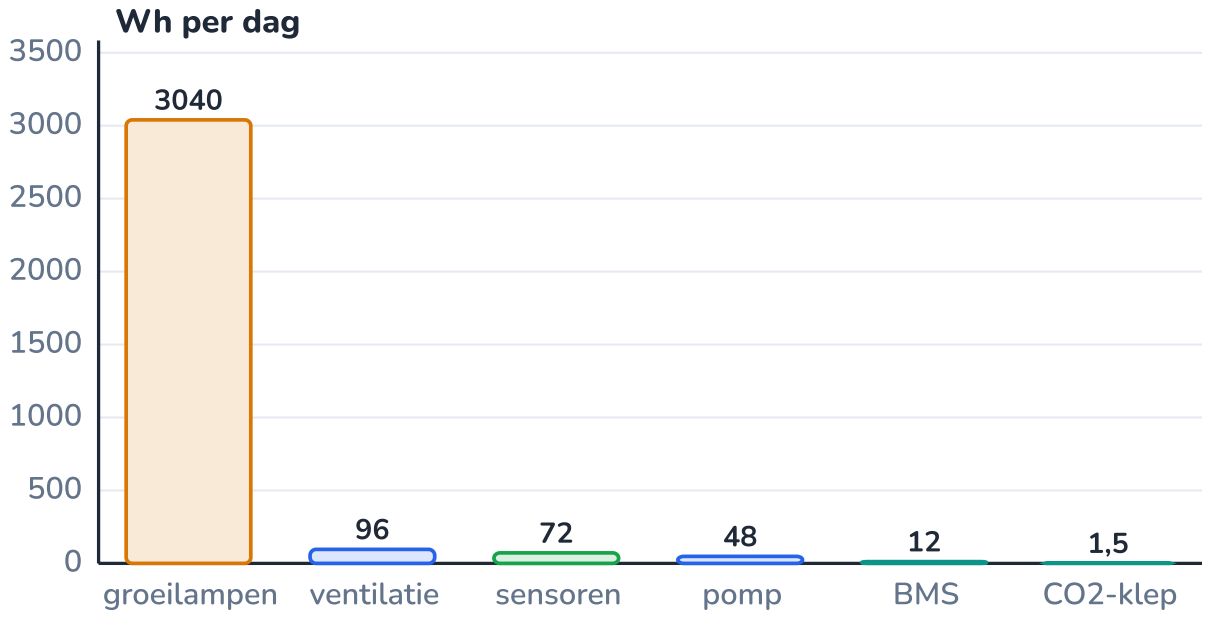
Een **energiebudget** is een lijst van alle verbruikers met hun vermogen en hoe lang ze per dag aanstaan. Vermogen maal tijd geeft de energie per verbruiker (hoofdstuk F21), de som het verbruik per dag. Voor de klimaatmodule komen de vermogens uit de vorige hoofdstukken; de lampen leggen we hier vast: **twee lampen van 95 W** (de zuinige lamp B uit hoofdstuk 4). Eén lamp geeft het kweekbed van 2,88 m² ongeveer 100 µmol lichtdeeltjes per m² per seconde; de slaproef van hoofdstuk 2 rekende met 200.

verbruiker	vermogen	tijd per dag	energie per dag
groeilampen (2 × 95 W, lamp B uit hoofdstuk 4)	190 W	16 h	3040 Wh
ventilatoren (afzuigen en circuleren)	5 W overdag, 2 W 's nachts	16 h + 8 h	96 Wh
microcontroller, sensoren, datalogger	3,0 W	24 h	72 Wh
pomp waterkringloop (eb en vloed, plan B, hoofdstuk 9)	12 W	4 h	48 Wh
laadregelaar en BMS (eigen verbruik)	0,5 W	24 h	12 Wh
CO_2-klep (hoofdstuk 14)	6 W	15 min	1,5 Wh
dakmotor (hoofdstuk 9)		± 30 s	0,2 Wh
totaal			**3270 Wh = 3,27 kWh**
zonder de lampen (basisverbruik)			230 Wh

Eén post overheerst alles: de lampen zijn goed voor **93 %** van het verbruik. Wie op de pomp of de sensoren bespaart, wint enkele procenten; wie de lampen een uur korter laat branden, wint meer dan alle andere verbruikers samen. Dat is de eerste les van elk energiebudget: zoek eerst de grootste post.

Wat niet in het budget zit: de verwarming

De module verliest in de winter ongeveer 20 W per °C (hoofdstuk 19): op een winterdag 7,8 kWh warmte, meer dan het dubbele van het hele budget. Die warmte komt van de verwarming van het station, niet uit de batterij van de module. De lampen helpen wel: hun 3,04 kWh wordt volledig warmte in de module.



De groeilampen verbruiken dertien keer zoveel als alle andere verbruikers samen.

CODE · 20.12

Een energiebudget-rekenaar in Python

12/21

Een gemiddelde dag per maand volstaat niet: of de batterij de nacht haalt, hangt af van *wanneer* er zon is en wanneer de lampen branden. Het programma rekent daarom elke maand uur per uur door, met het gemiddelde dagprofiel van PVGIS (in UTC; de lampen branden van 4 tot 20 uur UTC, dus van 6 tot 22 uur in de zomer). Per uur vergelijkt het de zonnestroom met het verbruik: wat tekortschiet, komt uit de batterij; wat overschiet, laadt de batterij (met 95 % rendement) of gaat verloren als ze vol is.

Python **energiebudget.py** 51 regels

```python
# energiebudget.py - zonnepanelen en batterij van de klimaatmodule, maand per maand
VERBRUIKERS = [                          # (naam, vermogen in W, uren per dag)
    ("groeilampen, 2 x 95 W", 190.0, 16),
    ("pomp waterkringloop", 12.0, 4),
    ("microcontroller en sensoren", 3.0, 24),
    ("ventilatoren overdag", 5.0, 16),
    ("ventilator 's nachts", 2.0, 8),
    ("CO2-klep", 6.0, 0.25),
    ("laadregelaar en BMS", 0.5, 24),
]
N_PANELEN, WP_PANEEL = 4, 350
BATTERIJ = 1229.0       # bruikbare energie (Wh): 120 Ah, 12,8 V, 80 % ontlaaddiepte
ETA_LADEN = 0.95        # rendement bij het laden van de batterij
MAANDEN = ["jan", "feb", "mrt", "apr", "mei", "jun", "jul", "aug", "sep", "okt", "nov", "dec"]
WH_PER_WP = [1.41, 2.11, 2.99, 3.69, 3.58, 3.54, 3.45, 3.31, 3.29, 2.5, 1.66, 1.25]
G = [   # PVGIS: gemiddelde dag, W/m2 op het paneel per uur (UTC), januari tot december
    [0, 0, 0, 0, 0, 0, 0, 0, 68, 177, 248, 273, 286, 248, 198, 120, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 17, 163, 265, 347, 364, 357, 346, 289, 211, 95, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 12, 127, 267, 378, 462, 501, 486, 464, 386, 286, 176, 42, 0, 0, 0, 0, 0, 0],
```

```
    [0, 0, 0, 0, 0, 8, 72, 211, 361, 494, 572, 611, 587, 537, 449, 350, 215, 84, 10, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 34, 96, 221, 360, 478, 534, 575, 572, 511, 426, 337, 214, 96, 35, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 11, 43, 97, 222, 346, 450, 520, 564, 552, 520, 450, 358, 235, 111, 49, 15, 0, 0, 0, 0],
    [0, 0, 0, 0, 4, 35, 86, 208, 340, 447, 499, 549, 565, 513, 452, 366, 243, 120, 48, 13, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 15, 72, 196, 334, 453, 519, 544, 528, 487, 420, 338, 224, 99, 25, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 55, 196, 349, 461, 539, 564, 535, 495, 404, 304, 179, 46, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 116, 250, 344, 438, 457, 434, 400, 305, 213, 72, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 16, 159, 233, 314, 332, 314, 278, 198, 102, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 57, 176, 233, 275, 269, 229, 165, 54, 0, 0, 0, 0, 0, 0, 0, 0]
]

E_dag = sum(P * t for naam, P, t in VERBRUIKERS)
print(f"verbruik: {E_dag:.0f} Wh per dag, waarvan {190 * 16} Wh voor de lampen")
P_licht = 190 + 12 * 4 / 16 + 3 + 5 + 6 * 0.25 / 16 + 0.5    # gemiddeld tijdens de 16 lichturen (W)
P_nacht = 3 + 2 + 0.5
kWp = N_PANELEN * WP_PANEEL / 1000

print("maand  zon (Wh)  uit batterij (Wh)  saldo (Wh)  oordeel")
for m in range(12):
    schaal = WH_PER_WP[m] / (sum(G[m]) / 1000)             # van instraling naar opbrengst
    zon = [g / 1000 * kWp * 1000 * schaal for g in G[m]]    # W, uur per uur
    last = [P_licht if 4 <= h < 20 else P_nacht for h in range(24)]   # lampen van 4 tot 20 uur UTC
    uit = sum(max(0.0, l - z) for z, l in zip(zon, last))   # uit de batterij
    over = sum(max(0.0, z - l) for z, l in zip(zon, last))  # naar de batterij, of verloren
    saldo = over * ETA_LADEN - uit
    if saldo < 0:
        oordeel = "te weinig zon"
    elif uit > BATTERIJ:
        oordeel = "batterij te klein"
    else:
        oordeel = "ok"
    print(f"{MAANDEN[m]:5s} {sum(zon):9.0f} {uit:14.0f} {saldo:13.0f}   {oordeel}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
verbruik: 3270 Wh per dag, waarvan 3040 Wh voor de lampen
maand  zon (Wh)  uit batterij (Wh)  saldo (Wh)  oordeel
jan        1974            1831         -1322   te weinig zon
feb        2954            1527          -376   te weinig zon
mrt        4186            1244           808   batterij te klein
apr        5166            1056          1749   ok
mei        5012             956          1608   ok
jun        4956             898          1557   ok
jul        4830             924          1436   ok
aug        4634            1024          1245   ok
sep        4606            1143          1213   ok
okt        3500            1436           147   batterij te klein
nov        2324            1729          -985   te weinig zon
dec        1750            1930         -1540   te weinig zon
```

Lees de kolommen zo. **Zon**: wat de vier panelen op een gemiddelde dag leveren. **Uit batterij**: wat de batterij tussen zonsondergang en de ochtend moet leveren, plus de uren waarin de zon te zwak is voor de lampen. **Saldo**: wat er over een hele dag overblijft na het terugladen; is het negatief, dan kan geen batterij helpen, want er is te weinig zon om ze te vullen. In opdracht 20.10 wordt het programma jouw ontwerpgereedschap.

UITLEG · 20.13

De batterij dimensioneren

13/21

Een batterij in een zonnesysteem heeft twee taken, en je rekent ze apart uit.

1. De nacht overbruggen (de dagelijkse cyclus). Alles wat na zonsondergang en voor het ochtendlicht verbruikt wordt, moet uit de batterij komen, en ook wat de zon op een zwak moment tekortkomt. De uurberekening van het vorige blad geeft dat getal per maand: de **nachtbehoefte**. De bruikbare energie van de batterij moet minstens zo groot zijn.

2. Grijze dagen overbruggen (autonomie). Als de zon een paar dagen bijna niets levert, moet de batterij de belangrijke verbruikers een aantal dagen N kunnen voeden:

BATTERIJ VOOR N DAGEN AUTONOMIE

$$E_{\text{bruikbaar}} \geq N \cdot E_{\text{dag}} \qquad E_{\text{nominaal}} = \frac{E_{\text{bruikbaar}}}{\text{DoD}} \qquad C = \frac{E_{\text{nominaal}}}{U_{\text{nom}}}$$

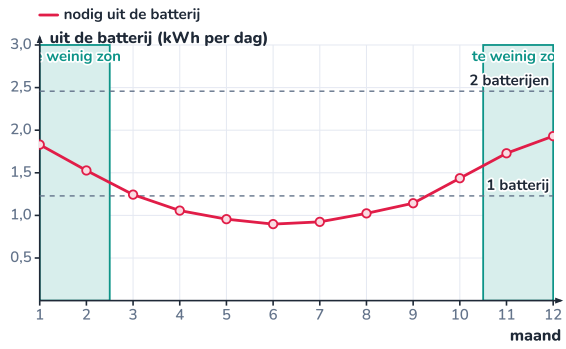
DoD: ontlaaddiepte (LiFePO4 0,8; lood 0,5, hoofdstuk 16); C in Ah, U_{nom} in V. Voor veroudering en kou neem je nog een marge van 10 à 25 %.

Daarbij komen drie praktische regels:

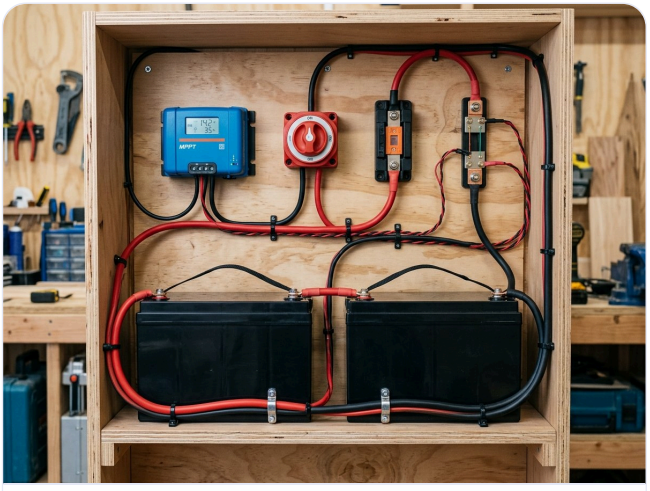
- **Laadrendement.** Om 1 kWh uit een LiFePO4-batterij te halen, moet je er ongeveer 1,05 kWh in steken (95 %). Die verliezen tellen mee bij de panelen, niet bij de batterij.
- **Laadstroom.** Een LiFePO4-batterij laadt je liefst met hoogstens 0,5 C: voor 120 Ah is dat 60 A. Vier panelen van 350 Wp leveren op een zonnige middag tot 1400 W, bij 14 V is dat 100 A.
- **Temperatuur.** Onder 0 °C mag LiFePO4 niet geladen worden (hoofdstuk 16): de batterij staat in de verwarmde module.

Een grotere batterij lost een tekort op

Een batterij verschuift energie in de tijd; ze maakt er geen. Als de panelen over een hele dag minder leveren dan de module verbruikt, loopt elke batterij leeg, hoe groot ook: ze wordt nooit meer volgeladen. Een grotere batterij helpt voor de nacht en voor een paar grijze dagen, niet voor een tekort dat weken duurt.



Nachtbehoefte per maand volgens de uurberekening: van april tot september past ze in één batterij, in maart en oktober niet meer.



Twee batterijen parallel, met kabels van gelijke lengte zodat ze de stroom eerlijk verdelen.

UITGEWERKT VOORBEELD · 20.14

Een batterij voor drie grijze dagen

14/21

(a) Hoe groot moet een LiFePO4-batterij (12,8 V, 80 % ontlaaddiepte) zijn om de **basisverbruikers** van de module (230 Wh per dag: pomp, ventilatie, sensoren) drie dagen zonder zon te voeden? (b) En om de hele module, lampen inbegrepen, drie dagen te voeden? (c) Hoeveel dagen autonomie geeft de batterij van 120 Ah voor de basisverbruikers en voor de hele module?

GEGEVEN

GEVRAAGD

- $E_{\text{basis}} = 230\ \text{Wh}$, $E_{\text{dag}} = 3270\ \text{Wh}$
- $N = 3$, DoD $= 0{,}8$, $U_{\text{nom}} = 12{,}8\ \text{V}$
- LiFePO4-pakket ± 110 Wh/kg (hoofdstuk 16)

- nominale energie, Ah en massa
- dagen autonomie van één batterij van 120 Ah

OPLOSSING

1. (a) Bruikbaar $3 \cdot 230 = 690\ \text{Wh}$; nominaal $690/0{,}8 = 863\ \text{Wh}$; $C = 863/12{,}8 = 67\ \text{Ah}$. De batterij van 120 Ah volstaat ruim.

2. (b) Bruikbaar $3 \cdot 3270 = 9810\ \text{Wh}$; nominaal $9810/0{,}8 = 12\,260\ \text{Wh}$; $C = 12\,260/12{,}8 = 958\ \text{Ah}$: acht batterijen van 120 Ah, samen ongeveer $12\,260/110 = 111\ \text{kg}$.

3. (c) Bruikbaar $0{,}8 \cdot 120 \cdot 12{,}8 = 1229\ \text{Wh}$: basis $1229/230 = 5{,}3$ dagen; hele module $1229/3270 = 0{,}38$ dag, ongeveer 9 uur.

ANTWOORD

(a) 67 Ah: één batterij van 120 Ah is genoeg. (b) 958 Ah, meer dan 100 kg batterij. (c) 5,3 dagen voor de basisverbruikers, maar maar 9 uur voor de hele module.

Klopt dit? Het verschil tussen (a) en (b) is de hele les: autonomie voor alles is onbetaalbaar en zinloos (in december zouden de panelen die acht batterijen nooit vullen). Daarom maakt een ontwerper onderscheid tussen verbruikers die altijd moeten werken en verbruikers die mogen wachten. Bij de klimaatmodule mogen de lampen wachten: sla overleeft een paar dagen weinig licht, maar geen dag zonder water of lucht.

UITLEG · 20.15

Een dag in juni en een dag in december

15/21

De twee grafieken tonen de uurberekening van de rekenaar voor een gemiddelde dag. Het ingekleurde vlak is de zonnestroom van de vier panelen, de rode trap het verbruik: 202 W zolang de lampen branden, 5,5 W 's nachts.

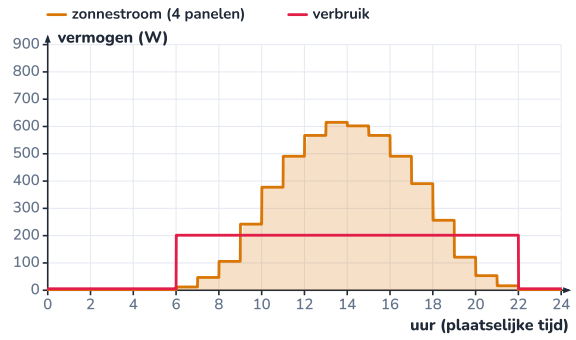
In juni valt het lichtschema bijna samen met de dag. Van ongeveer 9 tot 19 uur levert de zon meer dan de module vraagt: de batterij laadt, en daarna gaat er nog 1,6 kWh verloren omdat ze vol is. Vroeg in de ochtend en 's avonds is de zon te zwak, en 's nachts is er niets: dan komt 0,90 kWh uit de batterij.

In december is de zon er maar van ongeveer 9 tot 17 uur, en nooit sterk genoeg om de lampen en de batterij tegelijk te bedienen. De batterij zou 1,93 kWh moeten leveren, maar er komt over de hele dag maar 1,75 kWh zon binnen tegenover 3,27 kWh verbruik. Na een paar dagen is ze leeg, hoe groot ze ook is.

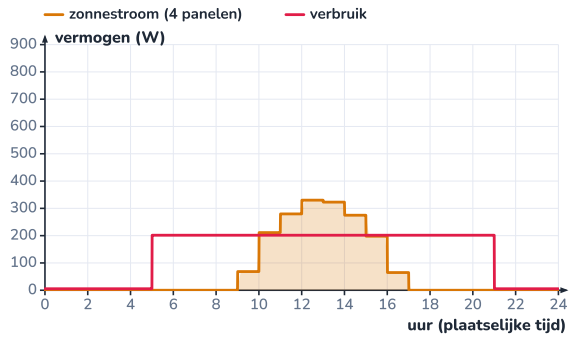
Tussen die twee uitersten liggen de andere maanden (tabel). Van april tot september past de nacht in één batterij, maar met weinig reserve. In maart en oktober is er overdag nog net genoeg zon, maar is de nacht te lang voor 1,2 kWh. Van november tot februari is er gewoon te weinig zon.

maand	zon (kWh/dag)	uit de batterij (kWh)	saldo (kWh)	met 1 batterij (1,23 kWh)
jan	1,97	1,83	−1,32	te weinig zon
feb	2,95	1,53	−0,38	te weinig zon
mrt	4,19	1,24	0,81	batterij te klein
apr	5,17	1,06	1,75	ok
mei	5,01	0,96	1,61	ok
jun	4,96	0,90	1,56	ok
jul	4,83	0,92	1,44	ok

maand	zon (kWh/dag)	uit de batterij (kWh)	saldo (kWh)	met 1 batterij (1,23 kWh)
aug	4,63	1,02	1,25	ok
sep	4,61	1,14	1,21	ok
okt	3,50	1,44	0,15	batterij te klein
nov	2,32	1,73	−0,98	te weinig zon
dec	1,75	1,93	−1,54	te weinig zon



Juni: overdag ruim overschot, 's avonds en 's nachts uit de batterij.



December: de zon komt nooit boven het verbruik van de lampen uit; de dag sluit met een tekort.

UITLEG · 20.16

Van zonlicht tot slalamp: de energiestromen

16/21

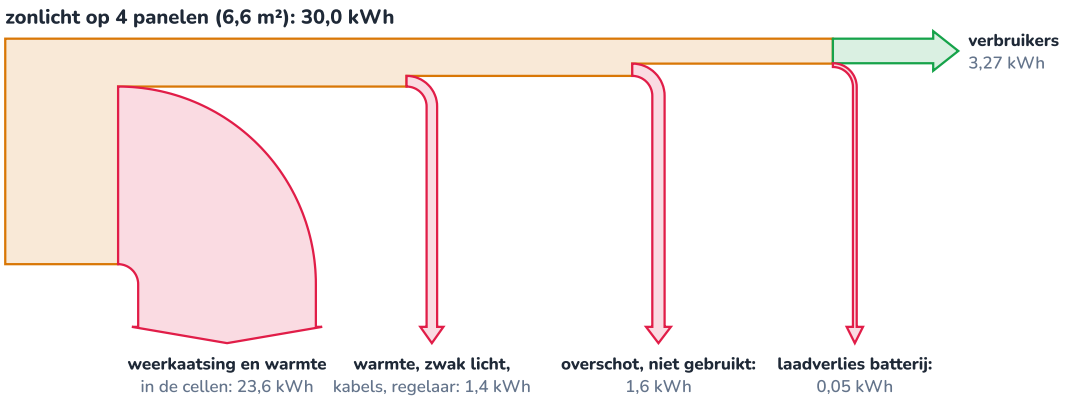
Volg de energie op een gemiddelde junidag in een Sankey-diagram (hoofdstuk F15). Op de vier panelen (6,6 m²) valt die dag 30,0 kWh zonlicht. Bijna vier vijfde daarvan wordt in de cellen weerkaatst of warmte: het rendement van 21 %. Warmte, zwak licht, kabels en de laadregelaar kosten nog eens 1,4 kWh.

Van de 4,96 kWh die de regelaar afgeeft, gebruikt de module 3,27 kWh; een deel daarvan gaat eerst in en uit de batterij, met een klein laadverlies. De rest, 1,6 kWh, kan nergens heen: de batterij is vol en de regelaar 'knijpt' de panelen af door ze van hun MPP weg te sturen.

Dat overschot is de keerzijde van een installatie die voor de winter ontworpen is. Het is geen verspilling van brandstof, wel van mogelijkheden: in de zomer kan het overschot een extra ventilator voeden, het buffervat bijwarmen (hoofdstuk 18) of naar het station gaan.

Het rendement van de hele keten

Van zonlicht tot verbruiker blijft er op een junidag ongeveer 11 % over, en dat komt vooral door het overschot dat niet gebruikt wordt. In december gebruikt de module élke wattuur: dan haalt de keten bijna het rendement van het paneel zelf, ongeveer 17 à 18 %.



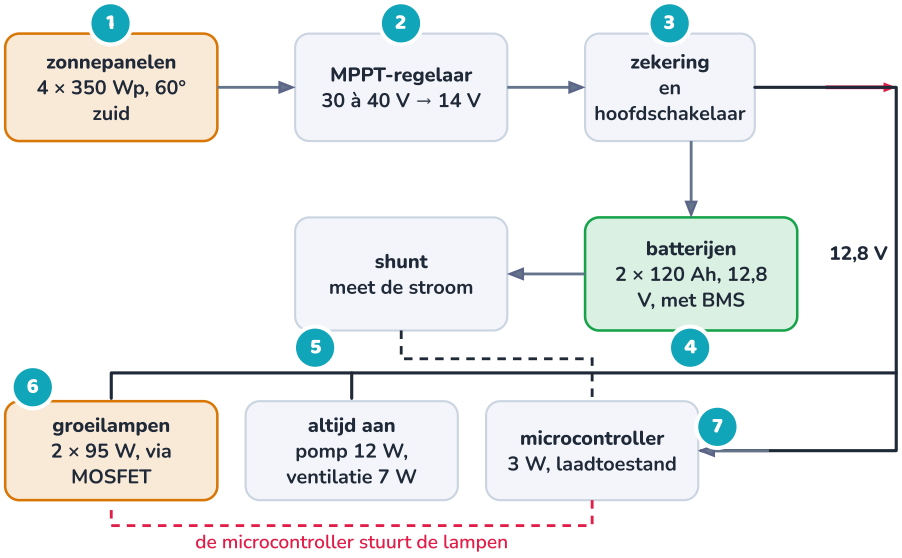
Bandbreedte evenredig met de energie (1 kWh = 6 beeldpunten); de nuttige energie voor de module is 3,27 kWh.

ONTLEED · 20.17

Het energiesysteem van de klimaatmodule

17/21

Een eilandsysteem op 12,8 V: de panelen laden via een regelaar de batterijen, en alle verbruikers hangen aan de batterij. Het blokschema toont de hoofdstroom (volle lijnen) en de meet- en stuurlijnen (stippellijnen).



Van paneel tot lamp: elke verbinding draagt ook een zekering (hier samengevat in één blok).

1 Zonnepanelen
Vier panelen van 350 Wp (1,40 kWp), 60° zuid, in twee rijen van twee op 3,61 m (hoofdstuk 6), op het platte dak van het station met ballast (hoofdstuk 10). Elk paneel geeft ongeveer 35 V bij zijn MPP.

2 MPPT-laadregelaar
Houdt de panelen op hun MPP en zet de spanning om naar de laadspanning van de batterij (± 14 V). Past de laadstroom aan: 100 A kan hij leveren, maar niet meer dan de batterijen verdragen (E24).

3 Zekering en hoofdschakelaar
Een kortsluiting aan een batterij van 12,8 V levert duizenden ampère: de zekering zit zo dicht mogelijk bij de batterij. Met de hoofdschakelaar maak je alles spanningsloos voor onderhoud.

4 Batterijen met BMS
Twee LiFePO4-batterijen van 120 Ah parallel: 2,46 kWh bruikbaar. Het BMS bewaakt elke cel en blokkeert het laden onder 0 °C (hoofdstuk 16).

5 Shunt
Een weerstand van enkele milliohm in de min-leiding: de spanning erover is evenredig met de stroom. De microcontroller telt zo de lading die in en uit gaat (coulombteller, hoofdstuk 16).

6 Groeilampen
De grootste verbruiker (93 %), en de enige die mag wachten. Een MOSFET (hoofdstuk 23) schakelt ze op bevel van de microcontroller; met PWM kan hij ze ook dimmen (hoofdstuk 25).

7 **Microcontroller**
Meet de laadtoestand, beslist of de lampen mogen branden (volgend blad) en logt alles (hoofdstuk 30). Pomp, ventilatie en sensoren staan altijd aan.

UITLEG · 20.18

Energiebeheer: wat mag eerst uitvallen?

Een eilandsysteem heeft een baas nodig die beslist wie energie krijgt als er te weinig is. Dat is het **energiebeheer**, een stukje code op de microcontroller. Het principe: deel de verbruikers in naar belang.

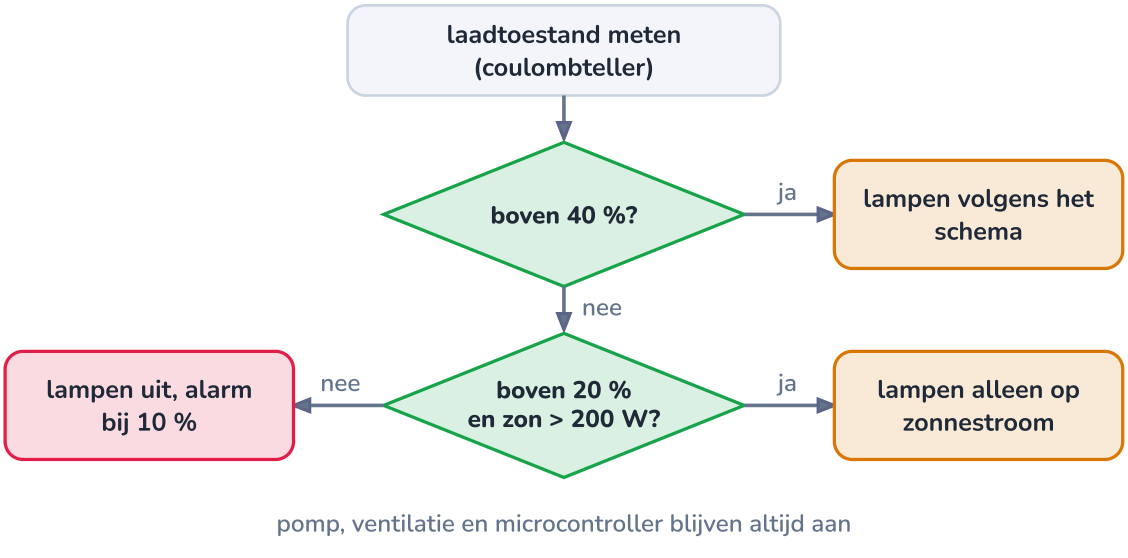
- **Altijd**: de microcontroller, de sensoren, de ventilatie en de pomp (230 Wh per dag). Zonder water of verse lucht is de teelt in een paar uur verloren.
- **Afschakelbaar**: de groeilampen (3040 Wh per dag). Sla die een dag minder licht krijgt, groeit trager, maar overleeft.

De microcontroller kent de laadtoestand dankzij de coulombteller. Boven 40 % branden de lampen volgens het schema. Tussen 20 en 40 % branden ze alleen als de zon ze rechtstreeks kan voeden. Onder 20 % blijven ze uit, en de reserve gaat naar de basisverbruikers; bij 10 % volgt een alarm. Zo'n regeling met drempels en toestanden programmeer je in hoofdstuk 32.

Met twee batterijen (3,07 kWh nominaal) blijft er onder 40 % nog ongeveer 1,1 kWh over tot het BMS afschakelt: genoeg om de basisverbruikers meer dan vier dagen in leven te houden, ook als er geen zon meer is. Zo'n diepe ontlading is een noodgeval, geen gewoonte: ze verkort de levensduur.

Een slimmere klok

Het energiebeheer kan ook vooruitkijken: als de weersverwachting voor morgen grijs is, laat het de lampen vanavond een uur korter branden. Zo'n voorspelling met data en een model maak je in Expert (hoofdstuk E41).



De lampen krijgen wat overblijft; de verbruikers die de planten in leven houden, krijgen altijd voorrang.

UITGEWERKT VOORBEELD · 20.19

Habitat: zonnepaneel en batterij van de klimaatmodule kiezen

19/21

Kies het piekvermogen van de panelen (van 350 Wp, op het zuiden onder 60°) en controleer of de batterij van 1,2 kWh bruikbaar volstaat. Gebruik het energiebudget (3,27 kWh per dag) en de uitvoer van de rekenaar.

GEGEVEN

- $E_{\text{dag}} = 3{,}27\,\text{kWh}$, waarvan $3{,}04\,\text{kWh}$ lampen
- PVGIS: Wh per Wp per dag per maand
- batterij: $1{,}23\,\text{kWh}$ bruikbaar, laadrendement 95 %

GEVRAAGD

- aantal panelen en kWp
- volstaat de batterij? zo niet: een oplossing

OPLOSSING

1. **Kritieke maand.** Voor december is $3{,}27/1{,}25 = 2{,}6\,\text{kWp}$ nodig: acht panelen, bij 12 V meer dan 200 A laadstroom, en in de zomer een enorm overschot. Onhaalbaar voor de module. Kies de teeltperiode van maart tot oktober: in oktober is $3{,}27/2{,}50 = 1{,}31\,\text{kWp}$ nodig.

2. **Panelen.** Vier panelen van 350 Wp: **1,40 kWp**. Volgens de rekenaar sluit de dag van maart tot oktober met een positief saldo (in oktober nog 0,15 kWh), van november tot februari met een tekort van 0,38 tot 1,54 kWh per dag.

3. **Batterij, nacht.** Van april tot september moet de batterij elke nacht 0,90 tot 1,14 kWh leveren: dat past in 1,23 kWh, met hoogstens 0,3 kWh reserve. In maart (1,24 kWh) en oktober (1,44 kWh) past het niet.

4. **Besluit.** 1,2 kWh bruikbaar **volstaat niet**: in maart en oktober is de nacht te lang, en het hele seizoen is er geen reserve voor een grijze dag. **Oplossing:** een tweede identieke batterij parallel (2 × 120 Ah = 240 Ah, 2,46 kWh bruikbaar, ± 28 kg). Dan past elke nacht van maart tot oktober, met 1,0 à 1,6 kWh reserve, en is de laadstroom van 100 A maar 0,42 C.

5. **Winter.** Van november tot februari helpt geen batterij. Het energiebeheer beperkt de lichturen tot wat de zon levert: in december ongeveer $(1{,}75 \cdot 0{,}95 - 0{,}23)/0{,}19 \approx 7{,}5$ uur, in februari ongeveer 13,5 uur. Wie de teelt in de winter op 16 uur licht wil houden, haalt het tekort van het net van het station (de windturbine van hoofdstuk E19, het energiebeheer van hoofdstuk E49).

ANTWOORD

Habitat kiest vier panelen van 350 Wp (1,40 kWp), 60° zuid, met een MPPT-regelaar van 100 A, en twee LiFePO4-batterijen van 120 Ah (2,46 kWh bruikbaar) met energiebeheer. Van maart tot oktober draait de module op een gemiddelde dag volledig op eigen zonnestroom; van november tot februari branden de lampen korter, of vult het station het tekort aan.

Klopt dit? Controle met de jaarbalans: de panelen leveren 1397 kWh per jaar, de module verbruikt 1193 kWh; zonder winterlicht (lampen 4 maanden gehalveerd) nog ongeveer 1010 kWh. Het ontwerp klopt in grote lijnen, en zijn zwakke plek (de winter) is bekend en benoemd. Dat is het verschil tussen dimensioneren en hopen.

NASLAG · 20.20

Overzicht: zonne-energie dimensioneren

20/21

invalshoek (direct licht)

$$G_{\text{paneel}} = G_n \cdot \cos\theta$$

symbool	betekenis	eenheid
G	instraling	W/m²
θ	hoek tussen stralen en normaal	°

rendement en wattpiek

$$\eta = \frac{P}{G \cdot A} \qquad P_{\text{piek}} = \eta_{\text{STC}} \cdot 1000 \cdot A$$

symbool	betekenis	eenheid
P_{piek}	piekvermogen	Wp
A	oppervlakte	m²

vermogen en celtemperatuur

$$P \approx P_{\text{piek}} \frac{G}{1000}\big(1 + \gamma(T_{\text{cel}} - 25)\big) \qquad T_{\text{cel}} \approx T$$

symbool	betekenis	eenheid
γ	temperatuurcoëfficiënt (± −0,0035)	per K

dagopbrengst

$$E_{\text{dag}} = P_{\text{piek}} \cdot \text{volle zonuren} \cdot PR$$

batterij voor N dagen autonomie

$$E_{\text{nominaal}} = \frac{N \cdot E_{\text{dag}}}{\text{DoD}} \qquad C = \frac{E_{\text{nominaal}}}{U_{\text{nom}}}$$

symbool	betekenis	eenheid
N	dagen autonomie	–
DoD	ontlaaddiepte	–
C	capaciteit	Ah

werkwijze: een eilandsysteem dimensioneren

1. Energiebudget: vermogen × tijd per verbruiker; splits in 'altijd' en 'afschakelbaar'.
2. Kritieke maand kiezen; piekvermogen = verbruik per dag / (Wh per Wp per dag van die maand).
3. Nachtbehoefte per maand (uurberekening) ≤ bruikbare energie van de batterij.
4. Autonomie: N dagen voor de verbruikers die altijd moeten werken.
5. Laadstroom (≤ 0,5 C), temperatuur, zekeringen, en een energiebeheer voor wat niet past.

Habitat: rekenwaarden uit dit hoofdstuk

energiebudget klimaatmodule	3,27 kWh per dag: groeilampen 2 × 95 W, 16 h (3,04 kWh); basis 0,23 kWh (pomp 48 Wh, sensoren 3 W, ventilatie 96 Wh, BMS 0,5 W, CO_2-klep); zonder verwarming
panelen	4 × 350 Wp (1,65 × 1,00 m, 21 %) = 1,40 kWp, 60° zuid, twee rijen op het platte dak; ± 1400 kWh per jaar
laadregelaar	MPPT, 12 V, tot 100 A
batterij	2 × LiFePO4 12,8 V 120 Ah parallel: 3,07 kWh nominaal, 2,46 kWh bruikbaar, ± 28 kg
energiebeheer	lampen boven 40 % volgens schema, 20-40 % alleen op zon, onder 20 % uit; alarm bij 10 %
winter (nov.-feb.)	tekort 0,4 à 1,5 kWh per dag: minder lichturen (dec. ± 7,5 h) of stroom van het station

HET GEHEEL · 20.21

Waar zonne-energie dimensioneren naartoe leidt

21/21

Dimensioneren verbindt **Energie** (hoeveel), **Elektriciteit** (spanning, stroom, laadstroom, zekeringen) en **Wiskundig modelleren** (de cosinus, gemiddelden, een uurmodel). Met de rekenaar zit je in **Programmeren**, met het energiebeheer in **Regelen**. En het is een schoolvoorbeeld van **Ontwerpen**: er is geen perfecte oplossing, alleen een verantwoorde keuze.

F17 Energie uit zon, wind en water
Wattpiek, volle zonuren, PVGIS en de energie van de meetkas.

F21 Elektrisch vermogen en energie
Vermogen maal tijd: de basis van elk energiebudget.

I6 Goniometrie en vectoren
De zonnehoogte, de invalshoek en de keuze van 60°; de rijafstand van 3,61 m.

I9 Arbeid, energie en vermogen
De pomp (12 W, plan B) en de dakmotor in het budget.

I16 Redox en batterijen
LiFePO4, ontlaaddiepte, rendement, BMS en de coulombteller.

I18 Warmte berekenen
Waarom de verwarming niet op de batterij kan: een warmtebuffer in de plaats.

I19 Warmtetransport en isolatie
Het warmteverlies van de module, dat buiten het elektrische budget valt.

I23 Halfgeleiders: diode, led en transistor
De MOSFET die de lampen schakelt, en de zonnecel als pn-overgang.

I25 Van analoog naar digitaal
De lampen dimmen met PWM in plaats van ze uit te schakelen.

I32 Toestandsmachines

Het energiebeheer als toestandsmachine met drempels.

I36 Hydroponie en voedselproductie

Opbrengst per kWh: hoeveel sla levert een kilowattuur licht?

I40 Integratieproject: de klimaatmodule

Het integratieproject: het energiesysteem bouwen en meten.

E3 Integralen: ophopen

De dagopbrengst als oppervlakte onder de vermogenskromme.

E24 Vermogenselektronica en elektrische machines

Hoe een MPPT-regelaar het maximum uit een paneel haalt.

E27 Elektrochemie: batterijen, brandstofcellen en elektrolyse

Waterstof als seizoensopslag: de zomer bewaren voor de winter?

E47 Duurzaamheid en levenscyclusanalyse

Hoe lang een paneel moet werken om zijn eigen productie terug te verdienen.

Habitat

Voor Habitat leggen we vast: het energiebudget van de klimaatmodule is 3,27 kWh per dag (93 % groeilampen). Vier panelen van 350 Wp (1,40 kWp), 60° zuid, laden via een MPPT-regelaar twee LiFePO4-batterijen van 120 Ah (2,46 kWh bruikbaar); de batterij van 1,2 kWh uit hoofdstuk 16 volstond niet. Een energiebeheer behandelt de lampen als afschakelbare last. Van maart tot oktober draait de module op eigen zon; van november tot februari branden de lampen korter of helpt het station bij. In opdracht 20.13 weeg je zelf de alternatieven af.

SAMENVATTING

In het kort

- Zonnestraling meet je in W/m² (heldere middag ± 1000, buiten de dampkring 1361); de energie per dag in kWh/m² is het aantal volle zonuren.
- Een paneel vangt direct licht met een factor $\cos\theta$; een steil paneel (60°) vangt de lage winterzon bijna volledig.
- Het rendement is $\eta = P/(G \cdot A)$, het piekvermogen het vermogen onder STC (1000 W/m², 25 °C). Een warm paneel levert minder (± −0,35 % per K).
- Een paneel levert het meeste vermogen bij één spanning, het MPP; een MPPT-laadregelaar houdt het daar.
- Met PVGIS reken je de opbrengst per maand: in Brussel levert 1 Wp ongeveer 1 kWh per jaar, maar in december drie keer minder per dag dan in april.
- Een energiebudget somt vermogen × tijd per verbruiker. Zoek de grootste post: bij de klimaatmodule de lampen (93 %).
- Een batterij dimensioneer je voor de nacht (uurberekening) en voor N dagen autonomie van wat altijd moet werken: E_nominaal = N · E_dag / DoD.
- Een batterij verschuift energie, ze maakt er geen: een seizoenstekort los je op met minder verbruik of een andere bron, niet met een grotere batterij.
- Habitat: 4 × 350 Wp en 2 × 120 Ah LiFePO4, met de lampen als afschakelbare last; in de winter minder licht of hulp van het station.

BEGRIPPEN

Wat je nu kent

instraling

Het vermogen van het zonlicht per vierkante meter, in W/m²; op een heldere middag ongeveer 1000 W/m².

volle zonuren

Het aantal uren zon van 1000 W/m² dat evenveel energie zou geven als het echte zonlicht van een dag: de instraling in kWh/m² per dag.

standaardtestcondities

Afgesproken omstandigheden om panelen te vergelijken: 1000 W/m² loodrecht, cel op 25 °C, spectrum AM1,5 (STC).

wattpiek

Het vermogen van een paneel bij standaardtestcondities, in Wp; zegt niets over de opbrengst op een gewone dag.

rendement van een paneel

Het deel van het invallende zonlicht dat elektrische energie wordt: $\eta = P/(G \cdot A)$.

MPP

Maximum power point: de spanning waarbij een paneel het grootste vermogen levert. Een MPPT-laadregelaar zoekt dat punt voortdurend.

energiebudget

Een lijst van alle verbruikers met hun vermogen, gebruiksduur en energie per dag; de basis om panelen en batterij te kiezen.

autonomie

Het aantal dagen dat een batterij de verbruikers kan voeden zonder dat er energie bijkomt.

afschakelbare last

Een verbruiker die het energiebeheer mag uitschakelen als er te weinig energie is, zodat de belangrijke verbruikers blijven werken.

HANDBOEK · HOOFDSTUK 21

De wetten van Kirchhoff

Twee wetten die elk netwerk doorrekenen, van een breadboard tot de voeding van een hele module

In Fundamental rekende je met één bron en weerstanden die netjes in serie of parallel stonden. Echte schakelingen zijn gemengd, hebben meer dan één bron, en de draden zelf hebben weerstand. Met twee wetten, die van Gustav Kirchhoff, reken je ze allemaal door: wat een knooppunt binnenkomt, gaat eruit, en wie een gesloten rondje maakt, komt op dezelfde spanning terug. Je leert ze toepassen met tekenafspraken, controleert je berekeningen met metingen, lost een netwerk op als stelsel (met de hand en met numpy) en rekent de hele sensorvoeding van de klimaatmodule na.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 8 lestijden

Elektriciteit en elektronica · Wiskundig modelleren

Wie een verdeelkast begrijpt, ziet knooppunten en mazen: waar stromen samenkomen en waar spanningen verdeeld worden.

Na dit hoofdstuk

- Je kan in een schakeling de knooppunten, takken en mazen aanduiden en het aantal onafhankelijke vergelijkingen bepalen.
- Je kan de knooppuntwet en de maaswet opschrijven met een gekozen stroomzin en omloopzin, en een negatieve uitkomst juist interpreteren.
- Je kan de vervangingsweerstand berekenen van om het even welke weerstanden in serie en parallel.
- Je kan een gemengde schakeling volledig oplossen: alle stromen, spanningen en vermogens.
- Je kan het vermogen in elke weerstand berekenen en een vermogensbalans opstellen als controle.
- Je kan een netwerk met meer dan één bron oplossen als stelsel van vergelijkingen, met de hand en met numpy.
- Je kan metingen gebruiken om een berekening te controleren, en omgekeerd een foute meting opsporen.
- Je kan de voeding van een groep sensoren narekenen, met de spanningsval in de draden en de gevolgen van een gedeelde massadraad.

VERHAAL · 21.1

Een student en de draden van de telegraaf

In 1845 was Gustav Kirchhoff 21 jaar en student in Königsberg, in het toenmalige Pruisen. Elektrische stroom was nog jong: Ohm had zijn wet pas achttien jaar eerder gepubliceerd, en in dezelfde jaren werden de eerste telegraaflijnen gespannen. Ingenieurs wilden weten hoe de stroom zich verdeelt over een netwerk van draden met meerdere batterijen en aftakkingen. Met de wet van Ohm alleen lukte dat niet.

Kirchhoff schreef twee regels op die voor **elk** netwerk gelden. De eerste: in een punt waar draden samenkomen, kan geen lading verdwijnen of ontstaan. De tweede: wie langs een gesloten weg rondgaat, komt op dezelfde 'hoogte' terug. Twee jaar later, in 1847, toonde hij hoe je voor een netwerk precies zoveel onafhankelijke vergelijkingen vindt als er onbekende stromen zijn. Dat werk is een van de wortels van de grafentheorie, de wiskunde van netwerken.

Later werd Kirchhoff in Heidelberg beroemd om iets heel anders: samen met Robert Bunsen ontdekte hij dat elk element een eigen lijnenspectrum heeft (hoofdstuk 12). Maar elke elektronicus kent hem van de twee wetten die in dit hoofdstuk centraal staan. Simulatieprogramma's die schakelingen doorrekenen, doen in de kern niets anders dan Kirchhoffs vergelijkingen opstellen en oplossen, voor duizenden onderdelen tegelijk.

Waarom dit hoofdstuk ertoe doet

De klimaatmodule heeft één 12,8 V-rail met lampen, een pomp en ventilatoren, een 5 V-rail met elf verbruikers, en sensoren aan meters kabel. Of alles de juiste spanning krijgt, of de zekeringen kloppen, en of een meting niet verstoord wordt door de stroom van de pomp: dat reken je na met Kirchhoff.

Batterijen, draden en aftakkingen: het soort netwerk waarvoor Kirchhoff zijn regels opschreef.

UITLEG · 21.2

Wat je al weet, en wat er nu bijkomt

In Fundamental leerde je vier regels voor serie en parallel, en de wet van Ohm. Ze kloppen, maar ze werken alleen als een schakeling netjes uit serie- en parallelstukken bestaat en er maar één bron is. De wetten van Kirchhoff zijn de **algemene** vorm van dezelfde regels.

wat je al weet (Fundamental)	wat er nu bijkomt
In serie loopt overal dezelfde stroom en tellen de spanningen op (F20).	De **maaswet**: in elke gesloten lus is de som van alle spanningen nul, ook met meerdere bronnen en in ingewikkelde netwerken.
Parallel tellen de stromen van de takken op: de knooppuntregel (F20).	De **knooppuntwet** met een tekenafspraak, ook als je vooraf niet weet welke kant de stroom op gaat.
$R_\text{v} = R_1 + R_2$ in serie; twee gelijke parallel geven de helft (F20).	$\frac{1}{R_\text{v}} = \frac{1}{R_1} + \frac{1}{R_2} + \ldots$ voor om het even welke weerstanden parallel.
$U = R \cdot I$ (F19) en $P = U \cdot I$ (F21).	Het vermogen in **elke** weerstand, en de vermogensbalans van het hele netwerk.
Stroom en spanning meten met een multimeter (F19).	Metingen gebruiken om een berekening te controleren, en een berekening om een meetfout te vinden.

Eén nieuw woord maakt alles eenvoudiger: de **potentiaal**. Kies één punt van de schakeling als referentie, meestal de minpool van de bron, en noem dat de **massa** (0 V; in een schema drie streepjes onder elkaar).

De potentiaal van een ander punt is de spanning tussen dat punt en de massa. De spanning over een onderdeel is dan gewoon het verschil tussen de potentialen aan zijn twee uiteinden: $U_{AB} = V_A - V_B$. Een microcontroller meet zo: een analoge ingang meet de potentiaal van één punt ten opzichte van zijn eigen GND.

Het beeld van de trap

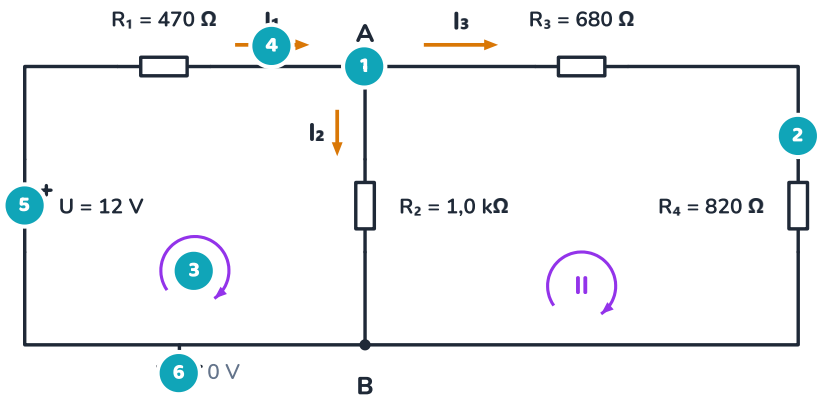
In F20 zag je spanning als hoogte: de bron tilt de lading op, en in elke verbruiker valt ze een stukje naar beneden. De potentiaal is precies die hoogte, gemeten vanaf de vloer (de massa). Een draad zonder weerstand is een vlakke overloop: overal dezelfde potentiaal.

ONTLEED · 21.3

Een netwerk ontleden: knooppunt, tak en maas

3/19

Voor je rekent, lees je het schema. Dit netwerk komt verder in het hoofdstuk terug als uitgewerkt voorbeeld. Zes begrippen zijn genoeg om elk netwerk te beschrijven.



R_1 staat in serie met twee parallelle takken: R_2, en R_3 + R_4. De stroompijlen en de omloopzinnen zijn gekozen.

1 Knooppunt
Punt waar drie of meer draden samenkomen, getekend als een bolletje: hier A en B. De hele onderste draad is één knooppunt, want er zit niets tussen: overal dezelfde potentiaal.

2 Tak
Het stuk tussen twee knooppunten, zonder vertakking. R_3 en R_4 vormen samen één tak: door beide loopt dezelfde stroom I_3. Dit netwerk heeft drie takken: bron met R_1, R_2, en R_3 met R_4.

3 Maas
Een gesloten lus. De vensters I en II zijn onafhankelijke mazen; de buitenste lus (bron, R_1, R_3, R_4) is ook een maas, maar haar vergelijking is de som van die van I en II.

4 Gekozen stroomzin
Per tak kies je een pijl. Klopt de keuze niet, dan wordt de stroom gewoon negatief. In deze kring is de zin duidelijk: van de pluspool weg.

5 Bron
De lange streep is de pluspool. Van de min- naar de pluspool stijgt de potentiaal met de bronspanning U.

6 Massa
Het referentiepunt: 0 V. Alle potentialen tel je vanaf hier. Waar je de massa legt, verandert niets aan de stromen en spanningen, alleen aan de getallen van de potentialen.

UITLEG · 21.4

De knooppuntwet: lading hoopt zich niet op

4/19

Lading kan in een draad niet verdwijnen en niet ontstaan, en in een knooppunt kan ze zich ook niet opstapelen: er is geen plaats voor. Wat er per seconde binnenkomt, moet er dus per seconde uit. Dat is

de eerste wet van Kirchhoff.

KNOOPPUNTWET (EERSTE WET VAN KIRCHHOFF)

$$\sum I_{\text{in}} = \sum I_{\text{uit}} \quad \Longleftrightarrow \quad \sum_k I_k = 0$$

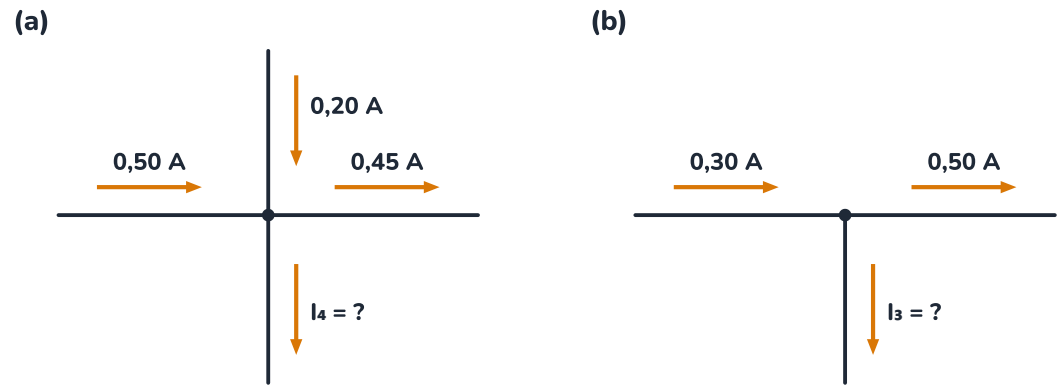
In de tweede vorm tel je een stroom die het knooppunt binnenkomt positief, een stroom die vertrekt negatief.

In knooppunt (a) komen 0,50 A en 0,20 A binnen en vertrekt 0,45 A. Dan is $0{,}50 + 0{,}20 = 0{,}45 + I_4$, dus $I_4 = 0{,}25\ \text{A}$, in de richting van de pijl.

Je weet niet altijd welke kant de stroom op gaat. Dan kies je een richting, tekent een pijl, en rekent. In knooppunt (b) is de pijl van I_3 naar buiten gekozen: $0{,}30 = 0{,}50 + I_3$, dus $I_3 = -0{,}20\ \text{A}$. Het minteken is geen fout: het zegt dat er 0,20 A loopt, maar **tegen** de gekozen pijl in. In (b) komt die stroom het knooppunt dus binnen.

De stroom kiest de weg met de kleinste weerstand

Stroom verdeelt zich over **alle** takken tussen twee knooppunten. De tak met de kleinste weerstand krijgt het grootste deel, maar ook een tak van 1 MΩ krijgt zijn (kleine) stroom. Alleen als één tak nul ohm heeft (een kortsluiting), krijgt die praktisch alles, en dan staat er geen spanning meer over de andere.



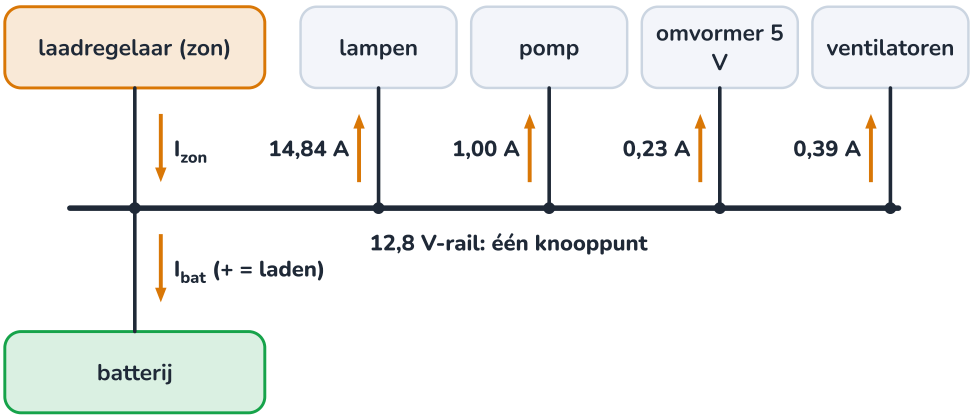
(a) Er gaat evenveel uit als erin komt: I_4 = 0,25 A. (b) Met de gekozen pijl wordt I_3 = –0,20 A: de stroom loopt het knooppunt in.

UITGEWERKT VOORBEELD · 21.5

Laadt of ontlaadt de batterij?

5/19

Alle 12,8 V-verbruikers van de klimaatmodule hangen aan één rail: de groeilampen (190 W), de pomp (1,0 A), de omvormer voor de sensoren (3,0 W) en de ventilatoren (5 W). De laadregelaar van de zonnepanelen levert I_{zon}. De batterij hangt aan dezelfde rail; we kiezen haar stroom positief als hij de batterij **in** gaat (laden). Bereken I_{bat} op een zonnige middag ($I_{\text{zon}} = 25{,}0\ \text{A}$) en op een grijze ochtend ($I_{\text{zon}} = 6{,}0\ \text{A}$), als alles aanstaat.



Eén knooppunt, zes stromen. De pijl van de batterij is gekozen; het teken van de uitkomst zegt of ze laadt of ontlaadt.

GEGEVEN

- $U = 12{,}8\ \mathrm{V}$; lampen 190 W, pomp 1,0 A, omvormer 3,0 W, ventilatoren 5 W
- $I_{\mathrm{zon}} = 25{,}0\ \mathrm{A}$ of $6{,}0\ \mathrm{A}$
- pijl van I_{bat}: de batterij in

GEVRAAGD

- I_{bat} in beide gevallen, en wat het teken betekent

OPLOSSING

1. Stroom per verbruiker met $I = P/U$: lampen $190/12{,}8 = 14{,}84\ \mathrm{A}$; omvormer $3{,}0/12{,}8 = 0{,}23\ \mathrm{A}$; ventilatoren $5/12{,}8 = 0{,}39\ \mathrm{A}$; pomp $1{,}00\ \mathrm{A}$.
2. De rail is één knooppunt. Erin: I_{zon}. Eruit: de vier verbruikers en I_{bat} (met de gekozen pijl). Knooppuntwet: $I_{\mathrm{zon}} = 16{,}47 + I_{\mathrm{bat}}$, met $16{,}47\ \mathrm{A}$ de som van de verbruikers.
3. Zonnig: $I_{\mathrm{bat}} = 25{,}0 - 16{,}47 = +8{,}53\ \mathrm{A}$.
4. Grijs: $I_{\mathrm{bat}} = 6{,}0 - 16{,}47 = -10{,}47\ \mathrm{A}$.

ANTWOORD

Op de zonnige middag laadt de batterij met 8,53 A. Op de grijze ochtend is de uitkomst negatief: de stroom loopt tegen de pijl in, de batterij ontlaadt met 10,47 A en levert samen met de zon de stroom voor de verbruikers.

Klopt dit? Het teken doet precies wat de coulombteller van hoofdstuk 16 doet: stroom erin positief, eruit negatief. Een laadregelaar die meer levert dan de verbruikers vragen, laadt de batterij; levert hij minder, dan vult de batterij het verschil aan. De lampen zijn 90 % van de stroom: daarom zijn zij de afschakelbare last van het energiebeheer (hoofdstuk 20).

UITLEG · 21.6

De maaswet: rondgaan en terugkomen op dezelfde hoogte

6/19

Neem in gedachten een lading mee langs een gesloten lus. In de bron wordt ze 'opgetild', in elke weerstand zakt ze wat. Als ze terug is bij haar vertrekpunt, is ze op dezelfde potentiaal: anders zou ze bij elke ronde energie winnen of verliezen uit het niets. Dat is de tweede wet van Kirchhoff.

MAASWET (TWEEDE WET VAN KIRCHHOFF)

$$\sum_{\text{maas}} \Delta V = 0$$

Langs een gesloten lus is de som van alle potentiaalstijgingen en -dalingen nul.

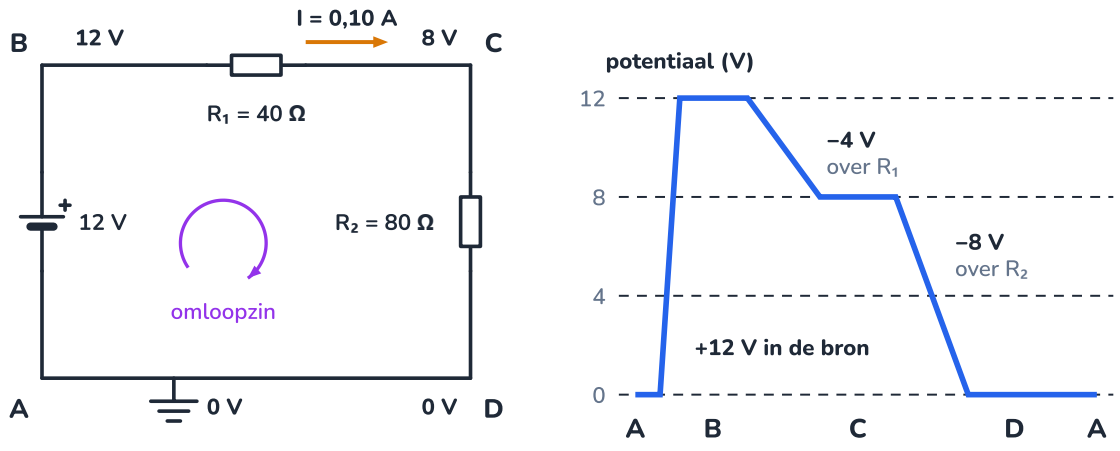
De tekenafspraken. Kies een omloopzin (hier met de klok mee) en loop de maas rond:

- door een **bron van − naar +**: de potentiaal stijgt, schrijf $+U$; van + naar −: $-U$;
- door een **weerstand in de zin van zijn stroompijl**: de potentiaal daalt, schrijf $-R \cdot I$; tegen de stroompijl in: $+R \cdot I$.

In de kring van de figuur, vanaf A met de klok mee: $+12 - 40\,I - 80\,I = 0$, dus $I = 12/120 = 0{,}10\ \text{A}$. De potentiaal gaat van 0 V naar 12 V (B), zakt 4,0 V in R_1 (C: 8,0 V) en 8,0 V in R_2 (D: 0 V). Dat is de regel 'in serie tellen de spanningen op' uit F20, nu als bijzonder geval van een algemene wet.

⚠ **Omloopzin en stroomzin zijn twee verschillende keuzes**

De stroompijl zegt welke richting je *positief* noemt voor de stroom in een tak. De omloopzin zegt in welke volgorde je de maas afloopt. Ze hoeven niet samen te vallen. Loop je tegen een stroompijl in door een weerstand, dan schrijf je $+R \cdot I$. Wie zich aan de afspraken houdt, krijgt altijd hetzelfde resultaat, welke zin hij ook kiest.



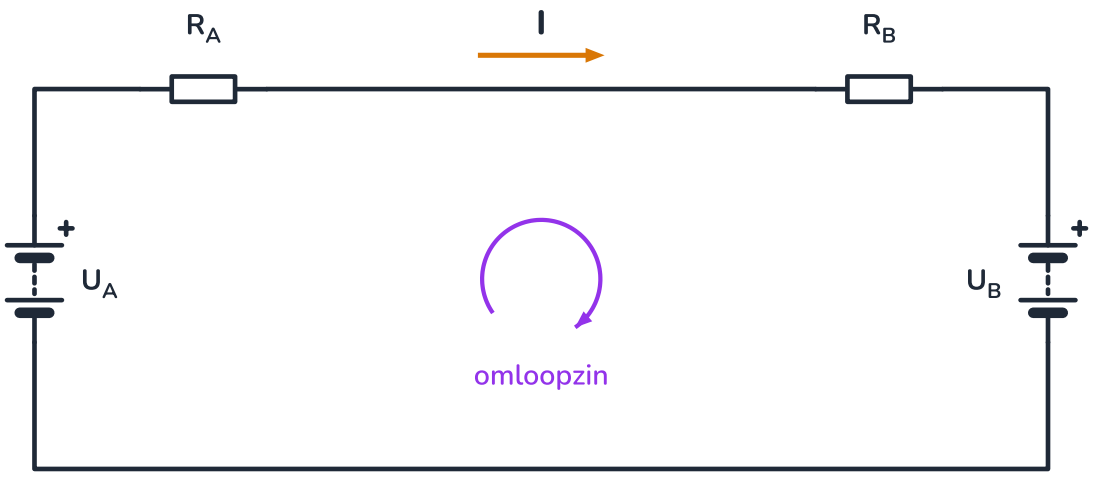
Links de maas met de omloopzin; rechts de potentiaal langs de kring: +12 V in de bron, −4 V en −8 V in de weerstanden.

UITGEWERKT VOORBEELD · 21.7

Twee batterijen parallel: de vereffeningsstroom

7/19

De klimaatmodule krijgt een tweede LiFePO4-batterij parallel aan de eerste (hoofdstuk 20). Batterij A is vol en heeft een bronspanning van 13,30 V, batterij B minder vol: 13,10 V. Elke batterij heeft, met haar kabel en BMS, een inwendige weerstand (orde van grootte): $R_A = 20\,\text{m}\Omega$ en $R_B = 25\,\text{m}\Omega$. Er is nog niets aangesloten. Welke stroom loopt er zodra je ze verbindt? Welke spanning meet je dan over de klemmen? En als B bijna leeg was (12,00 V)?



Eén maas met twee bronnen die tegen elkaar in staan: het verschil van 0,20 V drijft de stroom.

GEGEVEN

- $U_A = 13{,}30\,\text{V}$, $R_A = 0{,}020\,\Omega$
- $U_B = 13{,}10\,\text{V}$, $R_B = 0{,}025\,\Omega$

GEVRAAGD

- de stroom I
- de klemspanning
- I als $U_B = 12{,}00\,\text{V}$

OPLOSSING

1. Kies de stroompijl van A naar B over de bovenste draad, en loop de maas met de klok mee, vanaf de min van A.
2. Door A van − naar +: $+13{,}30$. Door R_A met de pijl mee: $-0{,}020\,I$. Door R_B met de pijl mee: $-0{,}025\,I$. Door B van + naar −: $-13{,}10$.
3. Maaswet: $13{,}30 - 0{,}020\,I - 0{,}025\,I - 13{,}10 = 0$, dus $I = \dfrac{0{,}20}{0{,}045} = 4{,}44\,\text{A}$.
4. Klemspanning (tussen de bovenste en de onderste draad): $U_A - R_A I = 13{,}30 - 0{,}020 \cdot 4{,}44 = 13{,}21\,\text{V}$. Controle langs B: $13{,}10 + 0{,}025 \cdot 4{,}44 = 13{,}21\,\text{V}$.
5. Met B op 12,00 V: $I = \dfrac{1{,}30}{0{,}045} = 29\,\text{A}$.

ANTWOORD

Er loopt 4,44 A van A naar B: de volle batterij laadt de andere bij, en beide klemmen staan op 13,21 V. Is B bijna leeg, dan wordt dat 29 A, meer dan de laadregelaar ooit levert.

Klopt dit? De stroom is positief: de gekozen pijl klopte. De stroom neemt vanzelf af, want de volle batterij zakt en de andere stijgt, tot hun spanningen gelijk zijn. Daarom laad je twee batterijen eerst apart vol voor je ze parallel zet, met gelijke kabels naar één verzamelpunt. Kortsluiting van één batterij zou $13{,}30/0{,}020 \approx 665$ A geven: een zekering vlak bij elke pluspool is geen luxe.

UITLEG · 21.8

Een echte bron: bronspanning en klemspanning

8/19

Het vorige voorbeeld gebruikte een model dat voor elke echte bron werkt: een **ideale bron** met spanning U_0 (de *bronspanning*, wat je meet zonder stroom) in serie met een kleine **inwendige weerstand** R_i.

Zodra de bron stroom levert, valt er een deel van de spanning over R_i, en blijft er aan de klemmen minder over. De maaswet geeft meteen de formule:

KLEMSPANNING VAN EEN BRON

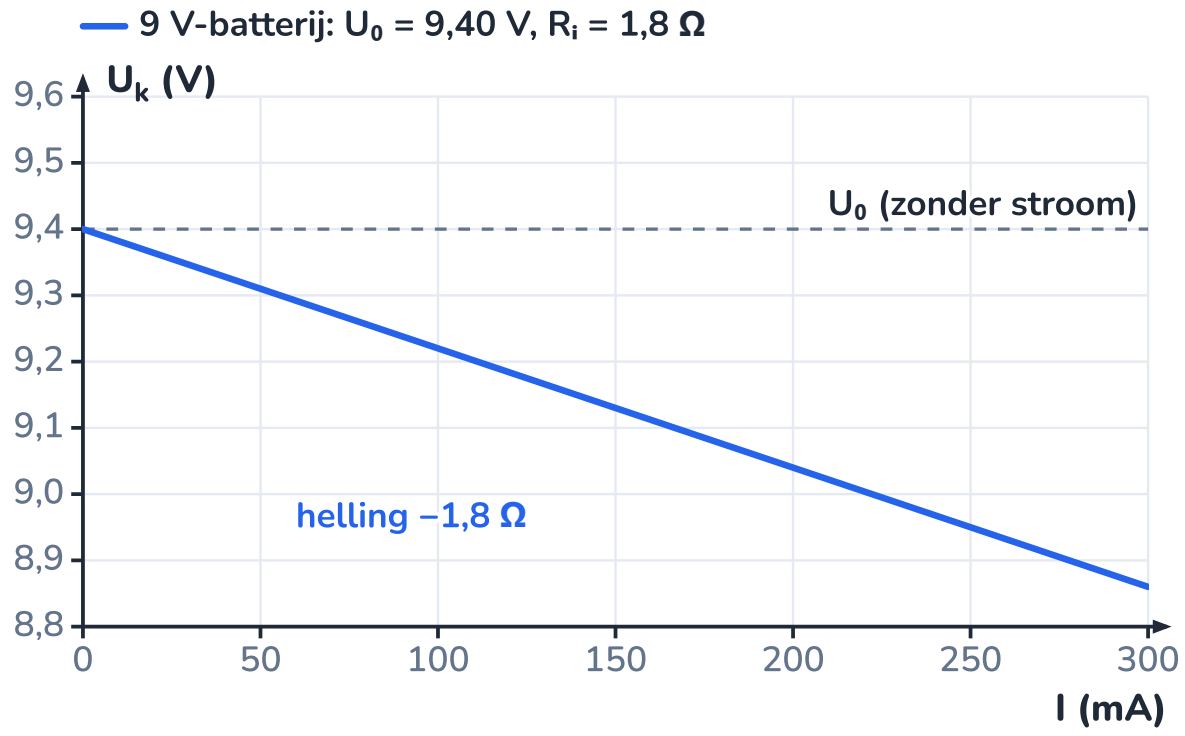
$$U_\mathrm{k} = U_0 - R_\mathrm{i} \cdot I$$

U_k: spanning tussen de klemmen (V); U_0: bronspanning (V); R_i: inwendige weerstand (Ω); I: geleverde stroom (A).

Zet je de klemspanning uit tegen de stroom, dan krijg je een dalende rechte: het snijpunt met de verticale as is U_0, de helling is $-R_\mathrm{i}$ (hoofdstuk 3). Een blokbatterij van 9 V heeft enkele ohm, een AA-cel enkele tienden van een ohm, een grote LiFePO4-batterij enkele milliohm. Daarom zakt een 9 V-batterij merkbaar als je er een motortje aan hangt, en kan een autobatterij honderden ampère leveren om een startmotor te laten draaien.

Kleine inwendige weerstand, grote kortsluitstroom

Bij kortsluiting is $U_\mathrm{k} = 0$, dus $I = U_0 / R_\mathrm{i}$. Voor een LiFePO4-batterij van 13,3 V met 20 mΩ is dat 665 A: genoeg om een schroevendraaier te laten smelten. Elke batterij krijgt daarom een zekering zo dicht mogelijk bij de pluspool (F18), en je werkt met geïsoleerd gereedschap.



Hoe meer stroom de bron levert, hoe meer spanning er in de bron zelf verloren gaat.

UITLEG · 21.9

Vervangingsweerstand: serie en parallel in het algemeen

9/19

Een groep weerstanden gedraagt zich naar buiten toe als één weerstand, de **vervangingsweerstand** R_v: bij dezelfde spanning trekt ze dezelfde stroom. De wetten van Kirchhoff geven de formules.

Serie. Door alle weerstanden loopt dezelfde stroom I. De maaswet zegt dat de spanningen optellen: $U = R_1 I + R_2 I + \cdots + R_n I = (R_1 + R_2 + \cdots + R_n)\, I$.

Parallel. Over alle weerstanden staat dezelfde spanning U. De knooppuntwet zegt dat de stromen optellen: $I = \frac{U}{R_1} + \frac{U}{R_2} + \cdots + \frac{U}{R_n}$. Deel door U:

VERVANGINGSWEERSTAND IN SERIE EN PARALLEL

$$R_\text{v} = R_1 + R_2 + \cdots + R_n \qquad \frac{1}{R_\text{v}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}$$

Voor twee weerstanden parallel: $R_\text{v} = \dfrac{R_1 R_2}{R_1 + R_2}$ ('product door som'). Voor n gelijke weerstanden R: $R_\text{v} = R/n$.

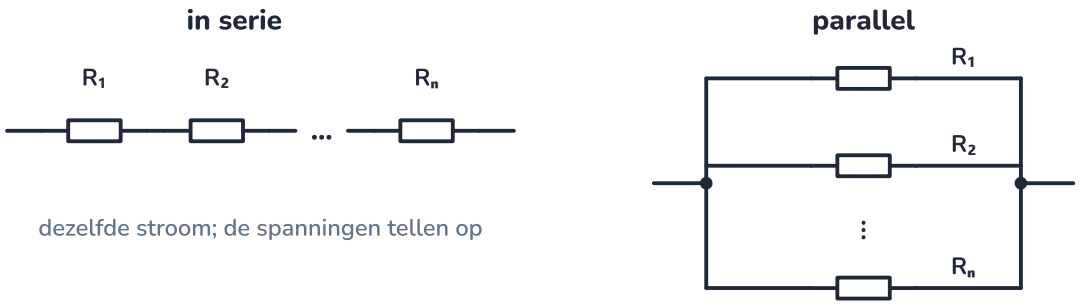
Parallel is R_v altijd **kleiner dan de kleinste** weerstand: elke extra tak is een extra weg voor de stroom. De grafiek toont het voor 100 Ω parallel met een tweede weerstand: hoe groot R_2 ook wordt, R_v komt nooit boven 100 Ω.

De stroomdeler. Over twee parallelle weerstanden staat dezelfde spanning, dus $R_1 I_1 = R_2 I_2$: de stromen verhouden zich *omgekeerd* als de weerstanden. Met de totale stroom I:

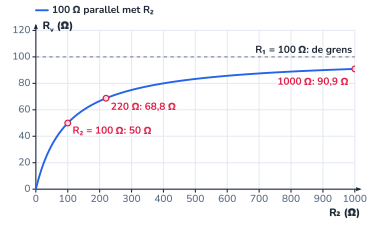
STROOMDELER (TWEE TAKKEN)

$$I_1 = I \cdot \frac{R_2}{R_1 + R_2} \qquad I_2 = I \cdot \frac{R_1}{R_1 + R_2}$$

Let op: in de teller staat de weerstand van de *andere* tak.



In serie één weg, parallel meerdere: de formules volgen uit de maaswet en de knooppuntwet.



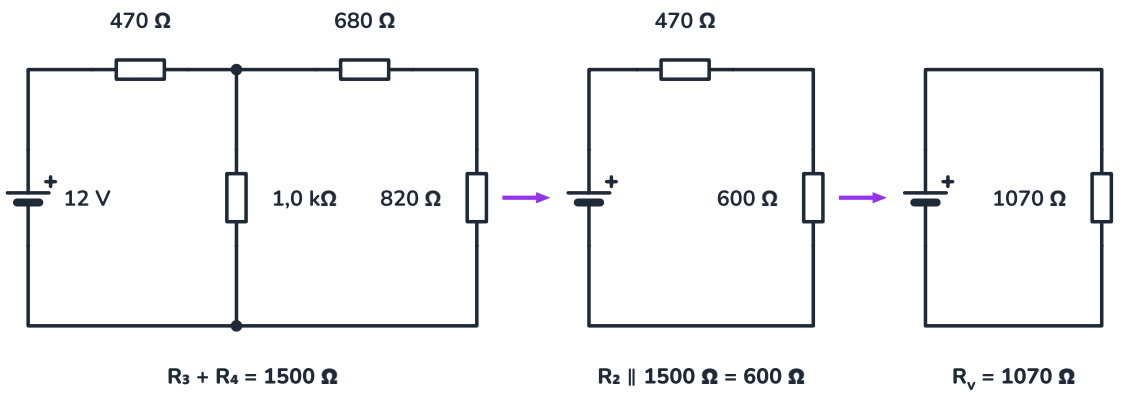
Hoe groot R_2 ook wordt, 100 Ω parallel met R_2 blijft onder 100 Ω.

UITGEWERKT VOORBEELD · 21.10

Een gemengde schakeling volledig oplossen

10/19

Bereken in het netwerk van het ontleedblad ($U = 12\,\text{V}$, $R_1 = 470\,\Omega$, $R_2 = 1{,}0\,\text{k}\Omega$, $R_3 = 680\,\Omega$, $R_4 = 820\,\Omega$) alle stromen en alle spanningen. Controleer met de knooppuntwet en de maaswet.



Vereenvoudigen van binnen naar buiten, daarna terugrekenen van buiten naar binnen.

GEGEVEN

- $U = 12\,\text{V}$
- $R_1 = 470\,\Omega$, $R_2 = 1000\,\Omega$, $R_3 = 680\,\Omega$, $R_4 = 820\,\Omega$

GEVRAAGD

- I_1, I_2, I_3; U_1 tot U_4

OPLOSSING

1. **Vereenvoudig van binnen naar buiten.** R_3 en R_4 staan in serie: $R_{34} = 680 + 820 = 1500\,\Omega$.
2. R_2 staat parallel met R_{34}: $R_\text{p} = \dfrac{1000 \cdot 1500}{1000 + 1500} = 600\,\Omega$.
3. R_1 staat in serie met die groep: $R_\text{v} = 470 + 600 = 1070\,\Omega$, dus $I_1 = \dfrac{12}{1070} = 0{,}01121\,\text{A} = 11{,}21\,\text{mA}$.
4. **Rek terug naar binnen.** $U_1 = 470 \cdot 0{,}01121 = 5{,}27\,\text{V}$ en de spanning over de parallelle groep $U_\text{p} = 600 \cdot 0{,}01121 = 6{,}73\,\text{V}$.
5. $I_2 = \dfrac{6{,}729}{1000} = 6{,}73\,\text{mA}$ en $I_3 = \dfrac{6{,}729}{1500} = 4{,}49\,\text{mA}$.
6. $U_3 = 680 \cdot I_3 = 3{,}05\,\text{V}$ en $U_4 = 820 \cdot I_3 = 3{,}68\,\text{V}$.

ANTWOORD

$I_1 = 11{,}21\,\text{mA}$, $I_2 = 6{,}73\,\text{mA}$, $I_3 = 4{,}49\,\text{mA}$; $U_1 = 5{,}27\,\text{V}$, $U_2 = 6{,}73\,\text{V}$, $U_3 = 3{,}05\,\text{V}$, $U_4 = 3{,}68\,\text{V}$.

Klopt dit? Knooppunt A: $11,21 = 6,73 + 4,49$ mA. Maas I: $12 - 5,27 - 6,73 = 0$. Maas II (met de klok mee, door R_3 en R_4 met de stroom mee, door R_2 ertegen in): $-3,05 - 3,68 + 6,73 = 0$. Alles klopt tot op de afronding.

UITLEG · 21.11

Vermogen in elke weerstand

11/19

Elke weerstand zet elektrische energie om in warmte. Zijn vermogen volgt uit $P = U \cdot I$ (F21), en met de wet van Ohm kan je dat op drie manieren schrijven:

VERMOGEN IN EEN WEERSTAND

$$P = U \cdot I = R \cdot I^2 = \frac{U^2}{R}$$

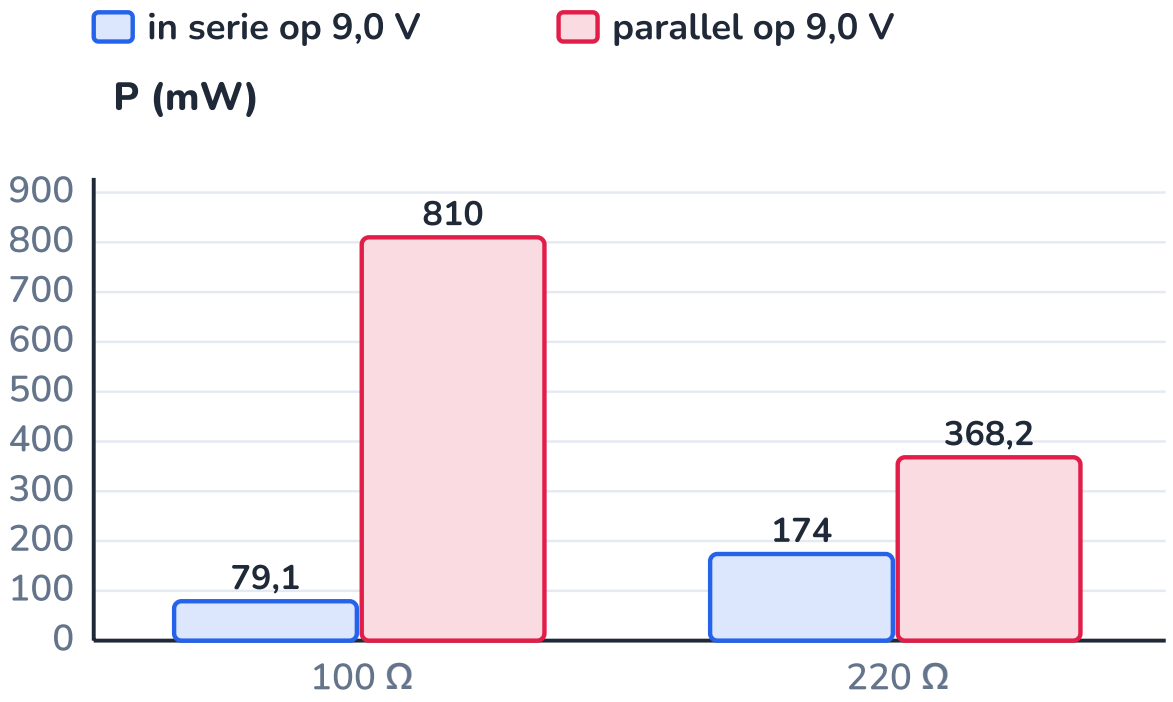
Kies de vorm met wat je kent: in serie ken je vaak de stroom (RI^2), parallel de spanning (U^2/R).

Twee gevolgen die veel mensen verrassen, met 100 Ω en 220 Ω op 9,0 V (grafiek):

- **In serie** is de stroom gelijk, dus krijgt de **grootste** weerstand het grootste vermogen ($P = RI^2$): 174 mW in 220 Ω tegenover 79 mW in 100 Ω.
- **Parallel** is de spanning gelijk, dus krijgt de **kleinste** weerstand het grootste vermogen ($P = U^2/R$): 810 mW in 100 Ω, 368 mW in 220 Ω.

De vermogensbalans. Energie gaat niet verloren: wat alle bronnen samen leveren, nemen alle weerstanden samen op. In het gemengde voorbeeld: $P_1 + P_2 + P_3 + P_4 = 59{,}1 + 45{,}3 + 13{,}7 + 16{,}5 = 134{,}6\,\mathrm{mW}$, en de bron levert $12 \cdot 11{,}21\,\mathrm{mA} = 134{,}6\,\mathrm{mW}$. Dat is een uitstekende controle op een hele berekening.

> ⚠ **Een gewone weerstand verdraagt 0,25 W**
>
> De weerstandjes in een doos met onderdelen zijn meestal gemaakt voor 0,25 W (¼ W). In de parallelschakeling hierboven wordt de weerstand van 100 Ω met 810 mW meer dan drie keer overbelast: hij wordt heet, verkleurt en kan doorbranden. Reken het vermogen dus altijd na, en kies zo nodig een weerstand van 1 W of 2 W.



In serie wordt de grootste weerstand het warmst, parallel de kleinste.

PROEF · 21.12

Kirchhoff nagemeten

12/19

ONDERZOEKSVRAAG

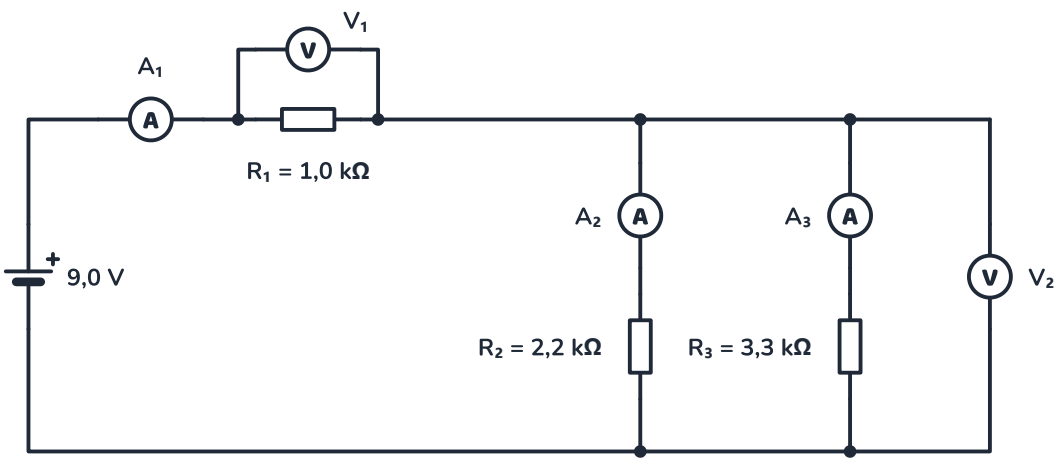
Kloppen de knooppuntwet en de maaswet ook met echte meters en weerstanden met een tolerantie?

MATERIAAL

voeding of 9 V-batterij · weerstanden 1,0 kΩ, 2,2 kΩ en 3,3 kΩ (tolerantie 5 %) · breadboard · multimeters (liefst drie) · draadjes

Veiligheid

Werk onder 12 V. Zet de multimeter op mA voor je hem in serie zet; nooit parallel over de bron.



De ampèremeters zitten in de takken, de voltmeters over R_1 en over de parallelle groep.

WERKWIJZE

1. Bouw de schakeling van het schema: R_1 in serie met R_2 en R_3, die parallel staan.
2. Bereken vooraf alle stromen en spanningen met de nominale waarden (stroom in mA, spanning in V).
3. Meet de bronspanning, de spanning over R_1 en over de parallelle groep.
4. Meet de stroom in de hoofdleiding en in elke tak. Heb je maar één meter, verplaats hem dan: open telkens de kring op die plaats.
5. Controleer: tellen de takstromen op tot de hoofdstroom? Tellen U_1 en U_2 op tot de bronspanning?

WAARNEMING

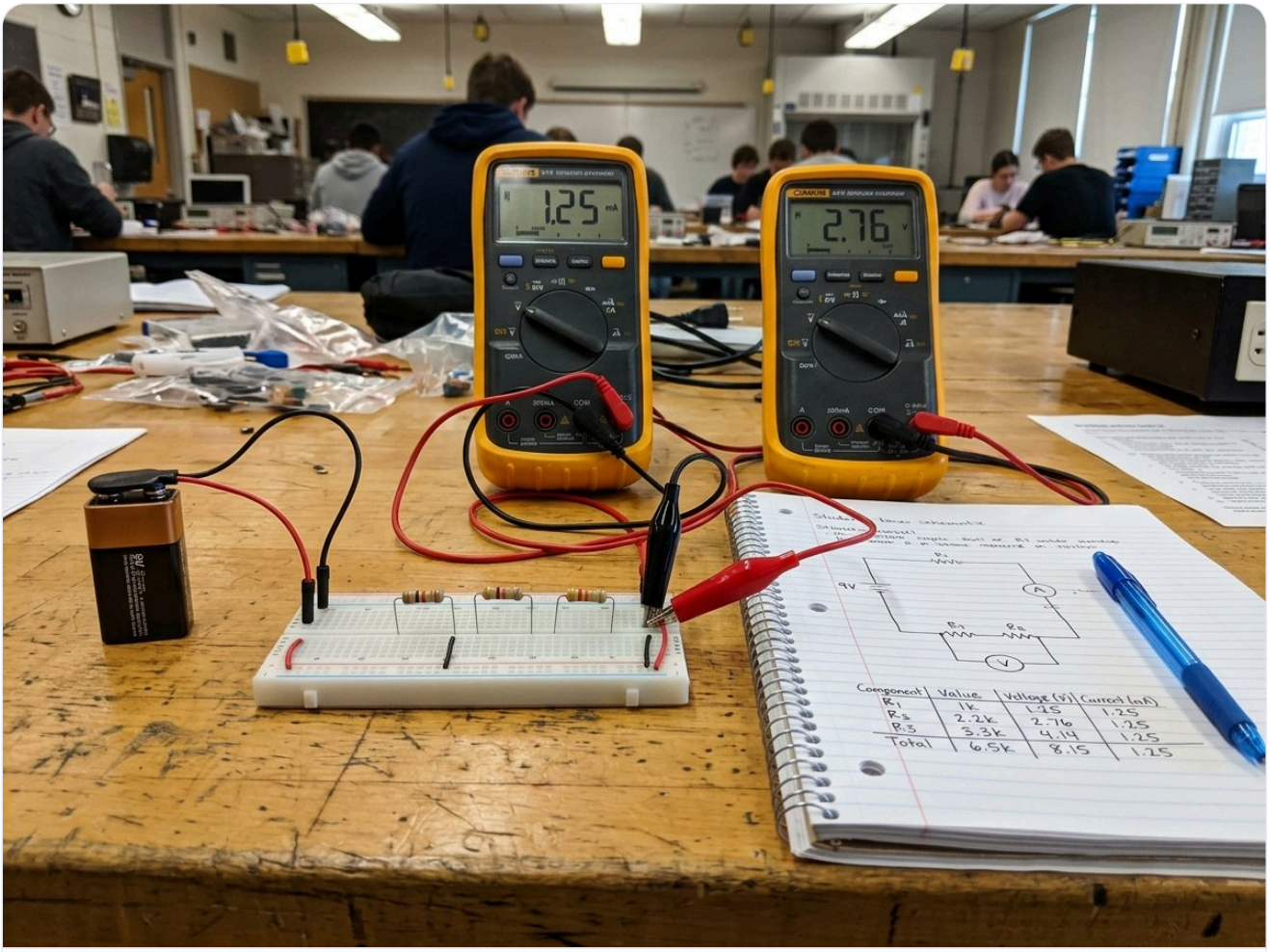
meter	berekend (nominaal)	gemeten
bron	9,00 V	9,02 V
A_1 (hoofdleiding)	3,88 mA	3,91 mA
A_2 (tak R_2)	2,33 mA	2,33 mA
A_3 (tak R_3)	1,55 mA	1,58 mA
V_1 (over R_1)	3,88 V	3,86 V
V_2 (over R_2 en R_3)	5,12 V	5,16 V

Voorbeeldmeting. Knooppuntwet: $2{,}33 + 1{,}58 = 3{,}91\,\text{mA}$. Maaswet: $3{,}86 + 5{,}16 = 9{,}02\,\text{V}$. Beide kloppen op het laatste cijfer.

VERKLARING

De wetten van Kirchhoff kloppen ook voor de metingen: ze zijn geen benadering, maar behoud van lading en energie. Waar berekening en meting verschillen, zit het verschil in de **gegevens**: de bron gaf 9,02 V in plaats van 9,00 V, en de weerstanden wijken tot 5 % af van hun opgedrukte waarde (hier bleek $R_1 = 0{,}988\,\text{k}\Omega$, $R_2 = 2{,}21\,\text{k}\Omega$, $R_3 = 3{,}27\,\text{k}\Omega$ met de ohmmeter). Met die echte waarden geeft de berekening precies de metingen.

Een multimeter op het mA-bereik heeft zelf een kleine weerstand (enkele tot enkele tientallen ohm). Tegenover kilo-ohms is dat verwaarloosbaar; in een kring met enkele ohm zou de meter zelf de stroom merkbaar verkleinen. Dat is de reden om voor zo'n proef weerstanden van enkele kilo-ohm te kiezen.



Drie weerstanden, twee meters en een schrift: meer heb je niet nodig om Kirchhoff te controleren.

UITLEG · 21.13

Van netwerk naar stelsel

13/19

Niet elk netwerk bestaat uit serie- en parallelstukken. Staan er twee bronnen in verschillende takken, of zit er een weerstand 'dwars' tussen twee takken (een brug), dan kan je niets meer samennemen. Dan schrijf je de wetten van Kirchhoff gewoon op en los je het stelsel op. Met b takken heb je b onbekende stromen, en dus b onafhankelijke vergelijkingen nodig:

ZOVEEL VERGELIJKINGEN ALS ONBEKENDEN

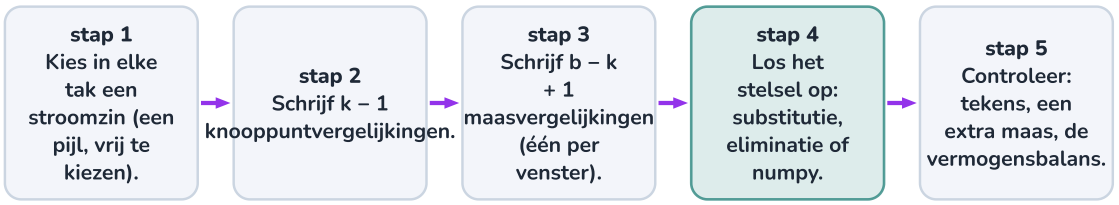
$$(k-1)\ \text{knooppuntvergelijkingen} \ + \ (b-k+1)\ \text{maasvergelijkingen} = b$$

k: aantal knooppunten; b: aantal takken. De laatste knooppuntvergelijking volgt uit de andere; $b-k+1$ is het aantal vensters van een vlak getekend schema.

Voor het netwerk van het ontleedblad: $k=2$ (A en B), $b=3$. Dus 1 knooppuntvergelijking en 2 maasvergelijkingen (vensters I en II): drie vergelijkingen voor I_1, I_2 en I_3.

Waarom niet gewoon meer vergelijkingen?

Je kan er meer opschrijven (de buitenste maas, het tweede knooppunt), maar die zijn een som van de andere: ze brengen niets nieuws. Ze zijn wel nuttig als **controle** achteraf.



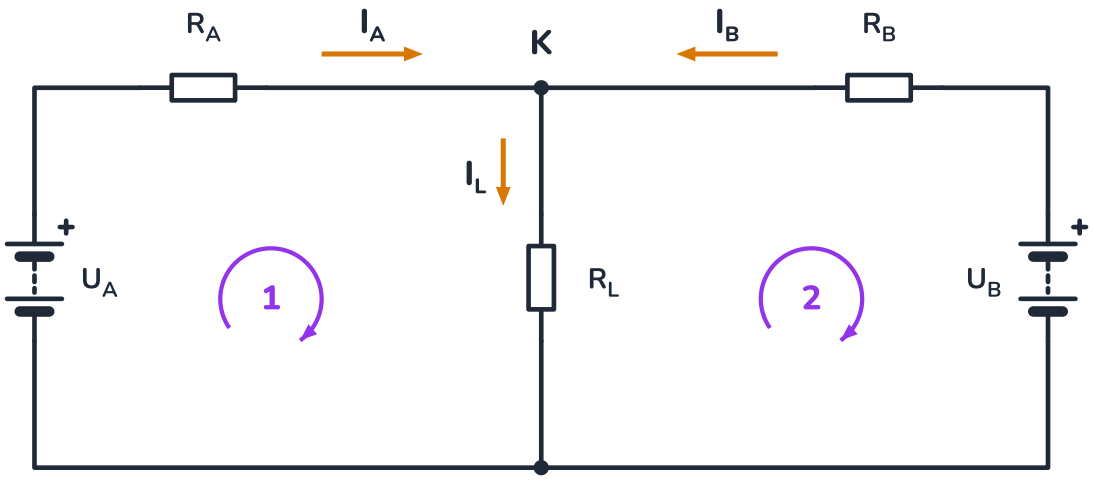
Het stappenplan werkt voor elk netwerk, ook als er niets te vereenvoudigen valt.

UITGEWERKT VOORBEELD · 21.14

Twee batterijen voeden samen de lampen

14/19

Dezelfde twee batterijen als eerder (13,30 V met 20 mΩ en 13,10 V met 25 mΩ) voeden nu samen de groeilampen. Stel de lampen voor als een weerstand $R_L = 0{,}90\,\Omega$ (190 W bij ongeveer 13 V). Hoeveel stroom levert elke batterij, en welke spanning krijgen de lampen?



Drie takken, twee knooppunten, twee vensters: drie vergelijkingen voor drie stromen.

GEGEVEN

- $U_A = 13{,}30\,\text{V}$, $R_A = 0{,}020\,\Omega$
- $U_B = 13{,}10\,\text{V}$, $R_B = 0{,}025\,\Omega$
- $R_L = 0{,}90\,\Omega$

GEVRAAGD

- I_A, I_B, I_L en de spanning over de lampen

OPLOSSING

1. Tel: $k = 2$ knooppunten (boven K, onder), $b = 3$ takken. Kies de pijlen zoals in de figuur: I_A en I_B naar K toe, I_L door de lampen naar beneden.
2. Knooppunt K: $I_A + I_B = I_L$.
3. Maas 1 (A en de lampen, met de klok mee): $13{,}30 - 0{,}020\,I_A - 0{,}90\,I_L = 0$.
4. Maas 2 (B en de lampen, tegen de klok in, ook vanaf de min van B): $13{,}10 - 0{,}025\,I_B - 0{,}90\,I_L = 0$.
5. **Substitutie.** Vul $I_L = I_A + I_B$ in: $0{,}92\,I_A + 0{,}90\,I_B = 13{,}30$ en $0{,}90\,I_A + 0{,}925\,I_B = 13{,}10$.

6 **Eliminatie.** Trek de tweede vergelijking af van de eerste: $0{,}020\,I_A - 0{,}025\,I_B = 0{,}20$, dus $I_A = 10 + 1{,}25\,I_B$. Invullen in de tweede: $0{,}90\,(10 + 1{,}25\,I_B) + 0{,}925\,I_B = 13{,}10$, zodat $2{,}05\,I_B = 4{,}10$ en $I_B = 2{,}00\ \text{A}$. Dan $I_A = 12{,}50\ \text{A}$ en $I_L = 14{,}50\ \text{A}$.

ANTWOORD

Batterij A levert 12,5 A, batterij B 2,0 A; de lampen krijgen 14,5 A bij $0{,}90 \cdot 14{,}5 = 13{,}05\ \text{V}$.

Klopt dit? Maas via A: $13{,}30 - 0{,}020 \cdot 12{,}5 = 13{,}05\ \text{V}$; via B: $13{,}10 - 0{,}025 \cdot 2{,}0 = 13{,}05\ \text{V}$: dezelfde railspanning. Het vermogen in de lampen is $13{,}05 \cdot 14{,}5 \approx 189\ \text{W}$: dat klopt met 190 W. De volste batterij levert het meest; zo lopen hun laadtoestanden vanzelf naar elkaar toe.

CODE · 21.15

Het stelsel met numpy

15/19

Met de hand is een stelsel van drie vergelijkingen nog te doen; een netwerk met twintig takken niet meer. Schrijf het stelsel daarom als één **matrixvergelijking** $A \cdot x = b$: elke rij van A bevat de coëfficiënten van één vergelijking, x is de lijst onbekende stromen, en b bevat wat rechts van het gelijkheidsteken staat. De bibliotheek numpy lost dat op met `np.linalg.solve`. Hoe dat binnenin werkt (de methode van Gauss) en wat een matrix precies is, leer je in hoofdstuk E5.

$$\begin{pmatrix} 1 & 1 & -1 \\ 0{,}020 & 0 & 0{,}90 \\ 0 & 0{,}025 & 0{,}90 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_L \end{pmatrix} = \begin{pmatrix} 0 \\ 13{,}30 \\ 13{,}10 \end{pmatrix}$$

Python **twee_batterijen.py** 25 regels

```python
# twee_batterijen.py - twee batterijen voeden samen de groeilampen: een stelsel van drie vergelijkingen
import numpy as np

UA, RA = 13.30, 0.020     # batterij A: bronspanning (V), inwendige weerstand met kabel (ohm)
UB, RB = 13.10, 0.025     # batterij B
RL = 0.90                 # de groeilampen als weerstand (ohm)

# onbekenden x = [IA, IB, IL], in de gekozen stroomzin van de figuur
A = np.array([[1.0, 1.0, -1.0],     # knooppunt K:  IA + IB - IL = 0
              [RA,  0.0,  RL],      # maas 1:       RA*IA + RL*IL = UA
              [0.0, RB,   RL]])     # maas 2:       RB*IB + RL*IL = UB
b = np.array([0.0, UA, UB])
IA, IB, IL = np.linalg.solve(A, b)
print(f"IA = {IA:.2f} A   IB = {IB:.2f} A   IL = {IL:.2f} A")
print(f"spanning over de lampen: {RL * IL:.3f} V")

# controle: wat de bronnen leveren, nemen de weerstanden op
P_bronnen = UA * IA + UB * IB
P_weerstanden = RA * IA**2 + RB * IB**2 + RL * IL**2
print(f"bronnen {P_bronnen:.2f} W, weerstanden {P_weerstanden:.2f} W")

print("\n  UB (V)   IA (A)   IB (A)   IL (A)")
for ub in (12.9, 13.0, 13.1, 13.2, 13.3):
    IA, IB, IL = np.linalg.solve(A, [0.0, UA, ub])
    print(f"{ub:8.2f} {IA:8.2f} {IB:8.2f} {IL:8.2f}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
IA = 12.50 A   IB = 2.00 A   IL = 14.50 A
spanning over de lampen: 13.050 V
bronnen 192.45 W, weerstanden 192.45 W

  UB (V)   IA (A)   IB (A)   IL (A)
   12.90    16.89    -2.49    14.40
```

```
13.00    14.70    -0.24    14.45
13.10    12.50     2.00    14.50
13.20    10.30     4.24    14.55
13.30     8.11     6.49    14.60
```

De vermogensbalans klopt tot op de laatste decimaal: een goede test voor elk programma dat netwerken oplost. De tabel toont wat er gebeurt als batterij B leger is: bij 12,9 V wordt I_B negatief, en laadt batterij A haar bij terwijl ze ook nog de lampen voedt. De lampen zelf merken bijna niets: hun stroom blijft rond 14,5 A.

UITLEG · 21.16

Habitat: de 5 V-voeding van alle sensoren narekenen

16/19

De microcontroller en alle sensoren van de klimaatmodule hangen parallel aan één **5 V-rail**, die gevoed wordt door een omvormer van 12 V naar 5 V (rendement 83 %, zoals in de meetkas). De rail is één knooppunt: de stroom die de omvormer levert, is de som van alle verbruikers. Voor elke verbruiker haal je de stroom uit de datasheet: het gemiddelde voor het energieverbruik, de piek om de omvormer en de draden te kiezen.

verbruiker op 5 V	gemiddeld (mA)	piek (mA)
microcontroller (Arduino Mega 2560, rekenwaarde)	50	50
CO_2-sensor (NDIR, MH-Z19B)	60	150
pH-module (hoofdstuk 15)	10	10
stroomsensor ACS712 (hoofdstuk 16)	10	13
druksensor MPX5010 (hoofdstuk 11)	5	10
SD-kaartmodule (datalogger)	10	100
twee statusleds (hoofdstuk 23)	10	20
temperatuur en RV (DHT22)	1,5	1,5
lichtsensor, LDR + 1 kΩ (hoofdstuk 22)	2	4,5
NTC in het reservoir (hoofdstuk 22)	0,25	0,25
EC-meting, gepulst (hoofdstuk 13)	0,10	2,3
som (knooppuntwet)	**158,8**	**361,6**

Vermogen. Aan de 5 V-kant: $5 \cdot 0{,}159 = 0{,}79\ \mathrm{W}$ gemiddeld, 1,81 W bij de pieken. De omvormer neemt aan de 12 V-kant $P_{\text{in}} = P_{\text{uit}}/\eta = 0{,}79/0{,}83 = 0{,}96\ \mathrm{W}$ gemiddeld (2,2 W piek), bij 12,8 V een stroom van 0,075 A. Een omvormer van 1 A aan de 5 V-kant heeft dus meer dan genoeg reserve, en de zekering van 1 A aan de 12 V-kant beschermt de draad.

Het budget. Hoofdstuk 20 rekende voor 'microcontroller en sensoren' met 3,0 W. De som van de datasheets geeft 0,96 W gemiddeld: het budget is ruim. Die ruimte is welkom: een display en een wifi-module voor de datalogger komen er later bij (opdracht 21.13), en een kleine omvormer werkt bij weinig belasting slechter dan 83 %.

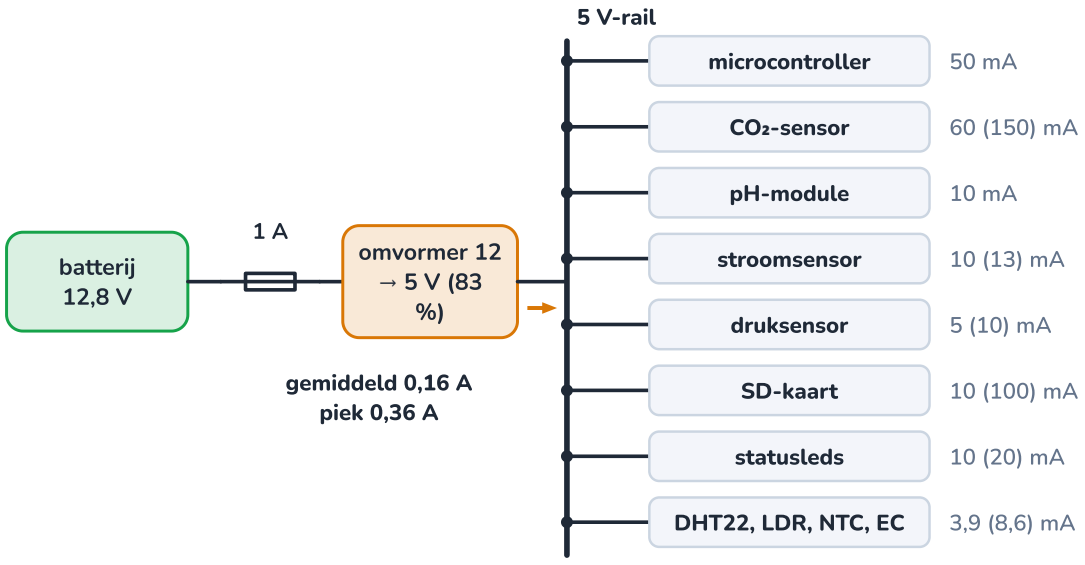
De sensorkast op drie meter. De CO_2-sensor, de temperatuursensor en de lichtsensor zitten in een kastje boven het kweekbed, 3,0 m van de microcontroller, aan een sensorkabel met aders van 0,14 mm². De weerstand van een draad volgt uit zijn lengte en doorsnede:

WEERSTAND VAN EEN DRAAD

$$R = \rho \cdot \frac{L}{A}$$

ρ: soortelijke weerstand, voor koper $0{,}0172\,\Omega \cdot \mathrm{mm}^2/\mathrm{m}$; L: lengte (m); A: doorsnede (mm²).

Eén ader: $R = 0{,}0172 \cdot 3{,}0/0{,}14 = 0{,}369\,\Omega$. Bij de piek van het kastje (156 mA, vooral de CO_2-sensor) valt er over de plus- én over de massader 0,057 V. De maaswet langs plusader, sensor en massader: $5{,}00 - 2 \cdot 0{,}057 = 4{,}89\,\mathrm{V}$ aan de sensoren. Dat ligt binnen de 4,5 tot 5,5 V die de CO_2-sensor vraagt.



Elf verbruikers parallel op de 5 V-rail; tussen haakjes de piekstroom. De omvormer levert de som.

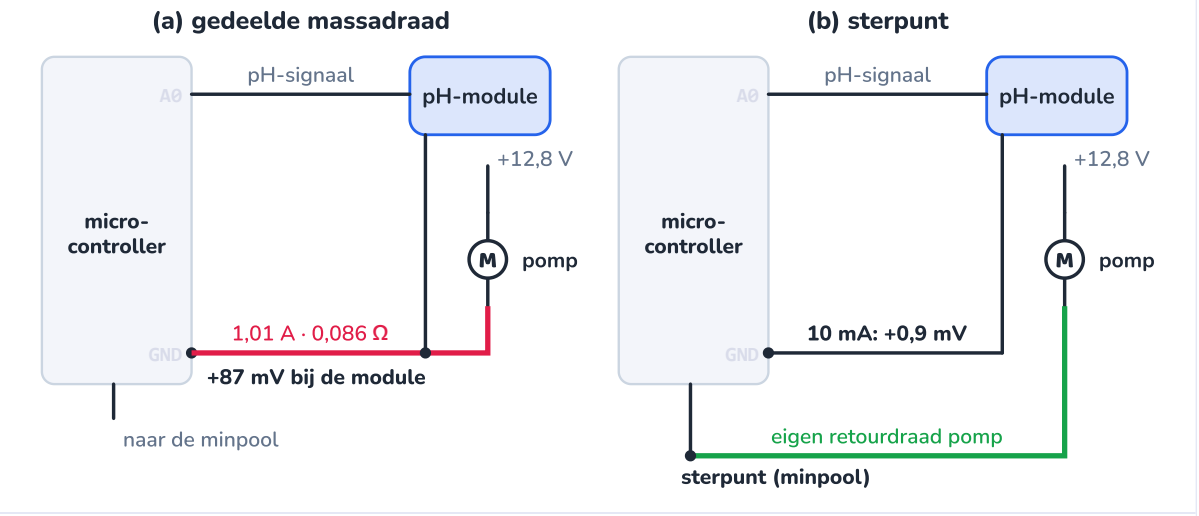
De sensorkast boven het kweekbed hangt aan drie meter kabel: ook die draden hebben weerstand.

UITGEWERKT VOORBEELD · 21.17

Habitat: waarom de pH verspringt als de pomp aanslaat

17/19

Bij de eerste test van de klimaatmodule verspringt de gemeten pH telkens met ongeveer een halve eenheid als de pomp aanslaat. De pH-module (hoofdstuk 15: 2,50 V bij pH 7,00, $-0{,}18\,\mathrm{V}$ per pH) en de pomp (1,0 A) staan allebei bij het reservoir. Om draad te sparen, lopen hun massadraden samen over één ader van 2,5 m en 0,5 mm² naar de massa van de microcontroller. Verklaar het verspringen en reken het na.



Links deelt de pH-module haar massadraad met de pomp; rechts heeft elk zijn eigen draad naar één sterpunt.

GEGEVEN

- gedeelde massadraad: $L = 2{,}5\,\text{m}$, $A = 0{,}5\,\text{mm}^2$, koper
- pomp 1,0 A, pH-module 10 mA
- pH-module: $-0{,}18\,\text{V}$ per pH

GEVRAAGD

- de verschuiving van de massa van de module
- de fout in de pH
- een oplossing

OPLOSSING

1. Weerstand van de gedeelde draad: $R = 0{,}0172 \cdot 2{,}5/0{,}5 = 0{,}086\,\Omega$.
2. Als de pomp draait, loopt er $1{,}0 + 0{,}01 = 1{,}01\,\text{A}$ door die draad naar de microcontroller. De massa van de module ligt dan $0{,}086 \cdot 1{,}01 = 0{,}087\,\text{V}$ **hoger** dan de massa van de microcontroller.
3. De module zet haar signaal ten opzichte van haar eigen massa. Maaswet langs signaaldraad, module en massadraad: de microcontroller meet het signaal plus 0,087 V.
4. Een hogere spanning betekent bij deze module een lagere pH: de fout is $0{,}087/0{,}18 = 0{,}48$ pH. De regeling denkt dat de oplossing zuurder wordt zodra de pomp draait.
5. **Oplossing: een sterpunt.** De pomp krijgt een eigen retourdraad rechtstreeks naar de minpool. Door de massadraad van de module loopt dan alleen haar eigen 10 mA: $0{,}086 \cdot 0{,}010 = 0{,}00086\,\text{V}$, of 0,005 pH.

ANTWOORD

De pompstroom maakt over de gedeelde massadraad 87 mV, en de pH-meting schuift daardoor 0,48 pH op. Met een eigen retourdraad voor de pomp (sterpunt) wordt de fout 0,005 pH: honderd keer kleiner.

Klopt dit? De sprong is even groot als de waarneming ('ongeveer een halve eenheid'), en ze verdwijnt als de pomp stilvalt: een sterk teken dat de verklaring klopt. Dezelfde regel geldt voor elke analoge sensor: grote stromen (lampen, pomp, ventilatoren) nooit door de massadraad van een meting laten lopen.

NASLAG · 21.18

Overzicht: de wetten van Kirchhoff

18/19

knooppuntwet

$$\sum I_{\text{in}} = \sum I_{\text{uit}}$$

symbool	betekenis	eenheid
I	stroom in een tak (met gekozen pijl)	A

maaswet

$$\sum_{\text{maas}} \Delta V = 0$$

symbool	betekenis	eenheid
ΔV	stijging (+) of daling (−) van de potentiaal	V

vervangingsweerstand in serie en parallel

$$R_{\text{v}} = R_1 + R_2 + \dots \qquad \frac{1}{R_{\text{v}}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots$$

symbool	betekenis	eenheid
R_{v}	vervangingsweerstand	Ω

stroomdeler

$$I_1 = I \cdot \frac{R_2}{R_1 + R_2}$$

symbool	betekenis	eenheid
I	stroom naar de twee takken	A

vermogen in een weerstand

$$P = UI = RI^2 = \frac{U^2}{R}$$

symbool	betekenis	eenheid
P	vermogen	W

klemspanning

$$U_{\text{k}} = U_0 - R_{\text{i}} I$$

symbool	betekenis	eenheid
U_0	bronspanning	V
R_{i}	inwendige weerstand	Ω

weerstand van een draad

$$R = \rho \frac{L}{A}$$

symbool	betekenis	eenheid
ρ	soortelijke weerstand (koper 0,0172)	Ω·mm²/m
L	lengte	m
A	doorsnede	mm²

je loopt door …	in de zin van de pijl / van − naar +	tegen de pijl / van + naar −
een weerstand R met stroom I	$-R \cdot I$	$+R \cdot I$
een bron U	$+U$ (van − naar +)	$-U$ (van + naar −)
een draad	0 (geen weerstand, geen spanning)	

Een negatieve stroom in de uitkomst: de stroom loopt tegen de gekozen pijl in. Een negatieve spanning: de potentiaal daalt waar je een stijging verwachtte.

stap	wat je doet
1	Knooppunten (k) en takken (b) tellen; per tak een stroompijl.
2	$k - 1$ knooppuntvergelijkingen.
3	$b - k + 1$ maasvergelijkingen, één per venster, met de tekenafspraken.
4	Oplossen: vereenvoudigen als het kan, anders substitutie of numpy.
5	Controle: een extra knooppunt of maas, de vermogensbalans, en de orde van grootte.

HET GEHEEL · 21.19

Waar Kirchhoff vandaan komt en waar hij heen gaat

19/19

De twee wetten zijn behoudswetten: de knooppuntwet is behoud van lading, de maaswet behoud van energie. Daarom duiken ze ook buiten de elektriciteit op, overal waar iets 'stroomt' onder invloed van een

'verschil': water door leidingen (druk en debiet, hoofdstuk 11), warmte door een wand (temperatuur en warmtestroom, hoofdstuk 19), zelfs verkeer op een kruispunt. En een netwerk oplossen is lineaire algebra: de leerlijn wiskundig modelleren en de leerlijn programmeren komen hier samen.

F19 Spanning, stroom en weerstand
De wet van Ohm, die je hier in elke tak gebruikt.

F20 Serie- en parallelschakeling
De vier regels voor serie en parallel: bijzondere gevallen van Kirchhoff.

F21 Elektrisch vermogen en energie
P = U · I, nu per weerstand en als vermogensbalans.

I3 Verbanden in data
De inwendige weerstand van een bron is de helling van een rechte door meetpunten.

I16 Redox en batterijen
De LiFePO4-batterij, de coulombteller en het teken van laden en ontladen.

I20 Zonne-energie dimensioneren
De tweede batterij en het budget van 3,0 W voor de sensoren.

I22 De spanningsdeler en sensoren
Twee weerstanden in serie als spanningsdeler: zo meet je met een sensor.

I24 Condensatoren en de RC-tijd
Kirchhoff blijft gelden als de spanning verandert, ook met condensatoren.

E5 Vectoren en matrices
Het stelsel als matrixvergelijking, en hoe de methode van Gauss het oplost.

E22 Operationele versterkers
De brug van Wheatstone en de instrumentatieversterker: netwerken met een meetdoel.

Habitat
De voeding van de klimaatmodule is nagerekend: een 5 V-rail met gemiddeld 0,16 A en pieken tot 0,36 A, een omvormer van 1 A, minder dan 1 W aan de 12 V-kant tegenover de 3,0 W van het budget, en 4,89 V aan de sensoren op drie meter kabel. De grootste les zit in de massa: grote stromen krijgen een eigen retourdraad naar een sterpunt. In opdracht 21.13 breid je de rail uit met een display en een wifi-module.

SAMENVATTING

In het kort

- Een knooppunt is een punt waar drie of meer draden samenkomen, een tak het stuk ertussen, een maas een gesloten lus.
- Knooppuntwet: wat een knooppunt binnenkomt, gaat eruit ($\sum I_{\text{in}} = \sum I_{\text{uit}}$); behoud van lading.
- Maaswet: langs een gesloten lus is de som van de potentiaalverschillen nul; behoud van energie. Bron van − naar +: $+U$; weerstand met de pijl mee: $-RI$.
- Kies de stroompijlen vrij: een negatieve uitkomst betekent dat de stroom tegen de pijl in loopt.
- Vervangingsweerstand: in serie $R_1 + R_2 + \ldots$; parallel $1/R_{\text{v}} = 1/R_1 + 1/R_2 + \ldots$, altijd kleiner dan de kleinste.
- Vermogen per weerstand: $P = RI^2 = U^2/R$; in serie wordt de grootste weerstand het warmst, parallel de kleinste. De bronnen leveren wat de weerstanden opnemen.
- Een echte bron: $U_{\text{k}} = U_0 - R_{\text{i}}I$. Een netwerk met b takken en k knooppunten geeft $k - 1$ knooppunt- en $b - k + 1$ maasvergelijkingen: los ze op met de hand of met numpy.
- Een gedeelde massadraad laat de stroom van een grote verbruiker in een sensormeting doorklinken: gebruik een sterpunt.

BEGRIPPEN

Wat je nu kent

knooppunt
Punt waar drie of meer geleiders samenkomen. Punten die alleen door een draad zonder onderdelen verbonden zijn, vormen samen één knooppunt.

tak
Stuk van een netwerk tussen twee knooppunten, zonder vertakking: door alle onderdelen van een tak loopt dezelfde stroom.

maas
Gesloten lus in een netwerk. Een venster van een vlak getekend schema is een onafhankelijke maas.

knooppuntwet
De som van de stromen die een knooppunt binnenkomen, is gelijk aan de som van de stromen die eruit gaan (behoud van lading).

maaswet
Langs een gesloten lus is de som van de spanningen nul: je komt terug op dezelfde potentiaal (behoud van energie).

potentiaal
De spanning van een punt ten opzichte van een gekozen referentiepunt (de massa, 0 V).

inwendige weerstand
Weerstand in een bron zelf; hierdoor zakt de klemspanning als de bron stroom levert: $U_{\mathrm{k}} = U_0 - R_{\mathrm{i}} I$.

vermogensbalans
Het vermogen dat de bronnen leveren, is gelijk aan de som van de vermogens in alle weerstanden.

sterpunt
Eén gemeenschappelijk massapunt waar elke verbruiker met een eigen retourdraad op aansluit, zodat grote stromen geen spanning opwekken in de massa van een sensor.

HANDBOEK · HOOFDSTUK 22

De spanningsdeler en sensoren

Hoe een microcontroller licht en temperatuur 'ziet': van weerstand naar spanning naar getal

Een LDR, een NTC en een draaiknop hebben één ding gemeen: hun weerstand verandert. Een microcontroller kan geen weerstand meten, alleen spanning. De brug tussen die twee is de spanningsdeler: twee weerstanden in serie, met een aftakking in het midden. In dit hoofdstuk leid je zijn formule af, kies je de juiste vaste weerstand, reken je een ADC-waarde om naar lux en graden, en leer je waarom een meter of een ingang de meting zelf kan verstoren. Je ijkt een NTC met een referentie en bouwt de lichtsensor die beslist of de groeilampen van de klimaatmodule aan moeten.

9× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 8 lestijden

L Elektriciteit en elektronica · O Onderzoeken en meten · P Programmeren en computationeel denken

Een lichtsensor onder een melkwit kapje beslist of de groeilampen aan moeten.

Na dit hoofdstuk

- Je kan de formule van de spanningsdeler afleiden met de wetten van Kirchhoff en ze in beide richtingen gebruiken.
- Je kan uitleggen hoe een LDR, een NTC en een potentiometer werken, en hun karakteristiek lezen en tekenen.
- Je kan een vaste weerstand kiezen zodat de sensor gevoelig is in het gebied waar het om gaat.
- Je kan een ADC-waarde van een microcontroller omrekenen naar spanning, weerstand en de gemeten grootheid (lux, °C).
- Je kan berekenen hoeveel een belasting de uitgang van een spanningsdeler verandert, en uitleggen waarom een ingang hoogohmig moet zijn.
- Je kan een sensor ijken met een referentie, ook als zijn karakteristiek niet recht is.
- Je kan een sensorprogramma in C++ schrijven dat een ADC-waarde omzet in een beslissing.

VERHAAL · 22.1

Twee toevallige ontdekkingen

In 1833 zag Michael Faraday iets vreemds bij zilversulfide: het geleidde **beter** naarmate het warmer werd. Bij metalen is het net omgekeerd; een warme koperdraad heeft een iets grotere weerstand (F19). Faraday had, zonder het te weten, een halfgeleider in handen: een stof waarin warmte extra ladingen vrijmaakt (hoofdstuk 23). Een NTC, de temperatuursensor van dit hoofdstuk, werkt nog altijd zo.

Veertig jaar later, in 1873, schreef de Engelse telegraafingenieur Willoughby Smith een brief aan het tijdschrift *Nature*. Hij gebruikte staafjes selenium als zeer grote weerstanden bij het testen van onderzeese telegraafkabels, en merkte dat hun weerstand sterk wisselde. De oorzaak bleek het licht: in een afgesloten doos was de weerstand groot, zodra het deksel openging, daalde ze. Dat is het principe van de LDR, de lichtgevoelige weerstand, en de voorloper van de eerste lichtmeters en fotocellen.

Beide ontdekkingen gaan over een weerstand die verandert met de omgeving. Wie die verandering kan meten, heeft een sensor. Daarvoor is één schakeling nodig, zo eenvoudig dat ze in bijna elk toestel zit: de spanningsdeler.

Waarom dit hoofdstuk ertoe doet

De groeilampen van de klimaatmodule zijn goed voor 93 % van haar energieverbruik (hoofdstuk 20). Als er genoeg daglicht is, hoeven ze niet te branden. Een goede lichtsensor spaart dus meer energie dan welke andere maatregel ook. Hetzelfde geldt voor de temperatuur van het reservoir: te warm water bevat weinig zuurstof voor de wortels.

Selenium tussen koperen klemmen: in het donker een grote weerstand, in het licht een veel kleinere.

Een microcontroller meet spanning, geen weerstand

Een analoge ingang van een Arduino Uno (A0 tot A5; de Mega van de klimaatmodule heeft er zestien, A0 tot A15, met dezelfde ADC) meet de **potentiaal** van één punt ten opzichte van GND (hoofdstuk 21), en zet die om in een getal. Dat doet de **ADC**, de analoog-digitaalomzetter. Hij verdeelt het gebied van 0 tot 5 V in $2^{10} = 1024$ stappen: het getal N gaat van 0 tot 1023, en één stap is $5\,\mathrm{V}/1024 = 4{,}88\,\mathrm{mV}$. Hoe dat binnenin werkt, zie je in hoofdstuk 25.

VAN SPANNING NAAR ADC-WAARDE (10 BIT)

$$N = \left\lfloor 1024 \cdot \frac{U}{U_{\mathrm{ref}}} \right\rfloor \qquad U \approx N \cdot \frac{U_{\mathrm{ref}}}{1024}$$

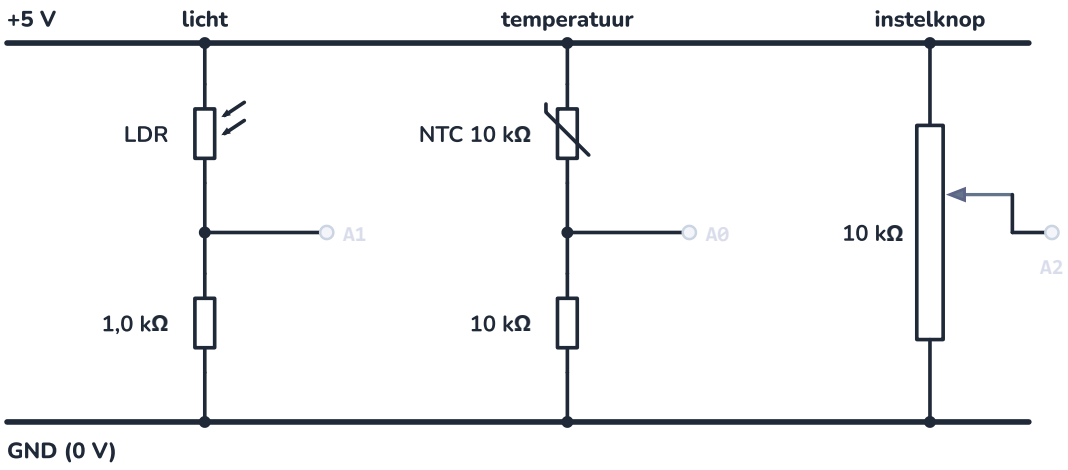
U_{ref} is bij een Uno de voedingsspanning van 5 V. De haken betekenen: naar beneden afronden.

Een LDR of een NTC levert geen spanning: het is een weerstand. Hang je hem gewoon aan een ingang, dan gebeurt er niets. Zet je hem in serie met een vaste weerstand tussen 5 V en GND, dan hangt de spanning op het punt ertussen af van de weerstand van de sensor. Die schakeling is de **spanningsdeler**. De figuur toont de drie sensoren van dit hoofdstuk, elk met hun eigen deler aan hun eigen ingang.

wat je al weet	wat er nu bijkomt
Een analoge ingang geeft een getal van 0 tot 1023 (F24, F25).	Wat dat getal precies is: een spanning, in stappen van 4,88 mV.
In serie verdeelt de spanning zich; de grootste weerstand krijgt het meest (F20).	De formule van de spanningsdeler, en hoe je ze omkeert om een weerstand te meten.
Een sensor ijken met een referentie (F25, hoofdstuk 3).	Een niet-lineaire sensor ijken door hem eerst recht te trekken.
Knooppunt- en maaswet (hoofdstuk 21).	Waarom de ingang van een microcontroller bijna geen stroom mag trekken.

⚠ 1023 of 1024?

In veel voorbeeldcode (ook in hoofdstuk 11 en 16) staat `U = N * 5.0 / 1023`. De datasheet van de ATmega328P, de chip op de Uno, rekent met 1024 stappen: $N = 1023$ betekent 'tussen 4,995 en 5 V'. Het verschil tussen beide formules is kleiner dan één stap (0,1 %). In dit hoofdstuk volgen we de datasheet.



LDR en NTC boven, een vaste weerstand onder; de potentiometer is zelf al een deler. Elke aftakking gaat naar een analoge ingang.

De spanningsdeler

Twee weerstanden R_1 en R_2 staan in serie over een spanning U_{in}. Zolang er niets aan de aftakking hangt (of iets dat geen stroom trekt), loopt door beide dezelfde stroom (knooppuntwet). De maaswet geeft $U_{\text{in}} = R_1 I + R_2 I$, dus $I = U_{\text{in}}/(R_1 + R_2)$. De spanning over R_2 is dan $R_2 I$:

SPANNINGSDELER (ONBELAST)

$$U_{\text{uit}} = U_{\text{in}} \cdot \frac{R_2}{R_1 + R_2}$$

U_{uit}: spanning over R_2 (tussen de aftakking en GND). De uitgang is altijd een deel van U_{in}: nooit meer, nooit negatief.

Drie dingen vallen op. De uitgang hangt alleen af van de **verhouding** van de weerstanden: 1 kΩ met 1 kΩ geeft dezelfde 2,5 V als 1 MΩ met 1 MΩ. De **grootste** weerstand krijgt het grootste deel van de spanning (F20). En de formule is eenvoudig om te keren: ken je U_{uit} en R_1, dan vind je R_2:

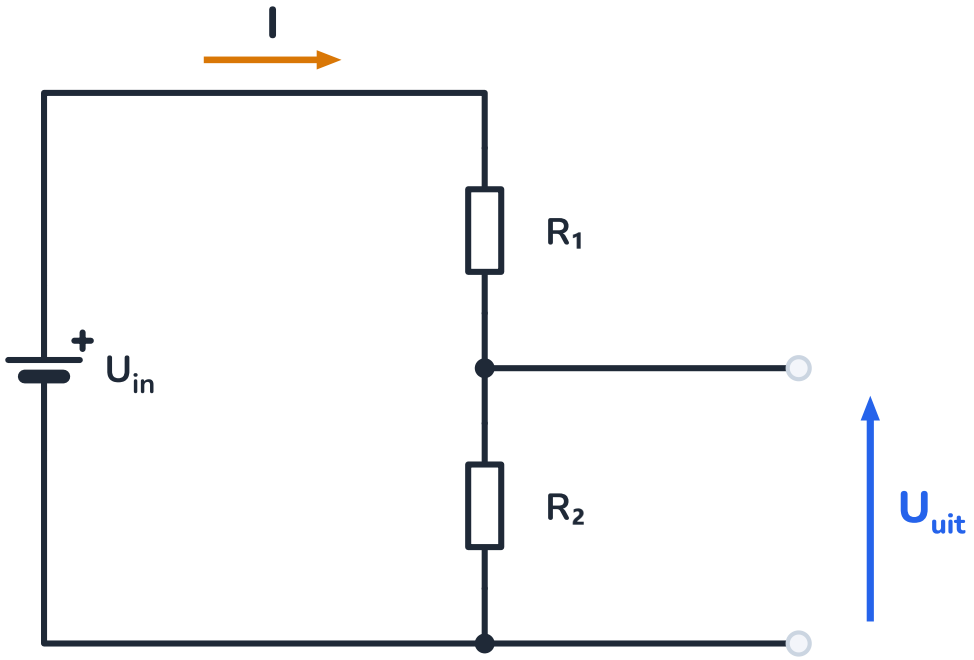
EEN ONBEKENDE WEERSTAND UIT DE DELER

$$R_2 = R_1 \cdot \frac{U_{\text{uit}}}{U_{\text{in}} - U_{\text{uit}}}$$

Zo meet een microcontroller een weerstand: hij kent R_1 en U_{in} en meet U_{uit}.

Ratiometrisch: de 5 V valt weg

Voedt de deler zich uit dezelfde 5 V die de ADC als referentie gebruikt, dan is $N/1024 = U_{\text{uit}}/U_{\text{in}} = R_2/(R_1 + R_2)$. De voedingsspanning valt uit de formule: of de USB-poort nu 4,8 of 5,1 V levert, het getal blijft hetzelfde. Daarom voed je een sensordeler altijd uit de 5 V van de microcontroller zelf, en niet uit een andere voeding.

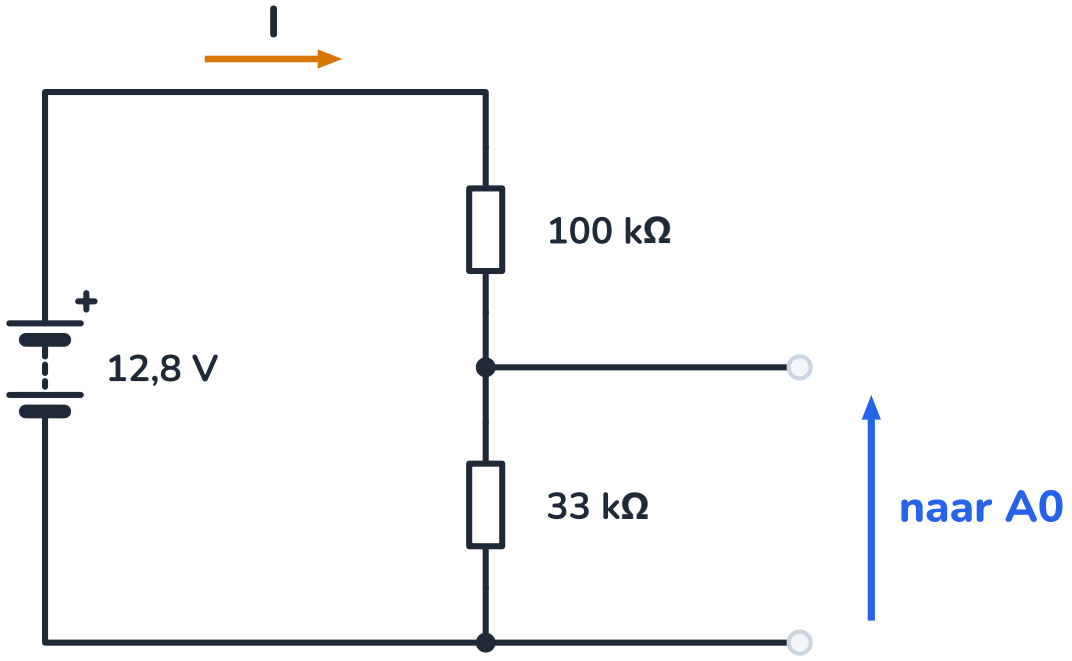


Door R_1 en R_2 loopt dezelfde stroom; de uitgang is de spanning over R_2.

UITGEWERKT VOORBEELD · 22.4

De batterijspanning meten

De microcontroller van de klimaatmodule volgt de spanning van de LiFePO4-batterij (hoofdstuk 16). Die schommelt tussen ongeveer 10 V (helemaal leeg) en 14,6 V (laden). Een ingang verdraagt hoogstens 5 V. Daarom hangt er een deler van 100 kΩ (boven) en 33 kΩ (onder) tussen de batterij en GND, met de aftakking op A0. (a) Welke spanning en welke ADC-waarde horen bij 12,8 V en bij 14,6 V? (b) Tot welke batterijspanning kan je meten? (c) Hoeveel stroom kost de deler? (d) Hoe groot is één ADC-stap, uitgedrukt in batterijspanning?



Een deler van 100 kΩ en 33 kΩ brengt tot 20 V batterijspanning terug tot 5 V.

GEGEVEN

- $R_1 = 100\,\text{k}\Omega$, $R_2 = 33\,\text{k}\Omega$
- ADC: 10 bit, $U_{\text{ref}} = 5{,}00\,\text{V}$

GEVRAAGD

- U op A0 en N bij 12,8 V en 14,6 V
- grootste meetbare spanning
- stroom
- stap

OPLOSSING

1. Verhouding: $\dfrac{R_2}{R_1 + R_2} = \dfrac{33}{133} = 0{,}248$.
2. (a) Bij 12,8 V: $U = 12{,}8 \cdot 0{,}248 = 3{,}176\,\text{V}$, $N = \lfloor 1024 \cdot 3{,}176/5 \rfloor = 650$. Bij 14,6 V: $U = 3{,}623\,\text{V}$, $N = 741$.
3. (b) De ingang mag 5,00 V krijgen: $U_{\text{bat}} = 5{,}00 \cdot \dfrac{133}{33} = 20{,}2\,\text{V}$.
4. (c) $I = \dfrac{14{,}6}{133\,000} = 0{,}110\,\text{mA}$; over een dag $2{,}6\,\text{mAh}$, tegenover 120 Ah in de batterij.
5. (d) Eén stap is 4,88 mV op A0, dus $4{,}88\,\text{mV} \cdot \dfrac{133}{33} = 19{,}7\,\text{mV}$ batterijspanning.

ANTWOORD

12,8 V geeft 3,18 V (N = 650), 14,6 V geeft 3,62 V (N = 741). Je kan meten tot ongeveer 20 V, de deler kost 0,11 mA, en de meting verspringt per 20 mV batterijspanning.

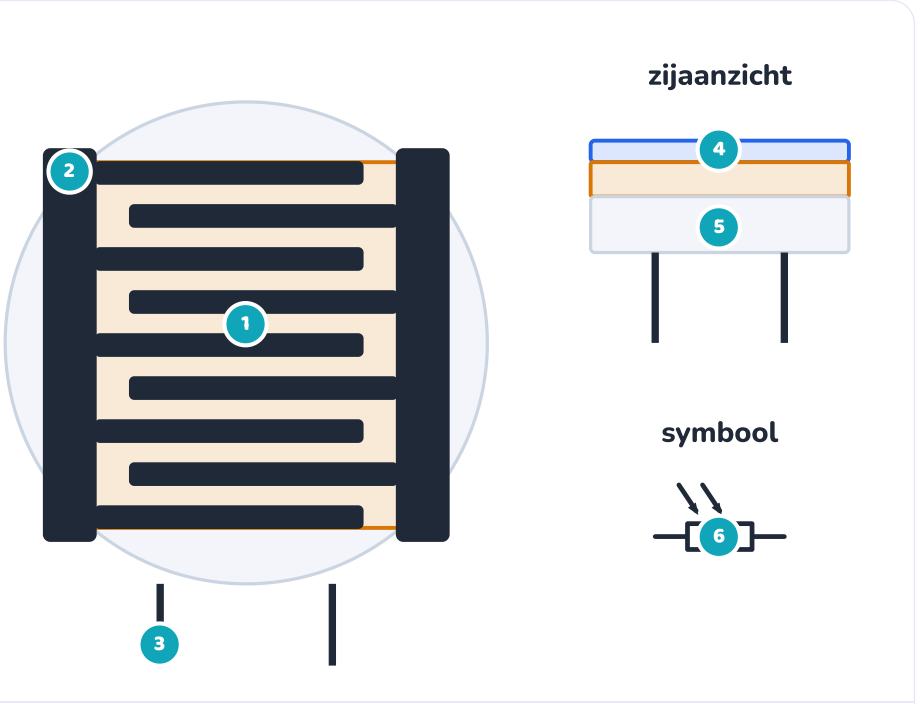
Klopt dit? Waarom zulke grote weerstanden? Met 1 kΩ en 330 Ω zou de deler 11 mA trekken, dag en nacht: 260 mAh per dag, voor niets. Maar te groot mag ook niet: de ADC wil een bron van hoogstens ongeveer 10 kΩ (verder in dit hoofdstuk). Deze deler zit daarboven, daarom komt er een condensator van 100 nF op A0 (hoofdstuk 24).

ONTLEED · 22.5

De LDR

5/20

Een LDR (*light dependent resistor*, fotoweerstand) is een keramisch schijfje met een laagje cadmiumsulfide (CdS). Het schijfje van de GL5528 is ongeveer 5 mm breed.



Links het bovenaanzicht: twee metalen kammen met daartussen een kronkelend spoor CdS.

1 CdS-spoor
Het oranje spoor tussen de kammen. In het donker zijn er in cadmiumsulfide bijna geen vrije ladingen: grote weerstand. Licht maakt elektronen vrij, en hoe meer licht, hoe meer (hoofdstuk 23).

2 Elektroden (kammen)
Twee in elkaar grijpende metalen kammen. Zo is het spoor kort en heel breed: dat geeft een kleine weerstand en een grote gevoelige oppervlakte.

3 Aansluitdraden
Twee draadjes, geen plus of min: een LDR is een gewone weerstand en werkt in beide richtingen.

4 Doorzichtige laklaag
Epoxy beschermt het CdS tegen vocht. De gevoeligheid is het grootst voor groen licht (540 nm), net als bij het menselijk oog.

5 Keramische drager
Isoleert, en voert de beetje warmte af: volgens de datasheet mag de LDR hoogstens 100 mW opnemen.

6 Symbool
Een weerstand met twee pijlen die naar binnen wijzen: er valt licht op.

De karakteristiek van een LDR

Hoe meer licht, hoe kleiner de weerstand, maar niet evenredig: tien keer meer licht maakt de weerstand ongeveer vijf keer kleiner. Zet je zowel de verlichtingssterkte E (in lux) als de weerstand op een logaritmische schaal (hoofdstuk 5), dan krijg je een **rechte**. Een rechte in een log-loggrafiek is een machtsfunctie:

MODEL VAN EEN LDR

$$R = R_{10} \cdot \left(\frac{E}{10\,\text{lx}} \right)^{-\gamma}$$

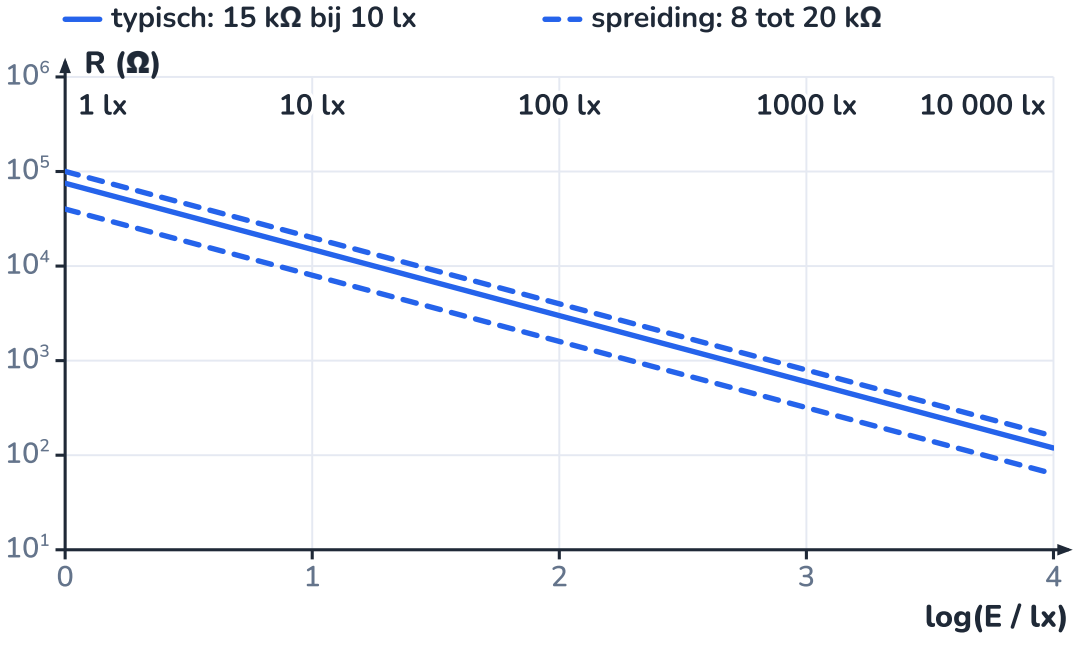
R_{10}: weerstand bij 10 lux; γ (gamma): de helling in de log-loggrafiek. Voor de GL5528: $R_{10} = 8$ tot $20\,\text{k}\Omega$, $\gamma = 0{,}7$; donker minstens 1 MΩ.

Met $R_{10} = 15\,\text{k}\Omega$: bij 100 lx is $R = 15 \cdot 10^{-0{,}7} = 2{,}99\,\text{k}\Omega$, bij 1000 lx $597\,\Omega$. Het model geldt goed tussen ongeveer 1 en 1000 lux; daarbuiten buigt de echte kromme af.

Een LDR is een goedkope, robuuste en gevoelige sensor, maar geen precisie-instrument. De **spreiding** is groot: R_{10} ligt ergens tussen 8 en 20 kΩ, dus zonder ijking is elke lux-waarde een schatting. Hij is **traag**: na een lichtflits duurt het tientallen milliseconden, van licht naar donker zelfs langer, voor de weerstand stabiel is. En hij is wat **temperatuurgevoelig**. Om te beslissen of het 'licht genoeg' is, is dat allemaal geen probleem.

Cadmium

Cadmiumsulfide is giftig. De Europese RoHS-regels beperken cadmium in elektrische toestellen, zodat LDR's in nieuwe producten zelden nog gebruikt worden: daar zitten fotodiodes, fototransistoren of digitale lichtsensoren (bv. de BH1750, die lux rechtstreeks via I²C doorgeeft, hoofdstuk 30). In de klas en voor een prototype is een LDR prima; een kapotte LDR hoort bij het elektronicaschroot.



In een log-loggrafiek is de karakteristiek een rechte met helling −0,7. Bij de stippellijnen: de spreiding volgens de datasheet.

De kronkelende lijn tussen de twee kammen is het lichtgevoelige cadmiumsulfide.

UITLEG · 22.7

Welke vaste weerstand kies je?

7/20

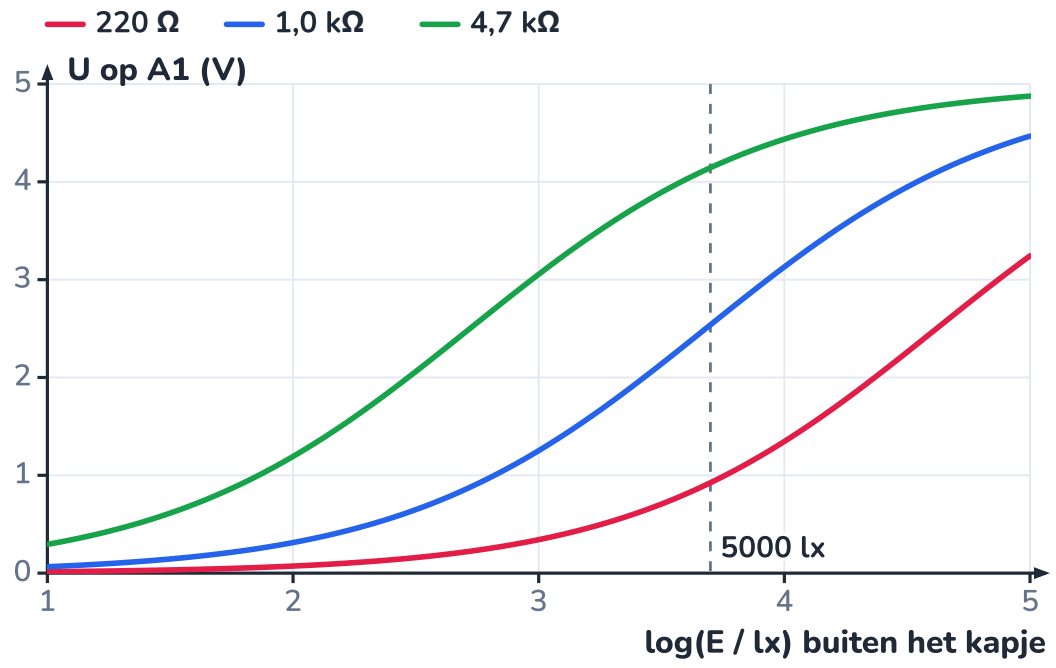
Zet de LDR bovenaan (tussen 5 V en de ingang) en een vaste weerstand R_v onderaan. Dan is $U = 5\,\text{V} \cdot \dfrac{R_\text{v}}{R_\text{v} + R_\text{LDR}}$: meer licht, kleinere R_LDR, **hogere** spanning. Zet je ze andersom, dan daalt de spanning met het licht. Beide werken; kies wat in je programma het logischst leest.

De keuze van R_v bepaalt **waar** de sensor gevoelig is. De grafiek toont het voor de klimaatmodule: met 220 Ω stijgt de spanning pas bij veel licht, met 4,7 kΩ is ze bij veel licht al bijna 5 V en verandert ze nauwelijks nog. Met 1,0 kΩ ligt het steilste stuk precies rond de 5000 lux waar de beslissing valt. Twee vuistregels:

- Wil je één waarde zo nauwkeurig mogelijk herkennen (een drempel), kies dan R_v gelijk aan de weerstand van de sensor bij die waarde. Dan staat er 2,5 V op de ingang, en daar is de deler het gevoeligst.
- Wil je een heel gebied bestrijken, van R_min tot R_max, kies dan $R_\text{v} = \sqrt{R_\text{min} \cdot R_\text{max}}$: dat geeft het grootste verschil in spanning tussen beide uiteinden (opdracht 22.15 bewijst het).

Het kapje

Op een zonnige dag valt er in de kas tot 100 000 lux. Daar zit de LDR ver buiten zijn model. Een melkwit kapje over de sensor verspreidt het licht (de sensor 'ziet' de hele hemel, niet één heldere wolk) en laat maar ongeveer 10 % door. Zo blijft de LDR tussen 10 en 10 000 lux. Welk percentage het kapje echt doorlaat, hoef je niet te weten: dat haal je eruit bij het ijken.



Met 1,0 kΩ ligt het steilste stuk bij 5000 lux daglicht, waar de lampen aan of uit moeten.

UITGEWERKT VOORBEELD · 22.8

Habitat: een lichtsensor voor de kasverlichting

8/20

De groeilampen van de klimaatmodule moeten bijspringen als er te weinig daglicht is. Eén lamp geeft het kweekbed ongeveer 100 µmol lichtdeeltjes per m² per seconde (hoofdstuk 20); daglicht van 5000 lux komt daarmee ongeveer overeen (zonlicht: ±54 lux per µmol/(m²·s)). Ontwerp de sensor: LDR (GL5528, $R_{10} = 15\,\text{k}\Omega$, $\gamma = 0{,}7$) onder een kapje dat 10 % doorlaat, met een vaste weerstand naar GND. (a) Kies R_v. (b) Welke ADC-waarde hoort bij de drempel van 5000 lux? (c) Hoe groot is de sensor bij volle zon (100 000 lux), en hoeveel vermogen neemt de LDR dan op?

GEGEVEN

- drempel: 5000 lx daglicht; kapje: 10 % doorlaat
- $R = 15\,\text{k}\Omega \cdot (E/10\,\text{lx})^{-0{,}7}$
- LDR tussen 5 V en A1, R_v tussen A1 en GND

GEVRAAGD

- R_v
- ADC-waarde bij de drempel
- spanning en vermogen bij volle zon

OPLOSSING

1. Op de LDR valt bij de drempel $0{,}10 \cdot 5000 = 500\,\text{lx}$: $R = 15\,000 \cdot 50^{-0{,}7} = 970\,\Omega$.
2. (a) Voor de grootste gevoeligheid bij de drempel kies je $R_\text{v} \approx R_\text{LDR}$: de E12-waarde 1,0 kΩ.
3. (b) $U = 5 \cdot \dfrac{1000}{1000 + 970} = 2{,}538\,\text{V}$ en $N = \lfloor 1024 \cdot 2{,}538/5 \rfloor = 519$.
4. (c) Volle zon: 10 000 lx op de LDR, $R = 119\,\Omega$; $U = 4{,}47\,\text{V}$ ($N = 914$). Stroom $I = 5/(1000 + 119) = 4{,}47\,\text{mA}$; vermogen in de LDR $RI^2 = 2{,}4\,\text{mW}$.

ANTWOORD

$R_\text{v} = 1{,}0\,\text{k}\Omega$; de drempel ligt bij ADC-waarde 519 (2,54 V). Bij volle zon geeft de sensor 4,47 V, en de LDR neemt 2,4 mW op, ver onder zijn 100 mW.

Klopt dit? Tussen 4500 en 5500 lux verandert de ADC-waarde van 500 naar 536: meer dan dertig stappen voor 20 % licht. Gevoelig genoeg; de spreiding van de LDR en het kapje is veel groter, en die haal je weg door te ijken. Het hele bereik staat in de tabel hieronder.

DAGLICHT (LX)	OP DE LDR (LX)	R_{LDR} (Ω)	U OP A1 (V)	ADC
1000	100	2993	1,25	256
2000	200	1842	1,76	360
5000	500	970	2,54	519
10 000	1000	597	3,13	641
20 000	2000	368	3,66	748
100 000	10 000	119	4,47	914

UITLEG · 22.9

De NTC: weerstand die daalt als het warmer wordt

9/20

Een NTC (*negative temperature coefficient*) is een bolletje of schijfje van metaaloxiden, een halfgeleider zoals bij Faraday: hoe warmer, hoe meer vrije ladingen, hoe kleiner de weerstand. De verandering is groot, ongeveer 4 % per graad rond kamertemperatuur, en ze volgt goed een exponentiële functie (hoofdstuk 5):

B-MODEL VAN EEN NTC

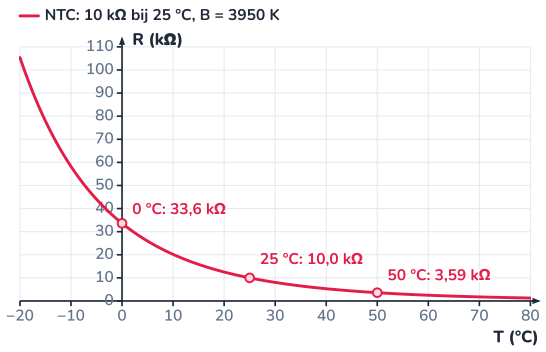
$$R(T) = R_{25} \cdot \mathrm{e}^{B\left(\frac{1}{T} - \frac{1}{T_{25}}\right)}$$

T in kelvin ($T = \theta + 273{,}15$); $T_{25} = 298{,}15\,\mathrm{K}$; R_{25}: weerstand bij 25 °C; B: de B-waarde in kelvin. De gangbare NTC van 10 kΩ heeft $B = 3950\,\mathrm{K}$.

Met $R_{25} = 10\,\mathrm{k\Omega}$ en $B = 3950\,\mathrm{K}$ is de weerstand bij 0 °C 33,6 kΩ, bij 50 °C 3,59 kΩ: een factor negen op vijftig graden. De tabel geeft ook meteen de spanning en de ADC-waarde in de deler met 10 kΩ, verder in dit hoofdstuk.

T (°C)	0	5	10	15	20	25	30	35	40	45	50
R (kΩ)	33,6	25,9	20,2	15,8	12,5	10,0	8,04	6,51	5,30	4,35	3,59
U op A0 (V)	1,15	1,39	1,66	1,94	2,22	2,50	2,77	3,03	3,27	3,48	3,68
ADC-waarde	234	285	339	396	454	512	567	620	669	713	753

Het B-model is nauwkeurig tot op enkele tienden van een graad tussen ongeveer 0 en 50 °C. Voor een groter gebied gebruiken fabrikanten een model met drie constanten, de vergelijking van Steinhart-Hart, of geven ze een tabel. De twee constanten R_{25} en B hebben elk een tolerantie (vaak 1 %): ook een NTC moet je ijken als je beter dan een halve graad wil meten.



Exact volgens het B-model: van 105 kΩ bij −20 °C tot 1,3 kΩ bij 80 °C.

In het roestvaststalen buisje zit een NTC van 10 kΩ, waterdicht ingegoten.

UITLEG · 22.10

Van ADC-waarde naar temperatuur

10/20

Zet de NTC bovenaan en een vaste weerstand van 10 kΩ onderaan. Warmer geeft een kleinere R_{NTC} en dus een **hogere** spanning. Omdat de deler en de ADC dezelfde 5 V gebruiken, valt die spanning weg, en haal je de weerstand rechtstreeks uit het getal N:

WEERSTAND UIT DE ADC-WAARDE (SENSOR BOVEN, VASTE WEERSTAND ONDER)

$$\frac{N}{1024} = \frac{R_{\text{v}}}{R_{\text{v}} + R_{\text{NTC}}} \quad \Longrightarrow \quad R_{\text{NTC}} = R_{\text{v}} \cdot \frac{1024 - N}{N}$$

Daarna keer je het B-model om (neem de natuurlijke logaritme van beide kanten):

TEMPERATUUR UIT DE WEERSTAND

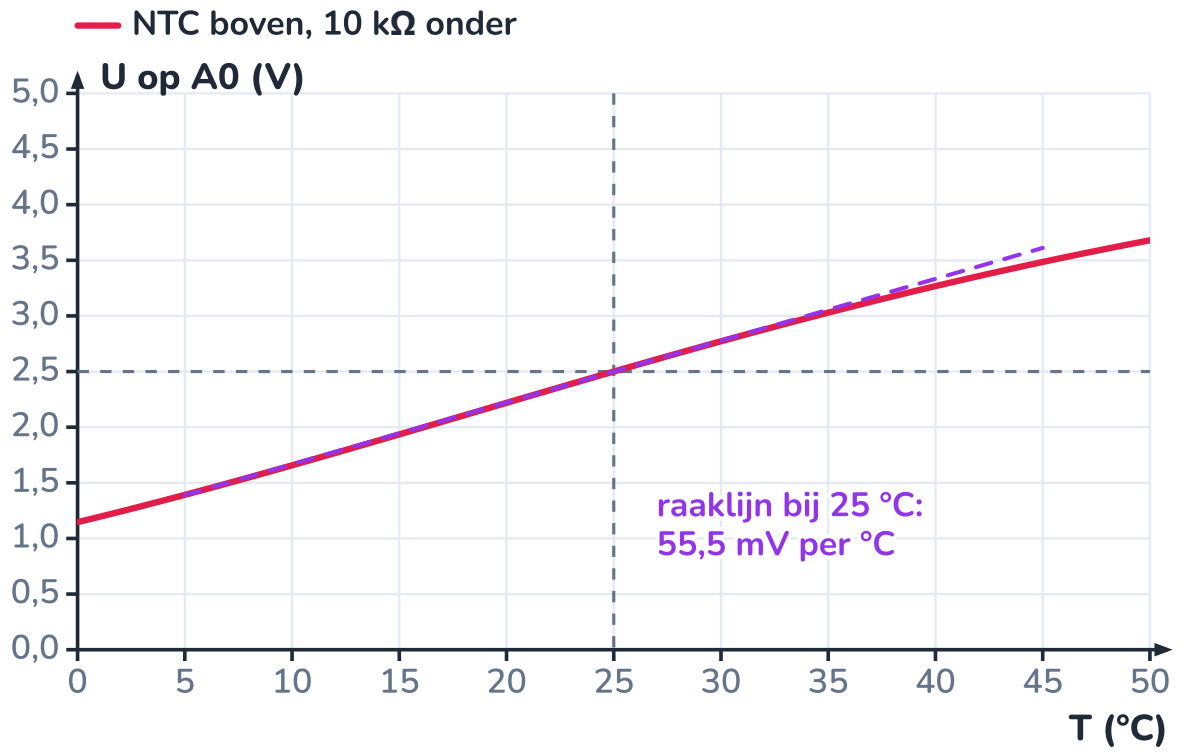
$$\frac{1}{T} = \frac{1}{T_{25}} + \frac{1}{B} \ln \frac{R_{\text{NTC}}}{R_{25}}$$

T in kelvin; trek er 273,15 van af voor graden Celsius.

Hoe fijn meet dit? De grafiek toont de spanning in functie van de temperatuur. Rond 25 °C, waar $R_{\text{NTC}} = R_{\text{v}}$, is ze het steilst: $55{,}5\,\text{mV}$ per graad. Eén ADC-stap van 4,88 mV is daar dus $0{,}088\,^{\circ}\text{C}$. Tussen 10 en 40 °C loopt de kromme bijna recht; daarbuiten wordt de sensor minder gevoelig.

Fijn meten is nauwkeurig meten

Een stap van 0,09 °C betekent niet dat de meting tot op 0,09 °C juist is. Resolutie (hoe fijn) en nauwkeurigheid (hoe juist) zijn verschillende dingen (hoofdstuk 1). Met een NTC van 1 % tolerantie en een vaste weerstand van 1 % kan je zonder ijking tot ongeveer een halve graad naast de echte temperatuur zitten, hoe fijn de ADC ook meet.



Rond 25 °C is de spanning bijna recht evenredig met de temperatuur; de stippellijn is de raaklijn.

UITGEWERKT VOORBEELD · 22.11

Welke temperatuur heeft het reservoir?

11/20

De NTC van het reservoir (10 kΩ, $B = 3950\,\mathrm{K}$) zit bovenaan in een deler met 10 kΩ, op A0. De microcontroller leest $N = 480$. Welke temperatuur heeft het water? Hoeveel verschilt de temperatuur als het getal één stap hoger is?

GEGEVEN

- $N = 480$; $R_v = 10{,}0\,\mathrm{k\Omega}$
- $R_{25} = 10{,}0\,\mathrm{k\Omega}$, $B = 3950\,\mathrm{K}$

GEVRAAGD

- R_{NTC} en T
- ΔT voor één stap

OPLOSSING

1. Spanning (ter controle): $U = 480 \cdot 5/1024 = 2{,}344\,\mathrm{V}$.
2. Weerstand: $R_{\mathrm{NTC}} = 10\,000 \cdot \dfrac{1024 - 480}{480} = 11\,333\,\Omega$.
3. $\ln(11\,333/10\,000) = 0{,}1252$, gedeeld door 3950: $3{,}169 \cdot 10^{-5}\,\mathrm{K^{-1}}$.
4. $\dfrac{1}{T} = \dfrac{1}{298{,}15} + 3{,}169 \cdot 10^{-5} = 3{,}38570 \cdot 10^{-3}\,\mathrm{K^{-1}}$, dus $T = 295{,}36\,\mathrm{K} = 22{,}2\,^{\circ}\mathrm{C}$.
5. Met $N = 481$: $R = 11\,289\,\Omega$ en $T = 22{,}30\,^{\circ}\mathrm{C}$.

ANTWOORD

Het water is 22,2 °C. Eén ADC-stap komt hier overeen met 0,087 °C.

Klopt dit? Het getal 480 ligt iets onder 512 (2,5 V): de NTC is iets groter dan 10 kΩ, dus iets kouder dan 25 °C. Klopt. De tabel van het vorige blad geeft bij 20 °C N = 454 en bij 25 °C N = 512: 480 ligt ertussen.

CODE · 22.12

C++: licht en temperatuur op de microcontroller

12/20

Dit programma leest beide sensoren, middelt telkens zestien metingen tegen ruis, rekent de getallen om met de formules van dit hoofdstuk en beslist of de lampen nodig zijn. Het vangt ook twee fouten op: een losse draad aan de NTC (dan trekt de vaste weerstand A0 naar 0) en een pikdonkere LDR (dan is $N = 0$ en zou je door nul delen).

C++ (Arduino) **sensoren.ino** 48 regels

```cpp
// sensoren.ino - NTC in het reservoir (A0) en lichtsensor van de kasverlichting (A1), Arduino Mega 2560 op 5 V
const float R_VAST_NTC = 10000.0;   // ohm, tussen A0 en GND (NTC tussen 5 V en A0)
const float R25 = 10000.0;          // ohm bij 25 graden C
const float B = 3950.0;             // K
const float T25 = 298.15;           // K
const float R_VAST_LDR = 1000.0;    // ohm, tussen A1 en GND (LDR tussen 5 V en A1)
const float R10 = 15000.0;          // ohm bij 10 lux op de LDR
const float GAMMA = 0.7;
const float KAPJE = 0.10;           // het melkwitte kapje laat 10 % van het licht door
const float DREMPEL_LUX = 5000.0;   // onder dit daglicht moeten de lampen bij

int gemiddeld(int pin) {            // 16 metingen middelen tegen ruis
  long som = 0;
  for (int i = 0; i < 16; i++) som += analogRead(pin);
  return som / 16;
}

float temperatuur() {
  int n = gemiddeld(A0);
  if (n < 5 || n > 1018) return NAN;              // draad los of kortsluiting
  float r = R_VAST_NTC * (1024.0 - n) / n;        // spanningsdeler omgekeerd
  float invT = 1.0 / T25 + log(r / R25) / B;      // B-model: log() is de natuurlijke logaritme
  return 1.0 / invT - 273.15;
```

```
}

float daglicht() {
  int n = gemiddeld(A1);
  if (n < 1) n = 1;                                  // pikdonker: vermijd delen door nul
  float r = R_VAST_LDR * (1024.0 - n) / n;
  float luxOpLdr = 10.0 * pow(r / R10, -1.0 / GAMMA); // R = R10 * (E / 10)^-gamma, omgekeerd
  return luxOpLdr / KAPJE;                           // het licht buiten het kapje
}

void setup() {
  Serial.begin(9600);
}

void loop() {
  float t = temperatuur();
  float e = daglicht();
  Serial.print("reservoir: ");
  Serial.print(t, 1);
  Serial.print(" C   daglicht: ");
  Serial.print(e, 0);
  Serial.print(" lx   lampen: ");
  Serial.println(e < DREMPEL_LUX ? "nodig" : "niet nodig");
  delay(2000);
}
```

Op de seriële monitor verschijnt elke twee seconden een regel zoals `reservoir: 22,2 C daglicht: 3981 lx lampen: nodig`. Twee dingen ontbreken nog. Rond 5000 lux zou de beslissing bij elke wolk heen en weer springen: een regeling met twee drempels (hysterese) lost dat op (opdracht 22.10 en hoofdstuk 31). En `delay()` legt het hele programma stil; in hoofdstuk 29 vervang je het door `millis()`.

UITLEG · 22.13

De potentiometer

13/20

Een potentiometer is een weerstand met een derde aansluiting: een contact, de **loper**, dat over de weerstand schuift als je aan de knop draait. Hang je de uiteinden aan 5 V en GND, dan is de loper de aftakking van een spanningsdeler die je met de hand instelt. Staat de loper op een fractie α van onderen, dan is $R_2 = \alpha R$ en $R_1 = (1-\alpha)R$:

ONBELASTE POTENTIOMETER (LINEAIR VERLOOP)

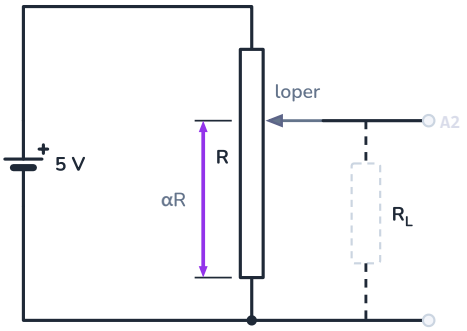
$$U_{\text{uit}} = \alpha \cdot U_{\text{in}}$$

In de klimaatmodule zit er een op het bedieningspaneel, aan A2: een **instelknop** voor de lichtdrempel. Het programma zet de ADC-waarde van 0 tot 1023 om naar een drempel tussen bijvoorbeeld 1000 en 20 000 lux. Door de potmeter van 10 kΩ loopt $5/10\,000 = 0{,}5\,\text{mA}$, in elke stand.

Gebruik je alleen de loper en één uiteinde, dan is de potentiometer een **regelbare weerstand** (*rheostaat*): van 0 tot 10 kΩ. Let dan op: in de stand 'nul' kan er een grote stroom lopen. Zet er een vaste weerstand in serie mee.

Lineair of logaritmisch

Potentiometers voor volumeknoppen hebben een **logaritmisch** verloop: bij de eerste helft van de draai verandert de weerstand weinig, daarna veel, omdat ons oor zo werkt. Voor een meetknop wil je een **lineair** verloop. De lettercode verschilt per fabrikant: kijk in de datasheet.



De loper deelt de weerstand R in αR (onder) en (1 – α)R (boven). In stippellijn: een belasting R_L.

Een draaiknop op het bedieningspaneel: achter de knop zit een potentiometer van 10 kΩ.

UITLEG · 22.14

Belasting: de deler verandert als je er iets aan hangt

14/20

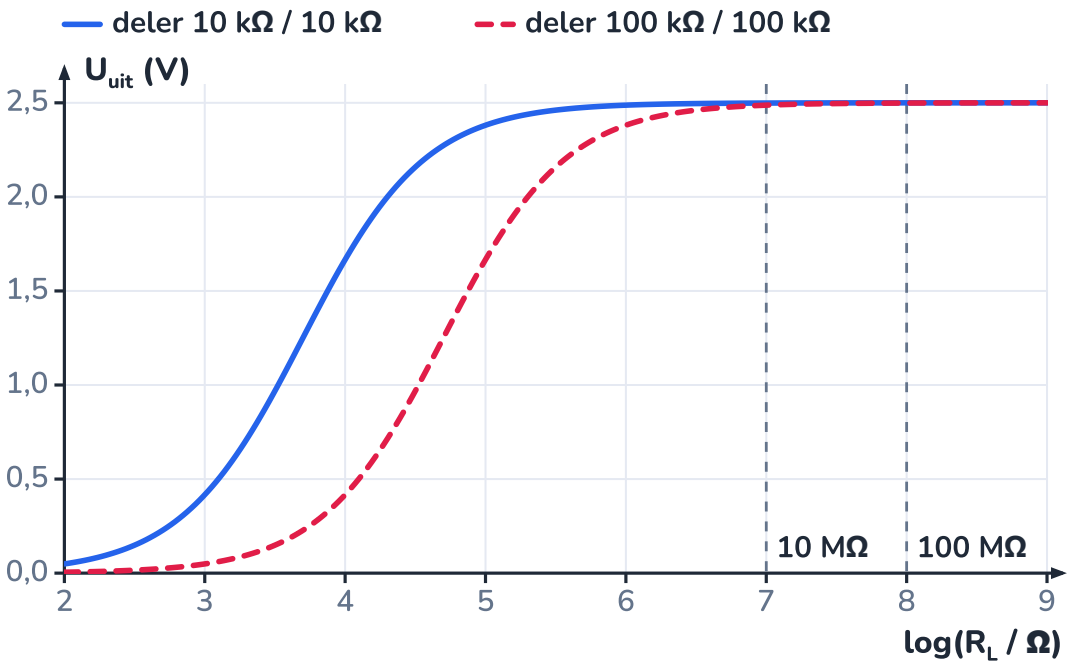
De formule van de spanningsdeler geldt alleen als er geen stroom uit de aftakking loopt. Hang je er een **belasting** R_L aan (een meter, een ingang, een led), dan staat R_L parallel met R_2, en die combinatie is kleiner dan R_2 (hoofdstuk 21). De uitgang zakt:

BELASTE SPANNINGSDELER

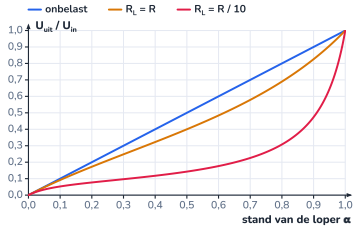
$$U_{\text{uit}} = U_{\text{in}} \cdot \frac{R_2 \parallel R_L}{R_1 + R_2 \parallel R_L} \qquad R_2 \parallel R_L = \frac{R_2 R_L}{R_2 + R_L}$$

Een handige manier om ernaar te kijken: een deler gedraagt zich als een bron met bronspanning U_{open} (de onbelaste uitgang) en een inwendige weerstand $R_{\text{uit}} = R_1 \parallel R_2$, zoals de batterij in hoofdstuk 21. Voor 10 kΩ met 10 kΩ is $R_{\text{uit}} = 5\,\text{k}\Omega$. Een belasting die veel groter is dan R_{uit}, verandert bijna niets: met $R_L = 100 \cdot R_{\text{uit}}$ zakt de uitgang ongeveer 1 %.

Waarom een ingang hoogohmig moet zijn. Een meetingang mag het signaal niet veranderen dat hij meet. Een analoge ingang van de ATmega328P gedraagt zich als een weerstand van ongeveer 100 MΩ: voor een deler van 10 kΩ is de fout 0,005 %, onmeetbaar klein. Een digitale multimeter heeft ongeveer 10 MΩ, een goedkope analoge meter soms maar 20 kΩ per volt. Er is één addertje: bij elke meting laadt de ADC een klein condensatortje (ongeveer 14 pF) op. De datasheet vraagt daarom een bron van hoogstens ongeveer 10 kΩ. Heb je een grotere R_{uit}, zoals de batterijdeler met 24,8 kΩ, dan zet je een condensator van 100 nF tussen de ingang en GND die dat opladen levert (hoofdstuk 24).



Links op de as een zware belasting, rechts een lichte (4 is 10 kΩ; bij 10 MΩ een multimeter, bij 100 MΩ de ADC-ingang). De deler van 100 kΩ zakt al waar die van 10 kΩ nog niets merkt.



Een potentiometer met een zware belasting verliest zijn rechte lijn: de uitgang zakt het meest in het midden.

UITGEWERKT VOORBEELD · 22.15

Een meter die de meting verandert

15/20

(a) Je controleert de batterijdeler (100 kΩ en 33 kΩ, batterij 13,2 V) door op A0 te meten, eerst met een digitale multimeter van 10 MΩ, dan met een oude meter van 1,0 MΩ. Wat lees je af, en hoeveel procent is dat te laag? (b) Iemand gebruikt de instelknop (10 kΩ, loper in het midden) om rechtstreeks een module te sturen met een ingangsweerstand van 10 kΩ. Welke spanning krijgt die module?

GEGEVEN

- (a) $U_{\text{bat}} = 13{,}2\ \text{V}$; $R_1 = 100\ \text{k}\Omega$, $R_2 = 33\ \text{k}\Omega$; $R_L = 10\ \text{M}\Omega$ of $1{,}0\ \text{M}\Omega$
- (b) potentiometer 10 kΩ op 5 V, $\alpha = 0{,}5$; $R_L = 10\ \text{k}\Omega$

GEVRAAGD

- afgelezen spanningen en de fout
- spanning op de module

OPLOSSING

1. (a) Onbelast: $U = 13{,}2 \cdot 33/133 = 3{,}275\ \text{V}$. Uitgangsweerstand $R_{\text{uit}} = 100 \parallel 33 = 24{,}8\ \text{k}\Omega$.
2. Met 10 MΩ: $U = 3{,}275 \cdot \dfrac{10\,000}{10\,000 + 24{,}8} = 3{,}267\ \text{V}$, 0,25 % te laag.

3 Met 1,0 MΩ: $U = 3{,}275 \cdot \dfrac{1000}{1000 + 24{,}8} = 3{,}196\ \text{V}$, 2,4 % te laag.

4 (b) Onderste helft: $5\,\text{k}\Omega \parallel 10\,\text{k}\Omega = 3{,}33\,\text{k}\Omega$; $U = 5 \cdot \dfrac{3{,}33}{5 + 3{,}33} = 2{,}0\ \text{V}$ in plaats van 2,5 V.

ANTWOORD

(a) 3,267 V met de goede meter, 3,196 V met de oude: 0,25 % en 2,4 % te laag. (b) 2,0 V in plaats van 2,5 V: 20 % te weinig.

Klopt dit? Omgerekend naar de batterij leest de oude meter 12,88 V in plaats van 13,2 V: een laadtoestand die er helemaal naast zit. De microcontroller zelf (±100 MΩ) meet 13,197 V. Vuistregel: de belasting minstens honderd keer groter dan de uitgangsweerstand. Moet een deler toch een zware belasting sturen, gebruik dan een buffer (een opamp, hoofdstuk E22).

PROEF · 22.16

Een NTC ijken in water

16/20

ONDERZOEKSVRAAG

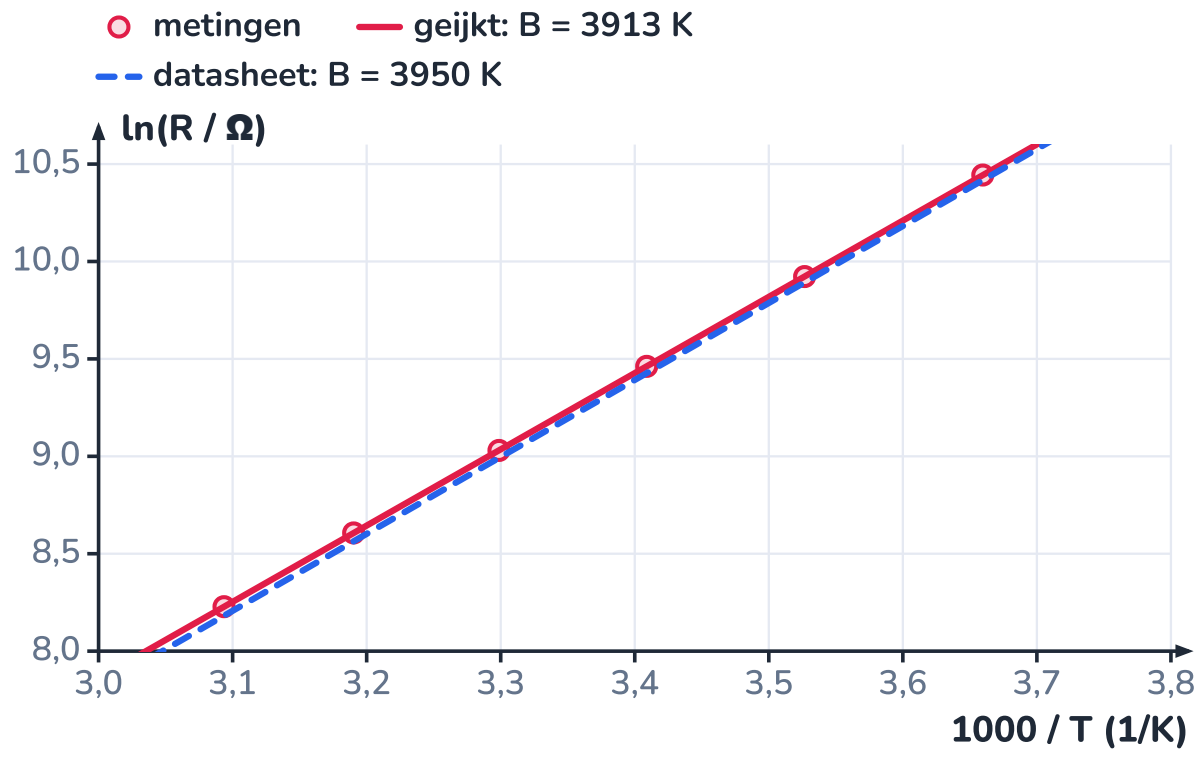
Hoe goed meet een NTC met de waarden uit de datasheet, en hoe maak je hem beter?

MATERIAAL

- NTC 10 kΩ ($B = 3950\,\text{K}$) in een waterdicht buisje, weerstand 10 kΩ, microcontroller
- referentiethermometer (±0,1 °C)
- bekers: ijswater, water van 10, 20, 30, 40 en 50 °C
- roerstaafje, computer

Veiligheid

Water van 50 °C is warm maar niet gevaarlijk; heter water gebruik je niet. Houd de microcontroller en de USB-kabel droog.



Gelineariseerd liggen de metingen op een rechte; de geijkte rechte ligt iets lager en minder steil dan die van de datasheet.

WERKWIJZE

1. Bouw de deler: NTC tussen 5 V en A0, 10 kΩ tussen A0 en GND. Laat de microcontroller elke seconde N doorsturen (gemiddelde van 16 metingen).
2. Hang de NTC en de referentiethermometer vlak naast elkaar in het ijswater. Roer, wacht tot beide stabiel zijn (minstens een minuut) en noteer T_{ref} en N.
3. Herhaal voor de andere bekers.
4. Reken elke N om naar R en naar T met de waarden van de datasheet. Vergelijk met T_{ref}.
5. Zet $\ln R$ uit tegen $1/T$ (in kelvin) en bepaal de best passende rechte.

WAARNEMING

T_{ref} (°C)	ADC-waarde	R (Ω)	T volgens datasheet (°C)	fout (°C)
0,1	231	34 329	−0,39	−0,49
10,4	337	20 386	9,79	−0,61
20,2	448	12 857	19,45	−0,75
30,0	558	8351	29,11	−0,89
40,3	662	5468	39,23	−1,07
50,1	745	3745	48,87	−1,23

Voorbeeldmeting. De NTC meet overal te laag, van een halve graad bij 0 °C tot meer dan een graad bij 50 °C.

VERKLARING

De fout is systematisch (hoofdstuk 1): altijd dezelfde kant op, en groeiend met de temperatuur. Ze komt uit de toleranties van R_{25}, van B en van de vaste weerstand. Het B-model zelf is niet het probleem, alleen zijn constanten. Neem de logaritme van het model: $\ln R = \ln R_{25} + B\,(1/T - 1/T_{25})$. Tegen $1/T$ uitgezet is dat een **rechte** met helling B (hoofdstuk 3: lineariseren). De best passende rechte door de metingen geeft $B = 3913\,\text{K}$ en $R_{25} = 10{,}37\,\text{k}\Omega$. Met die constanten valt elke meting binnen 0,02 °C van de referentie; wat overblijft, is de onzekerheid van de referentie zelf.

CODE · 22.17

Python: de ijking uitrekenen

17/20

Het programma rekent de proef na: van ADC-waarde naar weerstand, de rechte door $(1/T, \ln R)$ met `np.polyfit` (hoofdstuk 3 en 4), en daarna de temperaturen met de datasheet en met de geijkte constanten naast elkaar. De twee getallen die eruit komen, zet je in de C++-code in de plaats van 10000 en 3950.

Python **kalibratie.py** 24 regels

```python
# kalibratie.py - een NTC ijken: B en R25 uit metingen met een referentiethermometer
import math
import numpy as np

R_VAST = 10000.0                                  # vaste weerstand (ohm, nominaal)
T_ref = [0.1, 10.4, 20.2, 30.0, 40.3, 50.1]       # referentie (graden C)
N = [231, 337, 448, 558, 662, 745]                # ADC-waarden van de NTC-deler

R = [R_VAST * (1024 - n) / n for n in N]          # spanningsdeler omgekeerd
x = [1 / (t + 273.15) for t in T_ref]             # 1/T in 1/K
y = [math.log(r) for r in R]                      # ln R

B, b = np.polyfit(x, y, 1)                        # ln R = B * (1/T) + b is een rechte
R25 = math.exp(b + B / 298.15)
print(f"geijkt: B = {B:.0f} K, R25 = {R25:.0f} ohm")


def temperatuur(r, r25, b_waarde):
```

```
    return 1 / (1 / 298.15 + math.log(r / r25) / b_waarde) - 273.15


print(" T_ref    N  R (ohm)  T datasheet  T geijkt")
for t, n, r in zip(T_ref, N, R):
    print(f"{t:6.1f} {n:4d} {r:8.0f} {temperatuur(r, 10000, 3950):11.2f} {temperatuur(r, R25, B):9.2f}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
geijkt: B = 3913 K, R25 = 10375 ohm
 T_ref    N  R (ohm)  T datasheet  T geijkt
   0.1  231    34329       -0.39      0.09
  10.4  337    20386        9.79     10.41
  20.2  448    12857       19.45     20.21
  30.0  558     8351       29.11     30.01
  40.3  662     5468       39.23     40.29
  50.1  745     3745       48.87     50.09
```

Let op wat de ijking *niet* doet: ze meet niet de echte R_{25} van de NTC. De vaste weerstand wijkt ook af, en die afwijking zit mee in de geijkte R_{25}. Dat is geen fout, maar het principe van ijken: je stelt de hele meetketen bij, niet één onderdeel. Vervang je de vaste weerstand, dan ijk je opnieuw.

UITLEG · 22.18

Kalibreren: de hele meetketen

18/20

Tussen de temperatuur in het water en het getal op je scherm zitten vijf schakels, elk met een eigen foutbron. Je kan ze apart gaan uitmeten, maar dat is veel werk en je vergeet er altijd één. **Kalibreren** (ijken) vergelijkt het eindresultaat van de hele keten met een referentie die beter is dan wat je wil meten, en stelt het model bij tot ze overeenkomen. Drie manieren, van eenvoudig naar grondig:

- **Eenpuntsijking**: één referentiepunt, en je corrigeert een vaste afwijking (offset). Goed als de fout overal ongeveer even groot is, zoals bij de NTC-module in hoofdstuk 1.
- **Tweepuntsijking**: twee punten, voor een rechte (helling en offset), zoals de pH-module in hoofdstuk 15.
- **Model fitten**: meerdere punten, en de constanten van het model (hier B en R_{25}, voor de LDR R_{10} en γ) met de kleinste kwadraten bepalen. Trek het model eerst recht: $\ln R$ tegen $1/T$ voor de NTC, $\log R$ tegen $\log E$ voor de LDR.

Kies de ijkpunten in het gebied waar je gaat meten (voor het reservoir 10 tot 30 °C, niet 80 °C), en schrijf de datum van de ijking op. Sensoren verouderen, en een ijking heeft een houdbaarheid: bij de pH-module elke twee weken, bij een NTC eerder jaren.

De referentie moet beter zijn

Een ijking is nooit beter dan haar referentie. Een luxmeter-app op een smartphone kan er 20 % naast zitten; voor een drempel 'veel of weinig licht' is dat genoeg, voor een onderzoek naar de groei van sla niet. Een goede vuistregel: de referentie minstens drie keer nauwkeuriger dan wat je nodig hebt.

Vijf schakels, vijf foutbronnen. Kalibreren vergelijkt het begin en het einde van de keten in één keer.

NASLAG · 22.19

Overzicht: spanningsdeler, sensoren en ADC

19/20

spanningsdeler

$$U_{\text{uit}} = U_{\text{in}} \frac{R_2}{R_1 + R_2}$$

symbool	betekenis	eenheid
R_2	weerstand tussen uitgang en GND	Ω

uitgangsweerstand van een deler

$$R_{\text{uit}} = R_1 \parallel R_2 = \frac{R_1 R_2}{R_1 + R_2}$$

symbool	betekenis	eenheid
R_{uit}	inwendige weerstand van de deler als bron	Ω

ADC van 10 bit

$$N = \left\lfloor 1024 \frac{U}{U_{\text{ref}}} \right\rfloor$$

symbool	betekenis	eenheid
N	ADC-waarde, 0 tot 1023	–
U_{ref}	referentie (Uno: 5 V)	V

weerstand uit de ADC (sensor boven)

$$R_{\text{sensor}} = R_{\text{v}} \frac{1024 - N}{N}$$

symbool	betekenis	eenheid
R_{v}	vaste weerstand onder	Ω

LDR

$$R = R_{10} \left(\frac{E}{10\,\text{lx}} \right)^{-\gamma}$$

symbool	betekenis	eenheid
E	verlichtingssterkte	lx
γ	helling in log-log (GL5528: 0,7)	–

NTC (B-model)

$$R = R_{25}\, e^{B(1/T - 1/T_{25})} \qquad \frac{1}{T} = \frac{1}{T_{25}} + \frac{\ln(R/R$$

symbool	betekenis	eenheid
T	temperatuur	K
B	B-waarde	K

onderdeel	typische waarden	vaste weerstand
LDR GL5528	8-20 kΩ bij 10 lx, > 1 MΩ donker, γ = 0,7, hoogstens 100 mW	≈ R_{LDR} bij de drempel, of $\sqrt{R_{\min} R_{\max}}$
NTC 10 kΩ	$R_{25} = 10\,\text{k}\Omega$, $B = 3950\,\text{K}$, ±4 % per °C	10 kΩ (gevoeligst rond 25 °C)
potentiometer	1 kΩ tot 100 kΩ; lineair of logaritmisch	geen: zelf een deler
ADC Arduino Uno	10 bit, 0-1023, 4,88 mV per stap, ingang ±100 MΩ, bron liefst ≤ 10 kΩ	—

T (°C)	0	5	10	15	20	25	30	35	40	45	50
R (kΩ)	33,6	25,9	20,2	15,8	12,5	10,0	8,04	6,51	5,30	4,35	3,59
U op A0 (V)	1,15	1,39	1,66	1,94	2,22	2,50	2,77	3,03	3,27	3,48	3,68
ADC-waarde	234	285	339	396	454	512	567	620	669	713	753

HET GEHEEL · 22.20

Waar de spanningsdeler vandaan komt en waar hij heen gaat

20/20

De spanningsdeler is de eenvoudigste toepassing van Kirchhoff en tegelijk de meest gebruikte schakeling van de elektronica: in elke volumeknop, elke thermostaat, elke batterijmeter. Hij verbindt de leerlijn elektronica met onderzoeken en meten (een sensor is maar zo goed als zijn ijking) en met

programmeren (het getal moet een beslissing worden). Wiskundig oefen je hier exponentiële en machtsfuncties, logaritmen en het rechttrekken van een kromme.

F20 Serie- en parallelschakeling
In serie verdeelt de spanning zich: het idee achter de deler.

F25 Sensoren en automatisch regelen
De eerste sensoren en analoge waarden op de microcontroller.

I1 Meetfouten en nauwkeurigheid
Systematische fouten, resolutie tegenover nauwkeurigheid.

I3 Verbanden in data
Kalibreren met een rechte, en een kromme rechttrekken.

I5 Groei en verval: exponentiële modellen
Het B-model is een exponentiële functie; de LDR lees je op een logaritmische schaal.

I21 De wetten van Kirchhoff
Knooppunt- en maaswet, uitgangsweerstand als inwendige weerstand.

I23 Halfgeleiders: diode, led en transistor
Waarom een NTC en een LDR halfgeleiders zijn.

I24 Condensatoren en de RC-tijd
De condensator op een hoogohmige ADC-ingang, en ruis wegfilteren.

I25 Van analoog naar digitaal
Hoe de ADC binnenin werkt: resolutie, bereik en bemonsteren.

I31 Regelsystemen
Een drempel met hysterese, zodat de lampen niet bij elke wolk schakelen.

E22 Operationele versterkers
De brug van Wheatstone en een buffer die een deler niet belast.

Habitat
De klimaatmodule heeft nu haar lichtsensor: een LDR onder een melkwit kapje met 1,0 kΩ op A1, drempel 5000 lux daglicht (ADC-waarde 519). De NTC van het reservoir zit met 10 kΩ op A0 en is geijkt, en de batterijspanning komt via 100 kΩ / 33 kΩ met 100 nF op een derde ingang. In opdracht 22.10 krijgt de lichtsensor twee drempels, in opdracht 22.13 een plaats, een ijkplan en foutdetectie; in hoofdstuk 25 dimmen de lampen traploos met PWM.

SAMENVATTING

In het kort

- Een microcontroller meet spanning; een ADC van 10 bit geeft $N = \lfloor 1024\,U/U_{\text{ref}} \rfloor$, van 0 tot 1023, in stappen van 4,88 mV bij 5 V.
- Spanningsdeler: $U_{\text{uit}} = U_{\text{in}} \cdot R_2/(R_1 + R_2)$, onbelast. Met een sensor erin wordt een weerstand een spanning.
- Voed de deler uit dezelfde 5 V als de ADC: dan is $R_{\text{sensor}} = R_{\text{v}}(1024 - N)/N$, onafhankelijk van de voeding.
- LDR: $R = R_{10}(E/10\,\text{lx})^{-\gamma}$, een rechte in een log-loggrafiek; goedkoop, traag, grote spreiding.
- NTC: $R = R_{25}\text{e}^{B(1/T - 1/T_{25})}$; rond 25 °C met 10 kΩ ongeveer 55 mV per graad.
- Kies de vaste weerstand gelijk aan de sensorweerstand bij de belangrijkste waarde, of $\sqrt{R_{\text{min}}R_{\text{max}}}$ voor een heel gebied.
- Een belasting parallel aan R_2 laat de uitgang zakken; een deler is een bron met $R_{\text{uit}} = R_1 \parallel R_2$. Een ingang moet hoogohmig zijn (ADC ±100 MΩ), maar de bron liefst niet groter dan 10 kΩ.
- Kalibreren vergelijkt de hele meetketen met een betere referentie; trek een kromme eerst recht en bepaal de constanten van het model.

BEGRIPPEN

Wat je nu kent

spanningsdeler

Twee weerstanden in serie over een spanning; de spanning op de aftakking is $U_{\text{uit}} = U_{\text{in}} \cdot R_2/(R_1 + R_2)$.

LDR (fotoweerstand)

Weerstand van cadmiumsulfide waarvan de weerstand daalt als er meer licht op valt.

NTC (thermistor)

Halfgeleiderweerstand met een negatieve temperatuurcoëfficiënt: warmer geeft een kleinere weerstand.

B-waarde

Constante in het model $R = R_{25}\,\mathrm{e}^{B(1/T - 1/T_{25})}$ van een NTC, in kelvin; zegt hoe sterk de weerstand met de temperatuur verandert.

potentiometer

Weerstand met een verschuifbaar contact (de loper): een instelbare spanningsdeler.

ADC

Analoog-digitaalomzetter: zet een spanning om in een getal; op een Uno 10 bit, van 0 tot 1023, met 5 V als referentie.

ratiometrisch meten

De sensor en de ADC gebruiken dezelfde referentiespanning, zodat het getal alleen van de verhouding van de weerstanden afhangt.

uitgangsweerstand

De weerstand die een belasting 'ziet' als ze aan een deler hangt; voor een deler is dat $R_1 \parallel R_2$.

kalibreren (ijken)

De aanwijzing van een meetketen vergelijken met een referentie en het model bijstellen tot ze overeenkomen.

I INTERMEDIATE · DEEL I6 · ELEKTRONICA

HANDBOEK · HOOFDSTUK 23

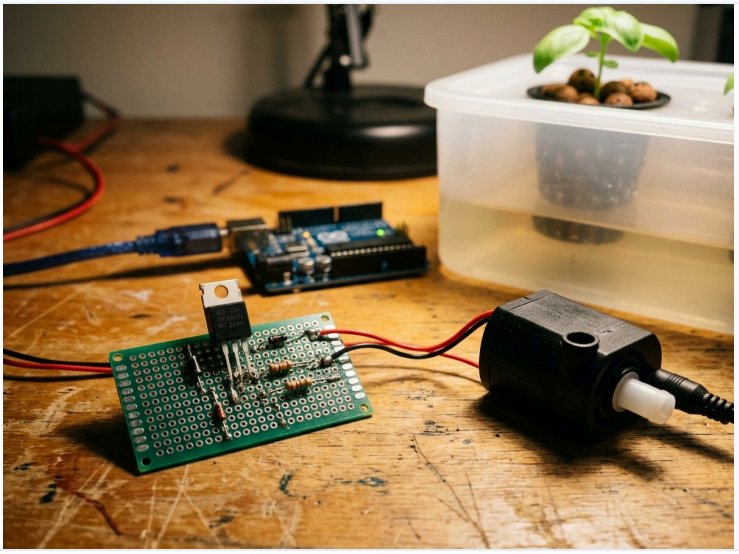
Halfgeleiders: diode, led en transistor

Van een laagje silicium naar een lampje, een gelijkrichter en een schakelaar die een pomp aanstuurt

Een microcontroller kan een led laten branden, maar geen pomp: zijn pinnen leveren hoogstens 20 mA bij 5 V, de pomp vraagt 1 A bij 12 V. De oplossing is een transistor: een halfgeleiderschakelaar die met een klein signaal een grote stroom stuurt. In dit hoofdstuk leer je wat een halfgeleider is en wat er gebeurt waar p-type en n-type silicium elkaar raken. Je meet en berekent de karakteristiek van een diode, maakt van wisselspanning gelijkspanning, rekent de weerstand voor een led uit en schakelt ten slotte de pomp van de klimaatmodule met een MOSFET, beschermd door een vrijloopdiode.

10× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 9 lestijden

L Elektriciteit en elektronica · M Materie en chemie



Drie onderdelen tussen de microcontroller en de pomp: een MOSFET, een diode en twee weerstanden.

Na dit hoofdstuk

- Je kan met het bindingsmodel van silicium uitleggen wat een halfgeleider is, en wat doteren tot n-type en p-type doet.
- Je kan uitleggen hoe een pn-overgang een sperlaag vormt en waarom een diode de stroom maar in één richting doorlaat.
- Je kan de karakteristiek van een diode lezen, meten en met het Shockley-model berekenen, en een werkpunt bepalen.
- Je kan uitleggen hoe een halvegolfgelijkrichter werkt en de uitgangsspanning berekenen.
- Je kan het verband tussen de kleur van een led en haar drempelspanning uitleggen en een voorschakelweerstand berekenen en kiezen.
- Je kan een npn-transistor en een n-MOSFET als schakelaar gebruiken en de basisweerstand, de verliezen en de juiste MOSFET kiezen.
- Je kan uitleggen waarom een spoel een spanningspiek geeft bij het uitschakelen en hoe een vrijloopdiode dat oplost.
- Je kan een 12 V-pomp veilig aansturen vanuit een microcontroller: schema, onderdelen, berekeningen en code.

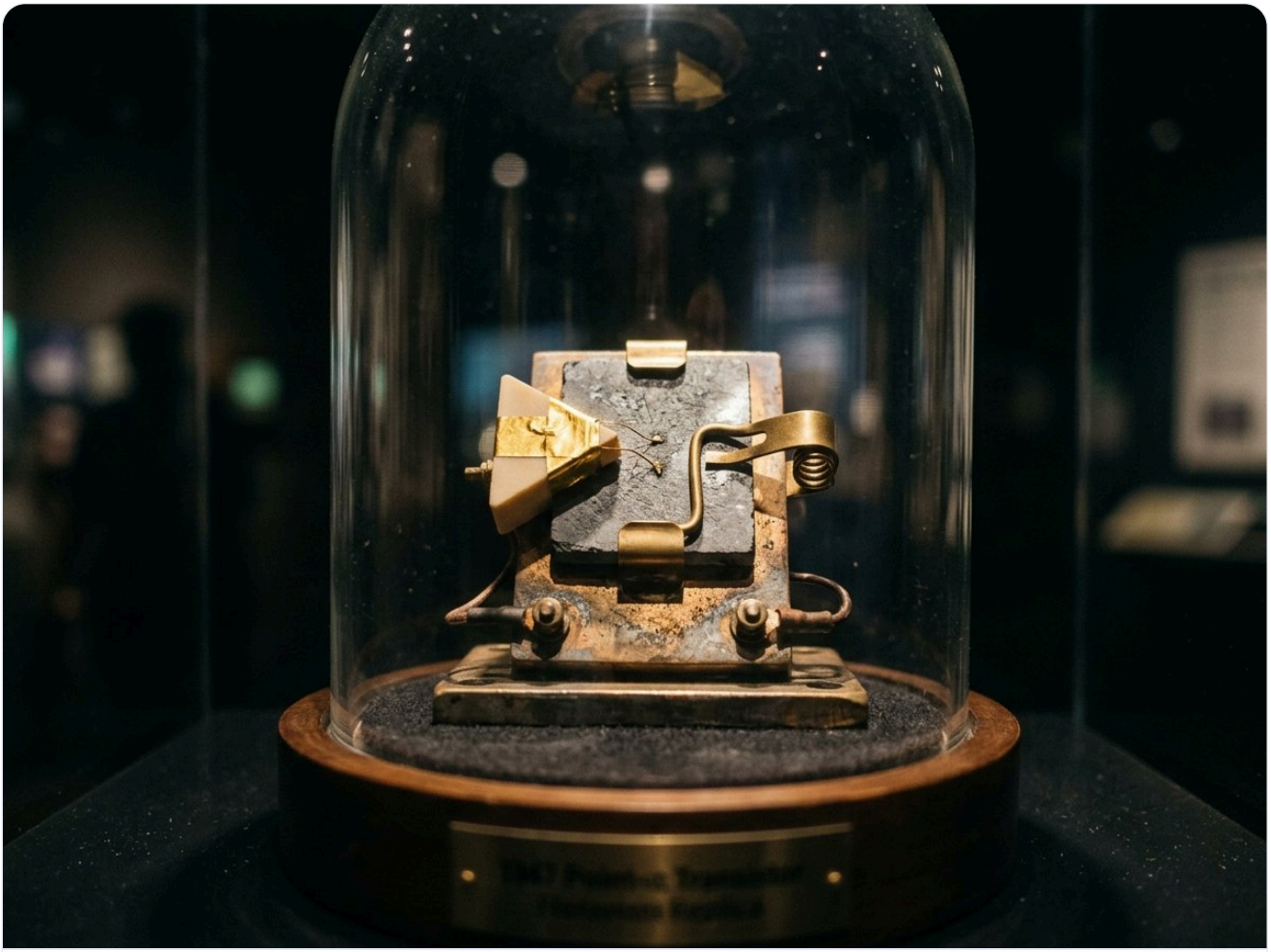
Een kerstcadeau van Bell Labs

In de jaren 1940 draaiden telefooncentrales en radio's op **elektronenbuizen**: glazen lampen met een gloeidraad, zo groot als een duim, die warm werden, veel stroom vroegen en geregeld doorbrandden. Bij Bell Labs, het onderzoekslabo van de Amerikaanse telefoonmaatschappij, zocht een ploeg een vervanger in **halfgeleiders**, stoffen als germanium en silicium die maar half geleiden.

In december 1947 drukten John Bardeen en Walter Brattain twee goudcontactjes, een fractie van een millimeter uit elkaar, op een blokje germanium. Een klein signaal op het ene contact stuurde een veel groter signaal door het andere: de eerste **transistor**. Een paar dagen voor Kerstmis toonden ze hem aan de directie. Hun collega William Shockley bedacht kort daarna een robuustere versie, de junctietransistor, gebouwd uit lagen p-type en n-type halfgeleider. De drie kregen er in 1956 de Nobelprijs natuurkunde voor. In 1959 bouwden Mohamed Atalla en Dawon Kahng, ook bij Bell Labs, de eerste **MOSFET**: de transistor die vandaag met miljarden tegelijk op elke processorchip staat (hoofdstuk 5).

Dit hoofdstuk volgt dezelfde weg: van de bindingen in een siliciumkristal naar de pn-overgang, de diode en de led, tot de transistor die de pomp van de klimaatmodule aan- en uitzet.

> **Waarom dit hoofdstuk ertoe doet**
>
> Alles wat de microcontroller van de klimaatmodule laat bewegen of branden (de pomp, de groeilampen, de ventilatoren, de CO_2-klep) vraagt veel meer stroom dan een pin kan leveren. Tussen de pin en de last zit telkens een transistor, en bij elke spoel een diode.



Een reconstructie van de eerste transistor: een plastic driehoekje met goudfolie op een blokje germanium.

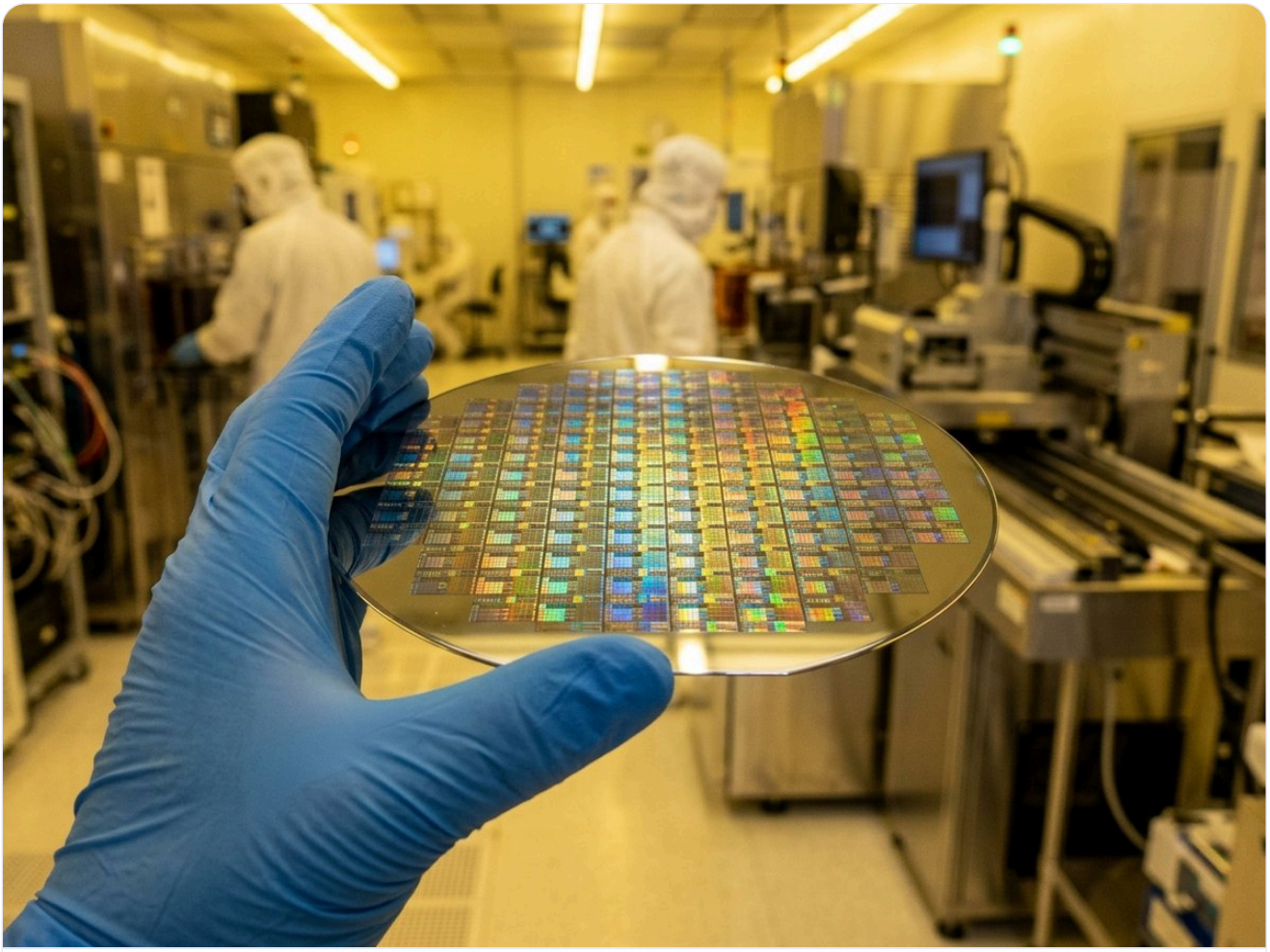
Halfgeleiders: geleiden op bestelling

Een metaal geleidt goed omdat er per atoom ongeveer één elektron vrij beweegt (F19). In een isolator zitten alle elektronen vast in bindingen. Een **halfgeleider** zit daartussen, en het bekendste voorbeeld is **silicium** (Si). Een siliciumatoom heeft vier valentie-elektronen (hoofdstuk 12). In een kristal deelt elk atoom die vier met zijn vier buren: vier atoombindingen, elk met twee elektronen (hoofdstuk 13). Bij het absolute nulpunt zit elk elektron vast: zuiver silicium is dan een isolator.

Bij kamertemperatuur trillen de atomen, en af en toe schudt de warmte een elektron los. Het is dan vrij, en op zijn oude plaats blijft een **gat** achter: een binding met één elektron te weinig. Een buurelektron kan in dat gat springen, en dan schuift het gat een plaats op. Een gat gedraagt zich als een positieve lading die beweegt. In zuiver silicium bij 20 °C is dat maar ongeveer één vrij elektron per $5 \cdot 10^{12}$ atomen, dus geleidt het slecht. Maar hoe warmer, hoe meer vrije ladingen: bij een halfgeleider **daalt** de weerstand met de temperatuur. Zo werkt de NTC van hoofdstuk 22.

stof	soortelijke weerstand (Ω·m, 20 °C)	soort
koper	$1{,}7 \cdot 10^{-8}$	geleider
silicium met 1 fosforatoom per miljoen	ongeveer 10^{-3}	gedoteerde halfgeleider
zuiver silicium	ongeveer $2 \cdot 10^{3}$	halfgeleider
glas	10^{10} tot 10^{14}	isolator

Het echte geheim zit in de onderste regels van de tabel: met een piepklein beetje vreemde atomen, **doteren**, kan je de geleiding van silicium over een enorm bereik instellen, en zelf kiezen of de stroom door vrije elektronen of door gaten gedragen wordt. Dat maakt het mogelijk om op één stukje silicium diodes en transistors naast elkaar te bouwen.



Een schijf zuiver silicium, gedoteerd en bewerkt tot honderden chips met elk miljarden transistors.

UITLEG · 23.3

Doteren: n-type en p-type

3/22

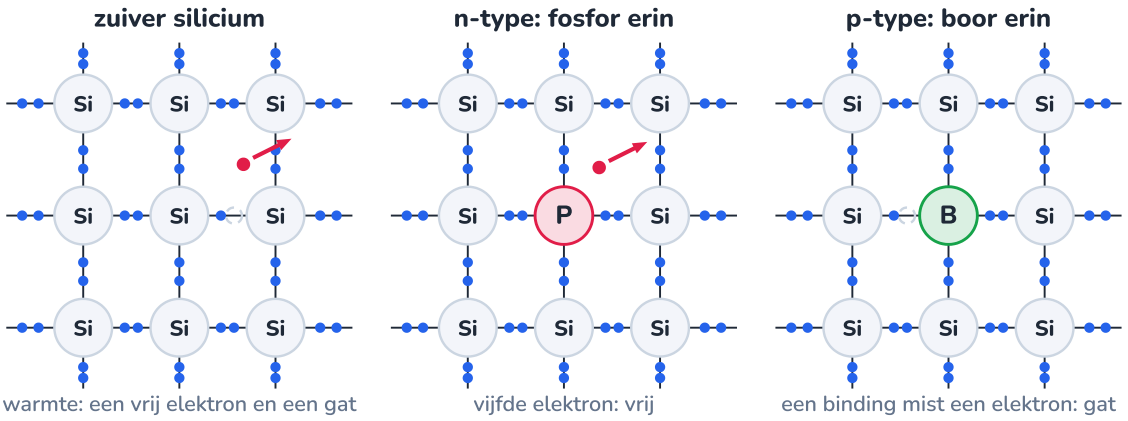
Vervang in het kristal één siliciumatoom op de miljoen door een **fosforatoom** (P). Fosfor heeft **vijf** valentie-elektronen. Vier ervan vormen de bindingen met de buren; het vijfde past nergens en is bij kamertemperatuur zo goed als vrij. Zo krijg je silicium met veel vrije elektronen: **n-type** (n van negatief, de lading van de beweeglijke dragers). Het fosforatoom zelf blijft achter als een vast positief ion.

Vervang je een siliciumatoom door een **booratoom** (B), met **drie** valentie-elektronen, dan mist er in één binding een elektron: een gat. Buurelektronen springen erin, en het gat zwerft door het kristal. Dat is **p-type** silicium (p van positief); het boor wordt een vast negatief ion.

Eén vreemd atoom per miljoen lijkt weinig, maar het maakt van een slechte geleider een redelijke: het voorbeeld hierna rekent het uit. Het materiaal blijft in zijn geheel neutraal: elke vrije lading heeft ergens een vast ion van het omgekeerde teken.

n-type silicium is negatief geladen

Nee. Een stuk n-type silicium is elektrisch neutraal: elk vrij elektron hoort bij een positief fosforion. 'n' zegt alleen dat de beweeglijke ladingen negatief zijn. Hetzelfde geldt voor p-type: neutraal, maar met beweeglijke gaten.



Elk streepje is een binding met twee elektronen (blauwe bolletjes). Fosfor brengt een vijfde elektron mee, boor laat een gat.

UITGEWERKT VOORBEELD · 23.4

Eén fosforatoom per miljoen

4/22

Een siliciumkristal bevat ongeveer $5 \cdot 10^{22}$ atomen per cm³. Zuiver silicium heeft bij kamertemperatuur ongeveer $1 \cdot 10^{10}$ vrije elektronen per cm³. (a) Hoeveel vrije elektronen per cm³ krijg je als je één atoom op de miljoen vervangt door fosfor? (b) Hoeveel keer meer is dat dan in zuiver silicium? (c) Koper heeft ongeveer $8{,}5 \cdot 10^{22}$ vrije elektronen per cm³. Hoeveel keer meer dan het gedoteerde silicium?

GEGEVEN

- $5 \cdot 10^{22}$ Si-atomen per cm³; zuiver: $1 \cdot 10^{10}$ vrije elektronen per cm³
- dotering: 1 op 10^6; koper: $8{,}5 \cdot 10^{22}$ per cm³

GEVRAAGD

- vrije elektronen per cm³ na doteren
- verhouding met zuiver silicium
- verhouding met koper

OPLOSSING

1. (a) Het aantal fosforatomen: $5 \cdot 10^{22} / 10^6 = 5 \cdot 10^{16}$ per cm³. Elk levert één vrij elektron: $5 \cdot 10^{16}$ vrije elektronen per cm³ (de 10^{10} van het zuivere silicium vallen daarbij weg).
2. (b) $\dfrac{5 \cdot 10^{16}}{1 \cdot 10^{10}} = 5 \cdot 10^6$: vijf miljoen keer meer.
3. (c) $\dfrac{8{,}5 \cdot 10^{22}}{5 \cdot 10^{16}} = 1{,}7 \cdot 10^6$: koper heeft nog altijd bijna twee miljoen keer meer vrije elektronen.

ANTWOORD

Eén fosforatoom per miljoen geeft $5 \cdot 10^{16}$ vrije elektronen per cm³: vijf miljoen keer meer dan zuiver silicium, maar nog bijna twee miljoen keer minder dan koper. De soortelijke weerstand daalt mee, van ongeveer $2 \cdot 10^3$ tot ongeveer $10^{-3}\,\Omega \cdot \mathrm{m}$.

Klopt dit? De getallen tonen waarom halfgeleiders zo bruikbaar zijn: met een minieme hoeveelheid vreemde atomen, die je per zone van de chip kan kiezen, stel je de geleiding over meer dan zes grootteordes in. De zuiverheid van het silicium zelf moet daarvoor extreem hoog zijn: beter dan één vreemd atoom per miljard.

De pn-overgang

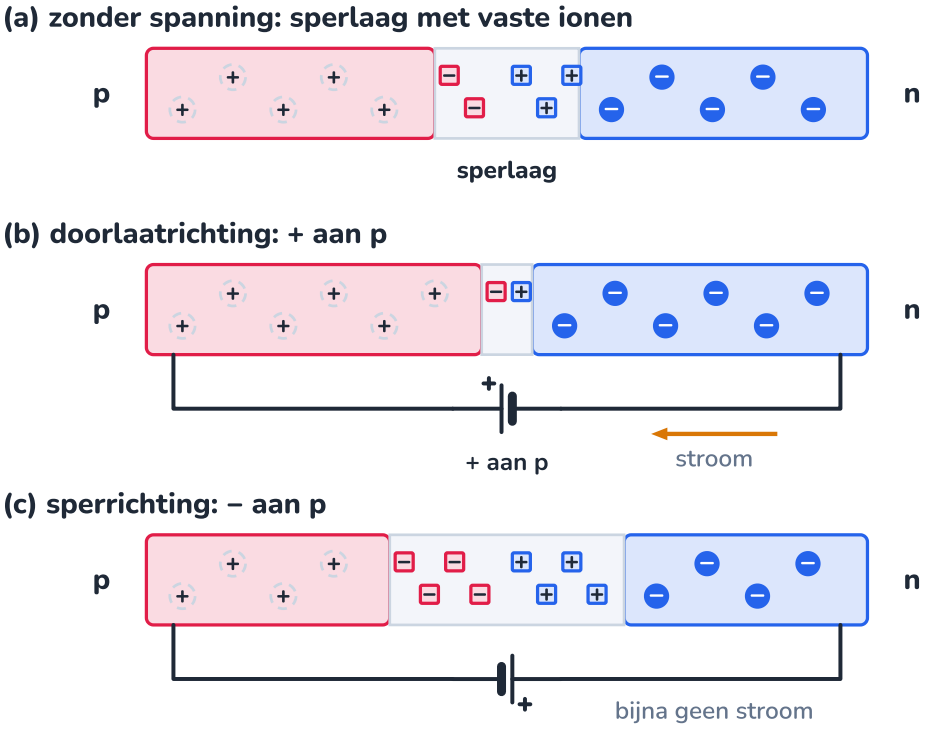
Maak in één kristal een p-gebied naast een n-gebied. Aan de grens bewegen vrije elektronen van de n-kant naar de p-kant, waar ze in gaten vallen, en omgekeerd. Zo ontstaat rond de grens een dunne laag zonder vrije ladingen, de **sperlaag**. Wat er achterblijft, zijn de vaste ionen: negatief aan de p-kant, positief aan de n-kant. Hun elektrisch veld duwt nieuwe elektronen terug: na een fractie van een seconde houdt het oversteken op. Over de sperlaag staat een 'drempel' van ongeveer 0,7 V bij silicium.

Doorlaatrichting. Zet de pluspool van een bron aan de p-kant. De bron duwt gaten en elektronen naar de grens: de sperlaag wordt smaller. Zodra de spanning groot genoeg is om de drempel te overwinnen (bij silicium ongeveer 0,6 à 0,7 V), stromen ze massaal over en ontmoeten ze elkaar: er loopt een grote stroom.

Sperrichting. Zet de minpool aan de p-kant. De bron trekt de vrije ladingen weg van de grens: de sperlaag wordt breder, en er loopt bijna niets, alleen een kleine lekstroom (bij een 1N4007 hoogstens 5 µA). Pas bij een heel grote spanning, de **doorslagspanning** (1000 V voor de 1N4007), breekt de sperlaag.

Een eenrichtingsklep

Een diode is voor stroom wat een terugslagklep is voor water: in één richting gaat hij open zodra er genoeg druk op staat, in de andere richting blijft hij dicht. Net als bij een klep kost het openhouden wat 'druk': de drempelspanning van ongeveer 0,7 V.



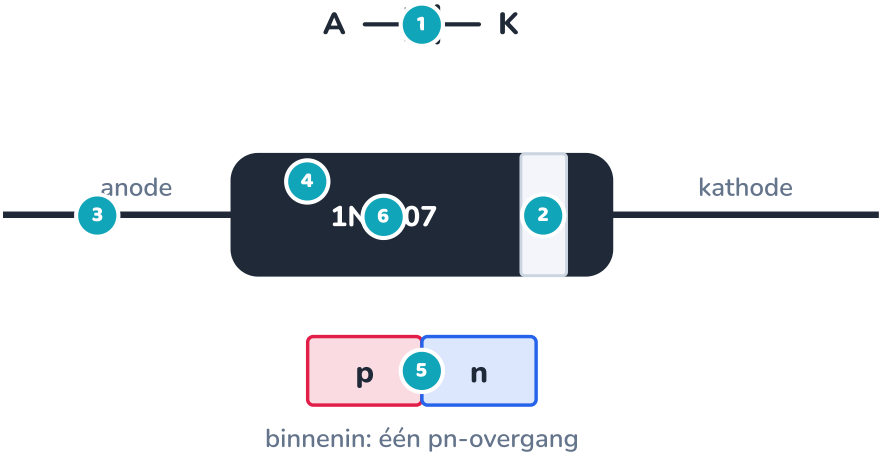
(a) De sperlaag bevat alleen vaste ionen. (b) In doorlaat wordt ze smal en loopt er stroom. (c) In sper wordt ze breder.

ONTLEED · 23.6

De diode

6/22

Een gewone siliciumdiode, de 1N4007, zoals je ze in elke doos met onderdelen vindt. Ze verdraagt 1 A en 1000 V in sperrichting.



Behuizing, ring en symbool: de ring en de streep van het symbool staan allebei aan de kathodekant.

1 Symbool
De driehoek wijst in de richting waarin stroom mag lopen (de conventionele stroomzin, F19); de streep is de kathode.

2 Ring = kathode
De zilveren of witte ring op de behuizing staat aan de kathodekant. Zo sluit je een diode juist aan zonder symbool.

3 Anode
De p-kant. In doorlaatrichting komt de stroom hier binnen: de anode ligt dan ongeveer 0,7 V hoger dan de kathode.

4 Behuizing
Zwart kunststof rond een klein siliciumchipje, met twee draden die ook de warmte afvoeren.

5 Binnenin
Eén pn-overgang: p aan de anode, n aan de kathode. De 'diode' is die overgang, de rest is verpakking.

6 Typeaanduiding
1N4001 tot 1N4007 zijn dezelfde diode voor 50 tot 1000 V; de 1N4148 is een kleine, snelle signaaldiode voor enkele honderden mA.

UITLEG · 23.7

De karakteristiek van een diode

7/22

Hoe hangt de stroom door een diode af van de spanning? De natuurkundige die de junctietransistor bedacht, William Shockley, leidde het af voor een ideale pn-overgang:

DIODEVERGELIJKING VAN SHOCKLEY

$$I = I_S\left(\mathrm{e}^{U/(n\,U_T)} - 1\right) \qquad U_T = \frac{kT}{q} \approx 25{,}7\,\mathrm{mV}\ (25\,^\circ\mathrm{C})$$

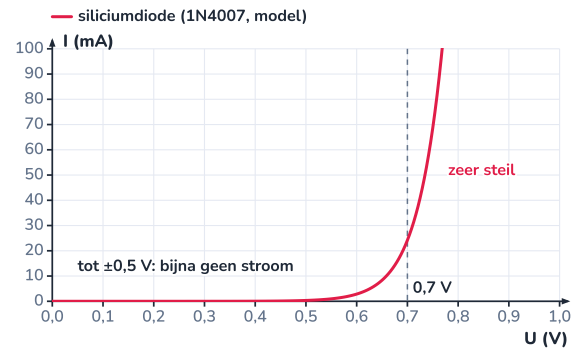
I_S: verzadigingsstroom (nanoampères of minder); n: idealiteitsfactor, tussen 1 en 2; U_T: thermische spanning, met k de constante van Boltzmann en q de lading van een elektron.

De grafieken tonen het rekenmodel van een 1N4007 ($I_S = 7{,}0\,\mathrm{nA}$, $n = 1{,}81$, en een serieweerstand van 0,034 Ω voor de draden en het silicium). Tot ongeveer 0,5 V loopt er bijna niets. Daarboven stijgt de stroom **exponentieel**: elke 107 mV extra geeft tien keer meer stroom. Op een lineaire schaal lijkt het daardoor alsof de diode bij 0,7 V plots 'opengaat'. Op een logaritmische schaal (hoofdstuk 5) is het een rechte.

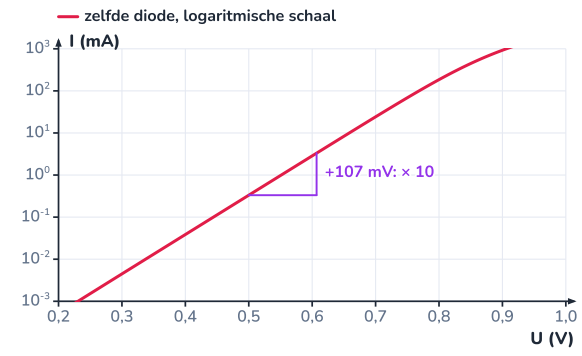
Daaruit komt de vuistregel die je bij elke berekening gebruikt: **een geleidende siliciumdiode heeft ongeveer 0,6 à 0,7 V over zich**, bijna los van de stroom: 0,55 V bij 1 mA, 0,66 V bij 10 mA, 0,77 V bij 100 mA. Bij 1 A wordt het 0,91 V (de datasheet zegt: hoogstens 1,1 V). Warmer maakt de drempel kleiner, ongeveer 2 mV per °C.

Een diode is een weerstand van 0,7 V

Een diode volgt de wet van Ohm niet: U/I is bij 1 mA 551 Ω en bij 100 mA 7,7 Ω. Wat bijna constant blijft, is de spanning, niet de verhouding. Daarom reken je met '0,7 V', niet met een vaste weerstand.



Exact volgens het model: bijna geen stroom tot 0,5 V, daarna een steile stijging.



Op een logaritmische stroomas is de karakteristiek een rechte: 107 mV per factor tien.

PROEF · 23.8

De karakteristiek van een diode en een led meten

8/22

ONDERZOEKSVRAAG

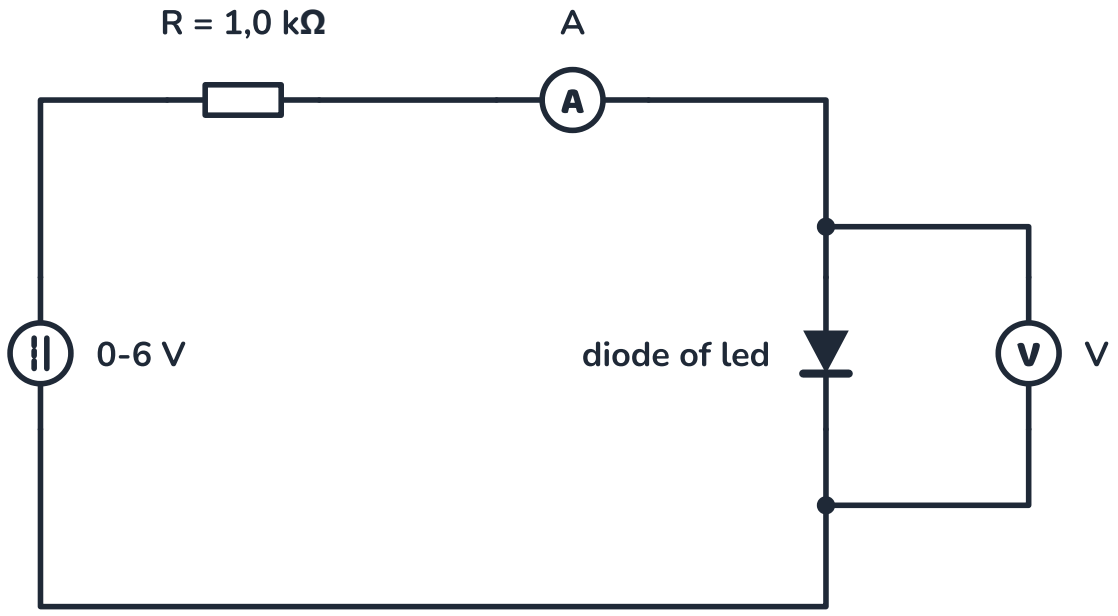
Bij welke spanning begint een siliciumdiode te geleiden, en bij welke een rode led?

MATERIAAL

- regelbare voeding 0-6 V (of een batterij met een potentiometer van 1 kΩ als deler)
- weerstand 1,0 kΩ
- siliciumdiode 1N4007 en een rode led van 5 mm
- twee multimeters

Veiligheid

De weerstand van 1,0 kΩ beperkt de stroom tot enkele mA: zonder die weerstand gaat een led meteen stuk.



De weerstand van 1,0 kΩ legt de stroom vast; de voltmeter meet alleen over de diode.

WERKWIJZE

1. Bouw de kring: voeding, weerstand, ampèremeter en diode in serie; de voltmeter over de diode.
2. Zet de diode in doorlaatrichting (ring aan de minkant). Verhoog de spanning in kleine stappen en noteer telkens de stroom en de spanning over de diode, tot 30 mA.
3. Draai de diode om en meet opnieuw: wat zie je?
4. Herhaal met de led (lange poot aan de pluskant).
5. Zet beide karakteristieken in één grafiek, I verticaal.

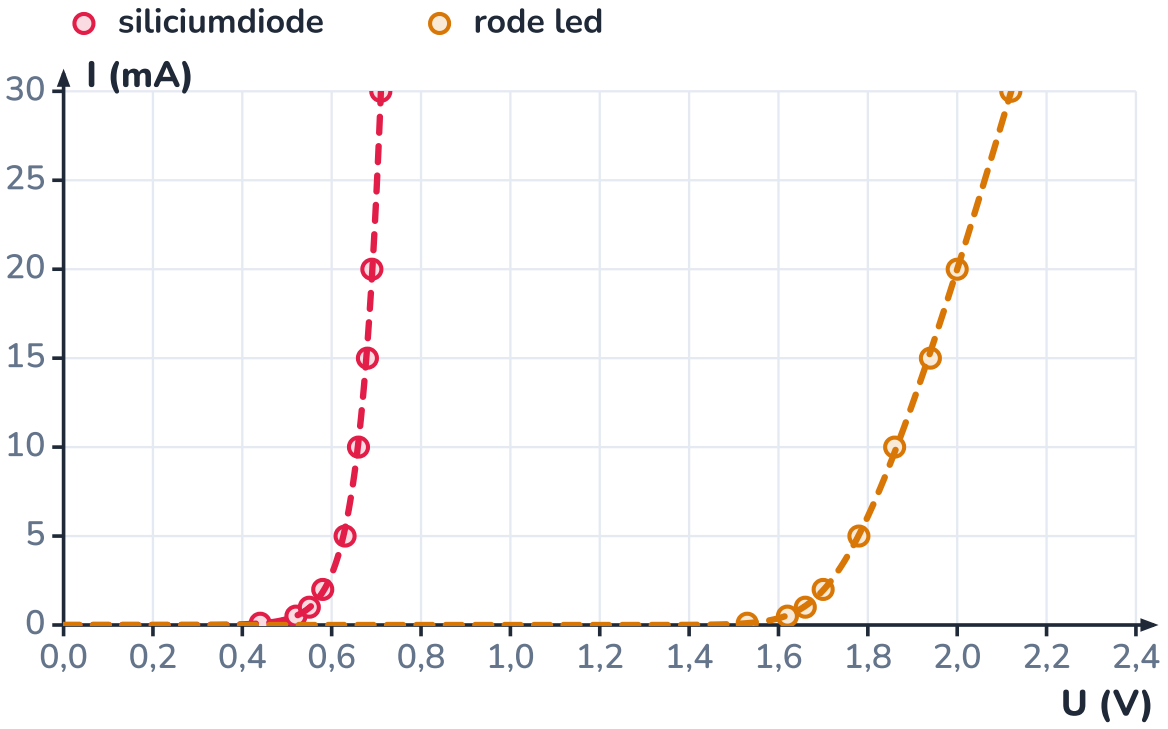
WAARNEMING

I (mA)	0,1	0,5	1	2	5	10	15	20	30
U diode (V)	0,44	0,52	0,55	0,58	0,63	0,66	0,68	0,69	
U rode led (V)	1,53	1,62	1,66	1,70	1,78	1,86	1,94	2,00	

Voorbeeldmeting. Omgekeerd aangesloten meet de ampèremeter 0,00 mA voor beide, en staat bijna de volle voedingsspanning over de diode. De led gloeit al zwak vanaf ongeveer 1,6 V.

VERKLARING

karakteristieken hebben dezelfde vorm: eerst bijna niets, dan een steile stijging. Bij de siliciumdiode ligt de knik rond 0,6 V, bij de rode led rond 1,8 V. Het verschil zit in het halfgeleidermateriaal: in een led is de drempel groter, en bij het oversteken van de overgang komt de energie vrij als licht in plaats van als warmte. Hoe groter de drempel, hoe meer energie elk lichtdeeltje meekrijgt (volgende bladen). Een led is dus ook een diode: omgekeerd aangesloten brandt ze niet.



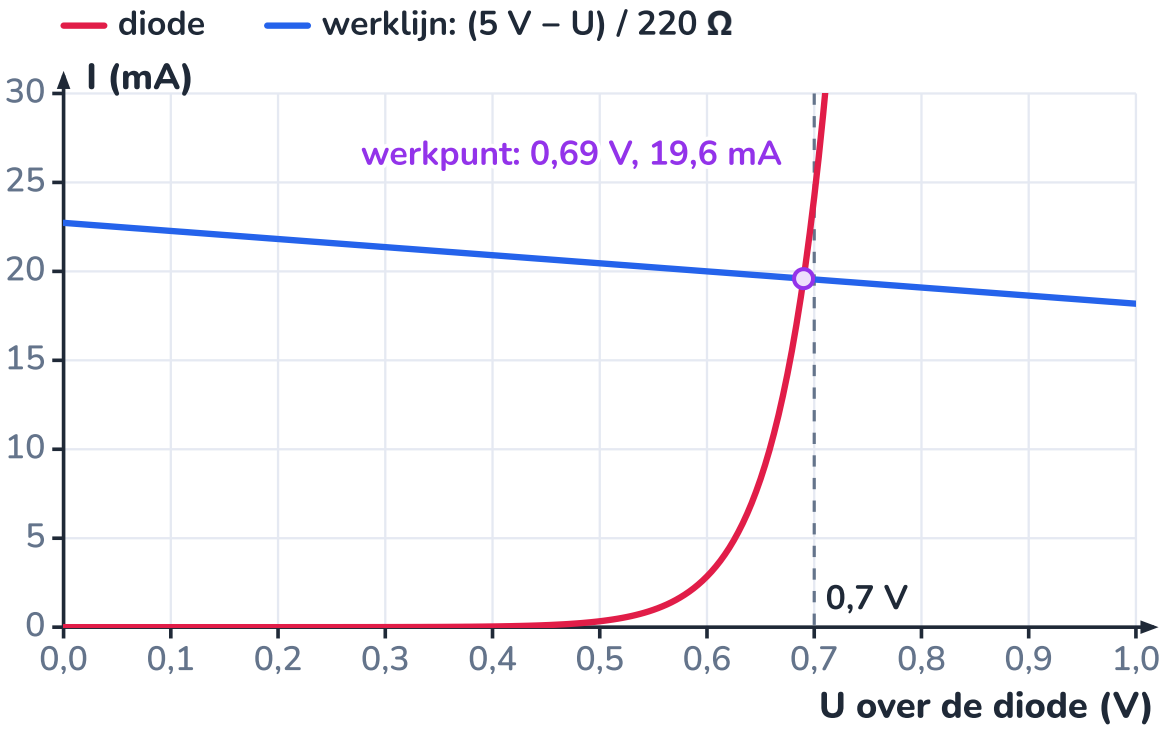
Meetpunten (bolletjes) en het rekenmodel (stippellijn) liggen op elkaar.

UITGEWERKT VOORBEELD · 23.9

Het werkpunt: een diode met een weerstand

9/22

Een siliciumdiode staat in doorlaatrichting in serie met een weerstand van 220 Ω op 5,0 V. Hoe groot is de stroom? Reken eerst met de vuistregel van 0,7 V, en bepaal daarna het exacte werkpunt met de karakteristiek.



De werklijn (maaswet) snijdt de karakteristiek (diode) in het werkpunt.

GEGEVEN

- $U = 5{,}0\,\text{V}$, $R = 220\,\Omega$
- karakteristiek van de diode (model)

GEVRAAGD

- I en U_D

OPLOSSING

1. Maaswet: $5{,}0 = U_D + 220 \cdot I$. Twee onbekenden, dus een tweede verband nodig: de karakteristiek.
2. **Vuistregel.** Neem $U_D = 0{,}7\,\text{V}$: $I = \dfrac{5{,}0 - 0{,}7}{220} = 19{,}5\,\text{mA}$.
3. **Grafisch.** Schrijf de maaswet als $I = \dfrac{5{,}0 - U_D}{220}$: een rechte, de **werklijn**. Bij $U_D = 0$ is $I = 22{,}7\,\text{mA}$, bij $U_D = 1{,}0\,\text{V}$ is $I = 18{,}2\,\text{mA}$. Teken ze in de grafiek van de diode.
4. Het snijpunt voldoet aan allebei: $U_D = 0{,}69\,\text{V}$ en $I = 19{,}6\,\text{mA}$ (met Python op het volgende blad tot op drie cijfers).

ANTWOORD

$I = 19{,}6\,\text{mA}$ **bij** $U_D = 0{,}69\,\text{V}$**. De vuistregel gaf 19,5 mA: amper 0,3 % verschil.**

Klopt dit? Een ideale diode (0 V) zou 22,7 mA geven: de 0,7 V weglaten is een fout van 16 %. Omdat de werklijn bijna horizontaal is en de karakteristiek bijna verticaal, maakt de precieze drempel weinig uit: een fout van 0,05 V in U_D verandert I maar met 0,2 mA. Daarom werkt de vuistregel zo goed.

CODE · 23.10

Python: het werkpunt uitrekenen

10/22

Het snijpunt van een exponentiële functie en een rechte kan je niet met een formule oplossen. Een computer doet het door te **halveren** (bisectie): hij begint met een interval waarin de stroom zeker ligt (tussen 0 en U/R), kijkt in het midden of de maaswet te veel of te weinig spanning geeft, en houdt de helft over waar het antwoord in ligt. Na 60 stappen is het interval kleiner dan de

nauwkeurigheid van de computer. Alle grafieken van dit hoofdstuk zijn zo berekend. In hoofdstuk E36 leer je snellere methoden, zoals die van Newton.

Python **diode.py** 31 regels

```python
# diode.py - de karakteristiek van een siliciumdiode en haar werkpunt met een weerstand
import math

UT = 1.380649e-23 * 298.15 / 1.602176634e-19     # thermische spanning bij 25 graden C (V)
IS, N, RS = 7.02767e-9, 1.80803, 0.0341512        # rekenmodel van een 1N4007


def u_van_i(i):
    """Spanning over de diode (V) bij stroom i (A): Shockley plus serieweerstand."""
    return N * UT * math.log(i / IS + 1) + RS * i


def werkpunt(u_bron, r):
    """Zoek door halveren de stroom waarbij u_diode + r * i = u_bron (maaswet)."""
    laag, hoog = 0.0, u_bron / r
    for _ in range(60):
        midden = (laag + hoog) / 2
        if u_van_i(midden) + r * midden > u_bron:
            hoog = midden
        else:
            laag = midden
    return laag, u_van_i(laag)


print("  I (mA)   U (V)")
for i_ma in (0.1, 1, 10, 100, 1000):
    print(f"{i_ma:8.1f}  {u_van_i(i_ma / 1000):.3f}")

i, u = werkpunt(5.0, 220)
print(f"werkpunt op 5 V met 220 ohm: {i * 1000:.2f} mA bij {u:.3f} V")
print(f"met 'U = 0,7 V': {(5.0 - 0.7) / 220 * 1000:.2f} mA")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
  I (mA)   U (V)
     0.1  0.444
     1.0  0.551
    10.0  0.658
   100.0  0.769
  1000.0  0.906
werkpunt op 5 V met 220 ohm: 19.59 mA bij 0.690 V
met 'U = 0,7 V': 19.55 mA
```

De tabel toont de exponentiële wet: elke stap van een factor tien in de stroom kost ongeveer 0,1 V. Pas bij 1 A speelt de serieweerstand mee.

UITLEG · 23.11

Gelijkrichten: van wisselspanning naar gelijkspanning

11/22

Een transformator of een dynamo levert **wisselspanning**: 50 keer per seconde van plus naar min en terug. Een led, een microcontroller of een batterij heeft **gelijkspanning** nodig. Het eenvoudigste middel is één diode in serie met de last: de **halvegolfgelijkrichter**.

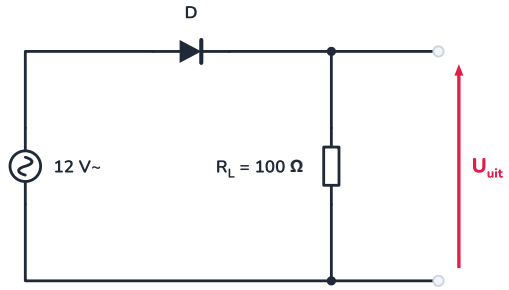
Een transformator van 12 V~ geeft een sinus met een topwaarde van $12 \cdot \sqrt{2} = 17{,}0\ \mathrm{V}$ (waarom 12 V de 'effectieve' waarde is, zie je in hoofdstuk E20). Tijdens de positieve helft staat de diode in doorlaat: de

last krijgt de ingang min ongeveer 0,7 V. Tijdens de negatieve helft spert de diode: de last krijgt niets, en de volle 17,0 V staat in sperrichting over de diode (voor een 1N4007 geen probleem).

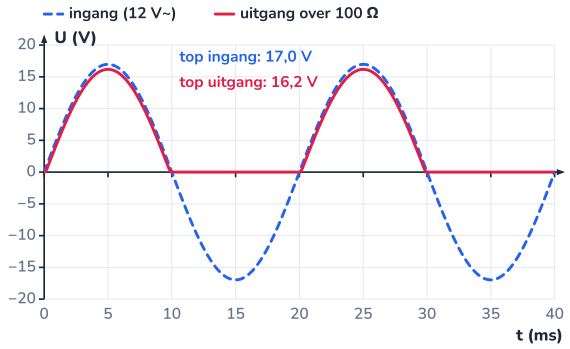
Met een last van 100 Ω is de piekstroom 0,16 A; de diode neemt dan 0,79 V, dus de uitgang komt tot 16,2 V. Het resultaat is een **pulserende** gelijkspanning: altijd dezelfde richting, maar de helft van de tijd nul. Gemiddeld is dat maar 5,0 V. Om er een bruikbare, vlakke gelijkspanning van te maken, zet je een condensator over de uitgang (hoofdstuk 24), en om ook de negatieve helft te gebruiken, vier diodes in een brug.

Diodes die je elke dag gebruikt

- **Ompolingsbeveiliging**: een diode in serie met de voeding van een toestel laat alleen stroom door als de batterij juist zit.
- **Blokkeerdiode**: een zonnepaneel met een diode in serie laadt de batterij overdag, maar de batterij kan 's nachts niet terugstromen in het paneel.
- **Bruggelijkrichter**: vier diodes in elke lader en adapter die van wisselspanning gelijkspanning maakt (opdracht 23.8).



De diode laat alleen de positieve halve golven door naar de last.



De uitgang volgt de positieve helft, 0,8 V lager; de negatieve helft valt weg.

UITLEG · 23.12

De led: licht uit een pn-overgang

12/22

In een led (*light emitting diode*) valt een elektron dat de overgang oversteekt in een gat, en geeft het zijn energie af als één lichtdeeltje, een **foton**. Silicium doet dat bijna niet (daar wordt de energie warmte); leds zijn gemaakt van verbindingen als galliumarsenide, galliumfosfide of indiumgalliumnitride. Hoeveel energie een foton krijgt, hangt af van het materiaal, en die energie bepaalt de kleur:

ENERGIE VAN EEN FOTON

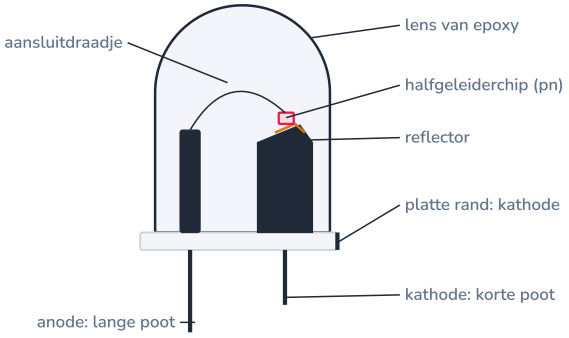
$$E_{\text{foton}} = \frac{h\,c}{\lambda} \qquad E\ (\text{eV}) \approx \frac{1240}{\lambda\ (\text{nm})}$$

$h = 6{,}626 \cdot 10^{-34}\ \text{J}\cdot\text{s}$ (constante van Planck), c: lichtsnelheid, λ: golflengte. 1 eV is de energie die een elektron krijgt als het 1 V doorloopt.

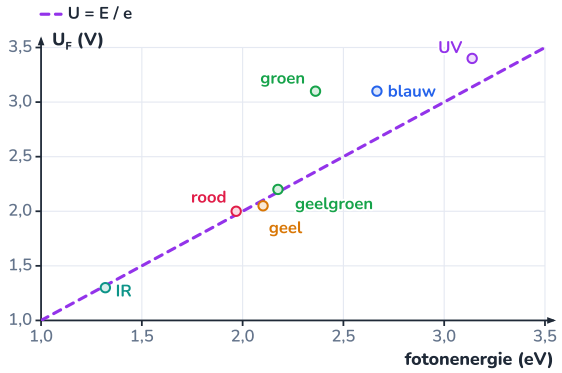
Om een foton van 2 eV te maken, moet elk elektron minstens ongeveer 2 V 'krijgen': de drempelspanning van een led ligt dicht bij de fotonenergie in volt. Rood licht ($\lambda \approx 630\,\text{nm}$, 1,97 eV) vraagt ongeveer 2 V, blauw (465 nm, 2,67 eV) ongeveer 3 V. De groene en blauwe leds van indiumgalliumnitride vragen nog wat meer; een witte led is een blauwe led met een geel oplichtend poeder (fosfor) erop.

kleur	golflengte (nm)	fotonenergie (eV)	materiaal	U_F bij 20 mA (V)
infrarood	±940	1,32	GaAs	1,2-1,4
rood	±630	1,97	AlGaInP	1,8-2,2
geel	±590	2,10	AlGaInP	1,9-2,2
geelgroen	±570	2,18	GaP	2,0-2,4
groen	±525	2,36	InGaN	2,8-3,4
blauw	±465	2,67	InGaN	2,8-3,4
wit (blauw + fosfor)	±455 (blauw)	2,72	InGaN	2,8-3,4
ultraviolet	±395	3,14	InGaN	3,2-3,6

Een led heeft een **richting**: de lange poot is de anode (plus), de korte poot en de platte kant van de rand zijn de kathode. Omgekeerd brandt ze niet, en veel leds verdragen in sperrichting maar een paar volt (vaak 5 V).



In de kop zit een piepklein chipje op een reflector; een draadje verbindt het met de anode.



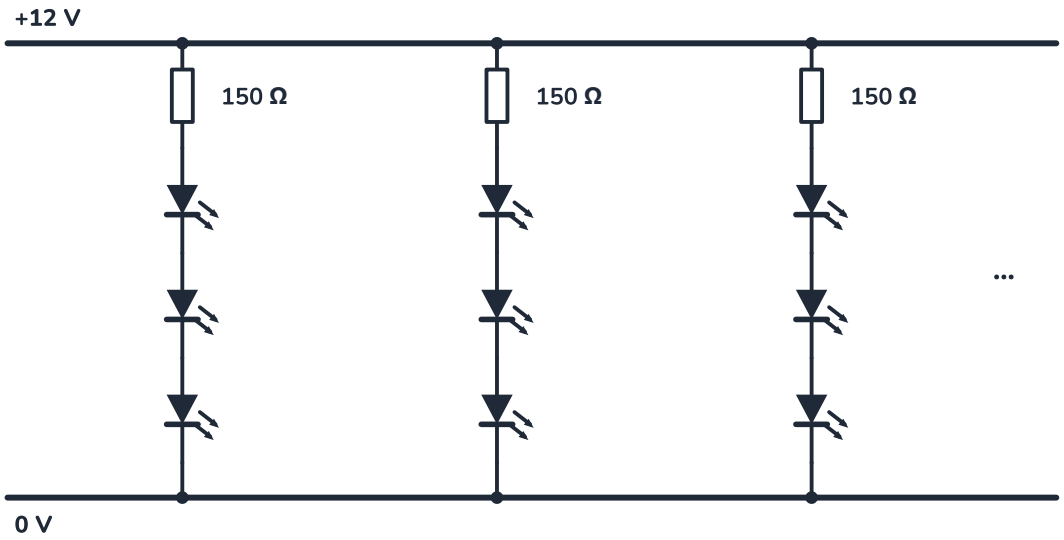
Hoe groter de fotonenergie, hoe groter de spanning; groene en blauwe InGaN-leds liggen hoger dan de lijn.

UITGEWERKT VOORBEELD · 23.13

De voorschakelweerstand berekenen

13/22

(a) Een rode statusled (2,0 V) moet aan een pin van 5,0 V branden met ongeveer 10 mA. (b) Een witte led (3,0 V) aan dezelfde pin, ook 10 mA. (c) Een ledstrip van 12 V bestaat uit segmenten van drie witte leds (3,0 V elk) met één weerstand, voor 20 mA. Bereken en kies telkens de weerstand uit de E12-reeks (… 100, 120, 150, 180, 220, 270, 330, 390, 470 …), en controleer het vermogen.



Elk segment: drie leds en één weerstand in serie; alle segmenten parallel tussen 12 V en 0 V.

GEGEVEN

- (a) $U = 5{,}0\,\text{V}$, $U_F = 2{,}0\,\text{V}$, $I = 10\,\text{mA}$
- (b) $U_F = 3{,}0\,\text{V}$
- (c) $U = 12\,\text{V}$, $3 \times 3{,}0\,\text{V}$, $I = 20\,\text{mA}$

GEVRAAGD

- R, gekozen waarde, echte stroom, vermogen

OPLOSSING

1 Maaswet: $U = U_F + R \cdot I$, dus $R = \dfrac{U - U_F}{I}$.

2 (a) $R = \dfrac{5{,}0 - 2{,}0}{0{,}010} = 300\,\Omega$. Kies de **grotere** buur, 330 Ω: $I = 3{,}0/330 = 9{,}1\,\text{mA}$. Vermogen in de weerstand $3{,}0 \cdot 0{,}0091 = 27\,\text{mW}$, in de led $2{,}0 \cdot 0{,}0091 = 18\,\text{mW}$.

3 (b) $R = \dfrac{5{,}0 - 3{,}0}{0{,}010} = 200\,\Omega$. Kies 220 Ω: $I = 9{,}1\,\text{mA}$; 18 mW in de weerstand.

4 (c) $R = \dfrac{12 - 3 \cdot 3{,}0}{0{,}020} = 150\,\Omega$: dat is een E12-waarde. Weerstand: $3{,}0 \cdot 0{,}020 = 60\,\text{mW}$; leds: $9{,}0 \cdot 0{,}020 = 180\,\text{mW}$.

ANTWOORD

(a) 330 Ω (9,1 mA), (b) 220 Ω (9,1 mA), (c) 150 Ω (20 mA). Alle vermogens liggen ver onder de 250 mW van een gewone weerstand. In een ledstrip gaat een kwart van het vermogen naar de weerstanden: $3{,}0/12 = 25\,\%$.

Klopt dit? Kies bij twijfel altijd de grotere weerstand: iets minder licht, maar de led en de pin blijven binnen hun grenzen (20 mA per pin, F24). De ledstrip van de meetkas (0,40 A) bestaat zo uit 20 van die segmenten: 60 leds, twee helften van 30 (F20).

UITLEG · 23.14

Waarom een led nooit zonder weerstand mag

14/22

De karakteristiek van een led is even steil als die van een diode, maar ligt hoger. Een rode led krijgt bij 1,86 V 10 mA, bij 2,0 V 20 mA, bij 2,12 V al 30 mA. Wie een led rechtstreeks op een spanning zet, bepaalt dus haar stroom niet: een verschil van een paar honderdsten van een volt, of een beetje opwarming (de drempel daalt), en de stroom verdubbelt. De weerstand lost dat op: over hem valt het

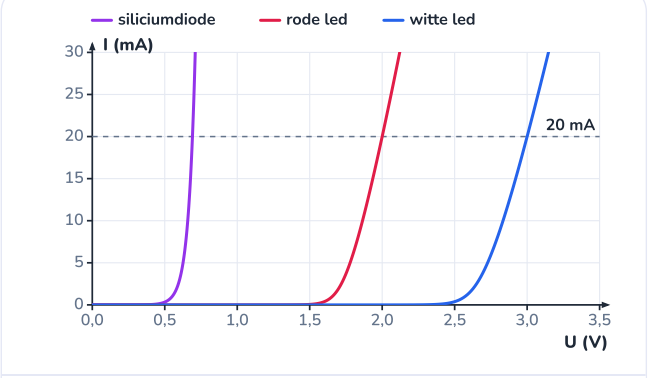
verschil tussen de voedingsspanning en de led, en die spanning legt met de wet van Ohm de stroom vast.

Maar ook met weerstand hangt de stroom af van de voedingsspanning. Een ledstrip van 12 V aan de LiFePO4-batterij van de klimaatmodule krijgt tussen 12,0 V (bijna leeg) en 14,6 V (laden). Volgens het model stijgt de stroom per segment dan van 20 naar 33 mA: 67 % te veel. De strip wordt warmer en veroudert sneller.

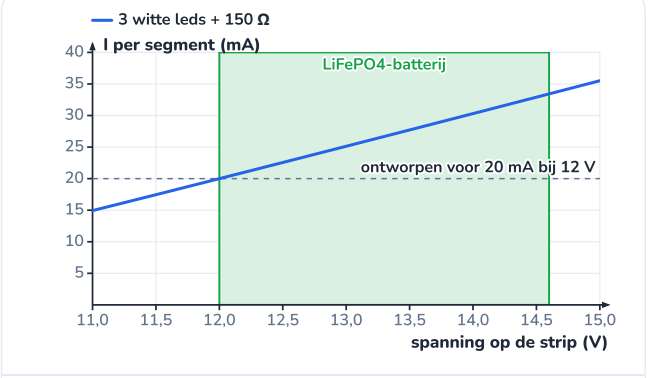
Een led van 3 V mag je op 3 V zetten

Alleen met een voeding die de **stroom** begrenst. De '3 V' in een datasheet is de spanning die hoort bij de nominale stroom, met een spreiding van enkele tienden van een volt tussen leds van dezelfde doos. Op een spanningsbron van 3 V kan de ene led nauwelijks branden en de andere doorbranden.

Daarom werken groeilampen en krachtige leds niet met een weerstand maar met een **stroombron** (*led-driver*): een schakeling die de stroom constant houdt, ongeacht de spanning. Hoe die werkt, zie je in hoofdstuk E24.



Hoe groter de drempel, hoe verder naar rechts; boven de drempel zijn alle drie even steil.



Met één weerstand per segment volgt de stroom de batterijspanning: 20 mA bij 12,0 V, 33 mA bij 14,6 V.

UITLEG · 23.15

De bipolaire transistor als schakelaar

15/22

Een **npn-transistor** bestaat uit drie lagen: n, p, n. De middelste, dunne p-laag is de **basis** (B), de buitenste zijn de **collector** (C) en de **emitter** (E). Er zijn dus twee pn-overgangen. Zolang er geen stroom in de basis loopt, spert een van die twee: tussen collector en emitter loopt niets, de transistor staat **uit**.

Stuur je een kleine stroom I_B in de basis (daarvoor moet de basis ongeveer 0,7 V boven de emitter liggen), dan loopt er een veel grotere collectorstroom: in het **actieve gebied** is

STROOMVERSTERKING

$$I_C = \beta \cdot I_B$$

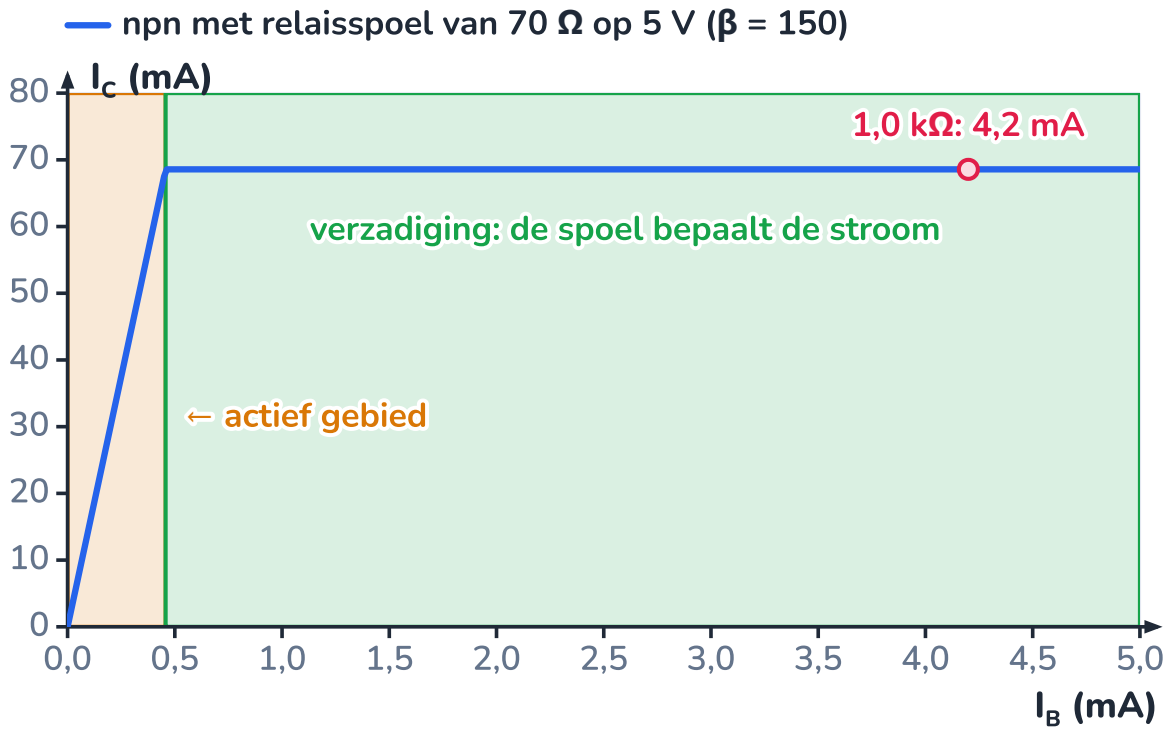
β (ook h_{FE}): typisch 50 tot 300, sterk verschillend van exemplaar tot exemplaar en afhankelijk van de stroom. De datasheet geeft een minimum.

Als schakelaar wil je geen versterker, maar 'helemaal aan' of 'helemaal uit'. Helemaal aan heet **verzadiging**: de collectorstroom wordt dan niet meer door β begrensd, maar door de last zelf, en tussen collector en emitter blijft maar een klein restje spanning over, $U_{CE(sat)} \approx 0{,}1$ à 0,3 V. Om zeker in verzadiging te zijn, geef je meer basisstroom dan strikt nodig: een **overstuurfactor** van 5 à 10 tegenover $I_C/\beta_{\min}$.

De basisstroom komt van de pin, via een **basisweerstand** R_B. Zonder die weerstand staat de pin rechtstreeks op een diode van 0,7 V (de basis-emitterovergang): de stroom wordt enorm, en pin of transistor gaat stuk.

BASISWEERSTAND VAN EEN TRANSISTORSCHAKELAAR

$$R_B = \frac{U_{\text{pin}} - U_{BE}}{I_B} \qquad I_B \approx (5 \text{ à } 10) \cdot \frac{I_C}{\beta_{\min}}$$



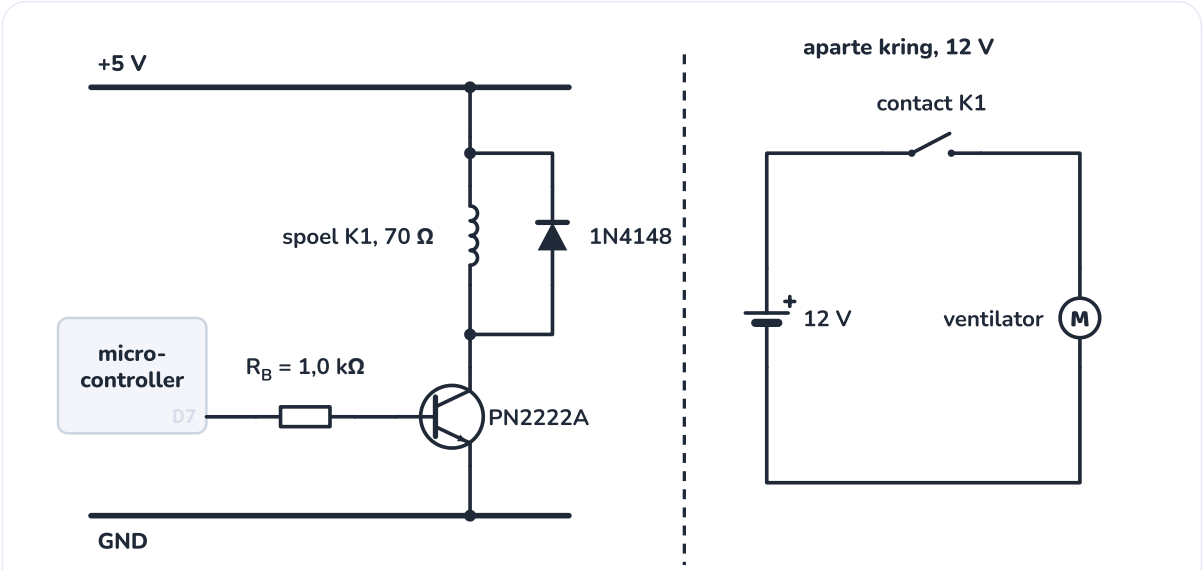
Vereenvoudigd model: tot 0,46 mA volgt I_C de basisstroom, daarna begrenst de spoel de stroom (verzadiging).

UITGEWERKT VOORBEELD · 23.16

Een relais schakelen met een npn-transistor

16/22

Een relais van 5 V (spoel 70 Ω) moet een aparte 12 V-kring met een ventilator schakelen. De spoel hangt tussen +5 V en de collector van een PN2222A; de emitter aan GND. De basis krijgt de pin D7 via een weerstand. Datasheet: $\beta \geq 100$ bij 150 mA, $U_{CE(sat)} \leq 0{,}3\ \text{V}$. (a) Hoe groot is de collectorstroom? (b) Kies de basisweerstand. (c) Hoeveel vermogen verliest de transistor?



Links de transistor die de spoel schakelt, met de diode over de spoel; rechts het contact in een aparte 12 V-kring.

GEGEVEN

- spoel 70 Ω op 5,0 V; $U_{CE(sat)} \approx 0{,}2\,\mathrm{V}$
- $\beta_{\min} = 100$; pin 5,0 V, $U_{BE} \approx 0{,}8\,\mathrm{V}$ in verzadiging

GEVRAAGD

- I_C
- R_B
- vermogen

OPLOSSING

1. (a) Maaswet door spoel en transistor: $I_C = \dfrac{5{,}0 - 0{,}2}{70} = 68{,}6\,\mathrm{mA}$.

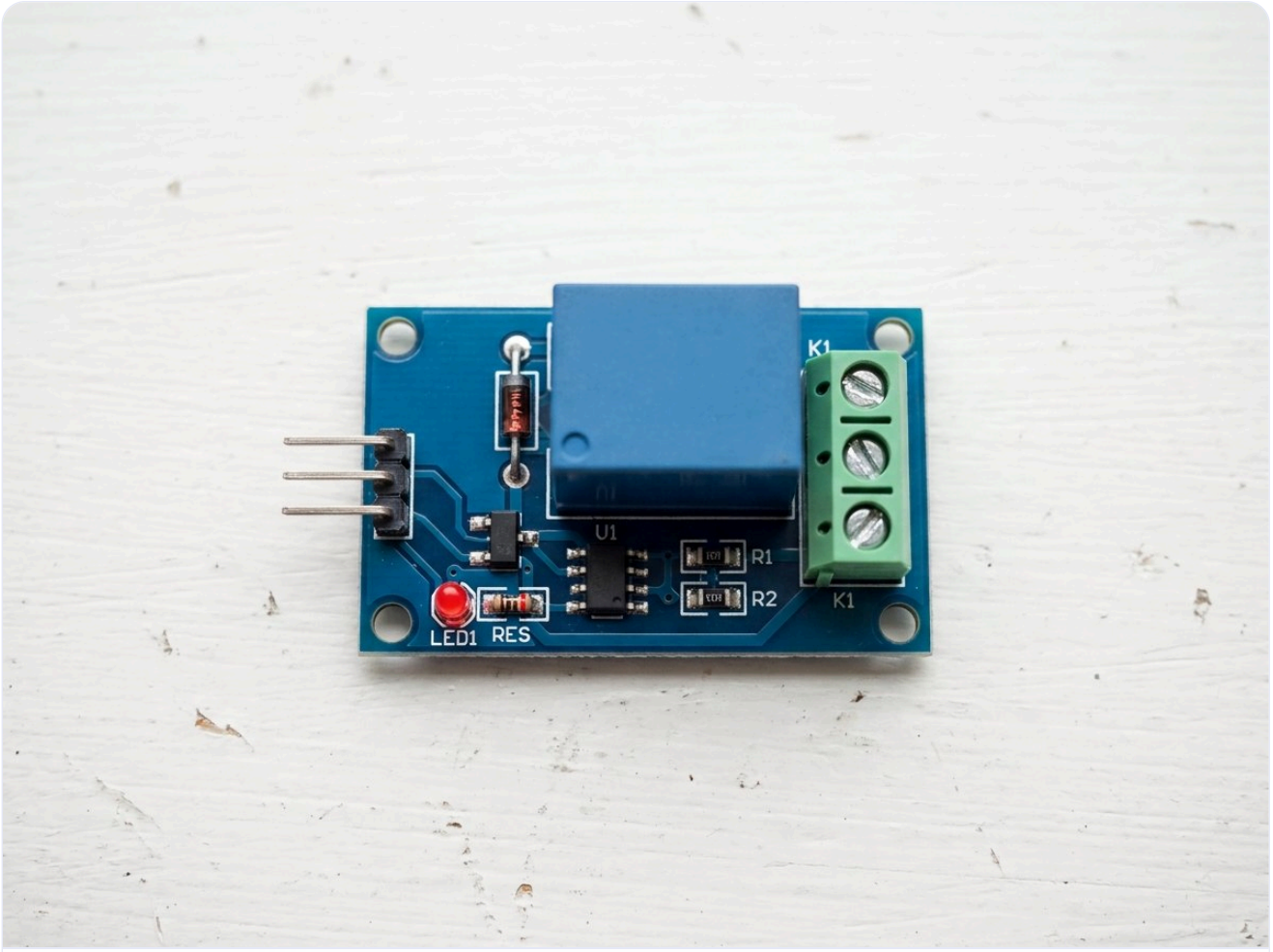
2. (b) Minimaal nodig: $I_B = 68{,}6/100 = 0{,}69\,\mathrm{mA}$. Met overstuurfactor 5 à 10: 3,4 à 6,9 mA. $R_B = \dfrac{5{,}0 - 0{,}8}{0{,}0042} = 1{,}0\,\mathrm{k\Omega}$ geeft 4,2 mA: overstuurfactor 6, en ruim onder de 20 mA van de pin.

3. (c) Transistor: $0{,}2 \cdot 0{,}0686 = 14\,\mathrm{mW}$, plus $0{,}8 \cdot 4{,}2 = 3{,}4\,\mathrm{mW}$ in de basis. De spoel zelf neemt $5^2/70 = 0{,}36\,\mathrm{W}$.

ANTWOORD

$I_C \approx 69\,\mathrm{mA}$; $R_B = 1{,}0\,\mathrm{k\Omega}$ **(4,2 mA); de transistor verliest ongeveer 17 mW en blijft koel.**

Klopt dit? Over de spoel staat een diode (1N4148) in sperrichting: bij het uitschakelen vangt ze de spanningspiek van de spoel op (verder in dit hoofdstuk). Het relais scheidt de twee kringen volledig: de 12 V van de ventilator komt nooit in de buurt van de microcontroller. Precies dat zit op de kant-en-klare 'relaismodules' voor microcontrollers: transistor, weerstand, diode en relais op één printje.



Een relaismodule: relais, transistor, weerstand en vrijloopdiode op één printje.

UITLEG · 23.17

De MOSFET: schakelen met spanning

17/22

Een **n-MOSFET** (*metal-oxide-semiconductor field-effect transistor*) heeft een **source** (S) en een **drain** (D), twee n-gebieden in een p-kristal, en ertussen, boven een flinterdun laagje isolerend oxide, de **gate** (G). Zonder spanning op de gate liggen er twee pn-overgangen tegen elkaar: er loopt niets. Maak je de gate positief tegenover de source, dan trekt ze elektronen naar het oppervlak onder het oxide. Boven de **drempelspanning** $U_{GS(th)}$ vormen die een geleidend n-**kanaal** van source naar drain: de MOSFET staat aan, en gedraagt zich als een kleine weerstand $R_{DS(on)}$.

Omdat de gate geïsoleerd is, loopt er in rust geen stroom in: minder dan 100 nA. De MOSFET is **spanningsgestuurd**. Alleen bij het schakelen moet de gate even geladen of ontladen worden: ze gedraagt zich als een kleine condensator (bij de IRLZ44N ongeveer 1700 pF, hoofdstuk 24).

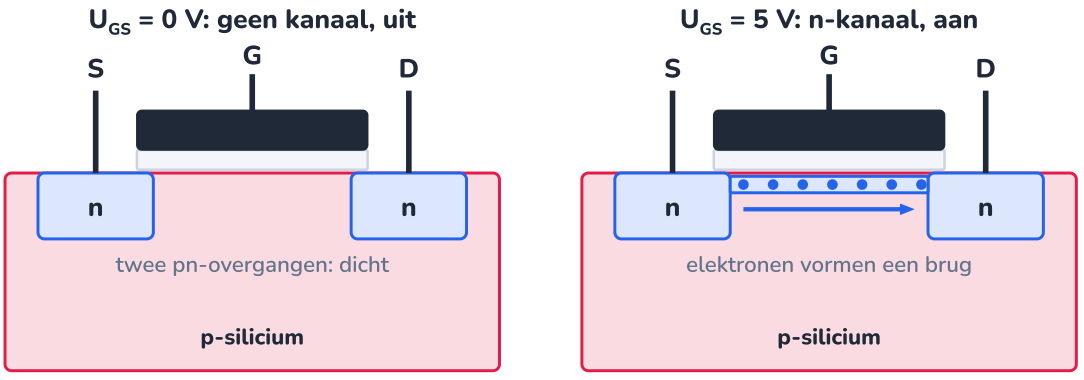
type	U_{DS} max	I_D max	$U_{GS(th)}$	$R_{DS(on)}$ volgens de datasheet
IRLZ44N	55 V	47 A	1,0-2,0 V	≤ 0,025 Ω bij 5 V; ≤ 0,035 Ω bij 4 V
IRF540N	100 V	33 A	2,0-4,0 V	≤ 0,044 Ω bij 10 V (bij 5 V niet gegeven)
2N7000	60 V	0,2 A	0,8-3,0 V	≤ 5,3 Ω bij 4,5 V

Logic level. De drempelspanning is *niet* de spanning waarbij de MOSFET goed aan staat, maar die waarbij hij net begint te geleiden (bij 250 µA). Wat telt, is $R_{DS(on)}$ bij de spanning die jouw pin levert. De IRLZ44N is gemaakt om vanuit 5 V te schakelen (*logic level*): hoogstens 0,025 Ω bij 5 V. De IRF540N

heeft zijn $R_{DS(on)}$ alleen bij 10 V: aan een pin van 5 V staat hij misschien maar half aan, en wordt hij heet. De 2N7000 schakelt wel vanuit 5 V, maar verdraagt maar 0,2 A.

Twee weerstanden horen erbij. Een **gateweerstand** van 100 à 220 Ω begrenst de korte laadstroom van de gate. Een **pull-down** van 10 kΩ tussen gate en source houdt de MOSFET uit zolang de pin nog niets doet (bij het opstarten en uploaden 'zweeft' een pin): anders kan de pomp onverwacht aanslaan.

	npn-transistor (PN2222A)	n-MOSFET (IRLZ44N)
aansluitingen	basis, collector, emitter	gate, drain, source
gestuurd door	stroom in de basis: $I_C \leq \beta I_B$	spanning op de gate: U_{GS}
stuurstroom	milliampères, zolang hij aanstaat	alleen even bij het schakelen; daarna < 100 nA
verlies als hij aan is	$U_{CE(sat)} \cdot I_C$, 0,1 tot 1 V maal de stroom	$R_{DS(on)} \cdot I^2$, bij 1 A 25 mW
goed voor	kleine lasten tot enkele honderden mA (relais, led, zoemer)	grote lasten (pomp, lampen, motoren), en PWM



Zonder gatespanning blokkeren twee pn-overgangen; met 5 V op de gate vormen elektronen een kanaal van source naar drain.

UITLEG · 23.18

De vrijloopdiode: een spoel wil blijven stromen

18/22

Een pomp, een relais en een magneetklep hebben één ding gemeen: een **spoel**. Een spoel verzet zich tegen elke verandering van de stroom (hoofdstuk 26 en E19): hoe sneller je de stroom verandert, hoe groter de spanning die ze opwekt.

SPANNING OVER EEN SPOEL

$$U_L = L \cdot \frac{\Delta I}{\Delta t}$$

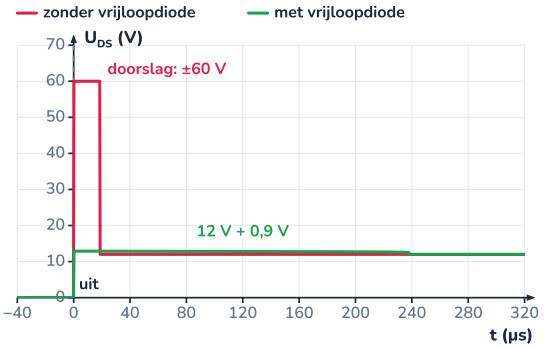
L: zelfinductie, in henry (H). Een klein pompmotortje: orde van grootte 1 mH; een relaisspoel: honderd keer meer.

Een MOSFET schakelt een stroom van 1 A uit in minder dan een microseconde. Met $L = 1\,\mathrm{mH}$ zou dat $10^{-3} \cdot 1/10^{-6} = 1000\,\mathrm{V}$ geven. Die spanning komt ook echt: de spoel duwt de drain omhoog tot er iets doorslaat. Bij een mechanische schakelaar is dat de lucht tussen de contacten (een vonk die de contacten wegbrandt), bij de IRLZ44N de MOSFET zelf, boven 55 V. De grafiek toont het model van de pomp (1 mH en 12 Ω): zonder diode staat de drain 19 µs lang op ongeveer 60 V, en de energie van de spoel ($\frac{1}{2}LI^2 = 0{,}5\,\mathrm{mJ}$) wordt warmte in de MOSFET. Bij elke uitschakeling opnieuw, en met PWM

(hoofdstuk 25) honderden keren per seconde. Die pieken jagen ook storingen door de draden, en dus door de metingen van de sensoren (hoofdstuk 21).

De oplossing is één diode over de spoel, in sperrichting ten opzichte van de voeding: kathode aan de plus. Zolang de MOSFET aan is, spert ze en doet ze niets. Als hij uitgaat, neemt de diode de stroom over: de stroom loopt rond door spoel en diode en dooft rustig uit in de weerstand van de spoel (238 µs in het model). De drain komt nooit hoger dan 12 V plus de drempel van de diode.

> ⚠ **Een model, geen meting**
>
> De grafiek rekent met een spoel van 1 mH en 12 Ω, en met doorslag bij 60 V. Een draaiende pomp wekt zelf ook een spanning op, waardoor de stroom nog sneller uitdooft. Het beeld blijft hetzelfde: zonder diode een hoge, korte piek, met diode 12,9 V. Meet het zelf met een oscilloscoop.



Zonder diode schiet de drain naar de doorslagspanning; met diode blijft ze net boven 12 V.

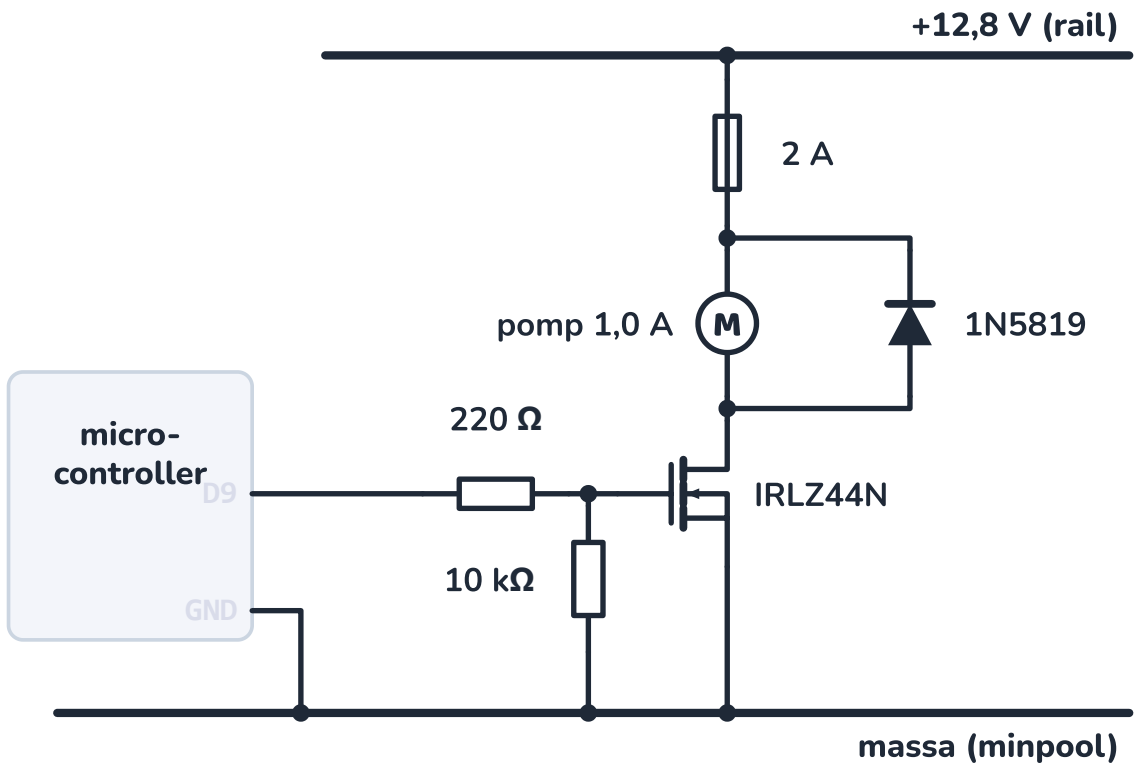
Zonder diode wordt de stroom in 19 µs afgebroken; met diode dooft ze uit in 238 µs.

UITGEWERKT VOORBEELD · 23.19

Habitat: de pomp van de waterkringloop schakelen

19/22

De pomp van de waterkringloop (12 V, 1,0 A) moet door de microcontroller aan- en uitgezet worden, vanuit pin D9 (5 V, hoogstens 20 mA). Ontwerp de schakeling: kies de schakelaar, reken het vermogen na, kies de weerstanden, de diode en de zekering. Kan het ook met een bipolaire transistor?



Low-side schakelen: de MOSFET tussen pomp en massa, de diode over de pomp, gemeenschappelijke massa.

GEGEVEN

- pomp 12 V, 1,0 A (12 W), aanloopstroom enkele ampère
- pin 5 V, hoogstens 20 mA
- IRLZ44N: $R_{DS(on)} \leq 0{,}025\,\Omega$ bij 5 V; $R_{\theta JA} = 62\,^\circ\mathrm{C/W}$

GEVRAAGD

- schema en onderdelen
- vermogen en opwarming van de MOSFET
- vergelijking met een bipolaire transistor

OPLOSSING

1. **Low-side**: de MOSFET zit tussen de pomp en de massa (source aan de massa), zodat de gatespanning gemeten wordt tegen dezelfde massa als de pin. De massa van de microcontroller en die van de 12,8 V-rail moeten dus verbonden zijn (liefst in het sterpunt, hoofdstuk 21).

2. **Vermogen**: $P = R_{DS(on)} \cdot I^2 = 0{,}025 \cdot 1{,}0^2 = 25\,\mathrm{mW}$. Opwarming zonder koelplaat: $0{,}025\,\mathrm{W} \cdot 62\,^\circ\mathrm{C/W} = 1{,}6\,^\circ\mathrm{C}$. Geen koelplaat nodig.

3. **Gate**: 220 Ω in serie (piek $5/220 = 23\,\mathrm{mA}$, maar maar enkele microseconden lang), 10 kΩ van gate naar massa ($0{,}5\,\mathrm{mA}$ als de pin hoog is).

4. **Vrijloopdiode** over de pomp, kathode aan +12,8 V: een Schottkydiode 1N5819 (40 V, 1 A, snel, ook voor PWM). Een 1N4007 werkt ook voor gewoon aan en uit.

5. **Zekering** van 2 A (traag) in de tak van de pomp, voor de aanloopstroom.

6. **Bipolair?** Een BD139 heeft bij 0,5 A een β van minstens 25; voor 1 A zou de pin minstens $1{,}0/25 = 40\,\mathrm{mA}$ moeten leveren, en voor verzadiging nog meer: te veel. En in verzadiging verliest hij typisch enkele tienden van een volt maal 1 A: tien keer meer dan de MOSFET.

ANTWOORD

Een IRLZ44N low-side aan D9 via 220 Ω, 10 kΩ pull-down, een 1N5819 over de pomp en een zekering van 2 A. De MOSFET verliest 25 mW en warmt 1,6 °C op; de pin levert alleen een korte laadstroom.

Klopt dit? Test in stappen: eerst met een lampje van 12 V in plaats van de pomp, en de spanning tussen drain en source meten (aan: enkele tientallen mV; uit: 12,8 V). Controleer de richting van de diode voor je de pomp aansluit: verkeerd om is ze een kortsluiting over de voeding zodra de MOSFET aangaat, en springt de zekering.

CODE · 23.20

C++: de pomp met droogloopbeveiliging

20/22

De pomp draait 10 minuten per uur: 4 uur per dag, zoals in het energiebudget van hoofdstuk 20, en genoeg om het reservoir van 60 L vier keer rond te pompen. Ze mag nooit droog draaien: de vlotterschakelaar van hoofdstuk 11 hangt aan pin 2 en de massa, met de interne pull-up (F24). Is de vlotter open (te weinig water), dan blijft de pomp uit, ook als het tijd is. De tijd wordt bijgehouden met `millis()`: het programma blijft zo op elk moment de vlotter lezen (hoofdstuk 29).

C++ (Arduino) **pomp.ino** 33 regels

```cpp
// pomp.ino - de pomp van de waterkringloop schakelen met een n-MOSFET (IRLZ44N) op pin 9
// Droogloopbeveiliging: vlotterschakelaar tussen pin 2 en GND (gesloten = genoeg water).
const int POMP = 9;
const int VLOTTER = 2;
const unsigned long AAN_MS = 10UL * 60 * 1000;       // 10 minuten pompen ...
const unsigned long CYCLUS_MS = 60UL * 60 * 1000;    // ... per uur: 4 uur per dag
unsigned long startCyclus = 0;
bool vorige = false;

void setup() {
  pinMode(POMP, OUTPUT);
  digitalWrite(POMP, LOW);             // uit bij het opstarten (en de pull-down van 10 kOhm op de gate)
  pinMode(VLOTTER, INPUT_PULLUP);      // vlotter open = HIGH = te weinig water
  Serial.begin(9600);
}

void loop() {
  unsigned long nu = millis();
  if (nu - startCyclus >= CYCLUS_MS) {
    startCyclus += CYCLUS_MS;          // een nieuw uur begint
  }
  bool tijdOmTePompen = (nu - startCyclus) < AAN_MS;
  bool genoegWater = (digitalRead(VLOTTER) == LOW);
  bool pompAan = tijdOmTePompen && genoegWater;
  digitalWrite(POMP, pompAan ? HIGH : LOW);

  if (pompAan != vorige) {             // alleen iets melden als er iets verandert
    if (pompAan) Serial.println("pomp aan");
    else if (!genoegWater) Serial.println("pomp uit: te weinig water in het reservoir!");
    else Serial.println("pomp uit");
    vorige = pompAan;
  }
}
```

Merk op dat de pomp bij het opstarten eerst uit gaat (`digitalWrite(POMP, LOW)`), en dat de pull-down op de gate haar ook uit houdt in de fractie van een seconde voordien. `unsigned long` en het aftrekken `nu - startCyclus` zorgen dat het programma ook na 49 dagen blijft werken, als `millis()` terug bij nul begint. In opdracht 23.12 voeg je een drukknop toe om de pomp met de hand te laten draaien.

NASLAG · 23.21

Overzicht: halfgeleiders

diode (Shockley)

$$I = I_S\left(e^{U/(nU_T)} - 1\right),\ U_T = kT/q$$

symbool	betekenis	eenheid
I_S	verzadigingsstroom	A
n	idealiteitsfactor (1-2)	–
U_T	25,7 mV bij 25 °C	V

fotonenergie

$$E = \frac{hc}{\lambda} \approx \frac{1240}{\lambda}$$

symbool	betekenis	eenheid
E	energie van een foton	eV
λ	golflengte	nm

voorschakelweerstand

$$R = \frac{U - U_F}{I}$$

symbool	betekenis	eenheid
U_F	drempelspanning van de led (of som van de leds in serie)	V

npn-transistor

$$I_C = \beta I_B \qquad R_B = \frac{U_{\text{pin}} - U_{BE}}{I_B}$$

symbool	betekenis	eenheid
β	stroomversterking	–
U_{BE}	±0,7 V (verzadiging ±0,8 V)	V

verlies in een MOSFET die aan staat

$$P = R_{DS(on)} \cdot I^2$$

symbool	betekenis	eenheid
$R_{DS(on)}$	weerstand tussen drain en source	Ω

spoel

$$U_L = L\frac{\Delta I}{\Delta t} \qquad E = \tfrac{1}{2}LI^2$$

symbool	betekenis	eenheid
L	zelfinductie	H
E	energie in de spoel	J

onderdeel	vuistregels
siliciumdiode	doorlaat ±0,6-0,7 V (tot ±0,9 V bij 1 A); sperstroom µA; 1N4007: 1 A, 1000 V; 1N4148: klein en snel
Schottkydiode	±0,2-0,6 V, snel; 1N5819: 1 A, 40 V; goed als vrijloopdiode
led	rood ±2,0 V, geel ±2,1 V, groen en blauw en wit ±3,0 V; 5 tot 20 mA; lange poot = anode
npn (PN2222A)	β ≥ 100 bij 150 mA; overstuurfactor 5-10; tot enkele honderden mA
n-MOSFET (IRLZ44N)	logic level; ≤ 0,025 Ω bij 5 V; 220 Ω naar de gate, 10 kΩ pull-down; low-side
spoel (pomp, relais, klep)	altijd een vrijloopdiode, kathode aan de plus

kleur	golflengte (nm)	fotonenergie (eV)	materiaal	U_F bij 20 mA (V)
infrarood	±940	1,32	GaAs	1,2-1,4
rood	±630	1,97	AlGaInP	1,8-2,2
geel	±590	2,10	AlGaInP	1,9-2,2
geelgroen	±570	2,18	GaP	2,0-2,4
groen	±525	2,36	InGaN	2,8-3,4
blauw	±465	2,67	InGaN	2,8-3,4
wit (blauw + fosfor)	±455 (blauw)	2,72	InGaN	2,8-3,4
ultraviolet	±395	3,14	InGaN	3,2-3,6

HET GEHEEL · 23.22

Waar halfgeleiders vandaan komen en waar ze heen gaan

Dit hoofdstuk verbindt de leerlijn materie (atoombouw, bindingen, kristallen) met de leerlijn elektronica: dezelfde vier valentie-elektronen die in hoofdstuk 12 en 13 een binding vormden, bepalen hier of er stroom loopt. Met de diode, de led en de transistor heb je de bouwstenen van elke moderne schakeling:

van de zonnecel (een grote pn-overgang die licht omzet in stroom, het omgekeerde van een led) tot de processor met miljarden MOSFETs.

F19 Spanning, stroom en weerstand
Stroom als bewegende lading, en de conventionele stroomzin van de diodepijl.

F24 De microcontroller
De eerste led met voorschakelweerstand, en de 20 mA van een pin.

I12 Atomen en het periodiek systeem
Valentie-elektronen van silicium, fosfor en boor.

I13 Chemische bindingen en stofeigenschappen
De atoombinding die in een halfgeleider soms loskomt.

I20 Zonne-energie dimensioneren
De zonnecel is een pn-overgang; de groeilampen schakelt een MOSFET.

I21 De wetten van Kirchhoff
Het werkpunt is een maaswet met een niet-lineair onderdeel; de gedeelde massa.

I22 De spanningsdeler en sensoren
LDR en NTC zijn halfgeleiders; de lichtsensor stuurt de lampen via een MOSFET.

I24 Condensatoren en de RC-tijd
Een gelijkrichter afvlakken, en de gate van een MOSFET als condensator.

I25 Van analoog naar digitaal
PWM: de MOSFET snel aan en uit om de pomp of de lampen te regelen.

I26 Motoren en actuatoren
Waarom een motor een spoel is, en de H-brug met vier transistors.

E24 Vermogenselektronica en elektrische machines
Omzetters en led-drivers: MOSFETs en diodes die energie verschuiven.

E28 Materiaalkunde
Kristalstructuur en dotering van halfgeleiders in detail.

Habitat
De pomp van de waterkringloop hangt nu aan pin 9 via een IRLZ44N (25 mW verlies), met een 1N5819 over de pomp, een zekering van 2 A en droogloopbeveiliging met de vlotter. Op dezelfde manier sturen pin 10 en 11 de CO_2-klep en de ventilatoren (opdracht 23.13). De groeilampen (14,8 A) vragen meer: in opdracht 23.17 beslis je tussen een relais en een MOSFET met koelplaat. In hoofdstuk 25 regelen dezelfde MOSFETs de lampen en de pomp traploos met PWM.

SAMENVATTING

In het kort

- Zuiver silicium geleidt slecht: alle vier valentie-elektronen zitten in bindingen. Warmte maakt enkele elektronen en gaten vrij.
- Doteren met fosfor (5 valentie-elektronen) geeft n-type met vrije elektronen; met boor (3) p-type met gaten. Het materiaal blijft neutraal.
- Aan een pn-overgang ontstaat een sperlaag. In doorlaatrichting (+ aan p) loopt er stroom boven ±0,6-0,7 V; in sperrichting bijna niets.
- Shockley: $I = I_S(\mathrm{e}^{U/(nU_T)} - 1)$; elke ±0,1 V meer geeft tien keer meer stroom. Rekenen met 0,7 V werkt omdat de karakteristiek zo steil is.
- Een diode maakt van wisselspanning een pulserende gelijkspanning; een brug gebruikt beide halve golven.
- Een led geeft licht met $E \approx 1240/\lambda$ eV per foton; haar drempel ligt dicht bij die energie in volt. Een led heeft altijd stroombegrenzing nodig: $R = (U - U_F)/I$.
- Een npn-transistor schakelt met basisstroom: $I_C = \beta I_B$, in verzadiging begrensd door de last; kies R_B met een overstuurfactor 5-10.
- Een n-MOSFET schakelt met gatespanning; kies een logic-level-type met kleine $R_{DS(on)}$ bij 5 V, met gateweerstand en pull-down.

- Een spoel geeft bij het uitschakelen een spanningspiek; een vrijloopdiode (kathode aan de plus) laat de stroom veilig uitdoven.

BEGRIPPEN

Wat je nu kent

halfgeleider
Stof die slecht geleidt als ze zuiver en koud is, maar veel beter als ze warm wordt, belicht wordt of gedoteerd is, zoals silicium.

doteren
Een heel klein aantal vreemde atomen in een halfgeleider brengen: fosfor geeft vrije elektronen (n-type), boor geeft gaten (p-type).

gat
Plaats waar een elektron in een binding ontbreekt; het gedraagt zich als een positieve lading die kan bewegen.

pn-overgang
Grens tussen p-type en n-type halfgeleider; er ontstaat een sperlaag zonder vrije ladingen.

diode
Halfgeleideronderdeel met één pn-overgang dat stroom doorlaat van anode naar kathode, en bijna niet omgekeerd.

werkpunt
De stroom en spanning waarop een schakeling zich instelt; grafisch het snijpunt van de karakteristiek met de werklijn.

led
Diode die licht geeft als er stroom in doorlaatrichting loopt; de kleur hangt af van het halfgeleidermateriaal.

voorschakelweerstand
Weerstand in serie met een led die de stroom vastlegt: $R = (U - U_F)/I$.

transistor
Halfgeleiderschakelaar of -versterker met drie aansluitingen, waarin een klein signaal een grote stroom stuurt.

MOSFET
Transistor die gestuurd wordt door de spanning op een geïsoleerde gate; als schakelaar heeft hij de weerstand $R_{DS(on)}$.

vrijloopdiode
Diode over een spoel (motor, relais, klep) die de stroom laat uitlopen als de spoel wordt uitgeschakeld, zodat er geen spanningspiek ontstaat.

HANDBOEK · HOOFDSTUK 24

Condensatoren en de RC-tijd

Lading opslaan, tijd maken en een drukknop die maar één keer drukt

Een condensator is twee geleiders met een isolator ertussen. Hij slaat lading en energie op, maar vooral: hij heeft **tijd** nodig om te laden of te ontladen. Met een weerstand erbij krijg je een klok van milliseconden tot minuten, met één getal dat alles vastlegt: de tijdconstante $\tau = R \cdot C$. In dit hoofdstuk reken je laden en ontladen exact uit, meet je de curve met een microcontroller, maak je van een pulserende spanning een vlakke, en zorg je dat de knop 'kasdak' op het bedieningspaneel van de klimaatmodule bij één druk ook echt maar één keer reageert.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 3× code · ± 8 lestijden

Elektriciteit en elektronica · Wiskundig modelleren

Achter de knop 'kasdak' zitten twee weerstanden, een condensator en een chipje: samen maken ze van één druk één signaal.

Na dit hoofdstuk

- Je kan uitleggen hoe een condensator lading opslaat en rekenen met $Q = C \cdot U$ in F, µF, nF en pF, ook vanuit een opdrukcode.
- Je kan het laden en ontladen via een weerstand berekenen met $U = U_0(1 - \mathrm{e}^{-t/\tau})$ en $U = U_0\,\mathrm{e}^{-t/\tau}$, met $\tau = R \cdot C$.
- Je kan de 63 %-regel en de 5τ-regel gebruiken en de tijd tot een drempelspanning berekenen.
- Je kan de energie in een condensator berekenen met $E = \frac{1}{2}CU^2$ en vergelijken met een batterij.
- Je kan uitleggen hoe een condensator een gelijkgerichte spanning afvlakt, en de rimpel schatten.
- Je kan een RC-curve meten met een microcontroller en er τ en C uit bepalen, ook door te lineariseren.
- Je kan een drukknop ontdenderen met een RC-filter en een Schmitt-trigger of in software, en die keuze verantwoorden.

VERHAAL · 24.1

De Leidse fles

In het midden van de achttiende eeuw was elektriciteit een salonattractie. Met een draaiende glazen bol en een wollen doek kon je vonken trekken, maar de lading verdween even snel als ze kwam. In 1745 en 1746 vonden twee onderzoekers, onafhankelijk van elkaar, een manier om ze te bewaren: Ewald von Kleist in Pommeren en Pieter van Musschenbroek in Leiden. Ze laadden een glazen fles met water, met een metalen draad erin, terwijl ze de fles in de hand hielden. Wie daarna tegelijk de draad en de buitenkant aanraakte, kreeg een schok die door arm en borst sloeg. Musschenbroek beschreef ze in een brief als zo hevig dat hij de proef niet wilde herhalen.

De **Leidse fles**, zoals ze in heel Europa ging heten, was de eerste condensator: twee geleiders (het water binnenin, later een laagje metaalfolie, en de hand of een folie buitenom) gescheiden door een isolator, het glas. Rijen flessen naast elkaar gaven grotere vonken. Bijna honderd jaar later onderzocht Michael Faraday hoe de isolator tussen de geleiders de opgeslagen lading beïnvloedt. De eenheid van capaciteit draagt zijn naam: de **farad**.

Vandaag zit er in elk elektronisch toestel een handvol tot enkele honderden condensatoren. Ze zijn meestal kleiner dan een rijstkorrel, en de meeste slaan zo weinig energie op dat je er niets van voelt. Maar het principe is nog altijd dat van de fles: twee geleiders, een isolator, en lading die blijft zitten tot ze een weg vindt.

Waarom dit hoofdstuk ertoe doet

In de klimaatmodule doen condensatoren drie dingen die je elke dag nodig hebt: ze houden de 5 V-rail stabiel, ze filteren storingen weg van de meetingangen, en ze maken tijd: de knop 'kasdak' op het bedieningspaneel wordt er betrouwbaar door.

Leidse flessen: glas met metaalfolie binnen en buiten, de eerste condensatoren.

UITLEG · 24.2

Twee platen, een isolator en een bron

2/21

Wat je al weet. Stroom is lading die beweegt (F19); in een kring loopt ze alleen als die gesloten is (F18). In hoofdstuk 5 zag je dat een condensator exponentieel ontlaadt, met $\tau = R \cdot C$. Hier leer je waarom.

Een **condensator** bestaat uit twee geleiders, de **platen**, met een isolator ertussen: het **diëlektricum**. Sluit je hem aan op een bron, dan duwt de bron elektronen naar de ene plaat en trekt ze weg van de andere. Door de isolator zelf gaat geen stroom, maar in de draden wel: de ene plaat wordt positief, de andere even sterk negatief. Tussen de platen ontstaat een elektrisch veld en een spanning U. Dat gaat door tot die spanning even groot is als die van de bron: dan is de condensator **geladen** en stopt de stroom.

Hoe meer spanning, hoe meer lading er op de platen zit, en die twee zijn recht evenredig. De evenredigheidsconstante is de **capaciteit** C:

LADING OP EEN CONDENSATOR

$$Q = C \cdot U \qquad 1\,\mathrm{F} = 1\,\mathrm{C/V}$$

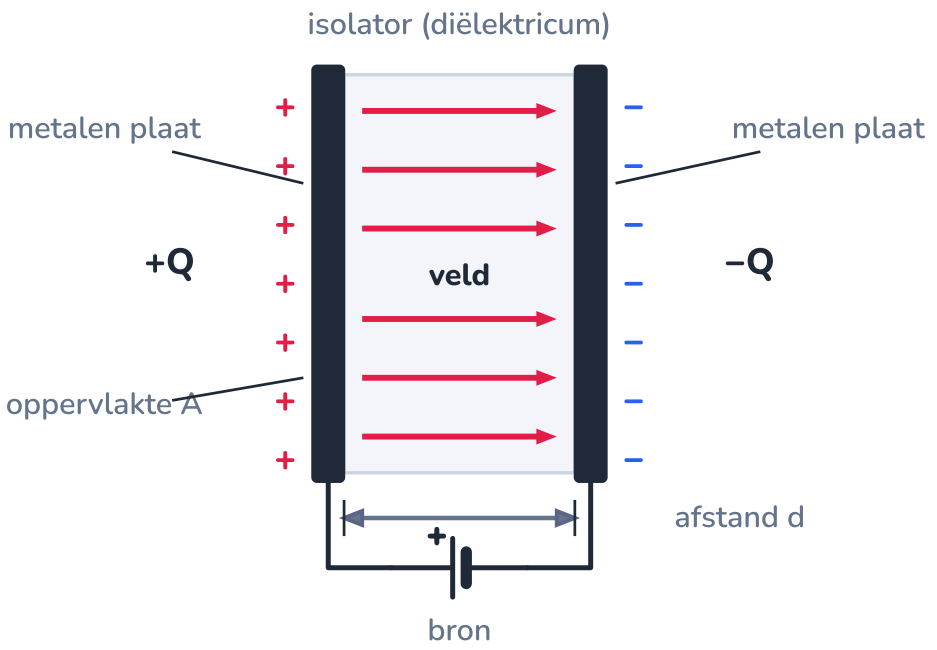
Q: lading op één plaat (de andere heeft $-Q$), in coulomb (C); U in volt; C in farad (F). Eén farad is enorm: in de praktijk werk je met microfarad ($1\,\mu\mathrm{F} = 10^{-6}\,\mathrm{F}$), nanofarad ($10^{-9}\,\mathrm{F}$) en picofarad ($10^{-12}\,\mathrm{F}$).

De capaciteit hangt af van de bouw: **grotere platen** en een **kleinere afstand** geven meer capaciteit, en een isolator die zich laat 'polariseren' (zijn eigen ladingen een beetje verschuift) ook. Lucht doet dat nauwelijks; sommige keramische materialen duizend keer meer dan lucht. Daarom zijn de meeste

condensatoren opgerold of gestapeld: lange, dunne platen met een flinterdunne isolator ertussen. Hoe je de capaciteit uit de afmetingen berekent ($C = \varepsilon A/d$), zie je in hoofdstuk E19.

Er loopt stroom door een condensator

Tijdens het laden meet een ampèremeter in de draad wel stroom, maar geen enkele lading gaat door de isolator: wat links op de plaat aankomt, vertrekt rechts van de andere plaat. Is de condensator geladen, dan loopt er niets meer: een condensator laat **geen gelijkstroom** door. Alleen een *veranderende* spanning geeft stroom in de kring.



De bron verschuift lading van de ene plaat naar de andere; tussen de platen ontstaat een veld.

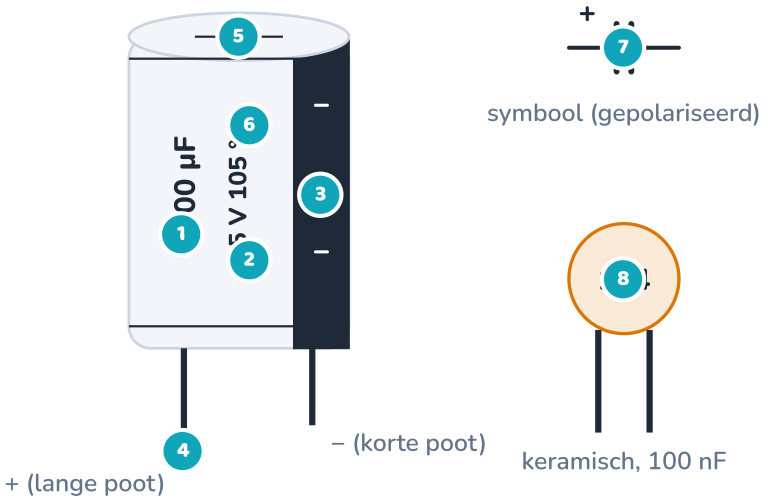
ONTLEED · 24.3

Een elco en zijn opschriften

3/21

Condensatoren zijn er in vier grote families. Welke je kiest, hangt af van hoeveel capaciteit je nodig hebt, bij welke spanning, en hoe nauwkeurig. Hieronder een **elektrolytische condensator** (elco), de grote cilinder in elke voeding, met een keramische condensator ernaast.

soort	typisch bereik	eigenschappen	waarvoor
keramisch (schijfje of SMD)	1 pF tot ±10 µF	klein, goedkoop, niet gepolariseerd; waarde verandert met temperatuur en spanning	ontkoppelen (100 nF), filters, ontdenderen
folie (polyester, polypropyleen)	1 nF tot ±10 µF	stabiel en nauwkeurig, niet gepolariseerd, groter	timing, filters, netspanning
elektrolytisch (elco)	1 µF tot ±100 000 µF	veel capaciteit per volume; **gepolariseerd**; ruime tolerantie (±20 %); veroudert	afvlakken, buffers bij motoren en voedingen
supercondensator	0,1 F tot ±3000 F	enorm veel capaciteit, maar maar ±2,7 V per cel; gepolariseerd	energiebuffer: klok, geheugen, korte stroomonderbreking



Een elco van 1000 µF, 25 V: de streep met mintekens en de korte poot zitten aan de minkant.

1 **Capaciteit**
1000 µF, met een ruime tolerantie: bij elco's vaak ±20 %. Een 'meting' van 900 µF is dus geen defect.

2 **Werkspanning**
25 V is de hoogste spanning die de elco blijvend mag krijgen. Kies ruim: voor een rail van 12,8 V (tot 14,6 V bij het laden) minstens 25 V.

3 **Minstreep**
Een elco is **gepolariseerd**: de isolator is een oxidehuidje dat alleen standhoudt als de min aan de minkant zit. Omgekeerd aangesloten lekt hij stroom, wordt heet en kan openbarsten.

4 **Lange poot = plus**
Nieuwe elco's hebben een langere pluspoot, zoals een led. Na het inkorten blijft alleen de streep over.

5 **Breukgroef**
Een ingeperst kruisje bovenaan: als de elco door een fout gas vormt, scheurt hij daar open in plaats van te ontploffen.

6 **105 °C**
De hoogste temperatuur. Een elco droogt uit: de levensduur staat in de datasheet (bv. 2000 h bij 105 °C) en wordt, als vuistregel, dubbel zo lang per 10 °C koeler.

7 **Symbool**
Twee plaatjes; bij een gepolariseerde condensator een + aan de pluskant (of één gebogen plaat).

8 **Keramisch, code 104**
Kleine condensatoren dragen een code van drie cijfers in picofarad: twee cijfers en het aantal nullen. 104 = 10 · 10^4 pF = 100 nF. Niet gepolariseerd.

UITGEWERKT VOORBEELD · 24.4

Rekenen met Q = C · U

4/21

(a) Welke waarde heeft een condensator met opdruk 473K? (b) Een elco van 470 µF staat op de rail van 12,8 V. Hoeveel lading zit erop, en hoeveel elektronen zijn er van de ene plaat naar de andere verschoven? (c) Hoe groot zou de capaciteit zijn van twee platen van 1 m² op 1 mm afstand in lucht? Hoe groot zouden die platen moeten zijn voor 1 F?

473K

47 · 10³ pF = 47 nF, ±10 %

Q = C · U

470 µF · 12,8 V = 6,02 mC

Opdrukcode en lading: telkens één stap, als je de voorvoegsels goed omzet.

GEGEVEN

- code 473K; $C = 470\,\mu\text{F}$, $U = 12{,}8\,\text{V}$
- $e = 1{,}602 \cdot 10^{-19}\,\text{C}$; lucht: $C = \varepsilon_0 A/d$ met $\varepsilon_0 = 8{,}85 \cdot 10^{-12}\,\text{F/m}$

GEVRAAGD

- waarde van 473K
- Q en het aantal elektronen
- capaciteit van de platen; oppervlakte voor 1 F

OPLOSSING

1. (a) 47 en drie nullen, in picofarad: $47\,000\,\text{pF} = 47\,\text{nF}$. De K betekent ±10 %.
2. (b) $Q = C \cdot U = 470 \cdot 10^{-6}\,\text{F} \cdot 12{,}8\,\text{V} = 6{,}02 \cdot 10^{-3}\,\text{C} = 6{,}02\,\text{mC}$.
3. Aantal elektronen: $\dfrac{6{,}02 \cdot 10^{-3}}{1{,}602 \cdot 10^{-19}} = 3{,}75 \cdot 10^{16}$.
4. (c) $C = \dfrac{8{,}85 \cdot 10^{-12} \cdot 1}{0{,}001} = 8{,}85 \cdot 10^{-9}\,\text{F} = 8{,}85\,\text{nF}$.
5. Voor 1 F moet de oppervlakte $1/(8{,}85 \cdot 10^{-9}) = 1{,}13 \cdot 10^{8}$ keer groter: $A = 113\,\text{km}^2$, een vierkant van ongeveer 10,6 km zijde.

ANTWOORD

473K = 47 nF ±10 %. De elco draagt 6,02 mC, dat zijn 3,8 · 10¹⁶ verschoven elektronen. Twee platen van 1 m² geven maar 8,85 nF; voor 1 F zou je 113 km² nodig hebben.

Klopt dit? De getallen tonen waarom echte condensatoren anders gebouwd zijn: een elco heeft een isolator van oxide die duizenden keren dunner is dan 1 mm, op een geruwde aluminiumfolie met een veel grotere oppervlakte dan je ziet. Zo past 470 µF in een cilindertje van een centimeter.

UITLEG · 24.5

Laden via een weerstand: waarom het steeds trager gaat

5/21

Zet de schakelaar in het schema op 1. Op het eerste moment is de condensator leeg: er staat 0 V over, dus de volle 5,0 V over de weerstand. De stroom is dan $I = 5{,}0\,\text{V}/10\,\text{k}\Omega = 0{,}50\,\text{mA}$. Die stroom brengt lading op de platen, en de spanning U_C stijgt. Maar daardoor blijft er minder spanning over de weerstand, en dus minder stroom: de condensator laadt steeds trager. Volgens de maaswet (hoofdstuk 21):

LAADSTROOM

$$I = \frac{U_0 - U_C}{R}$$

U_0: bronspanning; $U_0 - U_C$: wat er nog 'ontbreekt'.

In een kort tijdje Δt komt er een lading $I \cdot \Delta t$ bij, dus een spanning $\Delta U_C = I\,\Delta t/C = (U_0 - U_C) \cdot \Delta t/(RC)$. Met andere woorden: **in elk tijdje komt er een vaste fractie $\Delta t/(RC)$ bij van wat nog ontbreekt**. Een vaste fractie per stap: dat is exponentieel verval (hoofdstuk 5), hier van het tekort. De grafiek toont het met stappen van $0{,}2\,RC$: telkens 20 % van het resterende tekort erbij. Hoe kleiner je de stappen neemt, hoe dichter je bij de exacte kromme komt, en het getal e duikt vanzelf op:

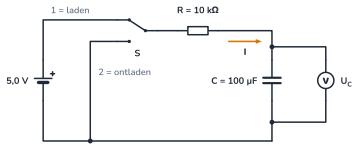
stap Δt	per stap bij	aantal stappen tot τ	na τ ontbreekt nog
0,50 τ	50,0 %	2	0,2500
0,20 τ	20,0 %	5	0,3277
0,10 τ	10,0 %	10	0,3487
0,01 τ	1,0 %	100	0,3660
0,001 τ	0,1 %	1000	0,3677
steeds kleiner			0,3679 = 1/e

Het product $R \cdot C$ bepaalt dus hoe snel alles gaat. Het heet de **tijdconstante** of **RC-tijd** τ (ohm maal farad is seconde). Voor de spanning en de stroom tijdens het laden vanaf 0 V geldt exact:

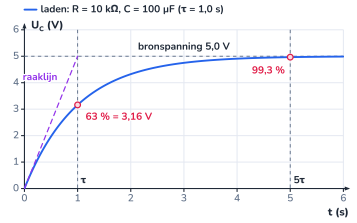
LADEN VAN EEN CONDENSATOR

$$U_C = U_0\left(1 - \mathrm{e}^{-t/\tau}\right) \qquad I = \frac{U_0}{R}\,\mathrm{e}^{-t/\tau} \qquad \tau = R \cdot C$$

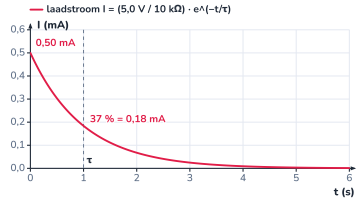
Met $R = 10\,\mathrm{k\Omega}$ en $C = 100\,\mu\mathrm{F}$ is $\tau = 10^4 \cdot 10^{-4} = 1{,}0\,\mathrm{s}$. De afleiding met afgeleiden volgt in hoofdstuk E4.



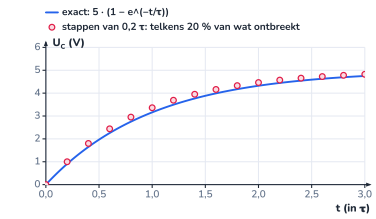
In stand 1 laadt de bron de condensator via R; in stand 2 ontlaadt hij via dezelfde R.



Na τ is 63 % bereikt, na 5τ 99,3 %. De raaklijn in het begin raakt de 5 V precies na τ.



De stroom begint op U_0/R en daalt met dezelfde τ: na 5τ loopt er praktisch niets meer.



Stappen van 0,2 τ (bolletjes) volgen de exacte kromme al goed; kleinere stappen nog beter.

UITLEG · 24.6

Ontladen, en de twee vuistregels

6/21

Zet de schakelaar op 2. De geladen condensator is nu zelf de bron: hij duwt stroom terug door de weerstand. Hoe lager zijn spanning, hoe kleiner die stroom, dus ook het ontladen gaat steeds trager, met dezelfde τ:

ONTLADEN VAN EEN CONDENSATOR

$$U_C = U_0\,\mathrm{e}^{-t/\tau} \qquad I = -\frac{U_0}{R}\,\mathrm{e}^{-t/\tau}$$

Het minteken: de stroom loopt de andere kant op dan bij het laden. De halveringstijd is $t_{1/2} = \tau \ln 2 \approx 0{,}69\,\tau$ (hoofdstuk 5).

Twee vuistregels die je elke dag gebruikt:

- **De 63 %-regel.** Na één τ is 63 % van de verandering gebeurd: bij het laden is 63 % van de eindspanning bereikt, bij het ontladen is nog 37 % over ($\mathrm{e}^{-1} = 0{,}368$). Zo lees je τ af uit een gemeten curve. Ook de **raaklijn** in het begin wijst τ aan: als de condensator in hetzelfde tempo zou blijven laden, was hij na precies één τ vol.
- **De 5τ-regel.** Na vijf tijdconstanten is het verschil kleiner dan 1 %: in de praktijk is de condensator dan vol of leeg. Wiskundig is hij dat nooit helemaal.

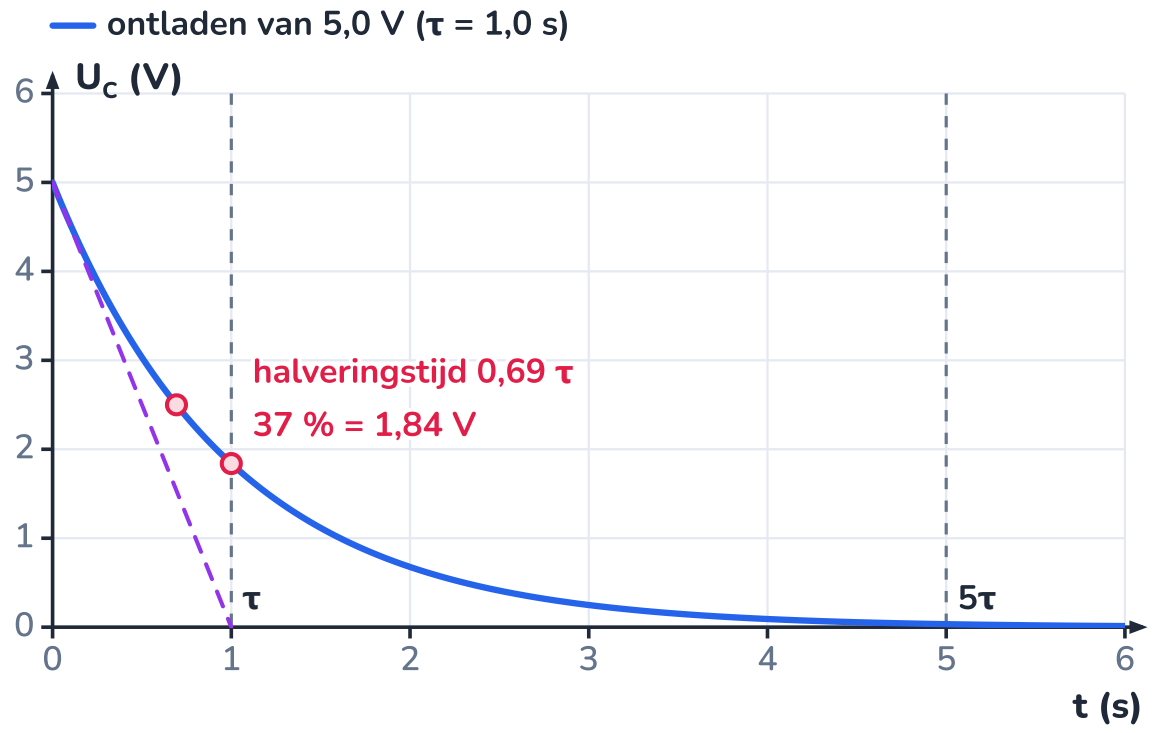
tijd	laden: al bereikt	ontladen: nog over
0,69 τ	50,0 %	50,0 %
1 τ	63,2 %	36,8 %
2 τ	86,5 %	13,5 %
3 τ	95,0 %	5,0 %
4 τ	98,2 %	1,8 %
5 τ	99,3 %	0,7 %

Vaak wil je weten **wanneer** de spanning een drempel U_d haalt, bijvoorbeeld de spanning waarbij een ingang kantelt. Neem de logaritme van de formule (hoofdstuk 5):

TIJD TOT EEN DREMPELSPANNING

$$t_{\text{laden}} = \tau \ln \frac{U_0}{U_0 - U_d} \qquad t_{\text{ontladen}} = \tau \ln \frac{U_0}{U_d}$$

Controle: voor $U_d = 0{,}63\,U_0$ geeft de eerste formule $\tau \ln(1/0{,}37) = \tau$.



Na τ is nog 37 % over; de halveringstijd is 0,69 τ. Ook hier wijst de raaklijn τ aan.

UITGEWERKT VOORBEELD · 24.7

Rekenen met τ: vertragen tot een drempel

7/21

Een condensator van 220 µF laadt via 47 kΩ vanaf 0 V op 5,0 V. (a) Bereken τ en de tijd tot hij 'vol' is. (b) Na hoeveel tijd haalt hij 3,0 V, de spanning waarboven een ingang van de microcontroller zeker 'hoog' leest ($0{,}6 \cdot 5\ \mathrm{V}$)? (c) Je wil dat dezelfde schakeling 2,0 s wacht voor ze de 3,0 V haalt, met dezelfde weerstand. Welke condensator kies je?

GEGEVEN

- $C = 220\ \mu\mathrm{F}$, $R = 47\ \mathrm{k\Omega}$, $U_0 = 5{,}0\ \mathrm{V}$
- drempel $U_d = 3{,}0\ \mathrm{V}$

GEVRAAGD

- τ en 5τ
- tijd tot 3,0 V
- C voor 2,0 s

OPLOSSING

1. (a) $\tau = R \cdot C = 47 \cdot 10^3 \cdot 220 \cdot 10^{-6} = 10{,}34\ \mathrm{s}$; vol na $5\tau = 51{,}7\ \mathrm{s}$.
2. (b) $t = \tau \ln \dfrac{5{,}0}{5{,}0 - 3{,}0} = 10{,}34 \cdot \ln 2{,}5 = 10{,}34 \cdot 0{,}916 = 9{,}47\ \mathrm{s}$.
3. (c) Er moet gelden $\tau \cdot 0{,}916 = 2{,}0\ \mathrm{s}$, dus $\tau = 2{,}18\ \mathrm{s}$ en $C = \tau / R = 2{,}18 / 47\,000 = 46\ \mu\mathrm{F}$.
4. De dichtstbijzijnde gangbare waarde is 47 µF: dan wordt het $47\,000 \cdot 47 \cdot 10^{-6} \cdot 0{,}916 = 2{,}02\ \mathrm{s}$.

ANTWOORD

$\tau = 10{,}34\ \mathrm{s}$ **(vol na ongeveer 52 s); 3,0 V na 9,47 s. Voor 2,0 s: ongeveer 47 µF, dat geeft 2,02 s.**

Klopt dit? Een RC-vertraging is eenvoudig, maar niet nauwkeurig: een elco heeft een tolerantie van ±20 %, en de spanning waarbij een ingang echt kantelt, ligt ergens tussen 1,5 en 3,0 V (dat is alles wat de datasheet belooft). Samen geeft dat gemakkelijk 30 %

spreiding. Waar het op de seconde aankomt, meet je de tijd in software met millis() (hoofdstuk 29); een RC gebruik je voor korte, onkritische tijden, zoals ontdenderen.

PROEF · 24.8

Een RC-curve meten met een microcontroller

8/21

ONDERZOEKSVRAAG

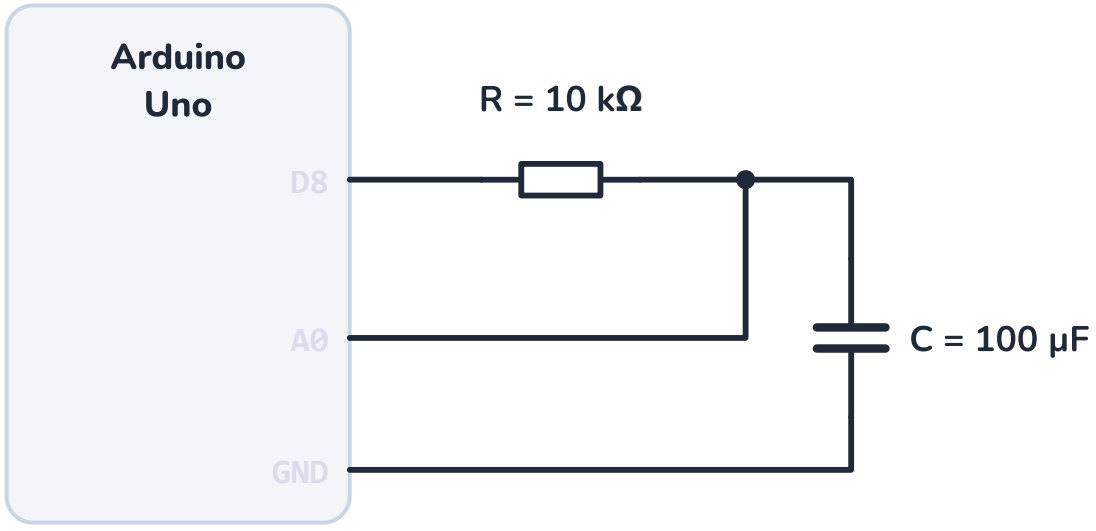
Laadt een condensator echt volgens $U_0(1 - \mathrm{e}^{-t/\tau})$? En hoe groot is zijn capaciteit werkelijk?

MATERIAAL

- Arduino Uno met USB-kabel
- weerstand 10 kΩ (±1 %)
- elco 100 µF (16 V of meer)
- breadboard en draadjes
- computer met de seriële monitor (of Python)

Veiligheid

Zet de elco met de min aan GND. De pin levert hoogstens 0,5 mA: veilig voor pin en condensator.



Een digitale pin is de bron (0 of 5 V), een analoge ingang meet de spanning over de condensator.

WERKWIJZE

1. Bouw de kring: D8 → 10 kΩ → knooppunt → elco → GND; het knooppunt ook naar A0.
2. Laad de schets rc_meting.ino op (volgende blad). Ze ontlaadt eerst 10 s, zet dan D8 hoog en meet 5 s lang om de 0,1 s.
3. Kopieer de uitvoer uit de seriële monitor naar een rekenblad of naar het Python-programma van het blad daarna.
4. Zet de ADC-waarde uit tegen de tijd. Bepaal τ met de 63 %-regel.
5. Bepaal τ ook door te linearisieren, en bereken telkens de capaciteit $C = \tau / R$.

WAARNEMING

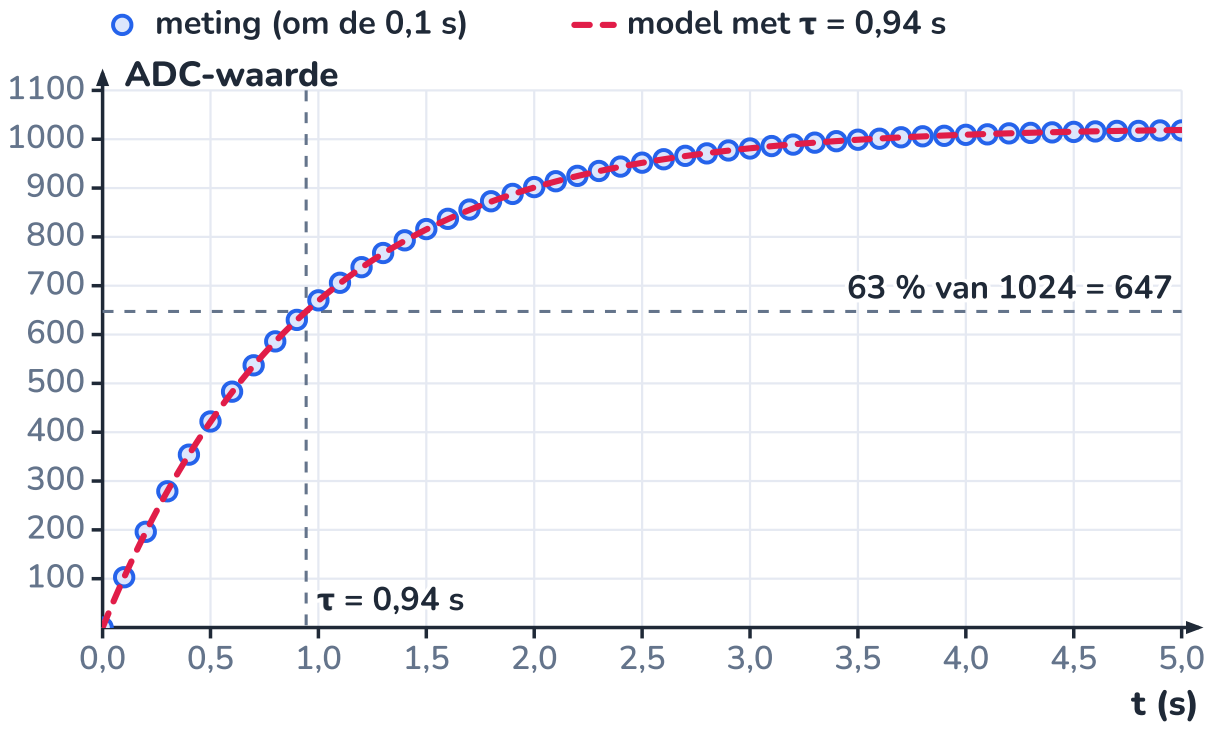
t (s)	0,0	0,3	0,6	0,9	1,0	1,5	2,0	3,0	4,0	5,0
N	0	279	483	630	670	816	902	981	1009	

Voorbeeldmeting (een deel van de 51 waarden). De waarde stijgt snel en dan steeds trager, en blijft net onder 1024. 63 % van 1024 is 647: dat wordt gehaald tussen 0,9 s (630) en 1,0 s (670). Lineair geïnterpoleerd: $\tau = 0{,}94\,\text{s}$.

VERKLARING

…tingen volgen de exponentiële laadcurve (grafiek). …$= 10\,\text{k}\Omega$ geeft $\tau = 0{,}94\,\text{s}$ een capaciteit van $94\,\mu\text{F}$, niet de 100 µF van de opdruk. Geen meetfout: een elco heeft een tolerantie van ±20 %, deze zit 6 % onder de nominale waarde. Lineariseren (volgende bladen) gebruikt alle punten tegelijk en geeft $\tau = 0{,}940\,\text{s}$.

Twee kleine fouten spelen mee. De pin levert niet precies 5,00 V, maar de ADC meet tegen dezelfde 5 V, zodat dat wegvalt (ratiometrisch, hoofdstuk 22). En de ADC rondt af naar beneden: daarom komt de waarde nooit boven 1023, en is een punt vlak bij 1024 onbetrouwbaar voor het linariseren.



De meetpunten liggen op de laadcurve met τ = 0,94 s; het 63 %-punt wijst τ aan.

CODE · 24.9

C++: de laadcurve meten

9/21

Het programma doet alles in `setup()`: het loopt maar één keer. Eerst staat D8 tien seconden laag, meer dan tien tijdconstanten: de condensator is dan zeker leeg. Dan gaat D8 hoog, en op de tijdstippen 0, 100, 200 … ms leest het A0. De uitvoer staat als `t_ms;N` in de seriële monitor, klaar voor een rekenblad.

C++ (Arduino) `rc_meting.ino` 29 regels

```
// rc_meting.ino - de laadcurve van een RC-kring meten
// D8 -- 10 kOhm -- knooppunt -- 100 uF -- GND; het knooppunt hangt ook aan A0.
const int LAAD = 8;
const int MEET = A0;
const unsigned long STAP_MS = 100;       // om de 0,1 s een meting
const int AANTAL = 51;                   // 0 tot 5,0 s

```

```ino
void setup() {
  Serial.begin(9600);
  pinMode(LAAD, OUTPUT);
  digitalWrite(LAAD, LOW);              // eerst volledig ontladen
  delay(10000);                         // 10 s = meer dan 10 tau
  Serial.println("t_ms;N");
  unsigned long start = millis();
  digitalWrite(LAAD, HIGH);             // t = 0: het laden begint
  for (int k = 0; k < AANTAL; k++) {
    while (millis() - start < k * STAP_MS) {
      // wachten tot het volgende meetmoment
    }
    int n = analogRead(MEET);
    Serial.print(k * STAP_MS);
    Serial.print(";");
    Serial.println(n);
  }
  digitalWrite(LAAD, LOW);              // daarna weer ontladen
}

void loop() {
}
```

Waarom een lege `while` en geen `delay(100)`? Een `analogRead` en het versturen van tekst kosten zelf tijd (ongeveer een milliseconde). Met `delay(100)` na elke meting schuiven de meetmomenten telkens iets op; door te wachten tot `millis() - start` het volgende veelvoud van 100 ms haalt, blijven ze op de juiste plaats. Voor een snellere RC-kring (kleiner dan 0,1 s) moet je sneller meten: dan loopt de lus met `micros()`.

CODE · 24.10

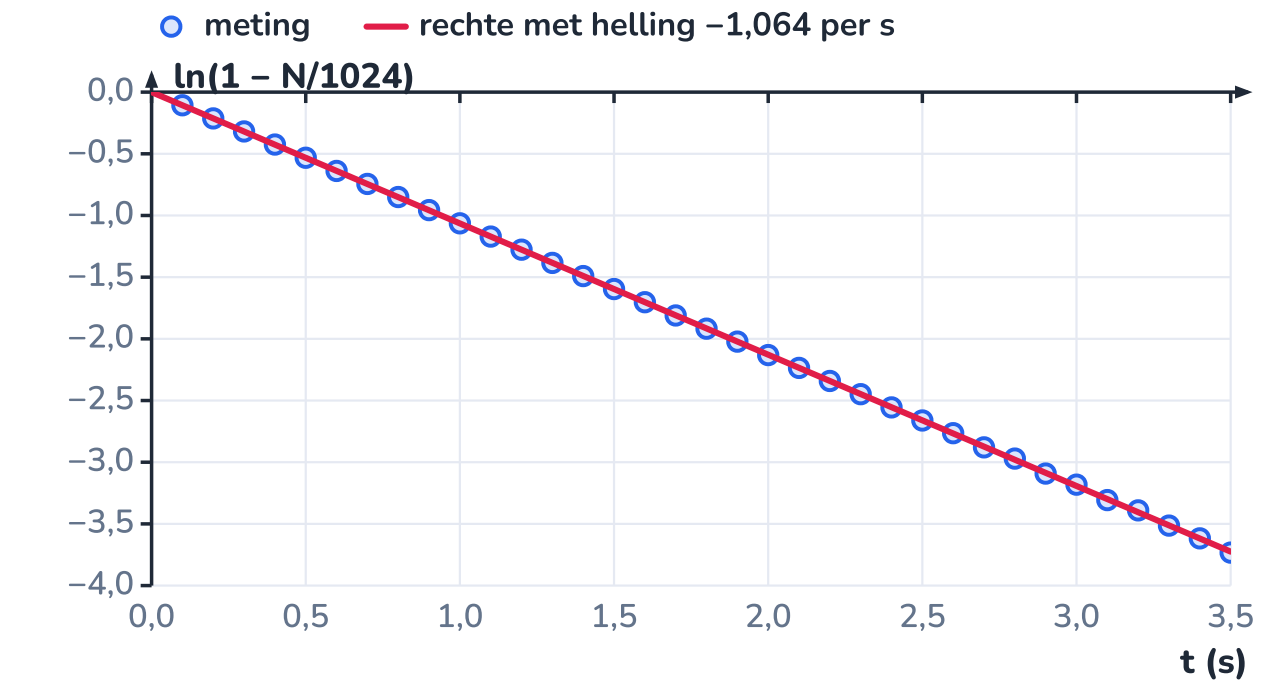
Python: τ uit de meting halen

10/21

Het 63 %-punt gebruikt maar twee metingen. Beter is een methode die alle punten gebruikt: **lineariseren** (hoofdstuk 3). Uit $N/1024 = 1 - \mathrm{e}^{-t/\tau}$ volgt $\ln(1 - N/1024) = -t/\tau$: zet je $\ln(1 - N/1024)$ uit tegen t, dan krijg je een rechte door de oorsprong met helling $-1/\tau$. Numpy zoekt de best passende rechte (`polyfit`, de kleinste kwadraten van hoofdstuk 3).

Python **tau.py** 22 regels

```python
# tau.py - de tijdconstante bepalen uit een gemeten laadcurve
import math
import numpy as np

t = np.arange(0, 5.01, 0.1)                     # s
N = np.array([0, 103, 196, 279, 354, 422, 483, 537, 586, 630, 670, 706, 738, 767, 793, 816, 837, 856, 873, 888, 902,
914, 925, 935, 944, 952, 959, 966, 971, 977, 981, 986, 989, 993, 996, 999, 1001, 1004, 1006, 1007, 1009, 1010, 1012,
1013, 1014, 1015, 1016, 1017, 1017, 1018, 1018])                            # ADC-waarden om de 0,1 s

# methode 1: het 63 %-punt (lineair interpoleren tussen twee metingen)
doel = (1 - math.exp(-1)) * 1024
k = int(np.argmax(N >= doel))                   # eerste meting boven het doel
tau1 = t[k - 1] + (doel - N[k - 1]) / (N[k] - N[k - 1]) * (t[k] - t[k - 1])

# methode 2: lineariseren. 1 - N/1024 = e^(-t/tau), dus ln(1 - N/1024) = -t/tau: een rechte
goed = (t > 0) & (N < 1000)                     # vlak bij 1024 zijn de afrondingsfouten te groot
y = np.log(1 - (N[goed] + 0.5) / 1024)          # + 0,5: het midden van de ADC-stap
helling, snijpunt = np.polyfit(t[goed], y, 1)
tau2 = -1 / helling

R = 10e3
print(f"63 %-punt:      tau = {tau1:.3f} s  ->  C = {tau1 / R * 1e6:.1f} uF")
print(f"lineariseren:   tau = {tau2:.3f} s  ->  C = {tau2 / R * 1e6:.1f} uF  ({goed.sum()} punten)")
print(f"snijpunt met de y-as: {snijpunt:.4f} (ideaal 0)")
```



Na lineariseren liggen de meetpunten op een rechte; de helling is $-1/\tau$.

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
63 %-punt:      tau = 0.943 s  ->  C = 94.3 uF
lineariseren:   tau = 0.940 s  ->  C = 94.0 uF  (35 punten)
snijpunt met de y-as: -0.0001 (ideaal 0)
```

Beide methoden geven ongeveer 94 µF. De punten boven $N = 1000$ laat het programma weg: daar is $1 - N/1024$ heel klein, en één stap afronding verandert de logaritme met tientallen procenten. Dat zie je ook in de grafiek: de punten liggen mooi op een rechte, tot ze onderaan beginnen te wiebelen.

UITLEG · 24.11

Energie in een condensator

11/21

Om lading op een condensator te brengen, moet de bron arbeid leveren: elke extra lading moet tegen de spanning in die er al staat. In het begin gaat dat gemakkelijk (0 V), op het einde tegen de volle U. Gemiddeld is dat tegen $\frac{1}{2}U$, dus is de opgeslagen energie $\frac{1}{2}QU$: de oppervlakte van de driehoek onder de rechte $Q = CU$.

ENERGIE IN EEN CONDENSATOR

$$E = \tfrac{1}{2}\,Q\,U = \tfrac{1}{2}\,C\,U^2$$

E in joule, C in farad, U in volt. Dubbel zoveel spanning geeft vier keer zoveel energie.

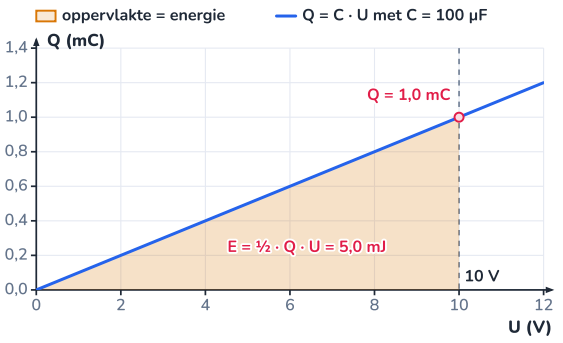
Hoeveel is dat? Een elco van 1000 µF op 25 V bevat $\frac{1}{2}\cdot 10^{-3}\cdot 25^2 = 0{,}31\ \text{J}$: genoeg voor een felle vonk als je hem kortsluit, niet genoeg om iets te laten draaien. Een **supercondensator** van 100 F op 2,7 V bevat $\frac{1}{2}\cdot 100\cdot 2{,}7^2 = 364{,}5\ \text{J}$, ongeveer 0,1 Wh. De LiFePO4-batterij van de klimaatmodule (1,54 kWh, hoofdstuk 16) bevat evenveel als ongeveer 15 200 zulke supercondensatoren.

Een condensator is dus **geen batterij**: per kilogram slaat hij veel minder energie op (een supercondensator orde van grootte 5 Wh/kg, een LiFePO4-batterij ongeveer 110 Wh/kg). Zijn sterkte is **snelheid**: hij kan zijn energie in milliseconden afgeven of opnemen, zo vaak je wil, zonder te slijten zoals

een batterij. Een flitser en een defibrillator laden daarom eerst langzaam een condensator op, en ontladen hem dan in één klap.

Condensatoren blijven geladen

Een condensator houdt zijn lading vast, ook als het toestel uit staat. In voedingen en omvormers die op netspanning werken, zitten elco's van honderden volt: open die nooit, ook niet na het uittrekken van de stekker. In de 12 V-installatie van Habitat is de spanning ongevaarlijk, maar een grote elco die je kortsluit met een schroevendraaier geeft een vonk die metaal wegbrandt.



Bij 100 µF en 10 V: Q = 1,0 mC en E = ½ · 1,0 mC · 10 V = 5,0 mJ.

Van picofarad tot honderd farad: een supercondensator naast gewone elco's en een keramisch exemplaar.

UITGEWERKT VOORBEELD · 24.12

Habitat: een supercondensator als buffer voor de 5 V-rail

12/21

Valt de voeding van de microcontroller even weg (een losse stekker, een omvormer die herstart), dan wil je dat de Mega nog lang genoeg draait om zijn datalogbestand netjes af te sluiten (hoofdstuk 30). De 5 V-rail trekt gemiddeld 0,16 A (hoofdstuk 21), en de Mega werkt op 16 MHz betrouwbaar tot 4,5 V. (a) Hoe lang houdt een elco van 1000 µF de rail boven 4,5 V? (b) En een supercondensator van 1,0 F? (c) Welk deel van de energie in de supercondensator gebruik je daarbij?

GEGEVEN

- $I = 0{,}16\ \text{A}$ (neem aan: constant); van 5,0 V tot 4,5 V
- $C = 1000\ \mu\text{F}$ of $1{,}0\ \text{F}$

GEVRAAGD

- tijd Δt voor beide
- gebruikte fractie van de energie

OPLOSSING

1. De lading die de rail verbruikt, komt van de condensator: $\Delta Q = I \cdot \Delta t$, en $\Delta Q = C \cdot \Delta U$. Dus $\Delta t = \dfrac{C \cdot \Delta U}{I}$.
2. (a) $\Delta t = \dfrac{10^{-3} \cdot 0{,}5}{0{,}16} = 3{,}1 \cdot 10^{-3}\ \text{s} = 3{,}1\ \text{ms}$: veel te kort.
3. (b) $\Delta t = \dfrac{1{,}0 \cdot 0{,}5}{0{,}16} = 3{,}1\ \text{s}$: ruim genoeg om een bestand te sluiten.
4. (c) Gebruikt: $\tfrac{1}{2} \cdot 1{,}0 \cdot (5{,}0^2 - 4{,}5^2) = 2{,}375\ \text{J}$ van de $\tfrac{1}{2} \cdot 1{,}0 \cdot 5{,}0^2 = 12{,}5\ \text{J}$: $19\ \%$.
5. Controle met het vermogen: gemiddeld $0{,}16\ \text{A} \cdot 4{,}75\ \text{V} = 0{,}76\ \text{W}$, maal $3{,}125\ \text{s}$ is $2{,}38\ \text{J}$. Klopt.

ANTWOORD

De elco overbrugt 3,1 ms, de supercondensator 3,1 s. Daarbij gebruik je maar 19 % van zijn energie: de rest zit onder 4,5 V en is voor de Mega onbruikbaar.

Klopt dit? Een supercondensator direct op de rail zetten, kan niet zomaar: leeg is hij bij het inschakelen een kortsluiting. Je laadt hem via een weerstand (met 10 Ω duurt dat 5τ = 50 s) en laat hem via een diode terug leveren; opdracht 24.15 werkt dat uit. Het programma moet ook weten dat de voeding wegviel: het meet de 12,8 V-rail op A3 (hoofdstuk 22).

UITLEG · 24.13

Afvlakken: een condensator na de gelijkrichter

13/21

In hoofdstuk 23 maakte één diode van wisselspanning een **pulserende** gelijkspanning: de helft van de tijd nul. Zet er een condensator over, en het beeld verandert. Tijdens de top laadt de diode de condensator tot bijna de topspanning. Daarna daalt de ingang, de diode spert, en de condensator levert zelf de stroom aan de last. Hij ontlaadt, maar veel te traag om ver te zakken: met 2200 µF en 100 Ω is $\tau = 0{,}22\,\text{s}$, elf keer langer dan een periode van 20 ms. Bij de volgende top laadt de diode hem weer bij.

Wat overblijft, is een gelijkspanning met een kleine **rimpel**. De simulatie (met het diodemodel van hoofdstuk 23 en een transformator met 1,0 Ω inwendige weerstand) geeft voor 2200 µF een uitgang tussen 13,87 en 14,99 V: een rimpel van 1,12 V rond 14,4 V.

Een snelle schatting. Tussen twee toppen levert de condensator bijna een hele periode $T = 1/f$ de laststroom I. Hij verliest dan een lading $I \cdot T$, dus een spanning:

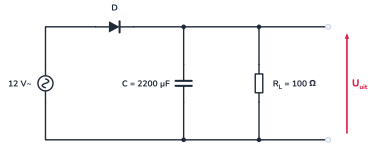
RIMPEL NA AFVLAKKEN

$$\Delta U \approx \frac{I}{f \cdot C}$$

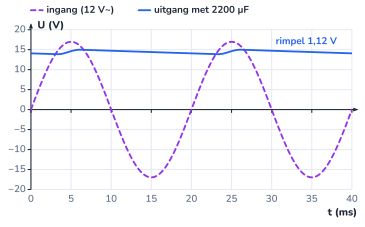
f: frequentie van de toppen: 50 Hz bij één diode, 100 Hz bij een brug (twee toppen per periode).

Voor 2200 µF: $\Delta U \approx 0,144/(50 \cdot 0{,}0022) = 1,31\,\text{V}$, iets meer dan de simulatie, omdat de condensator in werkelijkheid al laadt vóór de periode om is. De schatting is dus aan de veilige kant. Met 1000 µF wordt de rimpel 2,4 V, met 4700 µF 0,54 V; een brug met 2200 µF geeft 0,49 V.

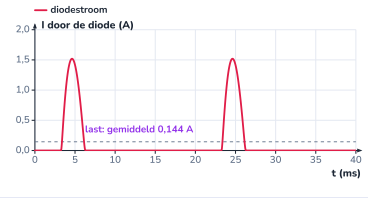
De prijs zie je in de tweede grafiek: de diode laadt de condensator bij in korte stoten van ongeveer 3,1 ms per periode. Omdat de hele lading in die korte tijd moet binnenkomen, is de piekstroom 1,5 A, tien keer de gemiddelde laststroom. Diode en transformator moeten dat aankunnen, en bij het aanzetten, als de condensator nog leeg is, is de stroomstoot nog groter.



De condensator staat parallel over de last, achter de diode.



De uitgang (volle lijn) daalt tussen de toppen maar een beetje: dat is de rimpel.



Gemiddeld levert de diode 0,14 A, maar in stoten van 1,5 A.

UITGEWERKT VOORBEELD · 24.14

Een afvlakcondensator kiezen

14/21

Een transformator van 12 V~ met een bruggelijkrichter moet een last van ongeveer 0,50 A voeden, met een rimpel van hoogstens 1,0 V. (a) Welke capaciteit heb je minstens nodig? Kies uit de reeks 2200, 3300, 4700, 6800, 10 000 µF. (b) Welke werkspanning moet de elco hebben? De netspanning mag 10 % boven 230 V liggen.

GEGEVEN

- brug: $f = 100\,\text{Hz}$; $I = 0{,}50\,\text{A}$; $\Delta U \leq 1{,}0\,\text{V}$
- transformator 12 V~ (topwaarde $12\sqrt{2}$); twee diodes van ±0,7 V in serie

GEVRAAGD

- C
- werkspanning

OPLOSSING

1. (a) $C \geq \dfrac{I}{f \cdot \Delta U} = \dfrac{0{,}50}{100 \cdot 1{,}0} = 5{,}0 \cdot 10^{-3}\,\text{F} = 5000\,\mu\text{F}$. Volgens de schatting is 4700 µF net te klein. De simulatie (last van 28 Ω, ongeveer 0,5 A) is milder: 4700 µF geeft 0,66 V rimpel, 6800 µF 0,46 V.
2. Een elco mag echter 20 % minder capaciteit hebben dan zijn opdruk, en verliest er nog wat bij het verouderen: een 4700 µF kan zo 3800 µF worden. Kies daarom **6800 µF**.
3. (b) Topwaarde: $12 \cdot \sqrt{2} - 2 \cdot 0{,}7 = 15{,}6\,\text{V}$. Met 10 % meer netspanning: $1{,}1 \cdot 12 \cdot \sqrt{2} - 1{,}4 = 17{,}3\,\text{V}$. Zonder last (niets trekt de condensator leeg) wordt het nog iets meer.
4. Een elco van 16 V is dus te krap; kies **25 V**.

ANTWOORD

Een elco van 6800 µF, 25 V. De schatting vraagt 5000 µF; volgens de simulatie geeft 6800 µF een rimpel van 0,46 V, met ruim marge voor tolerantie en veroudering.

Klopt dit? De simulatie geeft minder rimpel dan de schatting, omdat de condensator niet een hele halve periode alleen staat. Dat is goed nieuws: de formule is een veilige bovengrens. Denk ook aan de stroomstoten: in de simulatie laden de diodes de condensator bij met pieken van 2,1 A, ruim vier keer de laststroom. Kies een brug die ruim meer verdraagt dan de gemiddelde stroom (bv. 3 à 4 A).

UITLEG · 24.15

Denderen: waarom één druk tien keer drukt

15/21

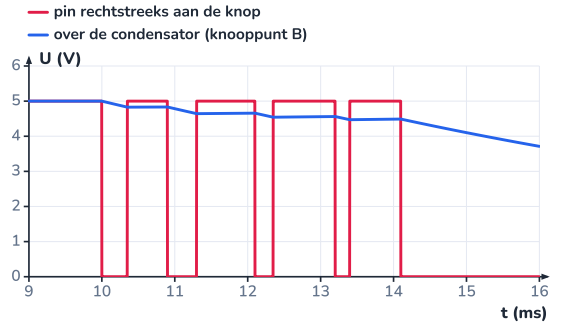
Een drukknop lijkt een perfecte schakelaar: open of dicht. Maar het contact is een metalen veertje dat tegen een ander stukje metaal klapt, en het veert terug. Gedurende enkele milliseconden gaat het contact tientallen keren open en dicht: het **dendert**. Voor een lamp maakt dat niets uit. Een microcontroller leest zijn ingang echter miljoenen keren per seconde, en ziet elke terugsprong als een nieuwe druk.

Hoe lang duurt dat? De ingenieur Jack Ganssle mat het voor achttien verschillende schakelaars. Zestien ervan denderden gemiddeld ongeveer 1,6 ms, en hoogstens 6,2 ms. Maar één mooie rode drukknop denderde bij het *loslaten* soms 157 ms. Veel sprongen waren korter dan een microseconde. Een ontdendering moet dus korte pulsjes negeren, en lang genoeg wachten voor de langzaamste knop die je verwacht.

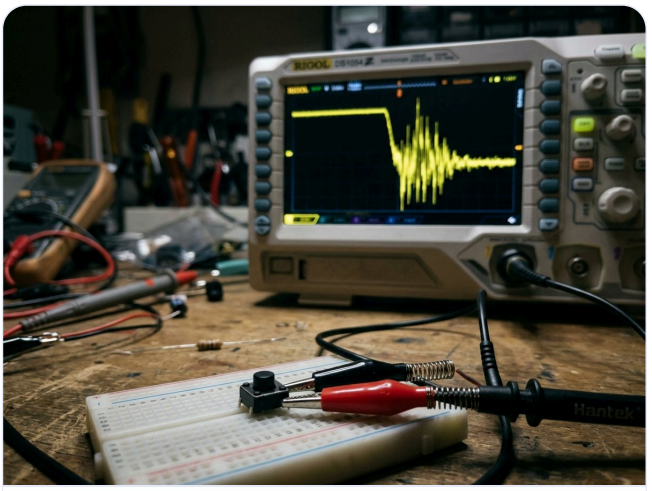
De grafiek toont het model van een versleten knop: 4,1 ms dender bij het indrukken. Rechtstreeks op een pin met pull-up (F24) zou het programma vijf keer 'ingedrukt' zien. Wisselt elke druk het kasdak tussen open en dicht, dan hangt de uitkomst af van toeval: een even aantal sprongen, en er gebeurt niets.

Er zijn drie oplossingen, die je vaak combineert:

1. **Een RC-filter**: een condensator die de snelle sprongen uitmiddelt, zodat de spanning maar traag verandert.
2. **Een Schmitt-trigger**: een ingang met twee drempels die van die trage spanning weer één nette flank maakt.
3. **Software**: een nieuwe toestand pas aanvaarden als het signaal een tijdje stil is gebleven.



Rechtstreeks: vijf keer laag in ruim 4 ms. Over de condensator: de spanning daalt rustig, met kleine knikjes.



Zo ziet dender eruit op een oscilloscoop: tientallen sprongen in enkele milliseconden.

UITLEG · 24.16

RC-filter en Schmitt-trigger

16/21

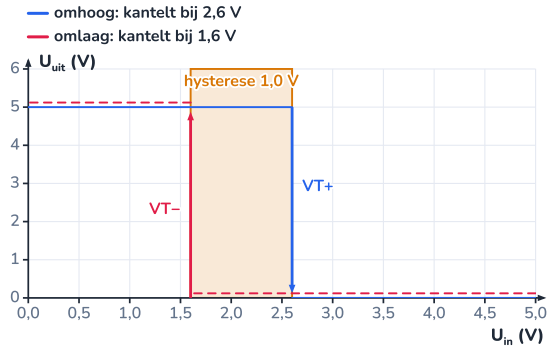
Het filter. In het schema van het volgende blad trekt R_1 het knooppunt A hoog; de knop trekt het naar 0 V. De condensator hangt via R_2 aan A. Wordt de knop ingedrukt, dan ontlaadt de condensator via R_2 (τ = 10 ms); laat je los, dan laadt hij via $R_1 + R_2$ (τ = 20 ms). Een terugsprong van een halve milliseconde verandert de spanning over de condensator maar met een paar procent: de dender verdwijnt in de trage kromme.

Het probleem. Die trage kromme komt nu wel ergens in het gebied tussen 'zeker laag' (onder 1,5 V) en 'zeker hoog' (boven 3,0 V). Een gewone ingang kan daar twijfelen en bij de kleinste storing heen en weer springen. Je hebt dus weer dender, nu elektronisch.

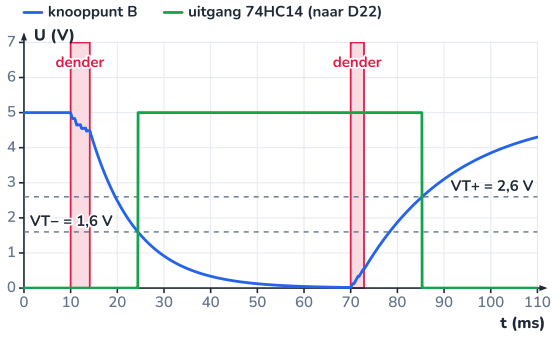
De oplossing: hysterese. Een **Schmitt-trigger** heeft twee drempels: omhoog kantelt hij pas bij V_{T+}, omlaag pas bij V_{T-}, en daartussen onthoudt hij zijn vorige toestand. Het is hetzelfde idee als de thermostaat met twee drempels van het dakraam (F25): één drempel geeft geklapper, twee drempels één nette schakeling. De 74HC14 bevat zes zulke ingangen, telkens met een omkering (hoog in, laag uit). Bij 4,5 V voeding geeft de datasheet typisch $V_{T+} = 2{,}38\,\text{V}$ en $V_{T-} = 1{,}40\,\text{V}$; bij 5 V rekenen we met ongeveer 2,6 en 1,6 V. De spreiding tussen exemplaren is groot: V_{T+} ligt ergens tussen ongeveer 1,8 en 3,5 V, V_{T-} tussen 1,0 en 2,2 V. Een ontwerp moet met die uitersten werken.

En de ingang van de microcontroller zelf?

De pinnen van de ATmega-chips op de Uno en de Mega hebben zelf een kleine Schmitt-trigger, maar de datasheet belooft alleen dat onder $0{,}3 \cdot V_{CC}$ 'laag' en boven $0{,}6 \cdot V_{CC}$ 'hoog' gelezen wordt; hoe groot de hysterese is, staat er niet in. Voor een knop op een lange kabel neem je daarom een echte Schmitt-trigger, of je controleert in software.



Tussen 1,6 en 2,6 V hangt de uitgang af van waar de ingang vandaan komt: dat is hysterese.



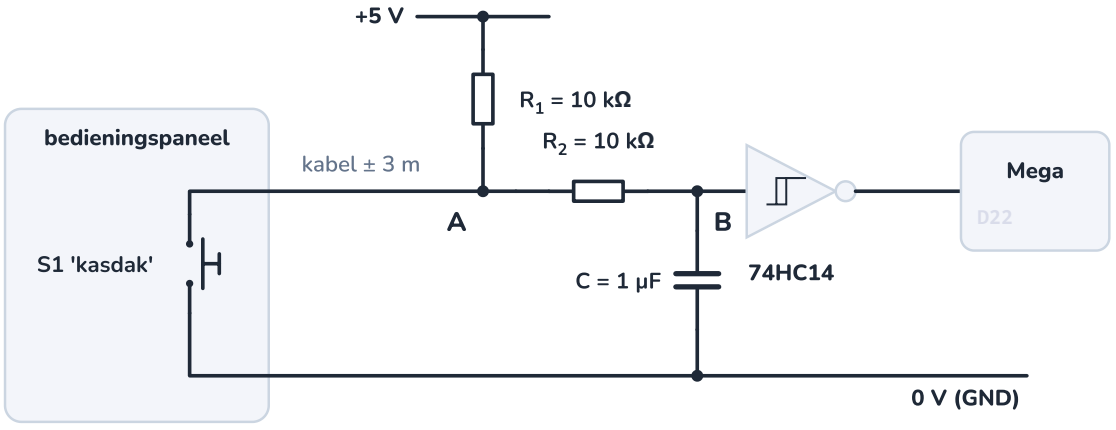
De condensatorspanning (blauw) kruist VT− 14,4 ms na de eerste sprong en VT+ 15,3 ms na het loslaten: telkens één nette flank.

UITGEWERKT VOORBEELD · 24.17

Habitat: de knop 'kasdak' op het bedieningspaneel ontdenderen

17/21

Naast de deur van de kweekruimte komt een bedieningspaneel. De eerste knop, S1 'kasdak', laat het schuifpaneel van het dak openen of sluiten: elke druk wisselt. Het paneel hangt op ongeveer 3 m kabel van de Mega, en in de buurt schakelen de pomp en de dakmotor. Ontwerp de ingang: kies R_1, R_2 en C, bereken de reactietijden met de rekenwaarden van de 74HC14, controleer ze met de uitersten, en kies een pin.



Knop op 3 m kabel, pull-up R1, filter R2-C en een Schmitt-trigger die omkeert: D22 is hoog als S1 ingedrukt is.

GEGEVEN

- knop naar GND; dender tot ±6 ms (Ganssle); 74HC14 bij 5 V: $V_{T+} \approx 2{,}6\,\text{V}$, $V_{T-} \approx 1{,}6\,\text{V}$
- uitersten: V_{T+} 1,8-3,5 V, V_{T-} 1,0-2,2 V
- reactie ruim onder 100 ms (dan voelt het 'meteen')

GEVRAAGD

- R_1, R_2, C
- reactietijden bij indrukken en loslaten
- pin en instellingen

OPLOSSING

1 **Keuze.** $R_1 = R_2 = 10\,\text{k}\Omega$ en $C = 1\,\mu\text{F}$ (keramisch). Ingedrukt loopt er maar $5\,\text{V}/10\,\text{k}\Omega = 0{,}5\,\text{mA}$ door de knop; de condensator ontlaadt via R_2 en stoot dus ook geen grote stroom door het contact.

2 **Indrukken**: ontladen van 5,0 V met $\tau = R_2 C = 10\,\text{ms}$ tot V_{T-}: $t = 10 \cdot \ln\dfrac{5{,}0}{1{,}6} = 11{,}4\,\text{ms}$.

3 **Loslaten**: laden met $\tau = (R_1 + R_2)C = 20\,\text{ms}$ tot V_{T+}: $t = 20 \cdot \ln \dfrac{5{,}0}{5{,}0 - 2{,}6} = 14{,}7\,\text{ms}$.

4 **Uitersten.** Indrukken: tussen $10\ln(5/2{,}2) = 8{,}2$ en $10\ln(5/1{,}0) = 16{,}1\,\text{ms}$. Loslaten: tussen $20\ln(5/3{,}2) = 8{,}9$ en $20\ln(5/1{,}5) = 24{,}1\,\text{ms}$. Altijd langer dan 6 ms dender, altijd ruim onder 100 ms.

5 **Pin.** De Mega heeft nog tientallen vrije digitale pinnen; D22 ligt op de dubbele rij aan de rand, handig voor een kabel naar het paneel. Omdat de 74HC14 omkeert, is D22 **hoog als de knop ingedrukt is**: `pinMode(22, INPUT)`, zonder interne pull-up.

ANTWOORD

$R_1 = R_2 = 10\,\text{k}\Omega$, $C = 1\,\mu\text{F}$ en een 74HC14 naar D22. Typisch reageert de ingang 11 ms na het indrukken en 15 ms na het loslaten; met de slechtste drempels blijft het tussen 8 en 24 ms.

Klopt dit? Het filter doet meer dan ontdenderen: een storingspiek van een paar microseconden die de pomp of de dakmotor in de kabel van 3 m opwekt, verandert de spanning over 1 µF nauwelijks. De 74HC14 heeft zes poorten: de andere vijf zijn voor latere knoppen van het paneel en voor de eindeloopschakelaars van het dak (hoofdstuk 26). De softwarecontrole van het volgende blad laten we toch aanstaan: dubbel zeker, en ze kost niets.

CODE · 24.18

C++: ontdenderen in software

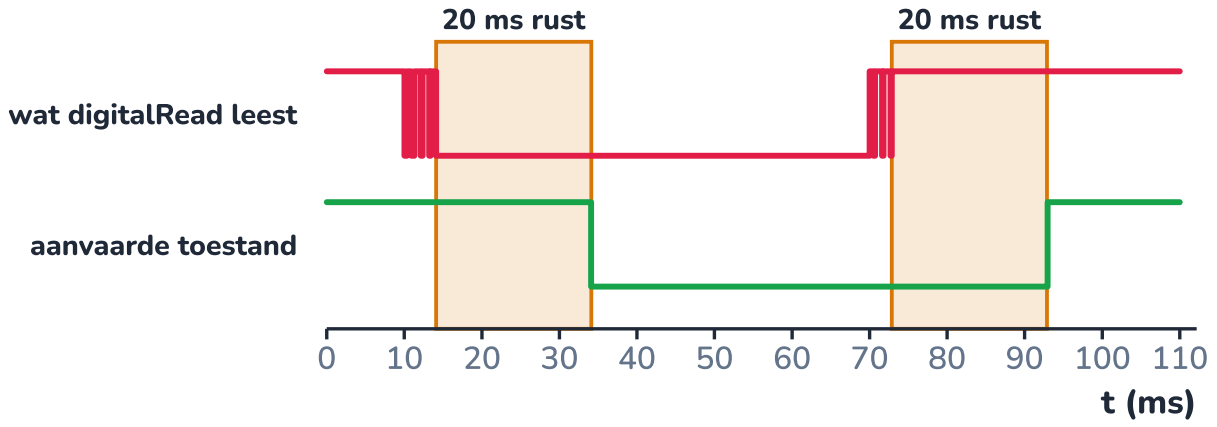
18/21

Zonder extra onderdelen kan het ook: de knop rechtstreeks tussen D22 en GND, met de interne pull-up. Het programma onthoudt wanneer de pin het laatst veranderde, en aanvaardt een nieuwe toestand pas als ze 20 ms onveranderd bleef. Het gebruikt `millis()` en geen `delay()`: de lus blijft ondertussen de vlotter, de pomp en de sensoren bedienen (hoofdstuk 23 en 29). Alleen de **flank** 'net ingedrukt' wisselt het kasdak, anders zou een knop die je vasthoudt het dak voortdurend laten wisselen.

C++ (Arduino) **knop_dak.ino** 40 regels

```cpp
// knop_dak.ino - de knop 'kasdak' ontdenderen in software
// Knop tussen D22 en GND, met de interne pull-up: los = HIGH, ingedrukt = LOW.
// (Met het RC-filter en de 74HC14 van het bedieningspaneel: INPUT en INGEDRUKT = HIGH.)
const int KNOP = 22;
const int INGEDRUKT = LOW;
const unsigned long DENDER_MS = 20;      // zo lang moet het signaal stil blijven

int vorigeLezing = HIGH;                 // wat de pin de vorige keer gaf
int stabiel = HIGH;                      // de aanvaarde toestand
unsigned long laatsteWissel = 0;         // wanneer de pin voor het laatst veranderde
bool dakOpen = false;
unsigned int keerGedrukt = 0;

void setup() {
  pinMode(KNOP, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop() {
  unsigned long nu = millis();
  int lezing = digitalRead(KNOP);

  if (lezing != vorigeLezing) {          // er verandert iets: echt of dender
    laatsteWissel = nu;                  // opnieuw beginnen met wachten
    vorigeLezing = lezing;
  }

  if (nu - laatsteWissel >= DENDER_MS && lezing != stabiel) {
    stabiel = lezing;                    // 20 ms rust: dit is de echte toestand
    if (stabiel == INGEDRUKT) {          // alleen de flank 'net ingedrukt' telt
      keerGedrukt++;
      dakOpen = !dakOpen;                // elke druk wisselt: open <-> dicht
```

```
      Serial.print("druk ");
      Serial.print(keerGedrukt);
      Serial.println(dakOpen ? ": kasdak openen" : ": kasdak sluiten");
    }
  }

  // hier doet de lus ondertussen haar andere werk (pomp, vlotter, sensoren)
}
```



Elke sprong zet de wachttijd opnieuw op nul; pas na 20 ms rust (gele strook) wordt de nieuwe toestand aanvaard.

De grafiek toont wat het programma met de versleten knop doet: het reageert 20 ms na de laatste sprong, precies één keer. Bij het loslaten gebeurt hetzelfde, maar daar volgt geen actie. Een variant reageert meteen op de eerste flank en negeert daarna 20 ms alles: sneller, maar gevoeliger voor één enkele storingspiek. Met het RC-filter en de 74HC14 ervoor is de dender al weg; dan zet je `INGEDRUKT` op `HIGH` en de pin op `INPUT`.

UITLEG · 24.19

Condensatoren overal in de klimaatmodule

19/21

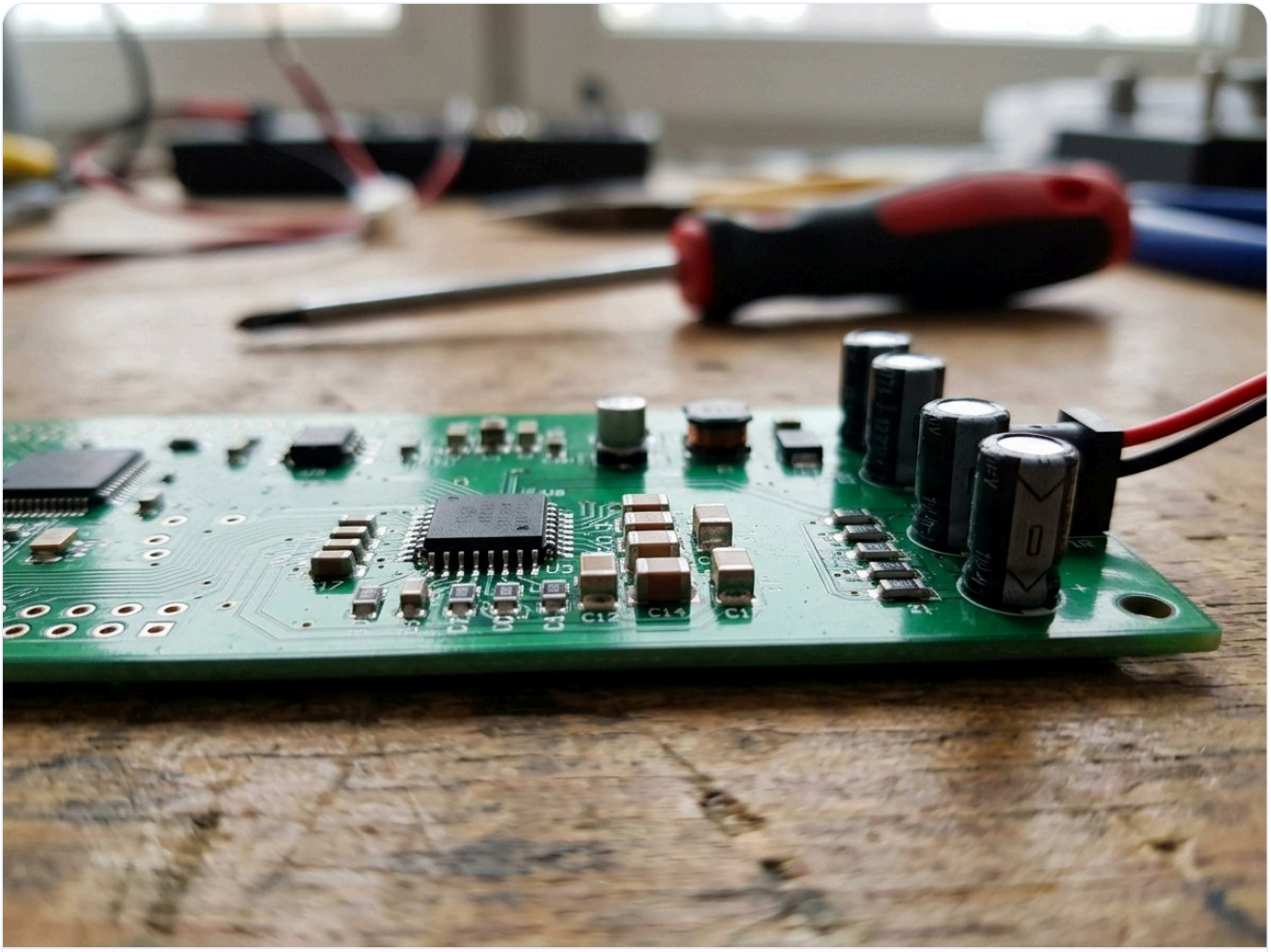
Kijk naar de print van de klimaatmodule en je vindt condensatoren op plaatsen die je nu kan verklaren. Telkens gaat het om hetzelfde: lading die klaarstaat, en een tijd $\tau = R \cdot C$.

waar	R	C	τ = R · C	wat de condensator doet
naast elke chip (Mega, sensoren)		100 nF		levert de korte stroomstootjes van de chip zelf, zodat de 5 V-rail niet hapert (ontkoppelen)
batterijdeler op A3 (hoofdstuk 22)	24,8 kΩ	100 nF	2,5 ms	levert de lading voor de ADC en dempt storingen
gate van de pomp-MOSFET (hoofdstuk 23)	220 Ω	±1,7 nF	0,37 µs	de gate zelf is een condensator: zo snel schakelt de MOSFET
knop 'kasdak' (dit hoofdstuk)	10 en 20 kΩ	1 µF	10 en 20 ms	ontdenderen
automatische reset van een Uno	10 kΩ	100 nF	1 ms	maakt van een spanningsstap een korte resetpuls
bij de H-brug van de dakmotor (hoofdstuk 26)		bv. 470 µF		buffer: vangt de stroompieken van de motor op

Ontkoppelen. Een chip trekt zijn stroom niet gelijkmatig, maar in stootjes van nanoseconden, telkens als zijn transistors schakelen. De draden naar de voeding hebben weerstand en gedragen zich een beetje als een spoel (hoofdstuk 26): de spanning aan de chip zou bij elke stoot even inzakken. Een condensator van 100 nF vlak naast de voedingspinnen levert die stootjes zelf, en wordt tussendoor rustig bijgeladen.

Filteren. Op de batterijdeler (hoofdstuk 22) laadt de ADC bij elke meting een eigen condensatortje van ongeveer 14 pF. De 100 nF op A3 levert die lading; de spanning zakt daarbij maar $14\,\mathrm{pF}/100\,\mathrm{nF}$, ruim een honderdste van een procent. Tegelijk vormen de 24,8 kΩ van de deler en de 100 nF een filter met $\tau = 2{,}5\,\mathrm{ms}$: storingspieken van de pomp, die maar microseconden duren, komen er nauwelijks door. Hoe zo'n filter frequenties scheidt, is het onderwerp van hoofdstuk E21.

Een puls maken. Open je de seriële monitor, dan herstart een Uno. Een lijn van de USB-chip (DTR) springt van 5 naar 0 V, en is via 100 nF met de RESET-pin verbonden, die een pull-up van 10 kΩ heeft. **De spanning over een condensator kan niet springen**: op dat moment was ze 0 V, dus springt RESET mee naar 0 V, en herstart de chip. Daarna laadt de condensator via 10 kΩ, en na enkele milliseconden ($\tau = 1\,\mathrm{ms}$) is RESET weer hoog. Zo maakt een condensator van een blijvende stap een korte puls.



Naast elke chip een ontkoppelcondensator, bij de voeding de grotere elco's.

Overzicht: condensatoren en RC

lading en capaciteit

$$Q = C \cdot U$$

symbool	betekenis	eenheid
Q	lading op een plaat	C
C	capaciteit	F
U	spanning	V

tijdconstante (RC-tijd)

$$\tau = R \cdot C$$

symbool	betekenis	eenheid
τ	na τ is 63 % van de verandering gebeurd	s
R	weerstand	Ω

laden vanaf 0 V

$$U_C = U_0\left(1 - \mathrm{e}^{-t/\tau}\right)$$

symbool	betekenis	eenheid
U_0	bronspanning	V
t	tijd sinds het begin	s

ontladen; stroom bij laden (+) en ontladen (–)

$$U_C = U_0\,\mathrm{e}^{-t/\tau} \qquad I = \pm\frac{U_0}{R}\,\mathrm{e}^{-t/\tau}$$

symbool	betekenis	eenheid
I	stroom in de kring	A

tijd tot een drempel (laden, ontladen)

$$t = \tau \ln\frac{U_0}{U_0 - U_d} \qquad t = \tau \ln\frac{U_0}{U_d}$$

symbool	betekenis	eenheid
U_d	drempelspanning	V

energie

$$E = \tfrac{1}{2} C\,U^2$$

symbool	betekenis	eenheid
E	opgeslagen energie	J

rimpel na afvlakken

$$\Delta U \approx \frac{I}{f \cdot C}$$

symbool	betekenis	eenheid
f	50 Hz (één diode) of 100 Hz (brug)	Hz

tijd	laden: al bereikt	ontladen: nog over
0,69 τ	50,0 %	50,0 %
1 τ	63,2 %	36,8 %
2 τ	86,5 %	13,5 %
3 τ	95,0 %	5,0 %
4 τ	98,2 %	1,8 %
5 τ	99,3 %	0,7 %

opdruk	betekenis	waarde
101	$10 \cdot 10^1$ pF	100 pF
472	$47 \cdot 10^2$ pF	4700 pF = 4,7 nF
104	$10 \cdot 10^4$ pF	100 000 pF = 100 nF
105	$10 \cdot 10^5$ pF	1 000 000 pF = 1 µF
J, K, M achteraan	tolerantie	±5 %, ±10 %, ±20 %

ontdenderen	hoe	goed voor
RC-filter + Schmitt-trigger	bv. 10 kΩ + 10 kΩ + 1 µF, 74HC14	lange kabels, storingen, interrupts
software	nieuwe toestand pas na 20 à 50 ms rust; millis(), geen delay()	korte draden, geen extra onderdelen

HET GEHEEL · 24.21

Waar de condensator vandaan komt en waar hij heen gaat

Dit hoofdstuk koppelt de leerlijn elektronica aan de wiskunde van hoofdstuk 5: hetzelfde exponentiële verband dat bacteriën, afkoelende koffie en radioactieve kernen beschrijft, legt vast hoe snel een condensator laadt. Met $\tau = R \cdot C$ heb je voor het eerst een onderdeel waarbij de **tijd** meespeelt: de kring van nu hangt af van wat er daarnet gebeurde.

F19 Spanning, stroom en weerstand
Stroom als bewegende lading: een condensator is een plaats waar lading zich ophoopt.

F25 Sensoren en automatisch regelen
Twee drempels tegen geklapper: dezelfde hysterese als de Schmitt-trigger.

I5 Groei en verval: exponentiële modellen
Exponentieel verval, halveringstijd en ln: de wiskunde van laden en ontladen.

I3 Verbanden in data
Lineariseren en de best passende rechte, om τ uit een meting te halen.

I22 De spanningsdeler en sensoren
De 100 nF op de batterijdeler en de lading van de ADC.

I23 Halfgeleiders: diode, led en transistor
De gelijkrichter die je hier afvlakt, en de gate van een MOSFET als condensator.

I25 Van analoog naar digitaal
Een RC-filter maakt van een PWM-signaal een gelijkspanning.

I26 Motoren en actuatoren
De spoel, het broertje van de condensator; de knop 'kasdak' stuurt de dakmotor.

I29 Microcontrollers programmeren in C++
millis() in plaats van delay(): de software-ontdendering in een groter programma.

E4 Differentiaalvergelijkingen en simulatie
De RC-kring als dU/dt = (U₀ − U)/τ, opgelost en gesimuleerd.

E19 Elektrische en magnetische velden, inductie
Het elektrisch veld tussen de platen en C = εA/d.

E21 Filters en signaalverwerking
RC-filters en de kantelfrequentie 1/(2πRC).

Habitat

De klimaatmodule krijgt een bedieningspaneel naast de deur van de kweekruimte. De knop S1 'kasdak' hangt via een RC-filter (10 kΩ, 10 kΩ, 1 µF) en een 74HC14 aan D22 van de Mega: hoog betekent ingedrukt, met een reactietijd van 11 à 15 ms. In hoofdstuk 26 stuurt die knop de dakmotor. Als buffer voor de 5 V-rail overbrugt een supercondensator van 1 F ruim 3 s: genoeg om het logbestand af te sluiten (opdracht 24.15).

SAMENVATTING

In het kort

- Een condensator is twee geleiders met een isolator ertussen. Hij slaat lading op: $Q = C \cdot U$, met C in farad (meestal µF, nF of pF).
- Door de isolator loopt geen stroom; in de kring wel, zolang de spanning verandert. Een geladen condensator laat geen gelijkstroom door.
- Laden via een weerstand gaat steeds trager: $U_C = U_0(1 - \mathrm{e}^{-t/\tau})$; ontladen: $U_C = U_0\,\mathrm{e}^{-t/\tau}$, met $\tau = R \cdot C$.
- Na één τ is 63 % van de verandering gebeurd, na 5τ meer dan 99 %. De tijd tot een drempel volgt uit de logaritme.
- Energie: $E = \frac{1}{2}CU^2$. Een condensator slaat weinig energie op, maar kan ze razendsnel afgeven: een buffer, geen batterij.
- Na een gelijkrichter vlakt een condensator de spanning af; de rimpel is ongeveer $I/(fC)$.
- Een drukknop dendert enkele milliseconden. Ontdender met een RC-filter en een Schmitt-trigger (twee drempels), of in software met millis().

BEGRIPPEN

Wat je nu kent

condensator
Onderdeel met twee geleiders (platen) en een isolator ertussen, dat lading en energie opslaat.

capaciteit (van een condensator)
Hoeveel lading een condensator opslaat per volt: $C = Q/U$, in farad (F).

farad
Eenheid van capaciteit: 1 F = 1 C/V. In de praktijk meestal µF, nF of pF.

diëlektricum
De isolator tussen de platen van een condensator; een goed diëlektricum vergroot de capaciteit.

elco
Elektrolytische condensator: veel capaciteit in een klein volume, maar gepolariseerd (plus en min).

RC-tijd
De tijdconstante $\tau = R \cdot C$ van een condensator die via een weerstand laadt of ontlaadt: na τ is 63 % van de verandering gebeurd.

rimpel
Het kleine, periodieke op en neer van een afgevlakte gelijkspanning.

dender
Het snel openen en sluiten van een mechanisch contact vlak na het indrukken of loslaten.

Schmitt-trigger
Ingang met twee drempels (hysterese) die van een trage of rommelige spanning een zuiver digitaal signaal maakt.

supercondensator
Condensator met een enorme capaciteit (farads) bij een lage spanning, gebruikt als kleine energiebuffer.

I INTERMEDIATE · DEEL I6 · ELEKTRONICA

HANDBOEK · HOOFDSTUK 25

Van analoog naar digitaal

Van een spanning naar een getal, en van een getal terug naar licht

De wereld is analoog: een temperatuur of een lichtsterkte kan elke waarde aannemen, op elk moment. Een microcontroller kent alleen enen en nullen. In dit hoofdstuk leer je hoe die twee elkaar vinden: hoe je met bits getallen schrijft, hoe een ADC een spanning in 1024 trapjes meet, hoe vaak je moet meten om niets te missen, en hoe een digitale pin met PWM toch 'een beetje' aan kan staan. Met dat laatste dim je de groeilampen van de klimaatmodule zo, dat ze het daglicht precies aanvullen.

8× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 8 lestijden

Elektriciteit en elektronica · Programmeren en computationeel denken · Wiskundig modelleren

Als de zon zakt, gaan de lampen niet plots aan, maar geleidelijk: zoveel als er aan daglicht ontbreekt.

Na dit hoofdstuk

- Je kan het verschil tussen een analoog en een digitaal signaal uitleggen: bemonsteren in de tijd en afronden in niveaus.
- Je kan getallen omzetten tussen decimaal, binair en hexadecimaal, en uitleggen wat een bit en een byte zijn.
- Je kan de resolutie en het bereik van een ADC berekenen, een spanning omrekenen naar een ADC-waarde en omgekeerd, en uitleggen hoe een ADC met opvolgende benadering werkt.
- Je kan een bemonsteringsfrequentie kiezen en herkennen wanneer aliasing optreedt.
- Je kan een PWM-signaal beschrijven met periode, frequentie en duty cycle, en de gemiddelde spanning en het vermogen berekenen.
- Je kan met analogWrite een led, een ledstrip of groeilampen dimmen via een MOSFET, en de verliezen schatten.

VERHAAL · 25.1

Een wereld in enen en nullen

1/19

In 1948 publiceerde de Amerikaanse wiskundige en ingenieur Claude Shannon een artikel met een bescheiden titel, 'A Mathematical Theory of Communication'. Hij toonde dat je elke boodschap (tekst, een telefoongesprek, een beeld) kan meten in één eenheid: het aantal ja-neevragen dat je nodig hebt om

ze te beschrijven. Die eenheid noemde hij, met een woord van zijn collega John Tukey, de **bit**, van *binary digit*.

Een jaar later bewees Shannon ook hoe vaak je een geluid of een ander signaal moet meten om het perfect te kunnen terugmaken: meer dan twee keer per trilling van de snelste toon erin. Harry Nyquist had het idee al in 1928 aangevoeld. Op die regel rust de muziek-cd uit 1982: 44 100 metingen per seconde, elk in 16 bits, genoeg voor tonen tot boven 20 000 Hz, de grens van het menselijk gehoor.

Een microcontroller doet op kleine schaal hetzelfde. Hij meet een spanning, maakt er een getal van 10 bits van, rekent ermee, en stuurt dan weer iets 'analoogs' aan: een led die half brandt, een motor die half zo snel draait. Hoe dat heen en terug gaat, is het onderwerp van dit hoofdstuk.

Waarom dit hoofdstuk ertoe doet

Elke sensor van Habitat komt binnen via een ADC, en elke actuator die meer kan dan aan en uit, wordt met PWM gestuurd. Wie weet hoe groot een ADC-stap is en hoe vaak je moet meten, weet ook welke metingen je kan vertrouwen.

Het oppervlak van een cd: miljarden putjes en vlakjes, samen 44 100 getallen van 16 bits per seconde en per kanaal.

UITLEG · 25.2

Analoog en digitaal

2/19

Wat je al weet. Een digitale pin is hoog of laag (F24); `analogRead` geeft een getal van 0 tot 1023 (F25), en dat getal hoort bij een spanning in stappen van 4,88 mV (hoofdstuk 22).

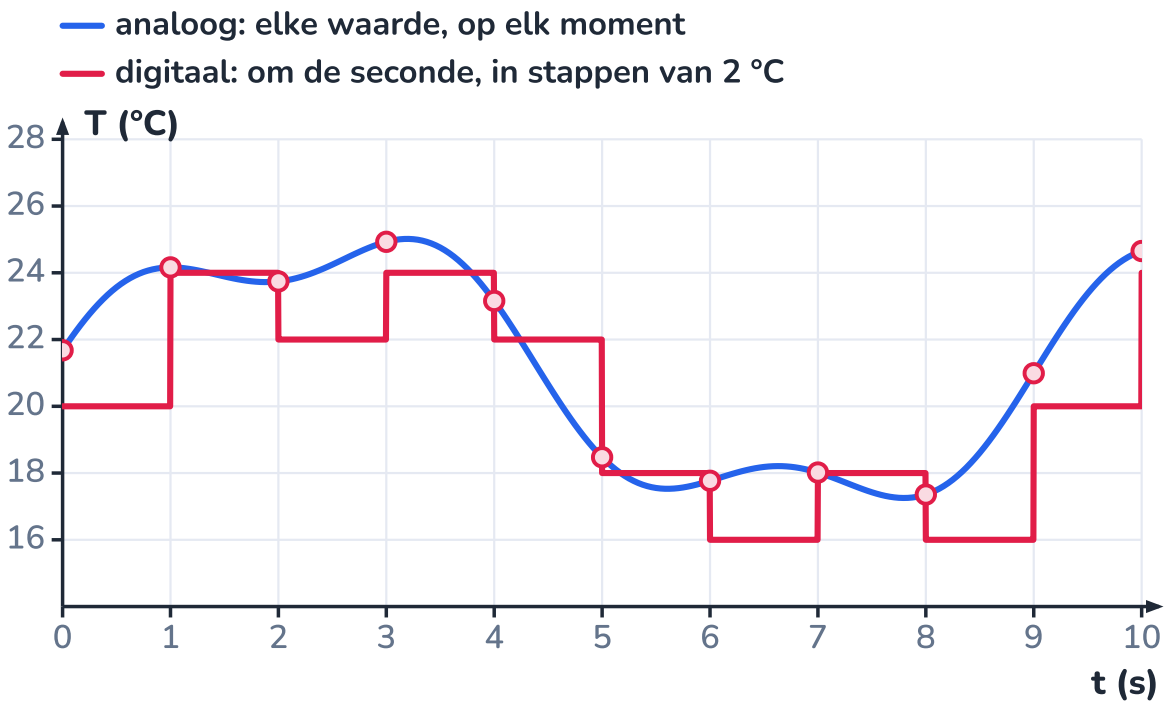
Een **analoog signaal** kan elke waarde aannemen, op elk moment: de temperatuur in de kweekruimte stijgt niet in sprongetjes van een graad, en ze bestaat ook tussen twee metingen. Een **digitaal signaal** is op twee manieren beperkt:

- **in de tijd**: je meet maar op bepaalde ogenblikken, bijvoorbeeld elke seconde (**bemonsteren**);
- **in de waarde**: je rondt af op een eindig aantal niveaus (**kwantiseren**), bij een ADC van 10 bit op 1024 niveaus.

De grafiek toont het met een bewust grove versie: om de seconde gemeten en afgerond op stappen van 2 °C. Wat tussen twee metingen gebeurt, is verloren, en elke meting is tot 2 °C te laag. Met meer niveaus en vaker meten komt de digitale versie dichter bij het origineel, maar het kost meer bits. De kunst is te kiezen wat genoeg is.

Waarom dan digitaal?

Een digitaal signaal kan je zonder verlies kopiëren, versturen, opslaan en bewerken: een 1 blijft een 1, ook na een lange draad met wat ruis. Een analoge spanning verandert bij elke stap een beetje. Daarom meet je zo dicht mogelijk bij de sensor, en reken je daarna digitaal verder.



Het analoge signaal (blauw) en de digitale versie (rood): een trap die pas bij de volgende meting verandert.

UITLEG · 25.3

Het binaire talstelsel

3/19

Ons talstelsel is **decimaal**: tien cijfers, en elke plaats is tien keer zoveel waard als de plaats rechts ervan. In 407 staat de 4 voor $4 \cdot 100$. Een computer heeft maar twee toestanden (spanning of geen spanning) en rekent daarom **binair**: twee cijfers, 0 en 1, en elke plaats is **twee** keer zoveel waard. Eén binair cijfer is een **bit**; acht bits samen zijn een **byte**.

Van binair naar decimaal: tel de gewichten op van de plaatsen met een 1. In de figuur: $1011\,0110_2 = 128 + 32 + 16 + 4 + 2 = 182$.

Van decimaal naar binair: deel telkens door 2 en noteer de rest. De resten, van onder naar boven gelezen, zijn de bits. Voor 200:

deling	quotiënt	rest
200 : 2	100	0
100 : 2	50	0
50 : 2	25	0
25 : 2	12	1
12 : 2	6	0
6 : 2	3	0
3 : 2	1	1
1 : 2	0	1

De resten van onder naar boven: **1100 1000**.

Met n bits schrijf je 2^n verschillende waarden, van 0 tot $2^n - 1$. Een byte gaat dus van 0 tot 255, en een ADC van 10 bit van 0 tot 1023:

bits	1	2	3	4	8	10	12	16
niveaus 2^n	2	4	8	16	256	1024	4096	65 536
grootste getal	1	3	7	15	255	1023	4095	65 535

Bits in je programma

Op de Uno en de Mega is een `int` 16 bits: van −32 768 tot 32 767 (één bit voor het teken). `millis()` geeft een `unsigned long` van 32 bits: tot 4 294 967 295 ms, ongeveer 49,7 dagen. Daarna begint de teller weer bij 0: daarom schrijf je altijd `nu - start >= wachttijd` (hoofdstuk 23 en 29). Ook tekst is binair: de letter A wordt verstuurd als de byte 65.

Elke plaats is twee keer zoveel waard als de plaats rechts ervan; de blauwe bits tellen mee.

UITGEWERKT VOORBEELD · 25.4

Omrekenen tussen decimaal, binair en hexadecimaal

4/19

(a) Schrijf 1023 binair. (b) Welk decimaal getal is $1100\,1000_2$? (c) Een Arduino-programma bevat `0x3FF` en `0xB6`. Welke getallen zijn dat? (d) Hoeveel bits heb je minstens nodig om de ADC-waarden van 0 tot 1023 op te slaan, en hoeveel bytes neemt dat in?

B is 11 = 1011, 6 = 0110: zo lees je hexadecimaal meteen binair.

GEGEVEN

- tabel met machten van 2; hexadecimale cijfers 0-9, A = 10 … F = 15

GEVRAAGD

- 1023 binair
- $1100\,1000_2$ decimaal
- 0x3FF en 0xB6
- bits en bytes voor 0-1023

OPLOSSING

1. (a) $1023 = 1024 - 1 = 2^{10} - 1$: tien enen, $11\,1111\,1111_2$. (Met delen door 2 kom je hetzelfde uit: elke rest is 1.)
2. (b) $1100\,1000_2 = 128 + 64 + 8 = 200$.
3. (c) **Hexadecimaal** (grondtal 16) gebruikt 0-9 en A-F. Eén hexcijfer is precies vier bits, dus een byte is twee hexcijfers. $3\mathrm{FF}_{16} = 3 \cdot 256 + 15 \cdot 16 + 15 = 1023$, of per vier bits: $0011\ 1111\ 1111_2$. $\mathrm{B6}_{16} = 11 \cdot 16 + 6 = 182 = 1011\,0110_2$.
4. (d) $2^9 = 512$ is te weinig, $2^{10} = 1024$ precies genoeg: 10 bits. Computers bewaren per hele byte, dus 2 bytes (16 bits, waarvan 6 ongebruikt).

ANTWOORD

$1023 = 11\,1111\,1111_2$; $1100\,1000_2 = 200$; **0x3FF = 1023 en 0xB6 = 182; 10 bits, dus 2 bytes per ADC-waarde.**

Klopt dit? Hexadecimaal is handig omdat je het zonder rekenen omzet naar binair: elk cijfer wordt vier bits. Je ziet het in de kleurcode van webpagina's (#FF8800: rood 255, groen 136, blauw 0) en in het geheugen van een microcontroller.

UITLEG · 25.5

De ADC: resolutie en bereik

5/19

Een **ADC** (analoog-digitaalomzetter) verdeelt een spanningsgebied, van 0 V tot de **referentiespanning** U_{ref}, in 2^n gelijke stappen. Twee getallen bepalen wat hij kan:

- het **bereik**: van 0 tot U_{ref}. Negatieve spanningen worden 0, spanningen boven U_{ref} worden 1023. (Boven de voedingsspanning mag een pin nooit komen: dan kan hij stukgaan.)
- de **resolutie**: de grootte van één stap, $U_{\text{ref}}/2^n$.

ADC VAN N BIT

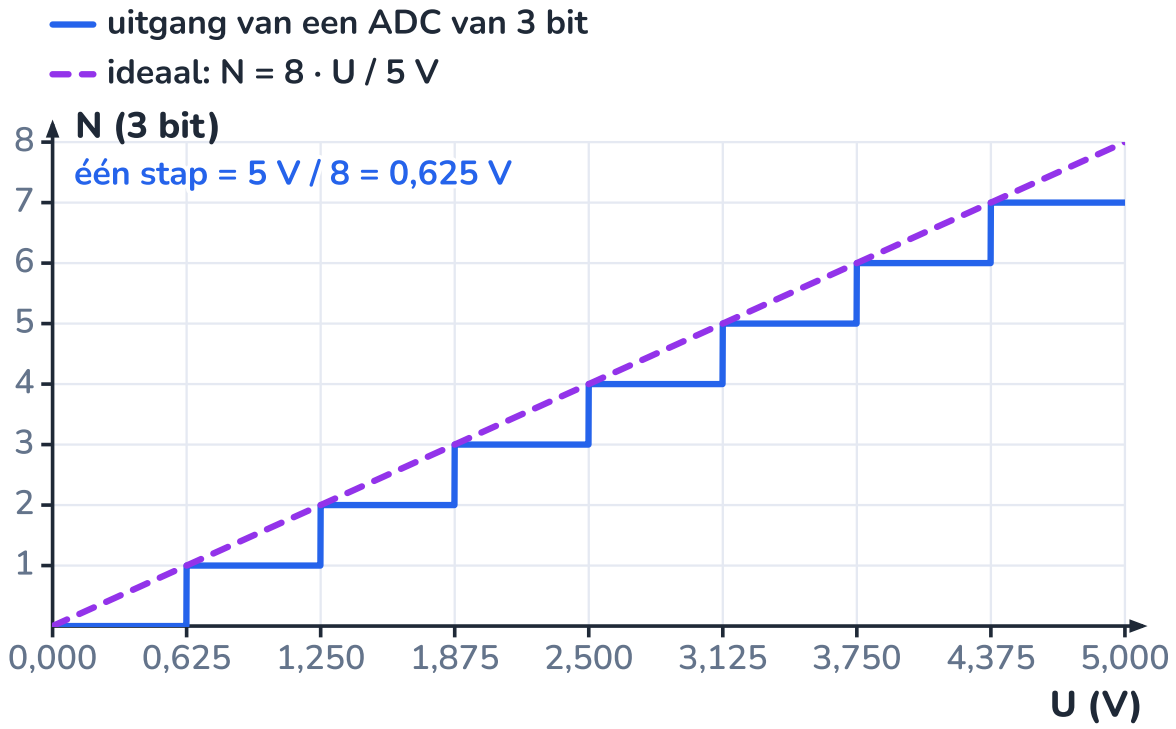
$$N = \left\lfloor 2^n \cdot \frac{U}{U_{\text{ref}}} \right\rfloor \qquad \text{stap} = \frac{U_{\text{ref}}}{2^n} = \frac{5\,\text{V}}{1024} = 4{,}88\,\text{mV}$$

$\lfloor\ \rfloor$: naar beneden afronden. Alles tussen N en $N+1$ stappen geeft hetzelfde getal: de meting zegt 'ergens in deze stap'. De fout door afronden is hoogstens één stap (of een halve, als je met het midden van de stap rekent).

De grafiek toont het principe met maar 3 bits: 8 niveaus, stappen van 0,625 V. Een ADC van 10 bits heeft hetzelfde trappetje, maar met 1024 treden van 4,88 mV.

Resolutie is niet hetzelfde als nauwkeurigheid (hoofdstuk 1). Een stap van 4,88 mV zegt hoe fijn je kan aflezen, niet of het getal klopt. Als de referentie 4,95 V is in plaats van 5,00 V, is elke meting 1 % te hoog, hoe fijn de trapjes ook zijn. En ruis op de sensor maakt de laatste bit vaak onzeker.

De referentie kiezen. Wie kleine spanningen meet, verspilt het grootste deel van de treden. Met `analogReference(INTERNAL)` meet een Uno tegen een ingebouwde referentie van ongeveer 1,1 V (op de Mega `INTERNAL1V1` of `INTERNAL2V56`): de stap wordt $1{,}1\,\mathrm{V}/1024 = 1{,}07\,\mathrm{mV}$. Het bereik krimpt dan wel tot 1,1 V. De interne referentie ligt per chip ergens tussen 1,0 en 1,2 V: ijk ze. En gooi de eerste metingen na het omschakelen weg.



Elke tree is 0,625 V breed; de trap ligt telkens onder de ideale rechte (naar beneden afronden).

UITGEWERKT VOORBEELD · 25.6

Habitat: hoe fijn meet de TMP36 van de meetkas?

6/19

De TMP36 op A1 van de meetkas (F25) geeft $U = 0{,}5\,\mathrm{V} + 0{,}010\,\mathrm{V/°C} \cdot T$. (a) Hoeveel graden is één ADC-stap met de standaardreferentie van 5 V? (b) En met de interne referentie van 1,1 V? Tot welke temperatuur kan je dan meten? (c) Welke ADC-waarde geeft 25,0 °C in beide gevallen? (d) Is 1,1 V een goed idee voor de kas, die op een zonnige dag 32 °C haalt?

GEGEVEN

- $U = 0{,}5 + 0{,}010 \cdot T$ (V, °C)
- 10 bit; $U_{\mathrm{ref}} = 5{,}0\,\mathrm{V}$ of $1{,}1\,\mathrm{V}$

GEVRAAGD

- °C per stap
- bereik met 1,1 V
- ADC-waarde bij 25 °C
- keuze

OPLOSSING

1. (a) Eén stap is 4,88 mV; de sensor geeft 10 mV per graad: $4{,}88/10 = 0{,}49\,^{\circ}\mathrm{C}$ per stap.
2. (b) Met 1,1 V: $1{,}1\,\mathrm{V}/1024 = 1{,}07\,\mathrm{mV}$, dus $0{,}107\,^{\circ}\mathrm{C}$ per stap, bijna vijf keer fijner. Hoogste meetbare spanning 1,1 V: $T = (1{,}1 - 0{,}5)/0{,}010 = 60\,^{\circ}\mathrm{C}$.

3 (c) Bij 25,0 °C is $U = 0{,}750\ \text{V}$. Met 5 V: $N = \lfloor 1024 \cdot 0{,}750/5 \rfloor = 153$. Met 1,1 V: $N = \lfloor 1024 \cdot 0{,}750/1{,}1 \rfloor = 698$.

4 (d) Ja: tot 60 °C is ruim genoeg voor de kas (32 °C), ook in de volle zon. Maar de TMP36 zelf is maar op ongeveer ±2 °C nauwkeurig (daarom de ijkcorrectie van −1,5 °C uit F25): de fijnere stap maakt de meting **vloeiender** (kleine veranderingen worden zichtbaar), niet **juister**.

ANTWOORD

0,49 °C per stap met 5 V, 0,11 °C met 1,1 V (bereik tot 60 °C); 25,0 °C geeft 153 of 698. De interne referentie loont voor een vloeiender verloop, niet voor een juistere temperatuur.

Klopt dit? Met 5 V kan een regeling op 28 °C (het dakraam, F25) alleen kiezen tussen 27,8 en 28,3 °C: net genoeg. Met 1,1 V gaat dat per 0,1 °C. Let op: de andere sensoren op de Uno (bodemvocht, LDR) hebben de 5 V-referentie nodig; wie wisselt tussen referenties, moet na elke omschakeling enkele metingen weggooien.

UITLEG · 25.7

Binnen in de ADC: tien vragen voor tien bits

7/19

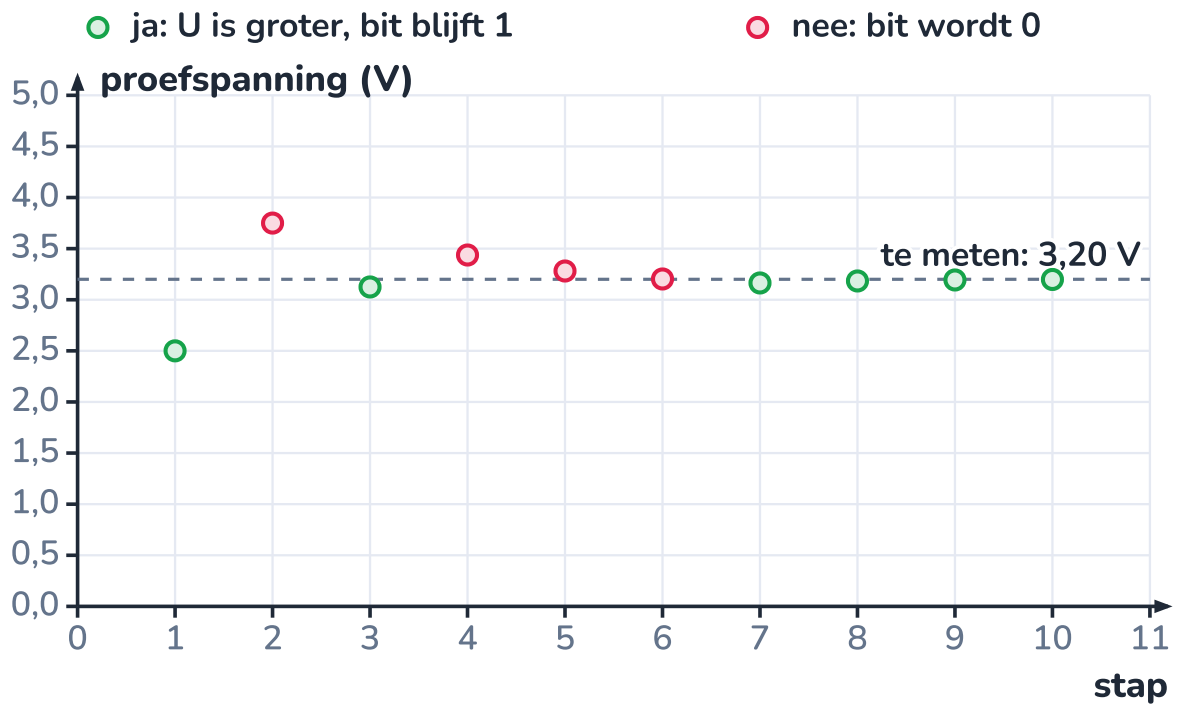
In hoofdstuk 22 bleef open hoe de ADC zijn getal vindt. De ADC van de ATmega werkt met **opvolgende benadering**, het spelletje 'raad mijn getal' met ja-neevragen. Hij heeft een kleine digitaal-analoogomzetter die een proefspanning maakt, en een **comparator** die zegt of de gemeten spanning groter of kleiner is.

1. Eerst houdt hij de ingangsspanning vast op een condensatortje van ongeveer 14 pF (hoofdstuk 24): de **sample and hold**.
2. Dan probeert hij het hoogste bit: is $U \geq 512$ stappen (2,5 V)? Ja: dat bit blijft 1. Nee: het wordt 0.
3. Daarna het volgende bit (256), dan 128, en zo verder tot het laatste bit (1).

Na tien vragen staan alle tien bits vast. Voor $U = 3{,}20\ \text{V}$:

vraag	bit	proefgetal	proefspanning	U ≥ proef?	tussenstand
1	9 (512)	512	2,5000 V	ja: 1	10 0000 0000
2	8 (256)	768	3,7500 V	nee: 0	10 0000 0000
3	7 (128)	640	3,1250 V	ja: 1	10 1000 0000
4	6 (64)	704	3,4375 V	nee: 0	10 1000 0000
5	5 (32)	672	3,2812 V	nee: 0	10 1000 0000
6	4 (16)	656	3,2031 V	nee: 0	10 1000 0000
7	3 (8)	648	3,1641 V	ja: 1	10 1000 1000
8	2 (4)	652	3,1836 V	ja: 1	10 1000 1100
9	1 (2)	654	3,1934 V	ja: 1	10 1000 1110
10	0 (1)	655	3,1982 V	ja: 1	10 1000 1111

Resultaat: $10\,1000\,1111_2 = 655$, en inderdaad is $\lfloor 1024 \cdot 3{,}20/5 \rfloor = 655$. Elke vraag halveert het gebied waarin het antwoord ligt: na tien vragen is dat één stap breed. Op een Arduino kost één `analogRead` 13 tikken van de ADC-klok van 125 kHz: 104 µs, dus hoogstens ongeveer 9600 metingen per seconde.



Elke proef halveert het overblijvende gebied: van 2,5 V naar 3,75, 3,125 … tot op 4,9 mV na 3,20 V.

UITLEG · 25.8

Bemonsteren: hoe vaak moet je meten?

8/19

Een microcontroller meet niet voortdurend, maar op vaste ogenblikken. Het aantal metingen per seconde is de **bemonsteringsfrequentie** f_s, in hertz; de tijd ertussen is $T_s = 1/f_s$. Hoe vaak is genoeg? Dat hangt af van hoe snel het signaal verandert. De temperatuur van het voedingsreservoir verandert in uren: eens per minuut is ruim. Een wisselspanning van 50 Hz verandert elke milliseconde: daar heb je honderden metingen per seconde nodig.

Te traag meten is erger dan weinig informatie. De grafiek toont een signaal van 9 Hz, gemeten met $f_s = 10\,\text{Hz}$. Tussen twee metingen gaat het signaal bijna één keer volledig rond. De metingen liggen daardoor netjes op een **traag** signaal van 1 Hz, dat er in werkelijkheid niet is. Dat heet **aliasing**: een snel signaal doet zich voor als een traag. Je kan het achteraf niet meer herkennen: de metingen van het signaal van 9 Hz zijn precies dezelfde als die van een signaal van 1 Hz. Je ziet het ook in films, waar de wielen van een auto soms achteruit lijken te draaien: de camera bemonstert 24 beelden per seconde.

DE REGEL VAN NYQUIST (INTUÏTIEF)

$$f_s > 2 \cdot f_{\text{max}}$$

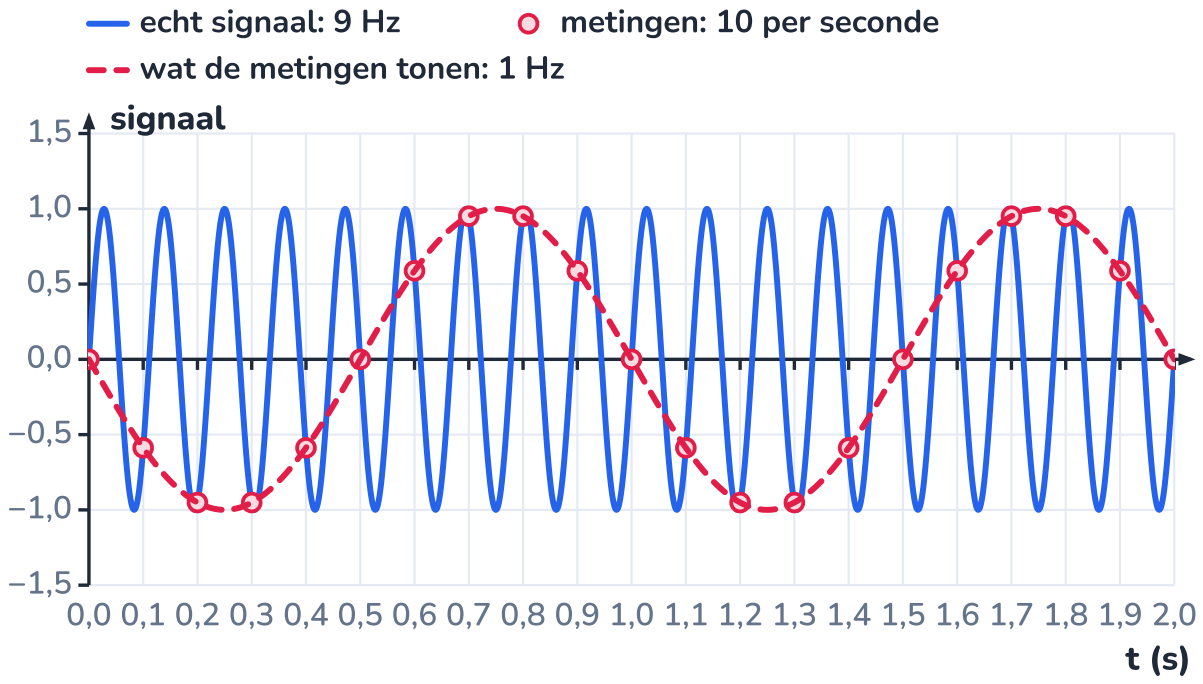
Meet meer dan twee keer per periode van de snelste verandering in het signaal. In de praktijk neem je vijf tot tien keer, of je filtert de snelle veranderingen eerst weg (een RC-filter, hoofdstuk 24). In hoofdstuk E21 zie je waarom de regel werkt en hoe je signalen in frequenties ontleedt.

De schijnbare frequentie is de afstand tot het dichtstbijzijnde veelvoud van f_s: 9 Hz ligt op 1 Hz van 10 Hz, dus je ziet 1 Hz. Een signaal van 50 Hz dat je precies 50 keer per seconde meet, lijkt zelfs stil te staan (0 Hz): je meet elke keer op hetzelfde punt van de golf.

toepassing	bemonstering	bits	datastroom
temperatuur van de kweekruimte	1 keer per 10 s	10	1 bit/s
klimaatmodule: 7 analoge sensoren (A0-A6)	1 keer per 10 s	7 × 16 (2 bytes)	120 960 bytes per dag

toepassing	bemonstering	bits	datastroom
telefoongesprek	8000 per s	8	64 000 bit/s
muziek-cd (stereo)	44 100 per s	2 × 16	1 411 200 bit/s

Meer metingen en meer bits kosten geheugen. De datalogger van de klimaatmodule (hoofdstuk 30) bewaart zeven analoge waarden om de 10 s: 2 bytes per waarde is 120 960 bytes per dag, ongeveer 44 MB per jaar. Als tekst (CSV) met datum en tijd per regel wordt dat al snel vier keer meer.



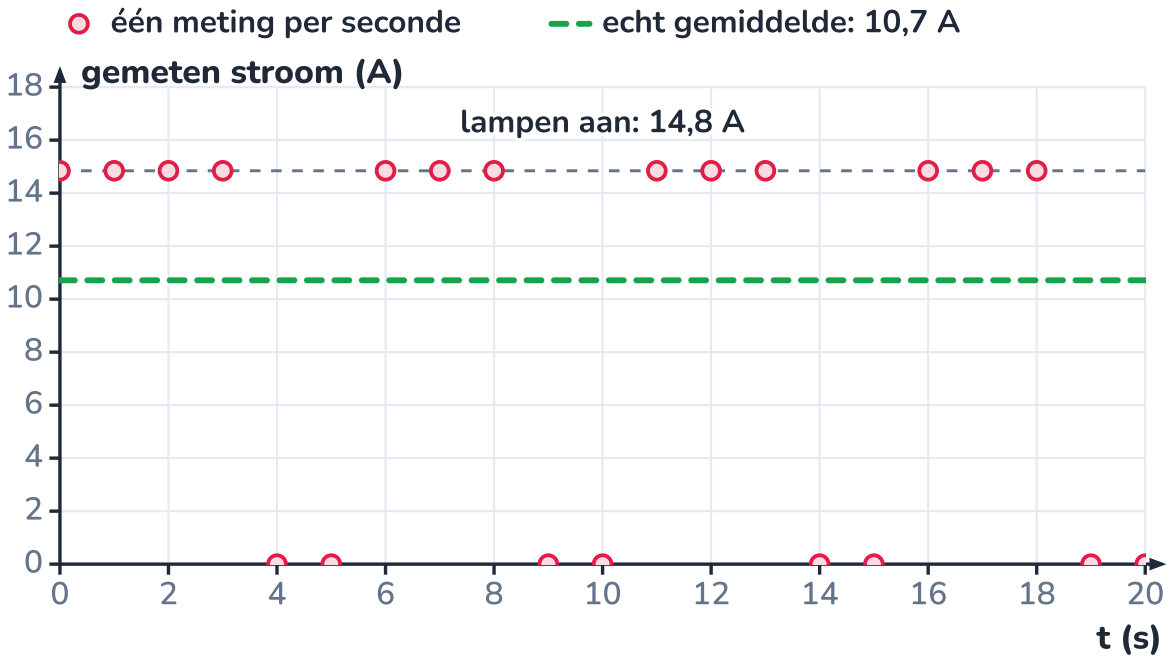
Tien metingen per seconde (bolletjes) van een signaal van 9 Hz liggen precies op een golf van 1 Hz (stippellijn).

UITGEWERKT VOORBEELD · 25.9

Habitat: de lampstroom meten terwijl de lampen gedimd zijn

9/19

De stroomsensor ACS712 op A4 meet de stroom van de batterij (hoofdstuk 16), en het programma telt de lading op om de laadtoestand te kennen. Het meet eenmaal per seconde. Straks worden de groeilampen gedimd met PWM van 490,2 Hz en een duty cycle van 72 % (verder in dit hoofdstuk): de lampstroom springt 490 keer per seconde tussen 0 en 14,8 A. (a) Wat meet het programma? (b) Hoe groot is de gemiddelde stroom echt? (c) Hoe los je het op?



Eén meting per seconde: telkens de volle stroom of niets, in een patroon van 5,1 s. Het gemiddelde (stippellijn) zie je niet.

GEGEVEN

- PWM: $f = 16\,\text{MHz}/(64 \cdot 510) = 490{,}196\,\text{Hz}$; $D = 0{,}722$
- lampen aan: $14{,}8\,\text{A}$; één meting per seconde

GEVRAAGD

- wat de metingen tonen
- het echte gemiddelde
- een betere meetmethode

OPLOSSING

1. (a) Elke meting valt op een willekeurige plaats in een PWM-periode: ofwel 'aan' (14,8 A), ofwel 'uit' (0 A). Tussen twee metingen verlopen er 490,196 perioden. Het deel achter de komma, 0,196, schuift het meetpunt elke seconde 19,6 % van een periode op. De metingen vormen zo een schijnbaar signaal van 0,196 Hz: een patroon dat zich elke 5,1 s herhaalt.

2. (b) Gemiddeld: $D \cdot I = 0{,}722 \cdot 14{,}8 = 10{,}7\,\text{A}$. Eén meting is dus 4,1 A te veel of 10,7 A te weinig.

3. (c) Meet niet één keer, maar een geheel aantal PWM-perioden lang en neem het gemiddelde: bijvoorbeeld 100 metingen in 20,4 ms, precies tien perioden (één `analogRead` duurt 104 µs, dus kies 204 µs tussen twee metingen). Of filter de sensorspanning eerst met een RC-filter, zoals in de proef van dit hoofdstuk.

ANTWOORD

Het programma ziet willekeurig 14,8 A of 0 A, in een patroon dat zich elke 5,1 s herhaalt, terwijl de echte stroom 10,7 A is. Gemiddeld over tien PWM-perioden meet je wel juist.

Klopt dit? Over lange tijd middelt de fout vaak een beetje uit, maar niet altijd: als de PWM-frequentie en het meetritme precies samenvallen, meet je altijd op hetzelfde punt, en is de laadtoestand na een dag volledig fout. Aliasing is geen zeldzaam randgeval: het ontstaat overal waar iets snel schakelt en iets anders traag meet.

UITLEG · 25.10

Van digitaal naar analoog: PWM

10/19

De Uno en de Mega hebben geen **DAC** (digitaal-analoogomzetter): een pin kan niet 2,5 V geven, alleen 0 of 5 V. Maar hij kan heel snel wisselen. Dat is **pulsbreedtemodulatie** (PWM): het signaal gaat met een vaste frequentie aan en uit, en je kiest welk deel van elke periode het aan is: de **duty cycle** D.

PWM

$$D = \frac{t_{\text{aan}}}{T} \qquad f = \frac{1}{T} \qquad U_{\text{gem}} = D \cdot U_{\text{hoog}}$$

T: periode; t_{aan}: hoe lang het signaal hoog is; U_{hoog}: de spanning als het hoog is (5 V aan een pin, 12,8 V over een last achter een MOSFET).

Met `analogWrite(pin, waarde)` kies je de duty cycle in 256 stappen (8 bits): 0 is altijd uit, 255 altijd aan, en daartussen is $D = \text{waarde}/255$. Niet elke pin kan het: alleen de pinnen met een tilde (~) op het bord, die aan een ingebouwde timer hangen.

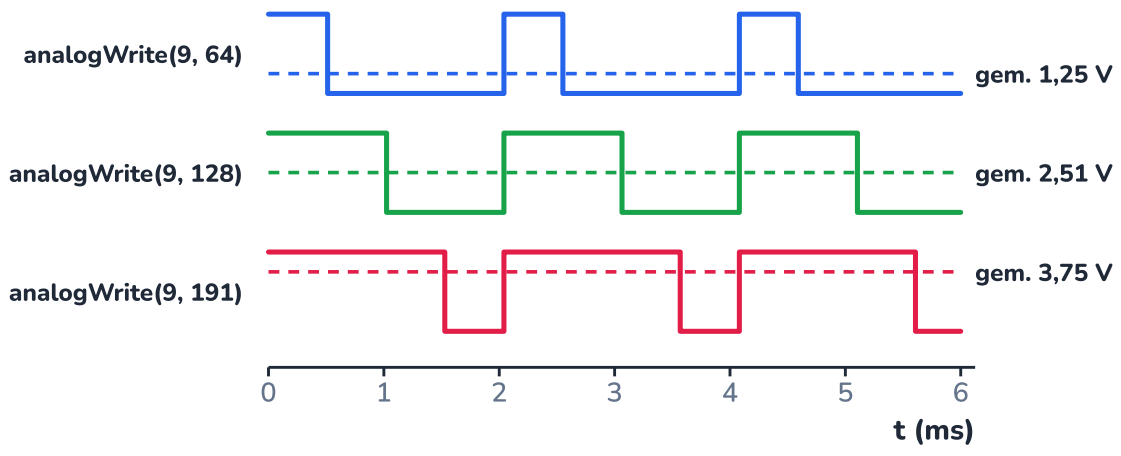
bord	PWM-pinnen	frequentie
Uno (meetkas)	3, 5, 6, 9, 10, 11	490 Hz; pin 5 en 6: 980 Hz
Mega (klimaatmodule)	2 tot 13 en 44 tot 46	490 Hz; pin 4 en 13: 980 Hz

Die frequenties zijn afgerond: de klok van 16 MHz, gedeeld door 64 en door 510, geeft $490{,}2\ \text{Hz}$ (periode 2,040 ms); op de pinnen met 980 Hz is het $16\ \text{MHz}/(64 \cdot 256) = 976{,}6\ \text{Hz}$. Die snelle pinnen delen hun timer met `millis()` en geven een iets grotere duty cycle dan waarde/255.

Waarom werkt dat? Omdat wat je aanstuurt traag is tegenover 2 ms. Een led flikkert echt 490 keer per seconde, maar je oog middelt dat uit: het ziet het gemiddelde licht. Een motor heeft massa en een spoel: zijn toerental en zijn stroom volgen het gemiddelde (hoofdstuk 26). En een RC-filter maakt er een echte gelijkspanning van (de proef hierna).

Half de spanning is half het vermogen

Voor een weerstand die aan PWM hangt, is de gemiddelde spanning $D \cdot U$, maar het gemiddelde vermogen is **niet** $(D \cdot U)^2/R$. Tijdens de aan-tijd krijgt de weerstand de volle spanning en het volle vermogen U^2/R, daarna niets: gemiddeld $D \cdot U^2/R$. Bij 50 % is dat de helft van het volle vermogen, niet een kwart. Hetzelfde voor leds met een voorschakelweerstand: vermogen en licht zijn evenredig met D.



Periode 2,04 ms (490 Hz). Hoe langer het signaal hoog is, hoe hoger het gemiddelde (stippellijn).

UITGEWERKT VOORBEELD · 25.11

De ledstrip van de meetkas dimmen

11/19

De ledstrip van de meetkas (12 V, 0,40 A, 4,8 W) hangt nu aan relais K1 op D7: alleen aan of uit. Je vervangt het relais door een IRLZ44N op pin 9 van de Uno, die nog vrij is en PWM kan. (a) Welke waarde geef je aan `analogWrite` voor 30 % licht? (b) Hoe

lang is de strip dan aan per periode? (c) Welk vermogen neemt de strip, en welke gemiddelde stroom trekt ze? (d) Hoeveel energie spaar je per winterdag (4 h licht, F17) als 30 % volstaat?

GEGEVEN

- ledstrip 12 V, 0,40 A; licht evenredig met de stroom
- Uno pin 9: 490,2 Hz, $D = \text{waarde}/255$

GEVRAAGD

- waarde
- t_{aan}
- vermogen en gemiddelde stroom
- energie per dag

OPLOSSING

1. (a) $0{,}30 \cdot 255 = 76{,}5$: kies 77 ($D = 77/255 = 30{,}2\,\%$).
2. (b) $T = 1/490{,}2\,\text{Hz} = 2{,}040\,\text{ms}$; $t_{\text{aan}} = 0{,}302 \cdot 2{,}040 = 0{,}616\,\text{ms}$.
3. (c) Tijdens de aan-tijd trekt de strip de volle 0,40 A (4,8 W), daarna niets. Gemiddeld: $I = 0{,}302 \cdot 0{,}40 = 0{,}121\,\text{A}$ en $P = 0{,}302 \cdot 4{,}8 = 1{,}45\,\text{W}$.
4. (d) Vol: $4{,}8 \cdot 4 = 19{,}2\,\text{Wh}$; gedimd: $1{,}45 \cdot 4 = 5{,}8\,\text{Wh}$. Je spaart $13{,}4\,\text{Wh}$ per winterdag, ongeveer een vijfde van het winterverbruik van de hele meetkas (62,9 Wh, F21).

ANTWOORD

`analogWrite(9, 77)`: **0,62 ms aan per periode van 2,04 ms; 1,45 W en 0,12 A gemiddeld; 13,4 Wh minder per winterdag.**

Klopt dit? De MOSFET schakelt de hele strip, en in de strip bepaalt de weerstand van elk segment (hoofdstuk 23) nog altijd de stroom: tijdens de aan-tijd is alles zoals zonder PWM. Ons oog vindt 30 % licht overigens niet 'een derde zo fel': het ziet het verschil tussen weinig en een beetje meer veel beter dan tussen veel en iets meer. Voor een lamp voor mensen maakt men daarom de stappen onderaan kleiner (een gammacurve).

PROEF · 25.12

Een PWM-signaal gladstrijken en meten

12/19

ONDERZOEKSVRAAG

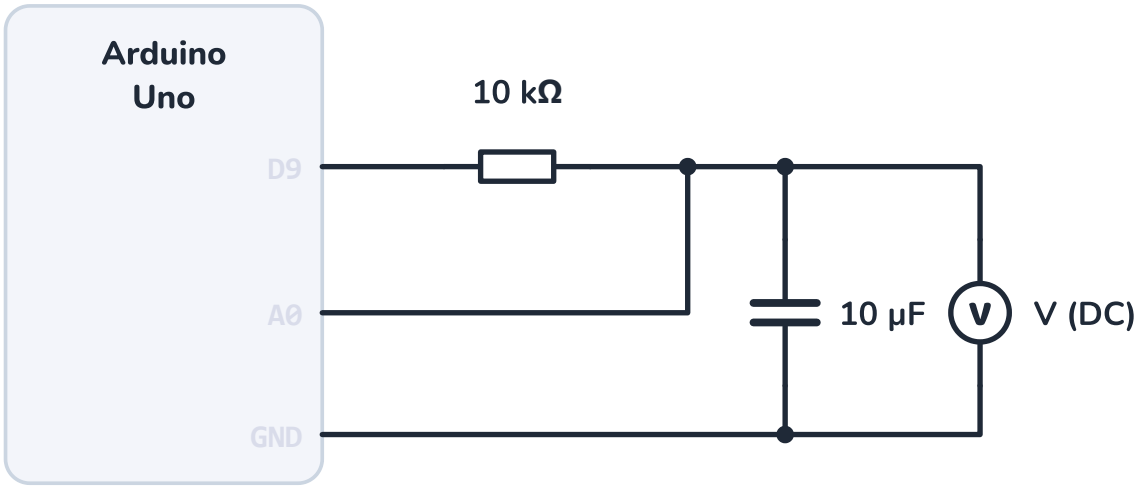
Wordt een PWM-signaal door een RC-filter echt een gelijkspanning van $D \cdot 5\,\text{V}$? En hoeveel rimpel blijft er over?

MATERIAAL

Arduino Uno · weerstand 10 kΩ, elco 10 µF · multimeter · breadboard, draadjes · eventueel een oscilloscoop

Veiligheid

Elco met de min aan GND. Alles op 5 V: geen risico.



Het RC-filter tussen pin 9 en A0; de voltmeter meet de gelijkspanning over de condensator.

WERKWIJZE

1. Bouw het filter: pin 9 → 10 kΩ → knooppunt → 10 µF → GND. Het knooppunt gaat naar A0 en naar de voltmeter (DC).
2. Schrijf een programma dat `analogWrite(9, waarde)` geeft en daarna 200 keer `analogRead(A0)` leest en het laagste en hoogste getal toont.
3. Meet voor de waarden 0, 64, 128, 191 en 255: de voltmeter en het laagste en hoogste A0-getal.
4. Vergelijk met $D \cdot 5\,\mathrm{V}$. Meet ook eens met de voltmeter rechtstreeks op pin 9, zonder filter.

WAARNEMING

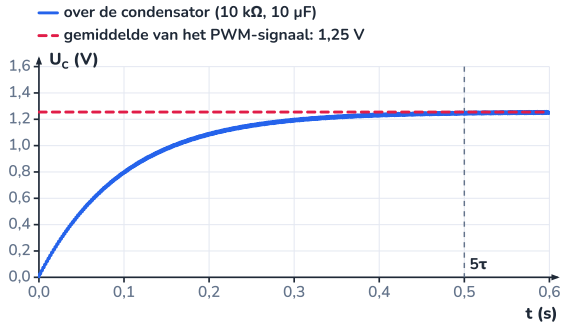
analogWrite	duty cycle	verwacht $D \cdot 5\,\mathrm{V}$	voltmeter	A0 (laagste-hoogste)	rimpel
0	0,0 %	0,000 V	0,00 V	0-0	
64	25,1 %	1,255 V	1,25 V	254-258	
128	50,2 %	2,510 V	2,51 V	511-516	
191	74,9 %	3,745 V	3,75 V	764-768	
255	100,0 %	5,000 V	5,00 V	1023-10	

Voorbeeldmeting (berekend met het exacte RC-model). De voltmeter geeft netjes $D \cdot 5\,\mathrm{V}$. A0 springt een paar stappen op en neer. Rechtstreeks op pin 9 toont de voltmeter bij waarde 128 ook ongeveer 2,5 V: een multimeter op DC middelt zelf al.

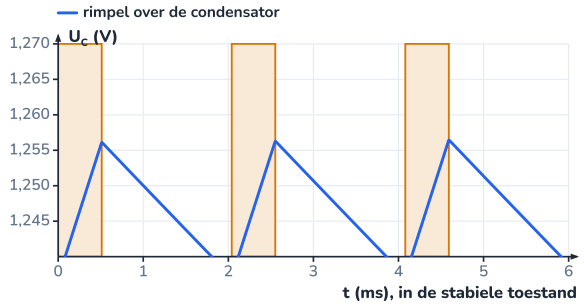
VERKLARING

$= 10\,\mathrm{k\Omega} \cdot 10\,\mu\mathrm{F} = 0{,}10\,\mathrm{s}$, vijftig keer …WM-periode (2,04 ms). Tijdens de aan-tijd laadt de condensator een beetje bij, tijdens de uit-tijd ontlaadt hij een beetje: gemiddeld blijft hij op $D \cdot 5\,\mathrm{V}$ hangen. Na ongeveer 5τ = 0,5 s is dat evenwicht bereikt. De rimpel is ongeveer $U \cdot D(1-D) \cdot T/\tau$: bij 25 % 19 mV, bij 50 % het grootst (ongeveer 26 mV, vijf ADC-stappen).

Zo maak je met PWM en een RC-filter een eenvoudige DAC. Een grotere τ geeft minder rimpel, maar een tragere reactie op een nieuwe waarde: dat is de afweging van elk filter (hoofdstuk E21).



Na ongeveer 0,5 s (5τ) schommelt de spanning rond het gemiddelde van 1,25 V.



Uitvergroot: 19 mV rimpel. In de gele stroken is pin 9 hoog en laadt de condensator.

UITLEG · 25.13

PWM voor motoren en pompen

13/19

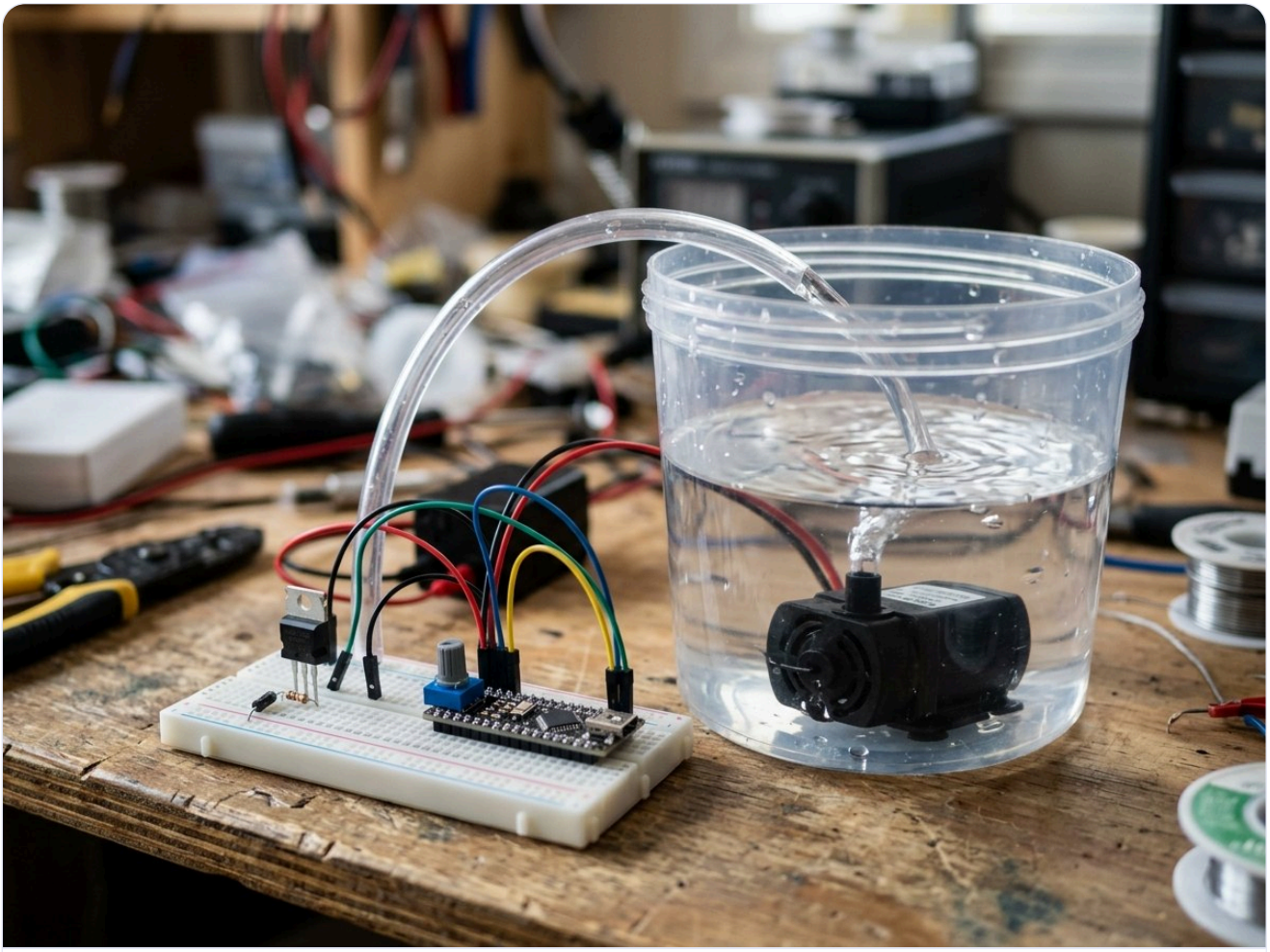
Ook een motor stuur je met PWM, via dezelfde MOSFET als in hoofdstuk 23. Tijdens de aan-tijd krijgt de motor de volle 12,8 V en neemt zijn stroom toe; tijdens de uit-tijd neemt de **vrijloopdiode** de stroom over en neemt ze langzaam af. De spoel van de motor houdt de stroom dus ongeveer constant, en de massa van de rotor houdt het toerental constant: de motor draait alsof hij de gemiddelde spanning $D \cdot 12{,}8\ \mathrm{V}$ kreeg. Zo regel je de pomp van de waterkringloop op D9 traploos, in plaats van 10 minuten per uur vol aan (hoofdstuk 23). Hoe toerental en spanning samenhangen, zie je in hoofdstuk 26.

Twee praktische punten:

- De vrijloopdiode moet **snel** zijn: bij PWM schakelt ze honderden keren per seconde. Daarom koos hoofdstuk 23 een Schottkydiode (1N5819).
- Een motor op 490 Hz **zingt**: de wikkelingen trillen mee en je hoort een toon van 490 Hz. Wie dat stoort, zet de timer op een frequentie boven het hoorbare (rond 31 kHz kan dat met een andere deler). Dan schakelt de MOSFET wel zestig keer vaker, en elke schakeling kost een beetje energie (volgend blad).

⚠ Niet voor alles

PWM werkt voor lasten die traag reageren (lampen voor het oog, motoren, verwarming). Een toestel met eigen elektronica, zoals de 5 V-omvormer of een wifi-module, wil een gewone gelijkspanning: 490 keer per seconde aan en uit laat het telkens opnieuw opstarten.



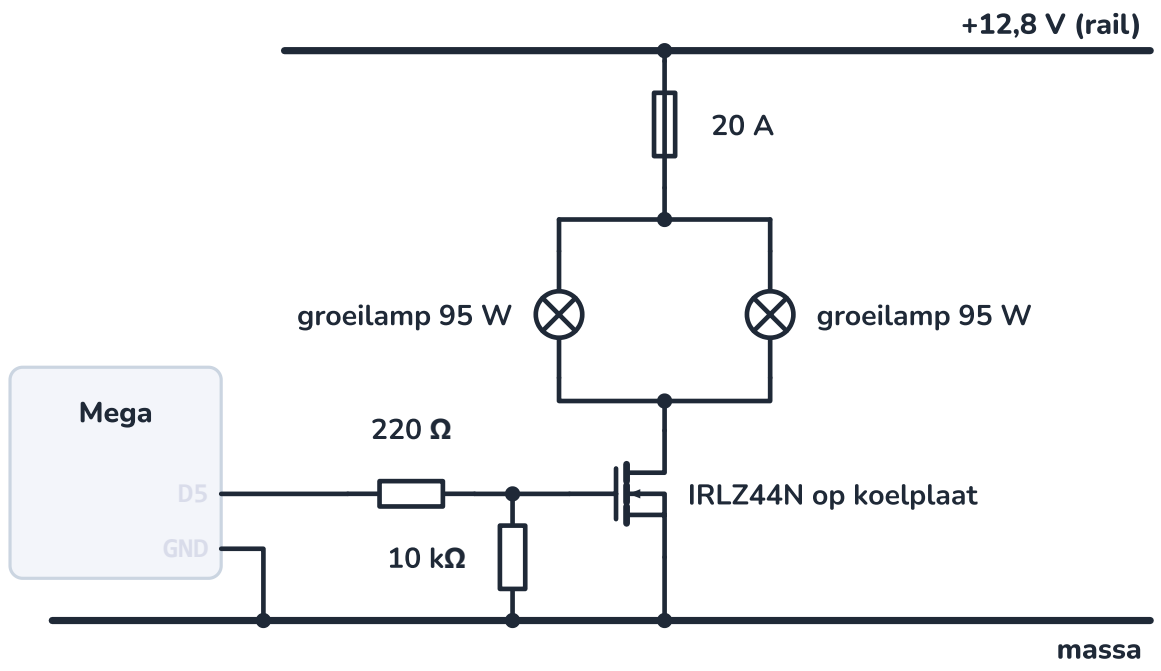
Met PWM en een potentiometer regel je het debiet van een pompje traploos.

UITGEWERKT VOORBEELD · 25.14

Habitat: de groeilampen traploos dimmen

14/19

In hoofdstuk 22 gingen de groeilampen aan onder 4000 lux en uit boven 6000 lux daglicht. Nu wil je dat het kweekbed tijdens de lichtperiode (16 uur) altijd **200 µmol/(m²·s)** krijgt: daglicht plus lamplicht. De twee lampen samen geven op 100 % precies 200 (hoofdstuk 22). Ze hangen via één IRLZ44N op een koelplaat (hoofdstuk 23) aan een PWM-pin van de Mega. (a) Kies de pin. (b) Welke PWM-waarde hoort bij 3000 lux daglicht? (c) Hoeveel vermogen nemen de lampen dan, en hoeveel verliest de MOSFET? (d) Hoe warm wordt hij?



Eén MOSFET schakelt beide lampen; een zekering van 20 A beschermt de tak. Leds zijn geen spoel: geen vrijloopdiode nodig.

GEGEVEN

- lampen: 190 W, 14,8 A op 12,8 V; $R_{DS(on)} = 0{,}025\,\Omega$; MOSFET + koelplaat $7\,^\circ\mathrm{C/W}$
- zonlicht: 54 lux per µmol/(m²·s); schakeltijd per flank ±3 µs
- vrije PWM-pinnen: D3-D8, D12, D13, D44-D46

GEVRAAGD

- pin
- PWM-waarde
- vermogen lampen en verlies MOSFET
- opwarming

OPLOSSING

1. (a) **D5**: 490 Hz (timer 3), met $D = \text{waarde}/255$. D4 en D13 geven 980 Hz en een iets afwijkende duty cycle; D44-D46 verliezen hun PWM als er ooit een servo bij komt (de servobibliotheek gebruikt op de Mega eerst timer 5); D6-D8 houden we vrij voor de dakmotor (hoofdstuk 26).

2. (b) Daglicht: $3000/54 = 55{,}6$ µmol. Tekort: $200 - 55{,}6 = 144{,}4$, dus $D = 144{,}4/200 = 0{,}722$ en de waarde $0{,}722 \cdot 255 \approx 184$. (Het programma leest via de LDR $N = 428$, rekent dat terug naar 2983 lux en komt op 185: één stap meer, door het afronden van de ADC.)

3. (c) Lampen: $0{,}722 \cdot 190 = 137\,\mathrm{W}$. Geleiding in de MOSFET: tijdens de aan-tijd $0{,}025 \cdot 14{,}8^2 = 5{,}51\,\mathrm{W}$, gemiddeld $0{,}722 \cdot 5{,}51 = 3{,}97\,\mathrm{W}$.

4. Schakelen: bij elke flank staan even spanning én stroom over de MOSFET: ongeveer $\tfrac{1}{2} \cdot 12{,}8 \cdot 14{,}8 \cdot 3 \cdot 10^{-6} = 0{,}28\,\mathrm{mJ}$, twee flanken per periode, 490 keer per seconde: $0{,}28\,\mathrm{W}$. Samen $4{,}3\,\mathrm{W}$.

5. (d) $4{,}3\,\mathrm{W} \cdot 7\,^\circ\mathrm{C/W} = 30\,^\circ\mathrm{C}$ boven de omgeving; op 100 % (geen schakelen) $5{,}51 \cdot 7 = 39\,^\circ\mathrm{C}$.

ANTWOORD

D5, `analogWrite(5, 184)` bij 3000 lux. De lampen nemen 137 W; de MOSFET verliest ongeveer 4,3 W en wordt 30 °C warmer dan de omgeving (op 100 %: 39 °C).

Klopt dit? De gate trekt alleen stroom bij het schakelen: hoogstens ongeveer 50 nC per flank, 980 flanken per seconde, gemiddeld minder dan 0,05 mA uit de pin. De LDR ziet het flikkeren van de lampen niet: hij reageert pas na enkele tientallen milliseconden, en hij hangt zo dat hij vooral daglicht ziet (hoofdstuk 22). En de stroommeting op A4 moet nu middelen over tien PWM-perioden (vorig voorbeeld). Eén voorwaarde bij de aankoop: de lampen moeten 'PWM-dimbaar' zijn, zoals ledpanelen van 12 V met een weerstand

per segment (net als de ledstrip) of met een led-driver die een dimingang heeft (hoofdstuk 23). Een lamp met een eigen voeding die bij elke puls opnieuw opstart, dim je zo niet.

UITLEG · 25.15

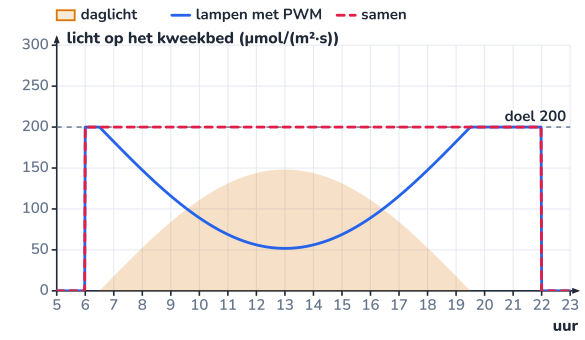
Habitat: een bewolkte lentedag, aan/uit tegenover gedimd

15/19

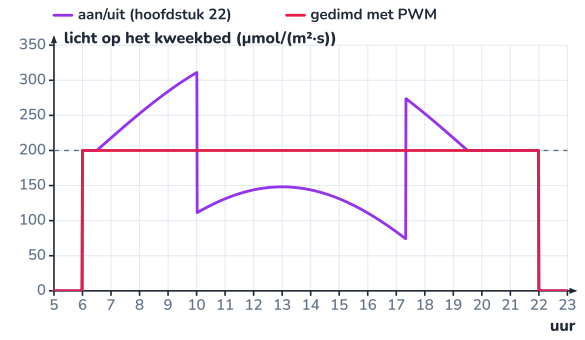
Wat levert dimmen op? Neem een bewolkte lentedag: de zon komt op om 6.30 u en gaat onder om 19.30 u, met hoogstens 8000 lux daglicht op het kweekbed. De lichtperiode loopt van 6 tot 22 u.

- **Aan/uit** (hoofdstuk 22): de lampen gaan uit om 10.01 u (boven 6000 lux) en weer aan om 17.20 u (onder 4000 lux). Ze verbruiken 1650 Wh. Het kweekbed krijgt tot 311 µmol als de lampen net nog aan zijn, en maar 111 als ze net uit zijn: een zaagtand.
- **Gedimd**: de lampen vullen precies aan tot 200. Ze verbruiken 1875 Wh, en het kweekbed krijgt de hele dag 200 µmol: 11,5 mol per m² over de dag, tegenover 10,7 met aan/uit.
- **Altijd vol** zou 3040 Wh kosten.

Dimmen spaart dus 38 % tegenover altijd vol aan. Op deze dag verbruikt het wel 225 Wh meer dan aan/uit, omdat aan/uit de planten tussen 10.01 en 17.20 u te weinig licht gaf. Op een zonnige dag (meer dan 10 800 lux) zijn de lampen gedimd ook uit. Het energiebeheer van hoofdstuk 20 wordt met PWM ook zachter: in plaats van de lampen uit te schakelen als de batterij laag staat, kan het de lampen op halve kracht zetten.



Het daglicht (geel) plus de gedimde lampen (blauw) geeft de hele lichtperiode 200 µmol/(m²·s).



Aan/uit geeft sprongen van 111 naar 311 µmol; gedimd blijft het kweekbed op 200.

CODE · 25.16

C++: de groeilampen dimmen

16/19

Het programma leest vijf keer per seconde de LDR, rekent het daglicht om naar µmol (hoofdstuk 22), berekent hoeveel de lampen moeten bijgeven en stuurt de PWM-waarde. Het regelt **zacht** bij: hoogstens één stap per 200 ms. Een wolk die even voor de zon schuift, laat de lampen dus niet meteen aanspringen, en je ziet nooit een sprong in het licht. Zonder echte klok (die komt in hoofdstuk 30) neemt het aan dat het om 6 u gestart is.

C++ (Arduino) **groeilampen.ino** 56 regels

```cpp
// groeilampen.ino - de groeilampen van de klimaatmodule dimmen met PWM (Arduino Mega)
// Daglicht: LDR onder een kapje op A1, 1,0 kOhm naar GND (hoofdstuk 22). Lampen: IRLZ44N op D5 (490 Hz).
const int LDR = A1;
const int LAMPEN = 5;
const float R_VAST = 1000.0;          // ohm
const float KAPJE = 0.10;             // het kapje laat 10 % door
const float DOEL = 200.0;             // umol/(m2 s) op het kweekbed: daglicht + lampen samen
const float LAMPEN_VOL = 200.0;       // wat de twee lampen op 100 % geven
const float LUX_PER_UMOL = 54.0;      // zonlicht
```

```ino
const unsigned long DAG_MS = 24UL * 3600UL * 1000UL;

int stand = 0;                        // huidige PWM-waarde, 0 tot 255
unsigned long vorige = 0;

float daglichtLux() {
  int n = analogRead(LDR);
  if (n < 1) n = 1;                                  // niet delen door nul in het donker
  float r = R_VAST * (1024.0 - n) / n;               // weerstand van de LDR (spanningsdeler omgekeerd)
  float eLdr = 10.0 * pow(r / 15000.0, -1.0 / 0.7);  // lux op de LDR (model uit hoofdstuk 22)
  return eLdr / KAPJE;                               // lux daglicht
}

int uurVanDeDag() {                  // eenvoudige klok: het programma start om 6 uur (echte klok: hoofdstuk 30)
  return (6 + (millis() % DAG_MS) / 3600000UL) % 24;
}

void setup() {
  pinMode(LAMPEN, OUTPUT);
  analogWrite(LAMPEN, 0);
  Serial.begin(9600);
}

void loop() {
  unsigned long nu = millis();
  if (nu - vorige < 200) return;      // vijf keer per seconde bijregelen
  vorige = nu;

  int doelStand = 0;
  int uur = uurVanDeDag();
  if (uur >= 6 && uur < 22) {         // lichtperiode van 16 uur
    float dag = daglichtLux() / LUX_PER_UMOL;          // daglicht in umol/(m2 s)
    float f = constrain((DOEL - dag) / LAMPEN_VOL, 0.0, 1.0);
    doelStand = (int)(f * 255 + 0.5);
  }
  // zacht bijregelen: hoogstens 1 stap per 200 ms (van 0 naar 255 in 51 s)
  if (doelStand > stand) stand++;
  else if (doelStand < stand) stand--;
  analogWrite(LAMPEN, stand);

  if (nu % 60000UL < 200) {           // eens per minuut melden
    Serial.print("uur ");
    Serial.print(uur);
    Serial.print(", PWM ");
    Serial.println(stand);
  }
}
```

Een test tegen een gesimuleerde dag geeft bij 3000 lux daglicht de waarde 185: de ADC rondt de LDR-spanning af, zodat het programma 2983 lux ziet (voorbeeld hierboven: 184). In het donker loopt de waarde in 51 s op van 0 naar 255. `constrain` houdt de fractie tussen 0 en 1: bij meer dan 10 800 lux zou ze negatief worden. De energiebeheerregels van hoofdstuk 20 voeg je toe in opdracht 25.13.

CODE · 25.17

Python: bits, ADC en aliasing narekenen

17/19

Drie kleine functies vatten het hoofdstuk samen: omzetten naar binair door te delen door 2, een ADC die naar beneden afrondt en nooit boven 1023 gaat, en de schijnbare frequentie na bemonsteren.

Python **bits.py** 33 regels

```python
# bits.py - binair, hexadecimaal, de ADC en bemonsteren in een paar functies

```

```python
def naar_binair(getal, bits=8):
    """Deel telkens door 2: de resten zijn de bits, van achter naar voor."""
    resten = []
    for _ in range(bits):
        resten.append(getal % 2)
        getal = getal // 2
    return "".join(str(b) for b in reversed(resten))


def adc(u, u_ref=5.0, bits=10):
    """Zoals de ADC: 2**bits niveaus, naar beneden afgerond, nooit meer dan het hoogste niveau."""
    return min(int(u / u_ref * 2 ** bits), 2 ** bits - 1)


def schijnbaar(f, fs):
    """Schijnbare frequentie na bemonsteren met fs: de afstand tot het dichtste veelvoud van fs."""
    k = round(f / fs)
    return abs(f - k * fs)


for g in (5, 182, 200, 255):
    print(f"{g:4d} = {naar_binair(g)} = 0x{g:02X}")
print(f"1023 = {naar_binair(1023, 10)} = 0x{1023:X}")

for u in (0.75, 1.25, 3.20, 4.999, 6.0):
    n = adc(u)
    print(f"U = {u:5.3f} V  ->  N = {n:4d} = {naar_binair(n, 10)}")

for f in (0.2, 9.0, 11.0, 50.0):
    print(f"{f:4.1f} Hz, 10 keer per seconde gemeten: lijkt {schijnbaar(f, 10.0):.1f} Hz")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
   5 = 00000101 = 0x05
 182 = 10110110 = 0xB6
 200 = 11001000 = 0xC8
 255 = 11111111 = 0xFF
1023 = 1111111111 = 0x3FF
U = 0.750 V  ->  N =  153 = 0010011001
U = 1.250 V  ->  N =  256 = 0100000000
U = 3.200 V  ->  N =  655 = 1010001111
U = 4.999 V  ->  N = 1023 = 1111111111
U = 6.000 V  ->  N = 1023 = 1111111111
 0.2 Hz, 10 keer per seconde gemeten: lijkt 0.2 Hz
 9.0 Hz, 10 keer per seconde gemeten: lijkt 1.0 Hz
11.0 Hz, 10 keer per seconde gemeten: lijkt 1.0 Hz
50.0 Hz, 10 keer per seconde gemeten: lijkt 0.0 Hz
```

Let op 6,0 V: dat is meer dan de referentie, en de ADC blijft op 1023 staan (in het echt mag een pin nooit boven 5 V komen). En 50 Hz, gemeten met 10 Hz, lijkt 0 Hz: een constante. Wie de netbrom meet met een veelvoud van 50 Hz, ziet ze niet.

NASLAG · 25.18

Overzicht: van analoog naar digitaal en terug

n bits

$$2^n \text{ niveaus, van } 0 \text{ tot } 2^n - 1$$

symbool	betekenis	eenheid
n	aantal bits	–

ADC

$$N = \left\lfloor 2^n \frac{U}{U_{\text{ref}}} \right\rfloor \qquad U \approx N \frac{U_{\text{ref}}}{2^n}$$

symbool	betekenis	eenheid
N	ADC-waarde	–
U_{ref}	referentiespanning (5 V, 1,1 V …)	V

resolutie

$$\text{stap} = \frac{U_{\text{ref}}}{2^n}$$

symbool	betekenis	eenheid
	10 bit, 5 V: 4,88 mV	V

bemonsteren (Nyquist, intuïtief)

$$f_s = \frac{1}{T_s} > 2 f_{\max}$$

symbool	betekenis	eenheid
f_s	bemonsteringsfrequentie	Hz
$f_{\max}$	snelste verandering in het signaal	Hz

PWM

$$D = \frac{t_{\text{aan}}}{T} \qquad U_{\text{gem}} = D \cdot U \qquad P_{\text{gem}} = D \cdot P_{\text{aan}}$$

symbool	betekenis	eenheid
D	duty cycle	–
T	periode (490 Hz: 2,04 ms)	s

bits	1	2	3	4	8	10	12	16
niveaus 2^n	2	4	8	16	256	1024	4096	65 536
grootste getal	1	3	7	15	255	1023	4095	65 535

decimaal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
binair	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

bord	PWM-pinnen	frequentie
Uno (meetkas)	3, 5, 6, 9, 10, 11	490 Hz; pin 5 en 6: 980 Hz
Mega (klimaatmodule)	2 tot 13 en 44 tot 46	490 Hz; pin 4 en 13: 980 Hz

Arduino	bereik	betekenis
`analogRead(A0)`	0-1023	10 bit; 104 µs per meting
`analogReference(INTERNAL)`		Uno: referentie ±1,1 V (Mega: INTERNAL1V1, INTERNAL2V56)
`analogWrite(9, v)`	0-255	8 bit PWM, $D = v/255$ op de 490 Hz-pinnen
`int` / `unsigned long`	16 / 32 bit	−32 768 tot 32 767 / 0 tot 4 294 967 295

HET GEHEEL · 25.19

Waar analoog en digitaal elkaar raken

Dit hoofdstuk is de brug tussen de elektronica (spanningen, stromen, transistors) en het programmeren (getallen, bits, lussen). Het verbindt ook de wiskunde: machten van 2, afronden, en de periode en frequentie van een signaal, die in Expert terugkomen als sinussen en Fourier.

F24 **De microcontroller**

Digitale pinnen: hoog of laag.

F25 **Sensoren en automatisch regelen**

analogRead en de eerste sensoren; de TMP36 van de meetkas.

I1 **Meetfouten en nauwkeurigheid**

Resolutie is niet hetzelfde als nauwkeurigheid.

I22 **De spanningsdeler en sensoren**

De ADC-omrekening en de LDR die hier de lampen stuurt.

I23 **Halfgeleiders: diode, led en transistor**

De MOSFET die met PWM schakelt, en de snelle vrijloopdiode.

I24 **Condensatoren en de RC-tijd**

Het RC-filter dat PWM gladstrijkt, en de sample-and-hold-condensator.

I26 **Motoren en actuatoren**

PWM voor het toerental van een motor; servo's met pulsen van 1 tot 2 ms.

I29 **Microcontrollers programmeren in C++**

analogRead, analogWrite, int en unsigned long in C++.

I30 **Datalogging en communicatie**

Hoeveel data de logger maakt, en hoe vaak hij meet.

E21 **Filters en signaalverwerking**

Nyquist en aliasing precies; frequenties in een signaal.

E23 **Digitale logica**

Bits worden poorten, flipflops en tellers.

E24 **Vermogenselektronica en elektrische machines**

PWM op hoge frequentie in omzetters en motorsturingen.

Habitat

De groeilampen hangen nu via één IRLZ44N op een koelplaat aan D5 van de Mega (490 Hz), met een zekering van 20 A. Ze vullen het daglicht aan tot 200 µmol/(m²·s) op het kweekbed. Op een bewolkte lentedag kost dat 1,88 kWh in plaats van 3,04 kWh. De batterijstroom op A4 wordt voortaan gemiddeld over tien PWM-perioden (100 metingen in 20,4 ms).

SAMENVATTING

In het kort

- Een digitaal signaal is bemonsterd in de tijd en afgerond op een eindig aantal niveaus.
- Binair: elke plaats is twee keer zoveel waard. Met n bits heb je 2^n waarden; een byte (8 bits) gaat van 0 tot 255. Eén hexadecimaal cijfer is vier bits.
- Een ADC van 10 bit deelt 0 tot U_{ref} in 1024 stappen: bij 5 V 4,88 mV. Resolutie is geen nauwkeurigheid. Hij vindt zijn getal met tien ja-neevragen.
- Meet meer dan twee keer per periode van de snelste verandering (Nyquist), liefst veel vaker; anders ontstaat aliasing: een snel signaal lijkt traag.
- PWM: een pin die snel aan en uit gaat. De duty cycle D bepaalt de gemiddelde spanning $D \cdot U$ en het gemiddelde vermogen $D \cdot P$.
- analogWrite geeft 256 standen; op de meeste pinnen 490 Hz. Achter een MOSFET dim je zo lampen en regel je motoren; een RC-filter maakt er een gelijkspanning van.

BEGRIPPEN

Wat je nu kent

bit

byte

Groep van 8 bits: 256 waarden, van 0 tot 255.

Kleinste eenheid van informatie: 0 of 1. Met n bits schrijf je 2^n verschillende waarden.

binair talstelsel

Talstelsel met grondtal 2: elke plaats is twee keer zoveel waard als de plaats rechts ervan.

hexadecimaal

Talstelsel met grondtal 16 (cijfers 0-9 en A-F); één hexadecimaal cijfer is vier bits.

resolutie (van een ADC)

De kleinste spanningsverandering die een ADC kan onderscheiden: $U_{\text{ref}}/2^n$, bij 10 bit en 5 V 4,88 mV.

bemonsteringsfrequentie

Hoeveel metingen per seconde een digitaal systeem van een signaal neemt.

aliasing

Een signaal dat te traag bemonsterd wordt, lijkt een ander, trager signaal.

PWM

Pulsbreedtemodulatie: een digitale uitgang die snel aan en uit gaat, zodat het gemiddelde tussen 0 en de volle spanning ligt.

duty cycle

Het deel van de periode waarin een PWM-signaal hoog is, in procent.

DAC

Digitaal-analoogomzetter: maakt van een getal een spanning.

HANDBOEK · HOOFDSTUK 26

Motoren en actuatoren

Van een kompasnaald die uitwijkt naar een dak dat zacht dichtschuift

Sensoren laten een microcontroller voelen; actuatoren laten hem iets doen. De meeste actuatoren steunen op één ontdekking: stroom maakt een magnetisch veld, en een stroom in een magnetisch veld voelt een kracht. In dit hoofdstuk leer je hoe een elektromagneet, een relais en een gelijkstroommotor werken, waarom het toerental van een motor met de spanning gaat en zijn koppel met de stroom, en hoe servo's en stappenmotoren een positie aanhouden. Met een H-brug laat je een motor in twee richtingen draaien. Zo kies en stuur je de motor die het schuifpaneel van het kasdak van de klimaatmodule opent en sluit.

7× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 9 lestijden

L Elektriciteit en elektronica · K Krachten, beweging en constructies · R Regelen en robotica

Een wormwielmotor, een tandriem en twee eindeloopschakelaars: zo schuift het kasdak open en dicht.

Na dit hoofdstuk

- Je kan met de rechterhandregel de richting van het magnetisch veld rond een stroomdraad en in een spoel bepalen.
- Je kan de kracht op een stroomdraad in een magnetisch veld berekenen ($F = B \cdot I \cdot l$) en haar richting bepalen.
- Je kan uitleggen hoe een relais en een gelijkstroommotor werken, en wat de commutator doet.
- Je kan met het model $U = RI + k\omega$ en $M = kI$ uitleggen waarom het toerental met de spanning en het koppel met de stroom samenhangt, en een motorkarakteristiek lezen.
- Je kan een servomotor en een stappenmotor aansturen en uitrekenen welke puls, hoek of stappen daarbij horen.
- Je kan een H-brug tekenen en uitleggen, met de verboden stand, en een motor met PWM in twee richtingen sturen.
- Je kan een motor met overbrenging kiezen voor een gegeven kracht en snelheid, en de aansturing ontwerpen.

VERHAAL · 26.1

Een kompasnaald die uitwijkt

In april 1820 gaf de Deense natuurkundige Hans Christian Ørsted in Kopenhagen een les over elektriciteit. Toen hij een stroom door een draad stuurde die boven een kompas hing, zwaaide de naald opzij. Elektriciteit en magnetisme, tot dan twee aparte raadsels, bleken één verschijnsel. Ørsted publiceerde het in juli; binnen enkele maanden toonde André-Marie Ampère in Parijs dat twee stroomdraden elkaar aantrekken of afstoten.

In 1821 liet Michael Faraday een draad met stroom rond een magneet draaien: de eerste elektromotor, een speeltje zonder nut, maar met een principe dat vandaag alles aandrijft. In 1834 bouwde Moritz Jacobi in Koningsbergen een motor die echt werk kon doen, en in 1838 voer er een boot met zijn motoren over de Neva.

Vandaag zitten er in een huis tientallen elektromotoren: in de koelkast, de wasmachine, de ventilator van de computer, de trilfunctie van een smartphone. In Habitat draaien ze de pomp, de ventilatoren en straks het kasdak. En het relais, dat al in de meetkas de ledstrip en de pomp schakelt, is gewoon een elektromagneet met een schakelaar eraan.

Waarom dit hoofdstuk ertoe doet

Een actuator kiezen is meer dan 'een motor bestellen': hij moet genoeg koppel geven, snel genoeg draaien, in twee richtingen kunnen, veilig stoppen en de juiste stroom vragen. Dat reken je hier uit voor de dakmotor van de klimaatmodule.

De proef van Ørsted: stroom door een draad boven een kompas, en de naald draait weg.

Stroom maakt een magnetisch veld

Wat je al weet. Een relais, een servo en een duwstang zetten een signaal van de microcontroller om in beweging (F25). Een spoel geeft een spanningspiek als je haar stroom onderbreekt, en daarom staat er een vrijloopdiode over (hoofdstuk 23). Tandwielen en een worm veranderen toerental en kracht (F12).

Rond elke stroomdraad ontstaat een **magnetisch veld**. De veldlijnen zijn cirkels rond de draad. Hun richting vind je met de **rechterhandregel**: omklem de draad met je rechterhand, duim in de richting van de (conventionele) stroom; je vingers wijzen de draairichting van het veld aan. In een tekening schrijf je een stroom naar je toe als ⊙ (de punt van een pijl) en een stroom van je weg als ⊗ (de veren).

De sterkte van het veld heet de **magnetische inductie** B, in **tesla** (T). Op een afstand r van een lange rechte draad met stroom I is

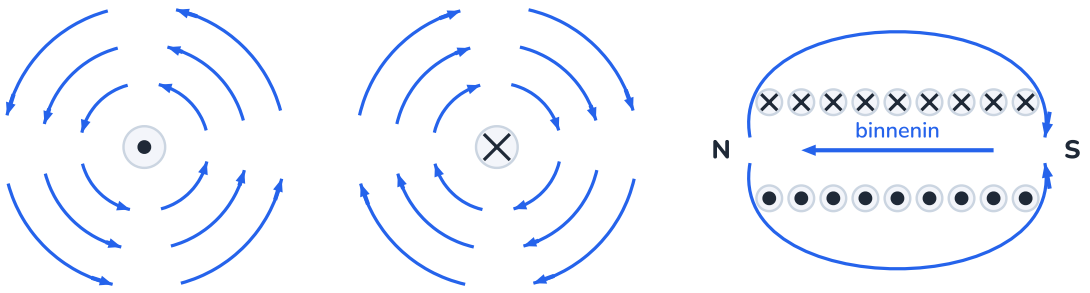
VELD VAN EEN LANGE RECHTE DRAAD

$$B = \frac{\mu_0 I}{2\pi r} \qquad \mu_0 = 4\pi \cdot 10^{-7}\,\mathrm{T \cdot m/A}$$

Voor 10 A op 1 cm: $B = 2 \cdot 10^{-7} \cdot 10/0{,}01 = 0{,}20\,\mathrm{mT}$, vier keer het veld van de aarde: genoeg om een kompas te laten uitwijken.

Wind je de draad op tot een **spoel**, dan tellen de velden van alle windingen op. Binnenin loopt een sterk, bijna gelijkmatig veld; buiten lijkt het op het veld van een staafmagneet, met een noord- en een zuidpool. De rechterhandregel voor een spoel: je vingers in de draairichting van de stroom, je duim wijst naar de noordpool. Voor een lange spoel met N windingen op een lengte l geldt $B = \mu_0 N I / l$: met 400 windingen, 0,50 A en 4 cm lengte is dat 6,3 mT. Een ijzeren kern maakt het honderden tot duizenden keren sterker, tot ongeveer 1,5 à 2 T: dan is het ijzer verzadigd. Dat is een **elektromagneet**: een magneet die je aan- en uitzet.

bron	magnetische inductie B
aardmagnetisch veld (België)	ongeveer 50 µT
draad met 10 A, op 1 cm	0,20 mT
spoel zonder kern: 400 windingen, 0,50 A, 4 cm lang	6,3 mT
koelkastmagneet, aan het oppervlak	orde van grootte 5 mT
in de luchtspleet van een motor, of een neodymiummagneet	0,2 tot 1 T



stroom naar je toe
veld tegen de klok in

stroom van je weg
veld met de klok mee

spoel: bovenaan de stroom het blad in
noordpool waar het veld eruit komt

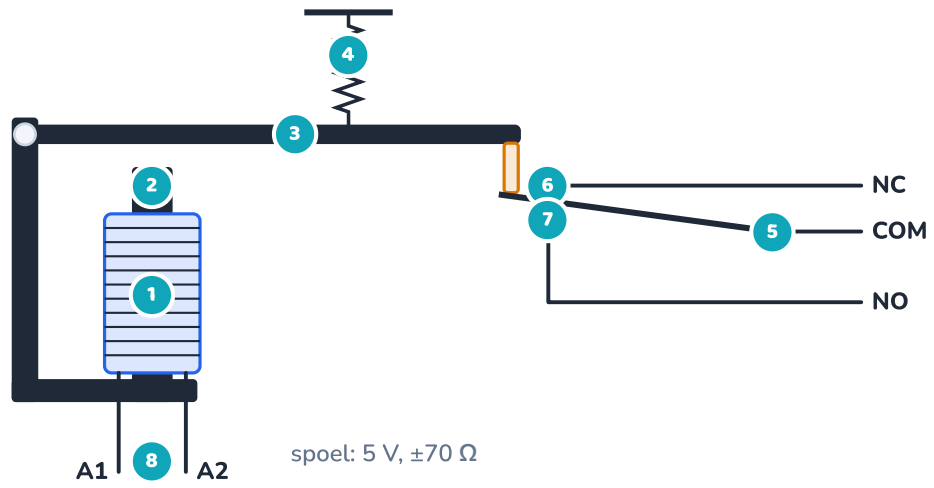
Links en midden: veldlijnen rond een draad, met de draairichting uit de rechterhandregel. Rechts: een spoel met haar noordpool.

ONTLEED · 26.3

Het relais

3/18

Een **relais** is een elektromagneet die een schakelaar bedient. In de meetkas schakelen vier relais de ledstrip, de pomp en de duwstang (F25). Het blauwe relais op de meeste relaismodules, het SRD-05VDC, heeft een spoel van 5 V (ongeveer 70 Ω, 71 mA) en contacten voor 10 A bij 250 V wisselspanning of 30 V gelijkspanning. Het grote voordeel: de **stuurkring** (5 V van de microcontroller) en de **geschakelde kring** (12 V of zelfs 230 V) raken elkaar nergens.



In rust trekt de veer het anker omhoog en raakt het contact COM de NC-kant; met stroom door de spoel klapt het anker naar beneden.

1 Spoel
Honderden windingen dun koperdraad. Met 5 V loopt er ongeveer 71 mA: te veel voor een pin, daarom een transistor of een relaismodule (hoofdstuk 23).

2 IJzeren kern
Maakt het veld van de spoel honderden keren sterker en leidt het naar het anker. Samen met het U-vormige juk vormt hij een bijna gesloten ijzeren kring.

3 Anker
Een scharnierend ijzeren plaatje. Loopt er stroom, dan trekt de kern het naar beneden; het duwt via een isolerend stift het contact om.

4 Veer
Trekt het anker terug als de stroom wegvalt. De datasheet belooft: aantrekken bij hoogstens 75 % van de nominale spanning (3,75 V), afvallen pas onder 10 % (0,5 V). Dat verschil is hysterese, zoals bij een Schmitt-trigger (hoofdstuk 24).

5 COM
Het gemeenschappelijke, beweegbare contact: een verende strip die ofwel NC ofwel NO raakt.

6 NC (normally closed)
In rust verbonden met COM. Gebruik het voor iets dat aan moet blijven als de sturing uitvalt.

7 NO (normally open)
Alleen verbonden met COM als de spoel bekrachtigd is. De ledstrip en de pomp van de meetkas hangen aan NO: zonder signaal zijn ze uit.

8 Aansluitingen van de spoel
A1 en A2. Over de spoel hoort een vrijloopdiode, kathode aan de plus (hoofdstuk 23); op een relaismodule zit die er al op.

UITLEG · 26.4

Een stroom in een magnetisch veld voelt een kracht

4/18

Het omgekeerde van Ørsted: zet een draad met stroom in het veld van een magneet, en de draad krijgt een duw. Die kracht staat loodrecht op de draad én loodrecht op het veld. Staat de draad loodrecht op de veldlijnen, dan is ze maximaal:

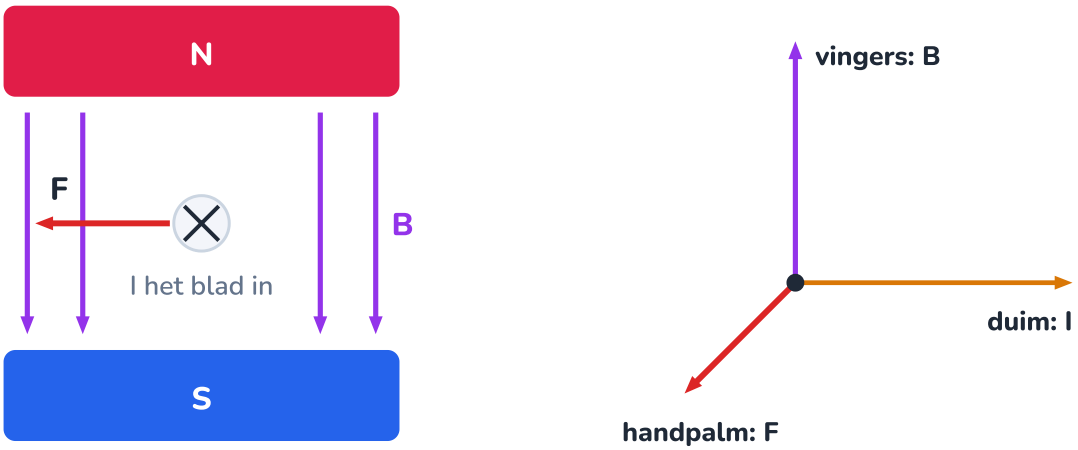
KRACHT OP EEN STROOMDRAAD (DRAAD LOODRECHT OP HET VELD)

$$F = B \cdot I \cdot l$$

F in newton, B in tesla, I in ampère, l de lengte van de draad in het veld (m). Evenwijdig met het veld is de kracht nul; schuin ertussen. Deze kracht op bewegende ladingen heet de Lorentzkracht (hoofdstuk E19).

De richting vind je met de **vlakke rechterhand**: steek je duim in de richting van de stroom en je gestrekte vingers in de richting van het veld (van N naar S); je handpalm duwt in de richting van de kracht. Keer je de stroom of het veld om, dan keert ook de kracht om. (Sommige boeken gebruiken de linkerhandregel van Fleming: die geeft dezelfde richting.)

Hoe groot? Een draad van 4 cm met 2,0 A in een veld van 0,080 T voelt $0{,}080 \cdot 2{,}0 \cdot 0{,}040 = 6{,}4\,\text{mN}$: het gewicht van 0,65 g. Klein, maar met honderden windingen, een sterke magneet en ijzer rond de spoel wordt het de kracht die een motor laat draaien.



Links: stroom het blad in, veld naar beneden, kracht naar links. Rechts: de drie richtingen van de vlakke rechterhand.

PROEF · 26.5

Een stroombalans op een keukenweegschaal

5/18

ONDERZOEKSVRAAG

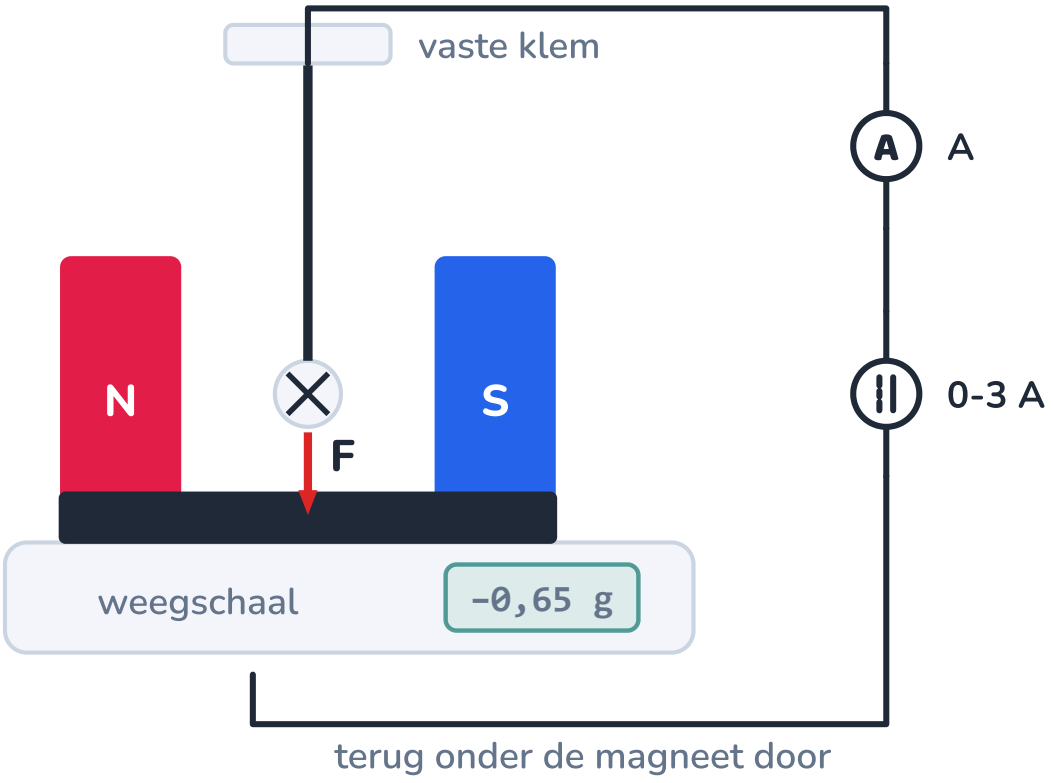
Is de kracht op een stroomdraad evenredig met de stroom? En hoe sterk is het veld van de magneet?

MATERIAAL

- U-magneet (of twee neodymiummagneten op een stalen U), polen 4,0 cm breed
- keukenweegschaal tot op 0,01 g
- stijve koperdraad in een statiefklem
- voeding die 3 A kan leveren, ampèremeter
- liniaal

Veiligheid

Houd de stroom maar enkele seconden aan: de draad en de voeding worden warm. Neodymiummagneten klappen hard tegen elkaar: niet met de vingers ertussen.



De draad hangt vast; de magneet staat op de weegschaal en voelt de tegenkracht.

WERKWIJZE

1. Zet de magneet op de weegschaal en zet ze op nul (tarra).
2. Hang de draad vast in de klem, zodat een recht stuk van 4,0 cm door de spleet loopt, loodrecht op het veld, zonder de magneet te raken.
3. Stuur 0,5 A door de draad en lees de weegschaal af. Verhoog in stappen van 0,5 A tot 3,0 A.
4. Keer de stroom om: wat verandert er?
5. Zet de aflezing uit tegen de stroom en bepaal B uit de helling.

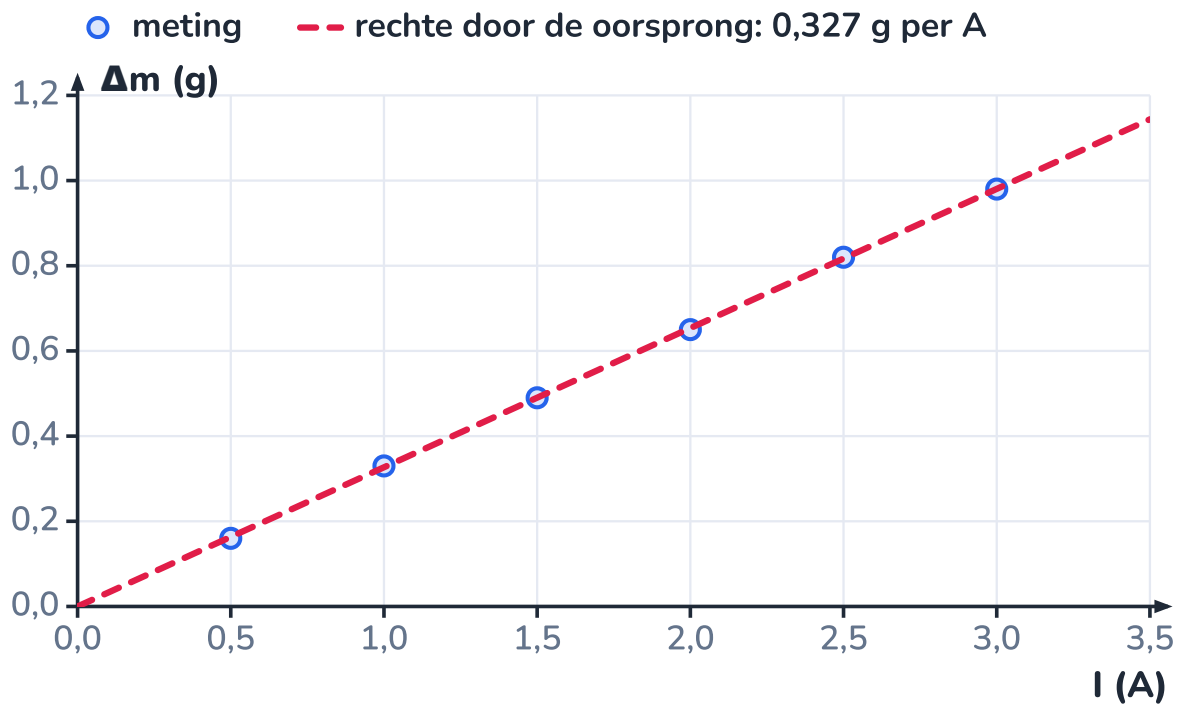
WAARNEMING

I (A)	0,5	1,0	1,5	2,0	2,5	3,0
aflezing (g)	−0,16	−0,33	−0,49	−0,65	−0,82	−0,98

Voorbeeldmeting. De weegschaal toont een negatief getal: de magneet lijkt lichter. Met de stroom omgekeerd wordt het getal even groot, maar positief.

VERKLARING

De magneet duwt de draad naar beneden. Volgens de derde wet van Newton (hoofdstuk 8) duwt de draad de magneet even hard omhoog: de weegschaal ziet een kleinere kracht. De punten liggen op een rechte door de oorsprong: F is recht evenredig met I. De helling is 0,327 g per A, dus per ampère $F = 0{,}327 \cdot 10^{-3}\,\text{kg} \cdot 9{,}81\,\text{N/kg} = 3{,}21\,\text{mN}$. Uit $F = BIl$: $B = \dfrac{F/I}{l} = \dfrac{3{,}21 \cdot 10^{-3}}{0{,}040} = 0{,}080\,\text{T}$.



Een rechte door de oorsprong: de kracht is evenredig met de stroom.

UITLEG · 26.6

De gelijkstroommotor

6/18

Zet een rechthoekige winding in een magnetisch veld en stuur er stroom door. De twee zijden die loodrecht op het blad staan, krijgen tegengestelde krachten: de ene omhoog, de andere omlaag. Samen geven ze een **koppel** (hoofdstuk 10): de winding draait. Het koppel is het grootst als de winding evenwijdig met het veld staat, en nul als ze er loodrecht op staat:

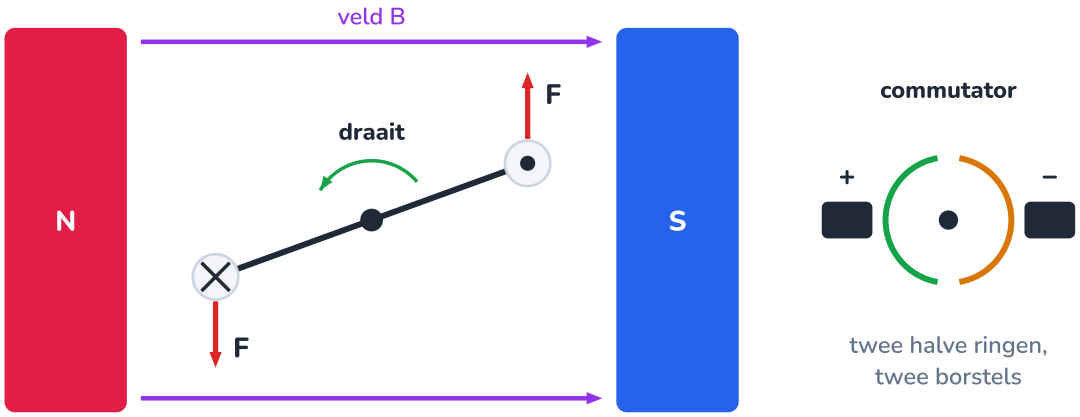
KOPPEL OP EEN SPOEL IN EEN VELD

$$M_{\max} = N \cdot B \cdot I \cdot A$$

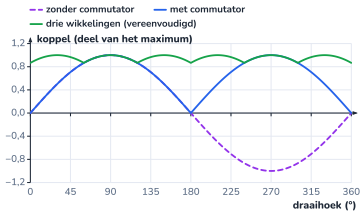
N: aantal windingen, A: oppervlakte van de winding (m²). Een schoolmotortje met 60 windingen van 2 × 3 cm in 0,25 T en 1,0 A: $60 \cdot 0{,}25 \cdot 1{,}0 \cdot 6 \cdot 10^{-4} = 9{,}0\,\text{mN} \cdot \text{m}$.

Het probleem: voorbij de loodrechte stand keren de krachten hun effect om, en de winding zou terugdraaien. Ze zou blijven schommelen rond die stand. **De oplossing** is de **commutator**: een ring op de as, in twee helften gespleten, waartegen twee **borstels** (vaak koolstof) de stroom aanvoeren. Na elke halve draai wisselt elke helft van borstel: de stroom in de winding keert om, net wanneer het koppel van teken zou veranderen. Zo blijft het koppel altijd in dezelfde zin.

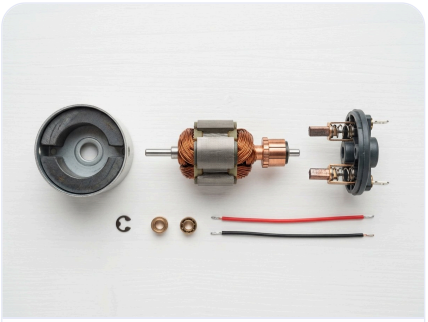
Een echte motor heeft niet één winding maar drie of meer, elk met hun eigen stukje commutator, op een ijzeren rotor: het koppel schommelt dan veel minder. De borstels slijten wel, en ze vonken een beetje. Borstelloze motoren (hoofdstuk E24) lossen dat op met elektronica in plaats van een commutator.



De winding (tussen ⊗ en ⊙) krijgt een kracht omlaag en een omhoog: ze draait tegen de klok in. De commutator keert de stroom elke halve draai om.



Zonder commutator keert het koppel om; met commutator blijft het positief; met drie wikkelingen is het bijna constant.



Een motortje uit elkaar: magneten, een rotor met drie spoelen, de commutator en de borstels.

UITLEG · 26.7

Toerental gaat met de spanning, koppel met de stroom

7/18

Een draaiende motor is ook een **dynamo**: de windingen bewegen door het veld en wekken zelf een spanning op (hoofdstuk E19). Die **tegen-emk** werkt de voedingsspanning tegen en is evenredig met het toerental ω (in rad/s). Wat overblijft, drijft de stroom door de weerstand R van de windingen. En het koppel is evenredig met de stroom. Met één motorconstante k (in N·m/A, of hetzelfde getal in V·s/rad):

MODEL VAN EEN GELIJKSTROOMMOTOR

$$U = R \cdot I + k \cdot \omega \qquad M = k \cdot I$$

$\omega = 2\pi n/60$ met n in toeren per minuut (tpm). In werkelijkheid gaat een klein deel van de stroom naar de wrijving: de nullaststroom I_0, dus $M = k(I - I_0)$.

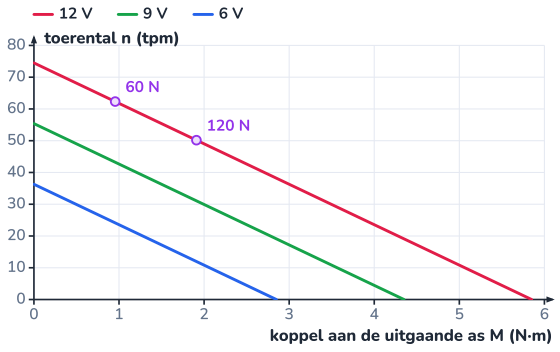
Daaruit volgen drie dingen die je bij elke motor terugziet:

- **Het koppel volgt de stroom.** Hoe zwaarder de last, hoe meer stroom de motor trekt, bij elke spanning dezelfde. Een motor die zwaar moet trekken, wordt warm (RI^2).
- **Het toerental volgt de spanning.** Onbelast is I klein, dus $\omega \approx U/k$: dubbele spanning, dubbel toerental. Onder belasting zakt het toerental een beetje, omdat RI van de spanning afgaat.
- **Stilstaan is gevaarlijk.** Bij $\omega = 0$ is er geen tegen-emk: de stroom is U/R, de **blokkeerstroom**. Dat gebeurt bij elke start, en als de motor vastloopt.

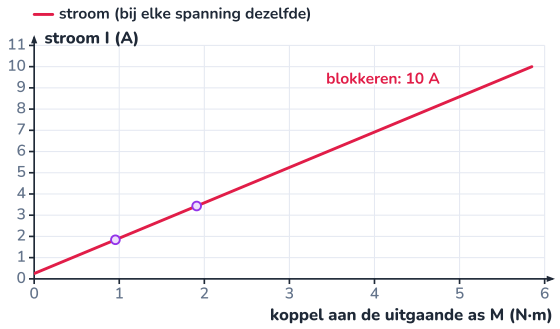
De grafieken tonen het model van de dakmotor van de klimaatmodule: een motor met $R = 1{,}2\,\Omega$, $k = 0{,}030\,\mathrm{N \cdot m/A}$ en een wormoverbrenging van 1 op 50 (koppel en toerental aan de uitgaande as). Bij 12 V draait hij onbelast 74,5 tpm; geblokkeerd levert hij 5,85 N·m en trekt hij 10 A. Tussen die twee daalt het toerental lineair met het koppel. Bij 6 of 9 V ligt dezelfde rechte lager, evenwijdig: de spanning verschuift het toerental. De stroom hangt alleen van het koppel af.

PWM en motoren

Met PWM (hoofdstuk 25) krijgt de motor gemiddeld $D \cdot U$: zo stel je het toerental in. Start je met een lage duty cycle en voer je die geleidelijk op, dan blijft de stroom bij het aanlopen klein: geen blokkeerstroom van 10,7 A, maar de stroom die de last vraagt.



Evenwijdige rechten: de spanning bepaalt het toerental, de last trekt het een beetje omlaag.



De stroom stijgt lineair met het koppel, bij elke spanning dezelfde: 1,84 A bij 60 N, 3,4 A bij 120 N.

UITGEWERKT VOORBEELD · 26.8

Een motor met overbrenging doorrekenen

8/18

De dakmotor heeft een wormoverbrenging van 1 op 50 met een rendement van 40 %. De motor zelf heeft $R = 1{,}2\,\Omega$, $k = 0{,}030\,\mathrm{N \cdot m/A}$ en een nullaststroom van 0,25 A. Aan de uitgaande as moet hij een koppel van 0,95 N·m leveren (de riem trekt 60 N, verder in dit hoofdstuk). Bereken (a) het koppel dat de motor zelf moet leveren, (b) de stroom, (c) het toerental van de motor en van de uitgaande as bij 12 V, (d) het rendement van het geheel.

GEGEVEN

- $i = 50$, $\eta_w = 0{,}40$; $R = 1{,}2\,\Omega$, $k = 0{,}030\,\mathrm{N \cdot m/A}$, $I_0 = 0{,}25\,\mathrm{A}$
- $M_{\text{uit}} = 0{,}955\,\mathrm{N \cdot m}$; $U = 12\,\mathrm{V}$

GEVRAAGD

- M_{motor}, I, n, rendement

OPLOSSING

1. (a) Een overbrenging vermenigvuldigt het koppel met i, maar de worm verliest 60 %: $M_{\text{motor}} = \dfrac{M_{\text{uit}}}{\eta_w \cdot i} = \dfrac{0{,}955}{0{,}40 \cdot 50} = 0{,}0477\,\mathrm{N \cdot m}$.

2. (b) $I = I_0 + \dfrac{M_{\text{motor}}}{k} = 0{,}25 + \dfrac{0{,}0477}{0{,}030} = 1{,}84\,\mathrm{A}$.

3. (c) $\omega = \dfrac{U - RI}{k} = \dfrac{12 - 1{,}2 \cdot 1{,}842}{0{,}030} = 326\,\mathrm{rad/s}$, of $326 \cdot 60/2\pi = 3116$ tpm. De as na de worm draait 50 keer trager: 62,3 tpm.

4. (d) Mechanisch: $M_{\text{uit}} \cdot \omega_{\text{uit}} = 0{,}955 \cdot 6{,}53 = 6{,}23\,\mathrm{W}$. Elektrisch: $12 \cdot 1{,}84 = 22{,}1\,\mathrm{W}$. Rendement $28\,\%$.

ANTWOORD

De motor levert 0,048 N·m bij 1,84 A en draait 3116 tpm; de uitgaande as 62,3 tpm. Van de 22,1 W gaat 6,2 W naar de riem: 28 %.

Klopt dit? Hoofdstuk 9 schatte 30 % rendement en 20 W elektrisch: het model geeft bijna hetzelfde. Waarom zo'n laag rendement? Omdat een zelfremmende worm altijd onder 50 % blijft: de wrijving tussen worm en wiel moet groot genoeg zijn om de last tegen te houden als de motor stilstaat, en diezelfde wrijving kost energie als hij draait. Voor een dak dat een halve minuut per dag beweegt, is dat een goede ruil.

UITLEG · 26.9

De H-brug: een motor in twee richtingen

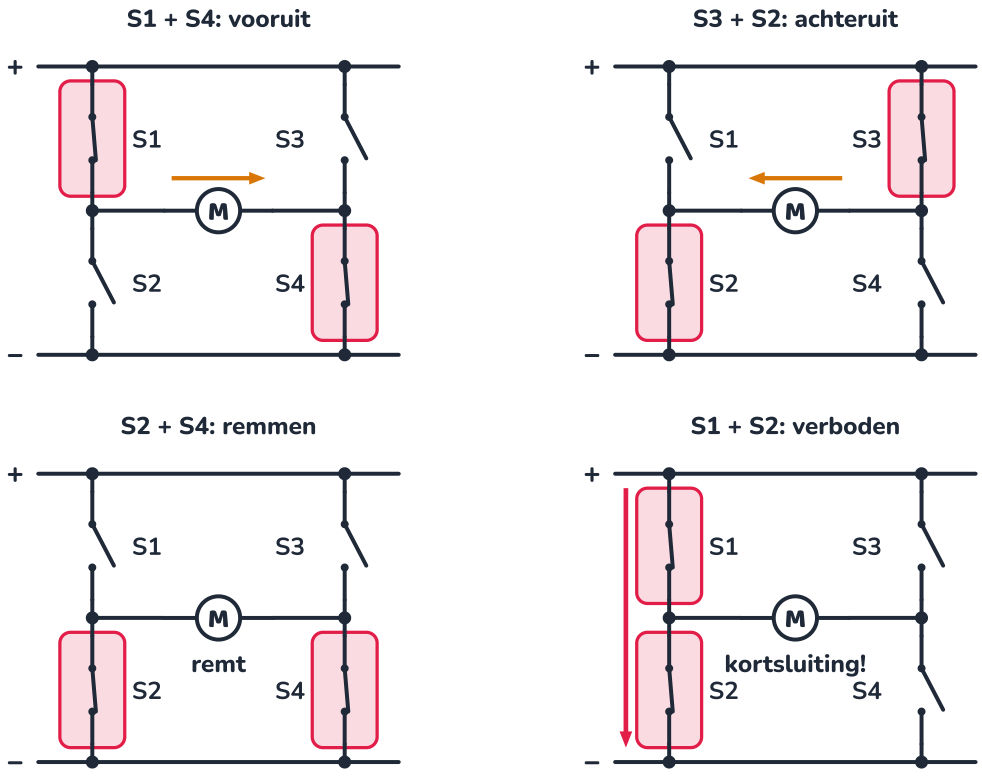
9/18

Een MOSFET low-side (hoofdstuk 23) laat een motor maar in één richting draaien. Voor twee richtingen moet de stroom ook omgekeerd door de motor kunnen. Daarvoor zet je vier schakelaars rond de motor, in de vorm van een H: de **H-brug**.

- **S1 en S4 dicht**: de stroom gaat van links naar rechts door de motor: vooruit.
- **S3 en S2 dicht**: van rechts naar links: achteruit.
- **S2 en S4 dicht** (beide onderkant): de motor is kortgesloten. Een draaiende motor is een dynamo; kortgesloten remt hij zichzelf hard af.
- **Alles open**: de motor bolt uit.
- **S1 en S2 dicht** (of S3 en S4): een directe kortsluiting van de voeding, door twee schakelaars, zonder de motor. Dit is de **verboden stand**: met een batterij van 12,8 V loopt er in een oogwenk honderden ampère; de schakelaars (transistors) branden door, of de zekering springt.

S1	S2	S3	S4	gevolg
dicht	open	open	dicht	vooruit (links +, rechts –)
open	dicht	dicht	open	achteruit
open	dicht	open	dicht	remmen: motor kortgesloten via de onderkant
open	open	open	open	vrijlopen (uitbollen)
dicht	dicht	×	×	**verboden**: kortsluiting over de voeding
×	×	dicht	dicht	**verboden**: kortsluiting over de voeding

Met transistors komt er een addertje bij: een transistor schakelt niet oneindig snel uit. Zet je S1 uit en S2 tegelijk aan, dan geleiden ze even allebei (*shoot-through*). Een goede H-brug wacht daarom een fractie van een microseconde tussen het uitzetten van de ene en het aanzetten van de andere: de **dode tijd**. En met PWM op een van de bovenste schakelaars regel je naast de richting ook het toerental.



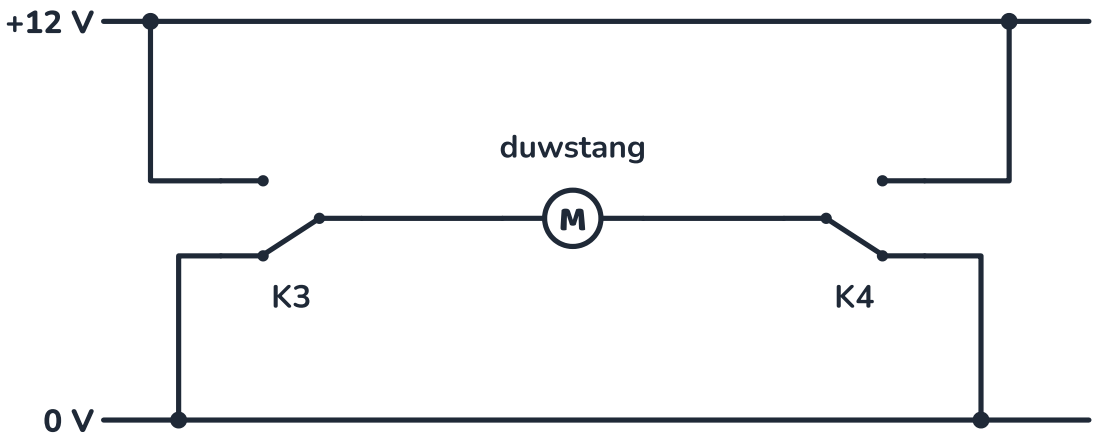
Gesloten schakelaars op een rode achtergrond. Vooruit en achteruit keren de stroom door de motor om; S1 + S2 is een kortsluiting.

UITGEWERKT VOORBEELD · 26.10

Twee wisselrelais als H-brug: de duwstang van de meetkas

10/18

De duwstang van het dakraam in de meetkas schuift uit bij de ene polariteit en in bij de andere. Ze hangt aan twee relais, K3 en K4 (F25). In F25 staat: 'nooit K3 en K4 tegelijk aan'. (a) Hoe moet je de twee wisselcontacten aansluiten zodat de duwstang in beide richtingen kan? (b) Wat gebeurt er in de vier combinaties van K3 en K4? (c) Kan deze schakeling een kortsluiting maken, zelfs als het programma een fout heeft?



In rust staan beide COM-contacten aan 0 V (NC). Wie een relais bekrachtigt, zet die kant op +12 V.

GEGEVEN

- twee relais met elk een wisselcontact (COM, NC, NO)
- duwstang: uit bij + links, in bij + rechts

GEVRAAGD

- schema
- gevolg van de vier combinaties
- kortsluiting mogelijk?

OPLOSSING

1 (a) De duwstang komt tussen de twee COM-contacten. Bij beide relais gaat NC naar 0 V en NO naar +12 V.

2 (b) Beide uit: beide aansluitingen aan 0 V, de duwstang staat stil (en is kortgesloten: ze remt). K3 aan: links +12 V, rechts 0 V: uitschuiven. K4 aan: omgekeerd: inschuiven. Beide aan: beide aan +12 V: geen spanningsverschil, de duwstang staat stil.

3 (c) Nee. Elk COM-contact raakt ofwel NC ofwel NO, nooit allebei: geen enkele combinatie verbindt +12 V rechtstreeks met 0 V. De verboden stand van de H-brug (S1 en S2 samen) kan hier mechanisch niet bestaan.

ANTWOORD

Motor tussen de twee COM's, NC's aan 0 V, NO's aan +12 V. Uit/uit en aan/aan: stil; één van beide aan: uit of in. Kortsluiting is onmogelijk; 'nooit tegelijk aan' is alleen nodig om geen stroom te verspillen.

Klopt dit? Dat is een mooi voorbeeld van **veilig ontwerpen**: een fout in de software kan hier niets kapotmaken. Relais hebben wel hun grenzen: ze klikken, slijten (orde van grootte honderdduizend schakelingen bij volle last) en kunnen geen PWM. Voor de trage duwstang van de meetkas, die een paar keer per dag beweegt, zijn ze ideaal; voor het kasdak van de klimaatmodule, dat zacht moet aanlopen, niet.

UITLEG · 26.11

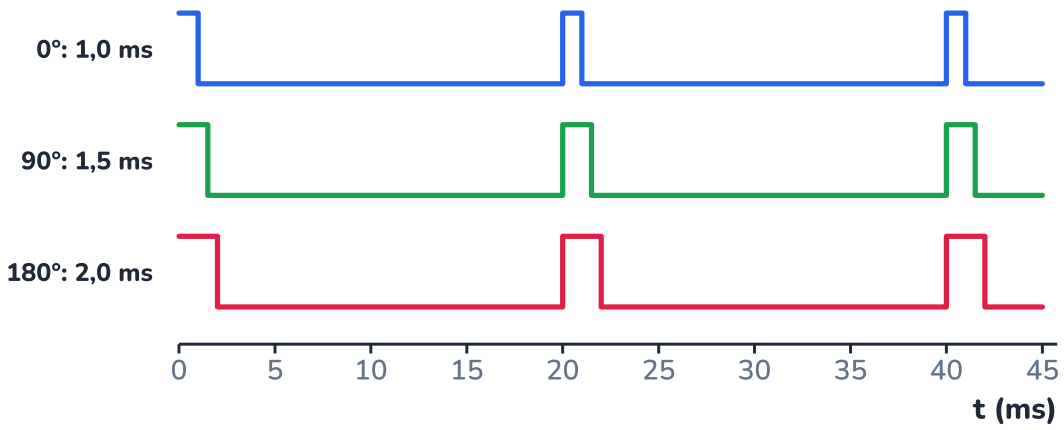
De servomotor: een hoek op bestelling

11/18

Een **servomotor** is een klein doosje met een gelijkstroommotor, een tandwielkast, een potentiometer op de uitgaande as en een regelaar. De regelaar vergelijkt de gewenste hoek met de gemeten hoek en laat de motor draaien tot ze gelijk zijn: een gesloten regelkring (hoofdstuk 31). Je geeft de gewenste hoek door met **pulsen**: elke 20 ms (50 Hz) één puls, en de lengte ervan is de hoek. De afspraak: 1,0 ms is de ene uiterste stand, 1,5 ms het midden, 2,0 ms de andere uiterste. Veel servo's draaien met pulsen van 0,5 tot 2,5 ms hun volle 180°.

De Arduino-bibliotheek `Servo` maakt die pulsen voor je. `servo.write(hoek)` zet een hoek van 0 tot 180° om in een puls van 544 tot 2400 µs: `write(90)` geeft 1472 µs. Wie precies wil zijn, gebruikt `servo.writeMicroseconds(1500)`. Op de Mega gebruikt de bibliotheek eerst timer 5: de pinnen 44, 45 en 46 kunnen dan geen PWM meer (daarom hangen de lampen op D5, hoofdstuk 25).

Een SG90, de kleine blauwe servo uit hoofdstuk F25, weegt 9 g, werkt op 4,8 à 6 V en levert ongeveer 0,18 N·m (1,8 kg·cm): genoeg voor een klepje, niet voor een dak. Een servo trekt stroom in stoten, tot enkele honderden mA bij het draaien: geef hem liever zijn eigen 5 V dan die van de microcontroller.



Dezelfde herhaling van 20 ms; alleen de pulsbreedte verandert, en daarmee de hoek.

UITLEG · 26.12

De stappenmotor: tellen in plaats van meten

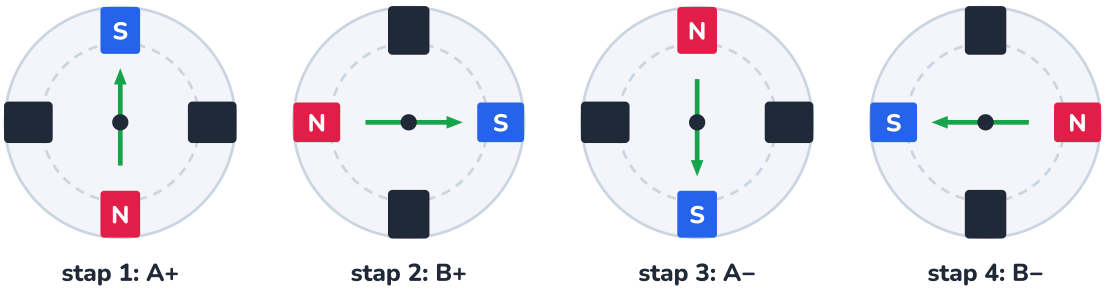
12/18

Een **stappenmotor** heeft een rotor met permanente magneten en een stator met meerdere spoelen. Zet je de spoelen één voor één aan, dan springt de rotor telkens een vaste hoek verder, naar de spoel die trekt. In de vereenvoudigde tekening zijn het vier stappen van 90°. Een gangbare **hybride stappenmotor** heeft een rotor met 50 tandjes en haalt zo **200 stappen per omwenteling: 1,8° per stap**.

Het grote voordeel: wie de stappen telt, kent de positie zonder sensor. 200 stappen per seconde is precies één omwenteling per seconde (60 tpm). Met **halve stappen** (400 per omwenteling) of **microstappen**, waarbij de driver de stroom in twee spoelen tussen twee standen verdeelt, gaat het fijner en rustiger. Een driver zoals de A4988 heeft maar twee ingangen nodig: **STEP** (elke puls één stap) en **DIR** (de richting), en kan tot 1/16 stap (3200 per omwenteling).

De nadelen: een stappenmotor trekt ook stroom als hij stilstaat (om zijn positie vast te houden), zijn koppel zakt bij hoge toerentallen, en als de last te zwaar is, **verliest hij stappen** zonder dat het programma het merkt. Een NEMA 17 (4,2 cm breed) houdt typisch ongeveer 0,4 N·m vast bij 1,7 A per spoel. Stappenmotoren zitten in 3D-printers, lasersnijders en doseerpompen: overal waar een positie klein en precies moet zijn, en de last voorspelbaar.

actuator	stuur je met	goed in	let op
gelijkstroommotor (met borstels)	MOSFET (één richting) of H-brug, PWM	eenvoudig, sterk, snel	positie onbekend; vrijloopdiode; aanloopstroom
servomotor	puls van 1-2 ms om de 20 ms	een hoek instellen (0-180°)	klein koppel; eigen voeding bij grote servo's
stappenmotor	driver met STEP/DIR	precieze positie zonder sensor	kan stappen verliezen; stroom ook in stilstand
relais	transistor of relaismodule	galvanisch gescheiden aan/uit, ook 230 V	traag, slijt, klikt; vrijloopdiode over de spoel
duwstang (lineaire actuator)	H-brug of twee wisselrelais	rechtlijnige beweging, met eindschakelaars	traag (mm per seconde)



Telkens een ander spoelpaar aan: de rotor (pijl, noordpool) draait mee, 90° per stap in dit model.

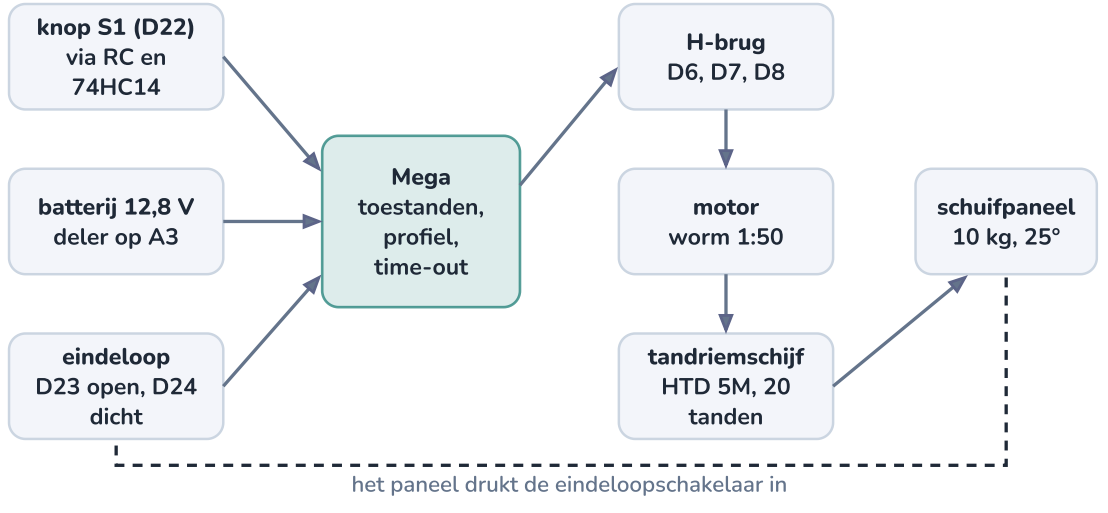
UITGEWERKT VOORBEELD · 26.13

Habitat: de motor van het kasdak kiezen

13/18

Het schuifpaneel van het kasdak (10 kg, rails onder 25°) moet dicht met 0,10 m/s, en de riem moet tot 60 N trekken; met een veiligheidsfactor 2 voor ijs en vuil minstens 120 N (hoofdstuk 7 en 8). Stroom valt soms weg: het paneel mag dan niet vanzelf openschuiven. De riem loopt over een tandriemschijf HTD 5M met 20 tanden (steek 5 mm). Drie motoren zijn te koop (tabel). (a) Welk koppel en toerental moet de as van de schijf leveren? (b) Welke motor kies je? (c) Wat is het werkpunt bij 60 en bij 120 N?

	A: tandwielmotor	B: wormwielmotor	C: planetaire motor
spanning	12 V	12 V	12 V
onbelast	30 tpm	74 tpm, 0,25 A	150 tpm
blokkeerkoppel	3,0 N·m	5,85 N·m, 10 A	1,2 N·m
overbrenging	tandwielen, zelfremmend: nee	worm 1:50, zelfremmend: ja	planetair, zelfremmend: nee



Van de knop tot het paneel: de Mega leest knop, batterij en eindeloopschakelaars, en stuurt de H-brug.

GEGEVEN

- $F = 60\,\mathrm{N}$ (normaal), $120\,\mathrm{N}$ (met veiligheidsfactor); $v = 0{,}10\,\mathrm{m/s}$

GEVRAAGD

- koppel en toerental aan de schijf
- keuze
- werkpunt

- schijf: 20 tanden van 5 mm: omtrek 100 mm, dus $r = 0{,}100/2\pi = 15{,}9\,\text{mm}$

OPLOSSING

1. (a) Koppel: $M = F \cdot r = 60 \cdot 0{,}0159 = 0{,}955\,\text{N} \cdot \text{m}$ (120 N: 1,91 N·m). Toerental: 0,10 m/s is 100 mm per seconde, precies één omwenteling per seconde: **60 tpm**. Mechanisch vermogen $P = F \cdot v = 6{,}0\,\text{W}$ (hoofdstuk 9).

2. (b) A draait onbelast maar 30 tpm: hoogstens 0,05 m/s, het dak zou 14 s nodig hebben. C is niet zelfremmend en haalt geblokkeerd maar 1,2 N·m, minder dan de 1,9 N·m van 120 N. **B**: zelfremmend, onbelast 74 tpm, geblokkeerd 5,85 N·m.

3. (c) Uit het model (vorig voorbeeld en de tabel hieronder): bij 60 N trekt B 1,84 A en draait de schijf 62,3 tpm (0,104 m/s) op 12 V; bij 120 N 3,43 A en 50,2 tpm. Beide keren komt het dak dicht.

4. Geblokkeerd zou B 5,85 N·m leveren: 368 N aan de riem. Dat is drie keer meer dan nodig en genoeg om de rail of de riem te beschadigen: de sturing moet stoppen aan de eindpunten en bij een blokkering (verder).

ANTWOORD

De schijf moet 0,95 N·m (tot 1,9 N·m) leveren bij 60 tpm. Wormwielmotor B voldoet: 1,84 A en 62 tpm bij 60 N, 3,4 A bij 120 N, en zelfremmend.

Klopt dit? Een motor kies je niet op zijn blokkeerkoppel, maar op zijn werkpunt: bij de gevraagde last moet hij het gevraagde toerental halen, zonder oververhit te raken. B werkt bij 60 N op een zesde van zijn blokkeerkoppel, vlak bij zijn beste rendement.

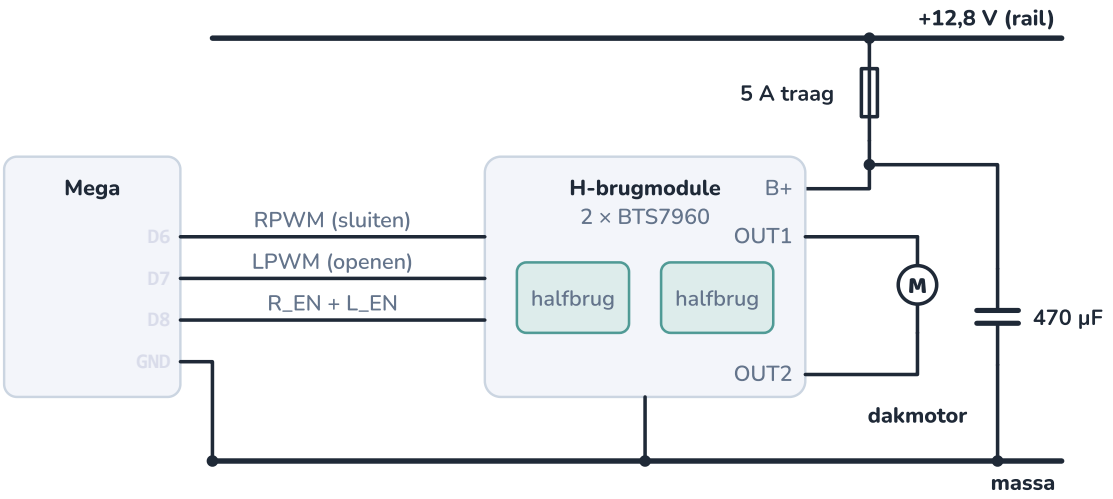
RIEMKRACHT	KOPPEL	STROOM	TOERENTAL	RIEMSNELHEID	OPGENOMEN	RENDEMENT
0 N	0,00 N·m	0,25 A	74,5 tpm	0,124 m/s	3,0 W	0 %
30 N	0,48 N·m	1,05 A	68,4 tpm	0,114 m/s	12,5 W	27 %
60 N	0,95 N·m	1,84 A	62,3 tpm	0,104 m/s	22,1 W	28 %
90 N	1,43 N·m	2,64 A	56,2 tpm	0,094 m/s	31,6 W	27 %
120 N	1,91 N·m	3,43 A	50,2 tpm	0,084 m/s	41,2 W	24 %

UITGEWERKT VOORBEELD · 26.14

Habitat: de dakmotor aansturen

14/18

De dakmotor B hangt aan de 12,8 V-rail, die tussen 12,0 V (bijna leeg) en 14,6 V (laden) kan liggen. (a) Welke H-brug en welke pinnen kies je? (b) Welke PWM-waarde geeft 0,10 m/s bij 12,8 V, en bij 14,4 V? (c) Hoe laat je de motor zacht aanlopen zoals in hoofdstuk 7? (d) Welke beveiligingen komen erbij?



Twee halfbruggen vormen samen een H-brug; D6 en D7 kiezen richting en snelheid, D8 schakelt de module in.

GEGEVEN

- model: $U = RI + k\omega$; bij 60 N is $I = 1{,}84\ \text{A}$
- motor voor 0,10 m/s: $\omega = 50 \cdot 0{,}10 / r = 314{,}2\ \text{rad/s}$
- vrije PWM-pinnen op de Mega: D3, D4, D6, D7, D8, D12, D13, D44-D46

GEVRAAGD

- H-brug en pinnen
- PWM-waarde
- aanloopprofiel
- beveiligingen

OPLOSSING

1 (a) Een module met twee BTS7960-halfbruggen: gemaakt voor autotoepassingen zoals een schuifdak, typisch 16 mΩ per schakelaar, dode tijd en beveiligingen ingebouwd, PWM tot 25 kHz. Door de brug gaat de stroom via twee schakelaars: bij 1,84 A verlies je $2 \cdot 0{,}016 \cdot 1{,}84^2 = 0{,}11\ \text{W}$. Pinnen: **D6** (RPWM: sluiten), **D7** (LPWM: openen), allebei timer 4 (490 Hz), en **D8** voor de twee enable-ingangen samen.

2 (b) Nodig: $U = 1{,}2 \cdot 1{,}84 + 0{,}030 \cdot 314{,}2 = 11{,}63\ \text{V}$ gemiddeld. Bij 12,8 V: $D = 11{,}63 / 12{,}8 = 0{,}909$, waarde 232. Bij 14,4 V: waarde 206. Het programma meet daarom de batterijspanning op A3 en past de waarde aan.

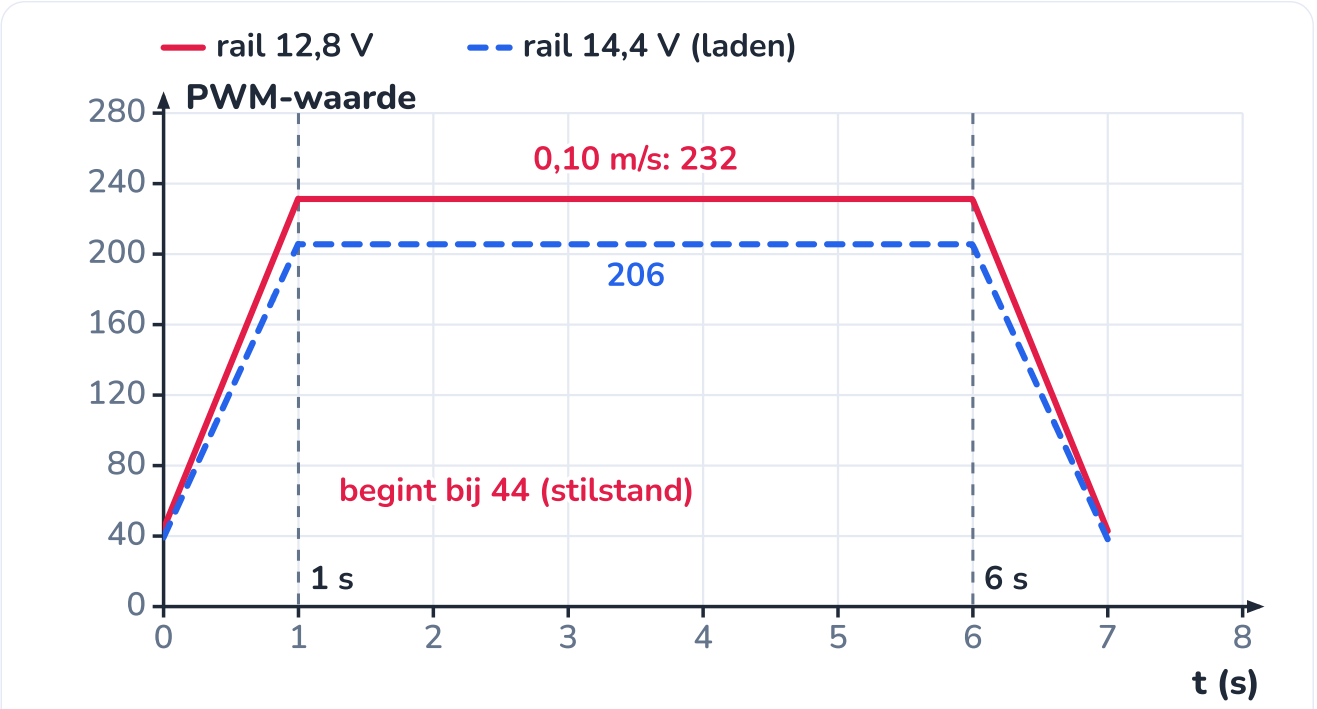
3 (c) Stilstaand heeft de motor al $1{,}2 \cdot 1{,}84 = 2{,}2\ \text{V}$ nodig om het paneel tegen de helling te houden. Het profiel: in 1 s van 2,2 V naar 11,6 V (0 naar 0,10 m/s), 5 s constant, in 1 s terug naar 2,2 V, en stoppen op de eindeloopschakelaar. Hard aanzetten zou een stroomstoot van $12{,}8 / 1{,}2 = 10{,}7\ \text{A}$ geven.

4 (d) Twee **eindeloopschakelaars** (D23 open, D24 dicht), als verbreekcontact: een losse draad geeft ook 'stop'. Een **time-out** van 12 s (normaal duurt sluiten ongeveer 7 s, met ijs op de rails ongeveer 9 s): daarna is het paneel geblokkeerd. Een **zekering** van 5 A traag (normaal hoogstens 3,4 A, geblokkeerd 10 A). Een **elco** van 470 µF / 25 V aan de module (hoofdstuk 24). En in software nooit beide PWM-ingangen tegelijk.

ANTWOORD

BTS7960-module aan D6/D7/D8; PWM-waarde 232 bij 12,8 V (206 bij 14,4 V), bijgesteld met de gemeten batterijspanning; in 1 s aanlopen vanaf 2,2 V; eindeloopschakelaars, time-out, zekering 5 A en een bufferelco.

Klopt dit? Een sluiting kost volgens het model 131 J (0,036 Wh): het dak weegt in het energiebudget niets (hoofdstuk 20). Bij het openen helpt de zwaartekracht, maar de zelfremmende worm laat het paneel niet vrij lopen: de motor moet het ook dan 'afwikkelen', met minder stroom.



Het profiel van hoofdstuk 7 als PWM-waarde: bij een vollere batterij is minder nodig voor dezelfde snelheid.

CODE · 26.15

C++: het kasdak openen en sluiten

15/18

Het programma is een kleine toestandsmachine (hoofdstuk 32) met drie toestanden: stil, sluiten en openen. De knop (via het filter van hoofdstuk 24) start een beweging, of stopt een lopende. De functie `motor()` zorgt dat nooit beide PWM-ingangen tegelijk een waarde krijgen. De spanning volgt het profiel, en wordt omgerekend naar een PWM-waarde met de gemeten batterijspanning.

C++ (Arduino) **dakmotor.ino** 102 regels

```cpp
// dakmotor.ino - het schuifpaneel van het kasdak openen en sluiten (Arduino Mega)
// H-brugmodule met twee BTS7960: RPWM = sluiten, LPWM = openen, R_EN + L_EN samen.
// Eindeloopschakelaars zijn verbreekcontacten naar GND: HIGH = eindpunt bereikt (of draad los: dan ook stoppen).
const int KNOP = 22;          // S1 'kasdak' via RC-filter en 74HC14: HIGH = ingedrukt (hoofdstuk 24)
const int EIND_OPEN = 23;
const int EIND_DICHT = 24;
const int PWM_DICHT = 6;
const int PWM_OPEN = 7;
const int ENABLE = 8;
const int BATTERIJ = A3;      // deler 100 kOhm / 33 kOhm (hoofdstuk 22)

const float U_NODIG = 11.63;  // gemiddelde spanning voor 0,10 m/s bij 60 N (model)
const float U_START = 2.21;   // net genoeg om het paneel in beweging te houden
const unsigned long AANLOOP_MS = 1000, REMMEN_VANAF_MS = 6000, TIMEOUT_MS = 12000;

enum Toestand { STIL, SLUITEN, OPENEN };
Toestand toestand = STIL;
bool dakOpen = false;
unsigned long start = 0;

int vorigeLezing = LOW, stabiel = LOW;          // software-ontdendering (hoofdstuk 24)
unsigned long laatsteWissel = 0;

void motor(int richting, int waarde) {          // richting: +1 sluiten, -1 openen, 0 stil (remmen)
  analogWrite(PWM_DICHT, richting > 0 ? waarde : 0);
  analogWrite(PWM_OPEN, richting < 0 ? waarde : 0);   // nooit beide tegelijk
}
```

```
28
29  float batterijspanning() {
30    return analogRead(BATTERIJ) * 5.0 / 1024.0 * (133.0 / 33.0);
31  }
32
33  bool knopGedrukt(unsigned long nu) {            // true bij de flank 'net ingedrukt'
34    int lezing = digitalRead(KNOP);
35    if (lezing != vorigeLezing) {
36      laatsteWissel = nu;
37      vorigeLezing = lezing;
38    }
39    if (nu - laatsteWissel >= 20 && lezing != stabiel) {
40      stabiel = lezing;
41      return stabiel == HIGH;
42    }
43    return false;
44  }
45
46  void stop(const char* reden) {
47    motor(0, 0);
48    toestand = STIL;
49    Serial.println(reden);
50  }
51
52  void setup() {
53    pinMode(KNOP, INPUT);
54    pinMode(EIND_OPEN, INPUT_PULLUP);
55    pinMode(EIND_DICHT, INPUT_PULLUP);
56    pinMode(PWM_DICHT, OUTPUT);
57    pinMode(PWM_OPEN, OUTPUT);
58    pinMode(ENABLE, OUTPUT);
59    motor(0, 0);
60    digitalWrite(ENABLE, HIGH);
61    Serial.begin(9600);
62    dakOpen = (digitalRead(EIND_OPEN) == HIGH);
63  }
64
65  void loop() {
66    unsigned long nu = millis();
67
68    if (knopGedrukt(nu)) {
69      if (toestand != STIL) {
70        stop("gestopt met de knop");              // een druk tijdens de beweging = stoppen
71      } else {
72        toestand = dakOpen ? SLUITEN : OPENEN;
73        start = nu;
74        Serial.println(toestand == SLUITEN ? "dak sluiten" : "dak openen");
75      }
76    }
77    if (toestand == STIL) return;
78
79    unsigned long t = nu - start;
80    if (toestand == SLUITEN && digitalRead(EIND_DICHT) == HIGH) {
81      dakOpen = false;
82      stop("dak dicht");
83      return;
84    }
85    if (toestand == OPENEN && digitalRead(EIND_OPEN) == HIGH) {
86      dakOpen = true;
87      stop("dak open");
88      return;
89    }
90    if (t > TIMEOUT_MS) {
91      stop("ALARM: eindpunt niet bereikt");       // geblokkeerd, ijs of een losse schakelaar
92      return;
93    }
94
95    // snelheidsprofiel: 1 s aanlopen, constant, vanaf 6 s afremmen tot kruipsnelheid (30 %), tot de eindeloopschakelaar
```

```
  float s = 1.0;
  if (t < AANLOOP_MS) s = t / (float)AANLOOP_MS;
  else if (t > REMMEN_VANAF_MS) s = max(0.3, 1.0 - (t - REMMEN_VANAF_MS) / (float)AANLOOP_MS);
  float u = U_START + (U_NODIG - U_START) * s;
  int waarde = constrain((int)(u / batterijspanning() * 255 + 0.5), 0, 255);
  motor(toestand == SLUITEN ? +1 : -1, waarde);
}
```

Getest tegen een gesimuleerd dak: de PWM-waarde loopt in 1 s op tot 232 (bij 12,8 V), het dak raakt de eindeloopschakelaar na 6,8 s (met ijs op de rails, 79 N: 8,7 s; openen gaat sneller, 5,8 s, omdat de zwaartekracht helpt), en een tweede druk tijdens de beweging stopt de motor. Het programma remt af tot 30 % van de snelheid en kruipt zo tot de schakelaar: het paneel stopt nooit net voor het eindpunt. Als de eindeloopschakelaar niet komt, stopt de time-out na 12 s. Merk op dat de knop hier ook een **noodstop** is: dat is een eis voor elke bewegende machine waar mensen bij kunnen.

CODE · 26.16

Python: het werkpunt van de dakmotor

16/18

Het model van dit hoofdstuk in een paar regels: uit de riemkracht volgt het koppel, uit het koppel de stroom, uit de spanning en de stroom het toerental.

Python **dakmotor.py** 29 regels

```python
# dakmotor.py - het werkpunt van de dakmotor (gelijkstroommotor met worm 1:50) uitrekenen
import math

R, K, I0 = 1.2, 0.030, 0.25          # ankerweerstand (ohm), motorconstante (N m/A = V s/rad), nullaststroom (A)
OVERBRENGING, RENDEMENT = 50, 0.40   # worm, zelfremmend
r = 0.100 / (2 * math.pi)            # tandriemschijf: 100 mm riem per omwenteling


def werkpunt(U, F):
    """Stroom, toerental en rendement bij spanning U (V) en riemkracht F (N)."""
    M = F * r                                          # koppel aan de uitgaande as (N m)
    I = I0 + M / (RENDEMENT * OVERBRENGING * K)        # koppel ~ stroom
    w_motor = (U - R * I) / K                          # toerental ~ spanning min R I (tegen-emk)
    n = w_motor / OVERBRENGING * 60 / (2 * math.pi)    # toeren per minuut aan de schijf
    v = w_motor / OVERBRENGING * r                     # snelheid van de riem (m/s)
    rendement = M * w_motor / OVERBRENGING / (U * I)
    return I, n, v, rendement


print("   F (N)   I (A)   n (tpm)   v (m/s)   rendement")
for F in (0, 30, 60, 90, 120):
    I, n, v, eta = werkpunt(12.0, F)
    print(f"{F:8d}{I:8.2f}{n:10.1f}{v:10.3f}{eta:11.0%}")

# welke gemiddelde spanning geeft 0,10 m/s bij 60 N?
I60 = werkpunt(12.0, 60)[0]
U = R * I60 + K * (0.10 / r * OVERBRENGING)
for rail in (12.0, 12.8, 14.4):
    print(f"rail {rail:4.1f} V: gemiddeld {U:.2f} V nodig -> PWM {round(U / rail * 255)}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
   F (N)   I (A)   n (tpm)   v (m/s)   rendement
       0    0.25      74.5     0.124          0%
      30    1.05      68.4     0.114         27%
      60    1.84      62.3     0.104         28%
      90    2.64      56.2     0.094         27%
     120    3.43      50.2     0.084         24%
rail 12.0 V: gemiddeld 11.63 V nodig -> PWM 247
```

```
rail 12.8 V: gemiddeld 11.63 V nodig -> PWM 232
rail 14.4 V: gemiddeld 11.63 V nodig -> PWM 206
```

Het rendement is het hoogst rond 60 N, en daalt bij lichte last (de nullaststroom weegt dan zwaar door) en bij zware last (de verliezen RI^2 groeien met het kwadraat van de stroom).

NASLAG · 26.17

Overzicht: elektromagnetisme en actuatoren

17/18

veld van een draad en van een lange spoel

$$B = \frac{\mu_0 I}{2\pi r} \qquad B = \frac{\mu_0 N I}{l}$$

symbool	betekenis	eenheid
B	magnetische inductie	T
μ_0	$4\pi \cdot 10^{-7}$	T·m/A

kracht op een stroomdraad (loodrecht op B)

$$F = B \cdot I \cdot l$$

symbool	betekenis	eenheid
l	lengte in het veld	m

koppel op een spoel (maximaal)

$$M = NBIA$$

symbool	betekenis	eenheid
A	oppervlakte van een winding	m²

gelijkstroommotor

$$U = RI + k\omega \qquad M = k(I - I_0)$$

symbool	betekenis	eenheid
k	motorconstante	N·m/A = V·s/rad
ω	hoeksnelheid, $2\pi n/60$	rad/s

overbrenging

$$n_{\text{uit}} = \frac{n}{i} \qquad M_{\text{uit}} = \eta \cdot i \cdot M$$

symbool	betekenis	eenheid
i	overbrengingsverhouding	–
η	rendement (worm zelfremmend: < 0,5)	–

van schijf naar riem

$$M = F \cdot r \qquad v = \omega \cdot r$$

symbool	betekenis	eenheid
r	straal van de schijf	m

actuator	stuur je met	goed in	let op
gelijkstroommotor (met borstels)	MOSFET (één richting) of H-brug, PWM	eenvoudig, sterk, snel	positie onbekend; vrijloopdiode; aanloopstroom
servomotor	puls van 1-2 ms om de 20 ms	een hoek instellen (0-180°)	klein koppel; eigen voeding bij grote servo's
stappenmotor	driver met STEP/DIR	precieze positie zonder sensor	kan stappen verliezen; stroom ook in stilstand
relais	transistor of relaismodule	galvanisch gescheiden aan/uit, ook 230 V	traag, slijt, klikt; vrijloopdiode over de spoel
duwstang (lineaire actuator)	H-brug of twee wisselrelais	rechtlijnige beweging, met eindschakelaars	traag (mm per seconde)

S1	S2	S3	S4	gevolg
dicht	open	open	dicht	vooruit (links +, rechts –)
open	dicht	dicht	open	achteruit
open	dicht	open	dicht	remmen: motor kortgesloten via de onderkant
open	open	open	open	vrijlopen (uitbollen)
dicht	dicht	×	×	**verboden**: kortsluiting over de voeding
×	×	dicht	dicht	**verboden**: kortsluiting over de voeding

regel	duim	vingers	handpalm
veld rond een draad	stroom	draairichting van het veld	

regel	duim	vingers	handpalm
spoel	noordpool	draairichting van de stroom	
kracht (vlakke hand)	stroom	veld (N → S)	kracht

HET GEHEEL · 26.18

Waar motoren vandaan komen en waar ze heen gaan

18/18

Dit hoofdstuk sluit het deel elektronica af, en verbindt het met de mechanica: de kracht, het koppel, de arbeid en het rendement van de hoofdstukken 8 tot 10 worden hier elektrisch. Een motor is de plaats waar een stroom een kracht wordt, en een dynamo (Expert) de plaats waar een kracht een stroom wordt.

F12 Hefbomen en mechanismen
Tandwielen, riemen en de wormoverbrenging die zelfremmend is.

F25 Sensoren en automatisch regelen
Servo, relais en de duwstang van de meetkas met twee relais.

I7 Beweging beschrijven
Het bewegingsprofiel van het kasdak: aanlopen, constant, afremmen.

I8 De wetten van Newton
De 60 N die de riem moet leveren, en de derde wet in de stroombalans.

I9 Arbeid, energie en vermogen
Vermogen, arbeid per sluiting en het rendement van motor en worm.

I10 Moment en evenwicht
Koppel als moment: M = F · r aan de tandriemschijf.

I23 Halfgeleiders: diode, led en transistor
De MOSFET, de vrijloopdiode en de transistor die een relais schakelt.

I24 Condensatoren en de RC-tijd
De ontdenderde knop die het dak bedient, en de bufferelco aan de H-brug.

I25 Van analoog naar digitaal
PWM regelt het toerental; servo's lezen de lengte van een puls.

I31 Regelsystemen
De servo als gesloten regelkring; open lus tegenover gesloten lus.

I32 Toestandsmachines
Stil, openen, sluiten, alarm: de dakmotor als toestandsmachine.

I33 Een mobiele robot
Twee motoren met H-bruggen voor een robot die rijdt en draait.

E19 Elektrische en magnetische velden, inductie
Lorentzkracht, inductie en de tegen-emk exact; de generator.

E24 Vermogenselektronica en elektrische machines
Borstelloze motoren en hun sturing, en H-bruggen op hoge frequentie.

Habitat

Het kasdak krijgt een wormwielmotor van 12 V (onbelast 74 tpm, blokkeerkoppel 5,85 N·m) op een tandriemschijf HTD 5M met 20 tanden: 60 tpm is 0,10 m/s. Bij 60 N trekt hij 1,84 A (22 W), bij 120 N 3,4 A. Hij hangt aan een H-brugmodule met twee BTS7960 op D6 (sluiten), D7 (openen) en D8 (enable), met eindeloopschakelaars op D23 (open) en D24 (dicht), een zekering van 5 A en een bufferelco van 470 µF. De knop S1 op D22 start en stopt; de PWM-waarde (232 bij 12,8 V) wordt bijgesteld met de batterijspanning op A3. Een sluiting kost ongeveer 0,04 Wh.

SAMENVATTING

In het kort

- Stroom maakt een magnetisch veld: cirkels rond een draad (rechterhandregel), een staafmagneetveld bij een spoel. IJzer versterkt het: een elektromagneet.

- Een stroomdraad in een veld voelt een kracht $F = BIl$, loodrecht op draad en veld (vlakke rechterhand: duim stroom, vingers veld, handpalm kracht).
- Een relais is een elektromagneet die een contact (COM naar NC of NO) omzet; stuurkring en lastkring zijn gescheiden.
- In een gelijkstroommotor draait een spoel in een veld; de commutator keert de stroom elke halve draai om.
- $U = RI + k\omega$ en $M = kI$: het toerental volgt de spanning, het koppel de stroom. Stilstaand trekt een motor de blokkeerstroom U/R.
- Een servo neemt een hoek aan volgens een puls van 1-2 ms om de 20 ms; een stappenmotor draait per stap een vaste hoek (vaak 1,8°).
- Een H-brug (vier schakelaars) stuurt een motor in twee richtingen; twee schakelaars aan dezelfde kant tegelijk dicht is een kortsluiting.

BEGRIPPEN

Wat je nu kent

magnetisch veld
Gebied waarin een magneet of een stroom een kracht voelt; de sterkte is de magnetische inductie B, in tesla (T).

rechterhandregel (veld)
Duim in de richting van de stroom: je gekromde vingers wijzen de draairichting van het veld rond de draad aan.

elektromagneet
Spoel rond een ijzeren kern die magnetisch is zolang er stroom door loopt.

relais
Schakelaar die bediend wordt door een elektromagneet; stuurkring en geschakelde kring zijn elektrisch gescheiden.

commutator
Gespleten ring op de as van een gelijkstroommotor die de stroom in de rotor elke halve draai omkeert.

tegen-emk
De spanning die een draaiende motor zelf opwekt, als een dynamo; ze werkt de voedingsspanning tegen en is evenredig met het toerental.

koppel (van een motor)
Het draaiende effect op de as, in N·m; bij een gelijkstroommotor evenredig met de stroom.

servomotor
Motor met overbrenging, positiesensor en regelaar, die een hoek aanneemt volgens de lengte van een stuurpuls.

stappenmotor
Motor die per stroompuls een vaste hoek draait (vaak 1,8°), zodat je de positie kent zonder sensor.

H-brug
Vier schakelaars rond een motor waarmee je de stroom in twee richtingen kan sturen.

zelfremmend
Overbrenging die de last niet kan terugdraaien, zoals een worm met een kleine spoedhoek.

HANDBOEK · HOOFDSTUK 27

Python: de basis

Van blokken naar tekst: variabelen, rekenen, keuzes, lussen en foutmeldingen die je leert lezen

In Fundamental stapelde je blokken tot een programma, en in de vorige hoofdstukken las je al korte Python-programma's. Nu schrijf je ze zelf. Je zet een blokprogramma om in tekst, leert rekenen, kiezen en herhalen in Python, en vooral: je leert de foutmeldingen lezen, zodat een fout geen muur is maar een aanwijzing. Aan het eind schrijf je een rekenhulp voor het energiebudget van de klimaatmodule.

10× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 2× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 5× code · ± 8 lestijden

P Programmeren en computationeel denken

Dezelfde simulatie twee keer: links in blokken, rechts in Python.

Na dit hoofdstuk

- Je kan een blokprogramma omzetten in een Python-programma en uitleggen wat de dubbelepunt en het inspringen doen.
- Je kan variabelen met een zinvolle naam maken en de types int, float, str en bool herkennen en in elkaar omzetten.
- Je kan rekenen met +, −, *, /, //, % en ** en de volgorde van de bewerkingen toepassen.
- Je kan resultaten netjes tonen met print() en f-strings, afgerond en met een decimale komma.
- Je kan uitleggen hoe input() werkt en invoer (ook met een komma) omzetten in een getal.
- Je kan keuzes programmeren met if, elif en else, met vergelijkingen en met and, or en not.
- Je kan herhalen met for en range() en met while, en een lus naspelen met een trace-tabel.
- Je kan een foutmelding lezen, de oorzaak vinden en de fout verbeteren, ook een fout zonder melding.
- Je kan een Python-rekenhulp schrijven en gebruiken voor het energiebudget van de klimaatmodule.

VERHAAL · 27.1

Een kerstproject dat wereldtaal werd

In december 1989 zocht de Nederlandse programmeur **Guido van Rossum** een project voor de kerstvakantie. Hij werkte bij het CWI, het onderzoeksinstituut voor wiskunde en informatica in Amsterdam, aan de taal ABC, die bedoeld was om programmeren makkelijk te maken. Hij begon aan een opvolger die even leesbaar moest zijn maar ook echt werk moest kunnen doen. De naam haalde hij niet bij de slang, maar bij de Britse comedyreeks *Monty Python's Flying Circus*.

In februari 1991 zette hij de eerste versie online. Andere programmeurs begonnen mee te bouwen, en Python groeide uit tot een van de populairste programmeertalen ter wereld. In december 2008 verscheen Python 3, dat een paar oude keuzes rechtzette; sinds 2020 wordt het oude Python 2 niet meer onderhouden. Alles in dit boek is Python 3.

Wat Python bijzonder maakt, is de leesbaarheid. Er zijn geen accolades of puntkomma's zoals in C++; de structuur zie je aan het **inspringen**. Een van de richtlijnen van de taal zegt het kort: "Readability counts." Een programma schrijf je één keer, maar lees je tientallen keren, ook als je een fout zoekt.

Wetenschappers en ingenieurs gebruiken Python voor bijna alles wat met data te maken heeft. De rekenbibliotheek numpy, die je in hoofdstuk 4 en 22 al tegenkwam, zat in de software van de eerste meting van zwaartekrachtgolven (2015) en van de eerste foto van een zwart gat (2019).

Waarom dit hoofdstuk ertoe doet

Blokken zijn ideaal om te leren denken als een programmeur, maar een blokprogramma van honderd regels wordt een onoverzichtelijke toren. In tekst schrijf je sneller, kan je code kopiëren, delen en vergelijken, en gebruik je bibliotheken voor grafieken, data en AI. De klimaatmodule en later het brein van het station draaien op zulke code.

Een programma lezen is net zo belangrijk als het schrijven.

UITLEG · 27.2

Wat je al kan, en hoe je Python draait

2/26

wat je al kan	wat er nu bijkomt
een algoritme als stroomdiagram tekenen en naspelen met een trace-tabel (F22)	hetzelfde algoritme in Python, en lussen naspelen met een trace-tabel
blokken stapelen: volgorde, herhaal, als … dan … anders, variabelen (F23)	dezelfde bouwstenen als tekst: regels, `for`, `while`, `if`, `=`
een microcontroller programmeren in C++ met `setup()` en `loop()` (F24, F25)	Python op de computer; C++ voor de Arduino komt terug in hoofdstuk 29
korte Python-programma's lezen: kalibreren (hoofdstuk 3), het energiebudget (hoofdstuk 20), een NTC ijken (hoofdstuk 22)	zulke programma's zelf schrijven, fouten lezen en verbeteren

Een Python-programma is een gewoon tekstbestand met de extensie `.py`. Je schrijft het in een **editor**. Voor beginners is **Thonny** handig (gratis, met Python erbij); **IDLE** zit bij elke installatie van Python; wie verder wil, gebruikt VS Code. Met de knop *Run* (of F5) geef je het bestand aan de **interpreter**: het programma dat je tekst leest en regel per regel uitvoert. Wat `print()` toont, en eventuele foutmeldingen, verschijnen in het uitvoervenster.

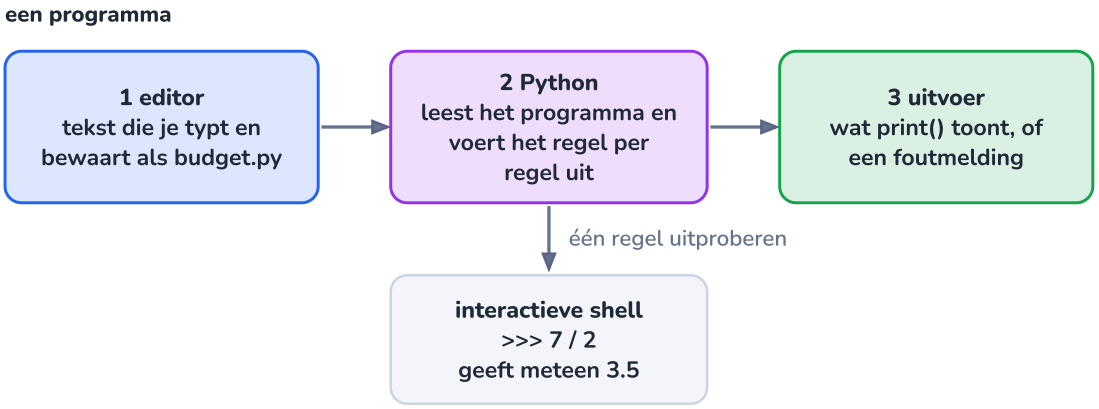
Onderaan in Thonny en IDLE zit ook de **interactieve shell**, met de prompt `>>>`. Daar typ je één regel en zie je meteen de uitkomst: ideaal om iets uit te proberen. Een programma bewaar je wel altijd in een bestand.

Interpreter tegenover compiler

Je Arduino-programma (F24) werd eerst volledig vertaald naar machinetaal (*compileren*) en dan naar het bordje gestuurd. Python leest je bestand, controleert de schrijfwijze en voert het meteen uit, regel per regel. Dat maakt uitproberen snel, maar een tikfout in een naam op regel 40 merk je pas als Python regel 40 bereikt. Alleen fouten in de schrijfwijze zelf ziet Python vooraf, vóór er iets draait.

Bestandsnamen

Geef je bestanden korte namen zonder spaties: `budget.py`, `kas_simulatie.py`. Noem een eigen bestand nooit zoals een bibliotheek (`random.py`, `math.py`, `csv.py`): dan laadt Python jouw bestand in plaats van de bibliotheek.



Een programma bewaar je als bestand; in de shell probeer je één regel uit.

CODE · 27.3

Van blokken naar tekst: de meetkas als simulatie

3/26

In F23 bouwde je een simulatie van de meetkas in blokken: het bodemvocht daalt elk uur 5 procentpunt, en onder 30 % geeft de pomp een beurt van 30 s (240 mL), die het vocht 16 procentpunt hoger maakt. Hier staat hetzelfde programma in Python. Lees het naast de blokken, regel per blok.

Python `kas_simulatie.py` 14 regels

```python
# kas_simulatie.py - de meetkas uit F23, nu in Python
vocht = 50              # bodemvocht (%)
water = 30              # water in het reservoir (L)
beurten = 0             # teller: aantal pompbeurten

for uur in range(48):   # een ronde is een uur
    vocht = vocht - 5   # de planten drinken
    if vocht < 30:
        vocht = vocht + 16          # pompbeurt van 30 s
        water = water - 0.24        # 240 mL
        beurten = beurten + 1

print("beurten:", beurten)
print("water over:", round(water, 2), "L")
```

wanneer op wordt geklikt
maak vocht 50
maak water 30
maak beurten 0
herhaal 48
verander vocht met -5
als vocht < 30 dan
verander vocht met 16
verander water met -0.24
verander beurten met 1
zeg voeg beurten: en beurten samen

Hetzelfde algoritme in blokken; de wachttijd uit F23 is weggelaten.

Uitvoer:

```
beurten: 14
water over: 26.64 L
```

blok (Scratch)	Python	wat er gebeurt
wanneer op de groene vlag wordt geklikt	(het programma starten)	Python begint gewoon bovenaan
maak [vocht] (50)	`vocht = 50`	de naam vocht wijst naar 50
herhaal (48)	`for uur in range(48):`	48 rondes; uur telt mee: 0, 1, ..., 47
verander [vocht] met (−5)	`vocht = vocht - 5`	nieuwe waarde = oude waarde − 5
als <vocht < 30> dan	`if vocht < 30:`	alleen als de voorwaarde waar is
de mond van een C-blok	inspringen (4 spaties)	wat ingesprongen staat, hoort bij de lus of de keuze
zeg (voeg [beurten:] en (beurten) samen)	`print("beurten:", beurten)`	tekst en waarde in de uitvoer

Drie dingen vallen op. Een **C-blok** wordt een regel die eindigt op een **dubbelepunt**, gevolgd door regels die **inspringen**: zo weet Python wat in de lus of in de keuze hoort. Een **maak**-blok en een **verander**-blok worden allebei een regel met `=`. En alles achter een `#` is **commentaar**: uitleg voor mensen, Python slaat het over. In Scratch gebruikte je 'wacht 0.2 sec.' om de simulatie te zien lopen; in Python doet `import time` bovenaan en `time.sleep(0.2)` in de lus hetzelfde.

ONTLEED · 27.4

Een Python-programma ontleed

4/26

Hetzelfde programma, zonder het commentaar achter de regels. De regelnummers staan er niet echt in: editors tonen ze in de kantlijn, en foutmeldingen verwijzen ernaar.

```
1  # kas_simulatie.py - de meetkas uit F23, nu in Python
2  vocht = 50
3  water = 30
4  beurten = 0
5
6  for uur in range(48):
7      vocht = vocht - 5
8      if vocht < 30:
9          vocht = vocht + 16
10         water = water - 0.24
11         beurten = beurten + 1
12
13 print("beurten:", beurten)
14 print("water over:", round(water, 2), "L")
```

Kleuren zoals in een editor: sleutelwoorden, ingebouwde functies, tekst en commentaar.

1 **Commentaar**
Alles na # tot het einde van de regel. Zet bovenaan wat het programma doet; leg uit *waarom*, niet wat er al staat.

2 **Toekenning**
`vocht = 50`: de naam vocht wijst voortaan naar de waarde 50. Het isgelijkteken betekent 'wordt', niet 'is gelijk aan'.

3 **For-lus**
`range(48)` levert 0, 1, ..., 47: de lus loopt 48 keer, en uur krijgt elke ronde de volgende waarde.

4 **Keuze met een voorwaarde**
`vocht < 30` is True of False. De dubbelepunt kondigt een blok aan dat alleen bij True draait.

5 **Inspringen**
Vier spaties: hoort bij de for. Acht spaties: hoort bij de if in de for. Het inspringen is de mond van het C-blok.

6 **Na de lus**
Niet ingesprongen: deze regel draait pas als de 48 rondes voorbij zijn, precies één keer.

7 **Functies aanroepen**
`round(water, 2)` geeft water afgerond op 2 decimalen; print() toont al zijn argumenten, gescheiden door een spatie.

UITLEG · 27.5

Variabelen: een naam voor een waarde

5/26

Een **variabele** is een naam die naar een waarde wijst. Je maakt ze met een **toekenning**: links de naam, dan `=`, rechts de waarde of een berekening. Python rekent eerst de rechterkant uit en hangt dan de naam aan het resultaat. Daarom is `vocht = vocht - 5` geen onzin: neem de huidige waarde van vocht (50), trek er 5 af, en laat vocht voortaan naar 45 wijzen. Korter kan ook: `vocht -= 5`, `beurten += 1`.

In F23 dacht je aan een variabele als een doos. In Python klopt een **naamkaartje** beter: de waarde bestaat, en de naam hangt eraan. Voor getallen maakt dat geen verschil, voor lijsten (hoofdstuk 28) wel.

Regels voor namen. Letters, cijfers en een liggend streepje; niet beginnen met een cijfer; geen spaties. Hoofdletters tellen: `Vocht` en `vocht` zijn twee verschillende namen. Woorden van de taal zelf (`for`, `if`, `while`, ...) mogen niet. Kies namen die zeggen wat erin zit, in kleine letters met streepjes: `verbruik_dag`, `wh_per_wp`. Een programma met `x`, `y` en `z` kan je volgende week zelf niet meer lezen. In dit boek zijn de namen Nederlands en de sleutelwoorden van Python Engels.

"= betekent is gelijk aan"

In de wiskunde is $x = x + 1$ een vergelijking zonder oplossing. In Python is `x = x + 1` een opdracht: 'x wordt x plus 1'. Wil je testen of twee waarden gelijk zijn, dan schrijf je **twee** isgelijktekens: `vocht == 30`. Dat geeft True of False en verandert niets.

Een nieuwe toekenning verlegt de pijl; de oude waarde 50 wordt niet meer gebruikt.

PROEF · 27.6

Rekent Python zoals jij? Voorspel, probeer, verklaar

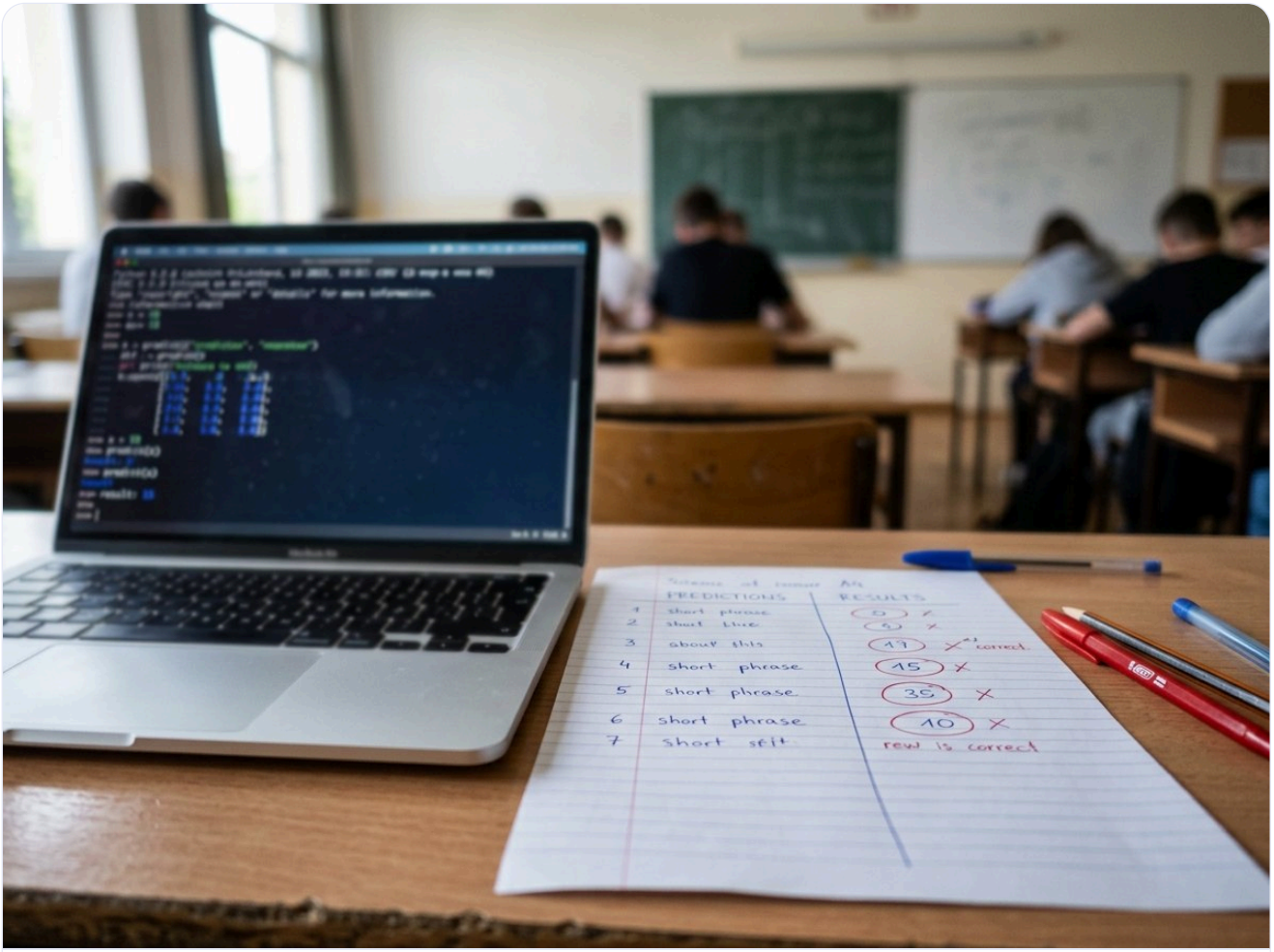
6/26

ONDERZOEKSVRAAG

Wat antwoordt Python op 7 / 2, 7 // 2, "7" + "2", 0,1 + 0,2 en 3269,5 > 3000?

MATERIAAL

- computer met Python 3 (Thonny of IDLE)
- papier en pen voor de voorspellingen



Eerst voorspellen op papier, dan pas typen: zo zie je waar Python anders rekent dan jij.

WERKWIJZE

1. Open de interactieve shell (het venster met >>>).
2. Schrijf voor elke regel uit de waarneming eerst op papier wat je verwacht. Let op de aanhalingstekens.
3. Typ de regels één voor één en druk telkens op Enter. Noteer wat Python antwoordt.
4. Omcirkel elke uitkomst die afwijkt van je voorspelling.
5. Vraag met `type(...)` het type op van de uitkomsten die je verrasten.

WAARNEMING

Zo antwoordt Python (Python 3.13):

```
>>> 7 / 2
3.5
>>> 7 // 2
3
>>> 10 / 5
2.0
>>> "7" + "2"
'72'
>>> "7" * 3
'777'
>>> 0.1 + 0.2
0.30000000000000004
>>> 3269.5 > 3000
True
>>> type(7 / 2)
<class 'float'>
>>> type(3269.5 > 3000)
<class 'bool'>
```

VERKLARING

Elke waarde heeft een **type**, en de bewerking hangt af van het type. `/` geeft altijd een kommagetal (`float`), ook als de deling opgaat. `//` geeft het gehele deel. Tussen aanhalingstekens staat **tekst** (`str`): `+` plakt tekst aan elkaar en `* 3` herhaalt ze. Een vergelijking levert `True` of `False` (`bool`).

En 0,1 + 0,2? Een computer bewaart kommagetallen in het tweetallig stelsel (hoofdstuk 25). Zoals 1/3 in het tientallig stelsel nooit precies opgaat (0,333…), gaat 0,1 in het tweetallig stelsel niet op. De fout zit in het zestiende cijfer: voor metingen onbelangrijk, maar test kommagetallen daarom nooit met `==`. Rond af of vergelijk met een kleine marge.

UITLEG · 27.7

Types: int, float, str en bool

7/26

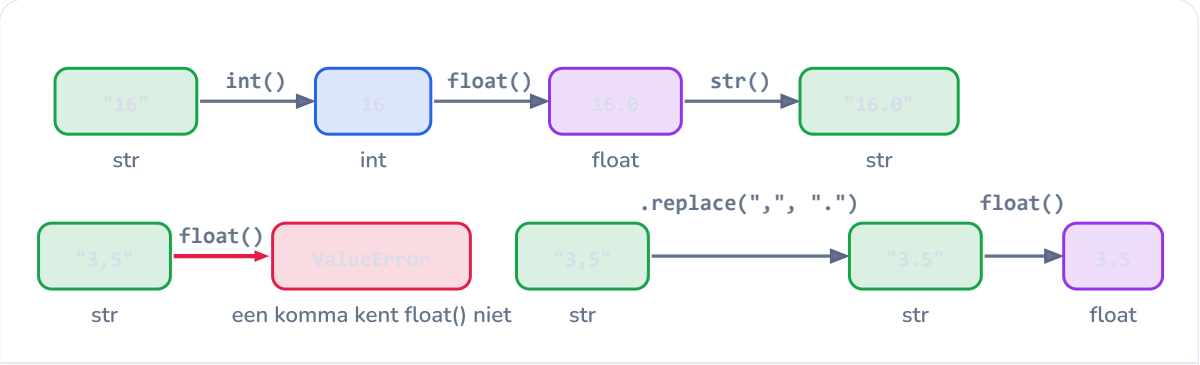
type	naam	voorbeelden	gebruik
`int`	geheel getal (*integer*)	`16`, `-3`, `0`, `2457`	tellen: beurten, dagen, lichturen, ADC-waarden
`float`	kommagetal (*floating point*)	`3.5`, `1400.0`, `2.99`, `1e-9`	meten en rekenen: vermogen, temperatuur, pH
`str`	tekst (*string*)	`"kas"`, `'16'`, `""` (lege tekst)	namen, berichten, alles wat input() binnenkrijgt
`bool`	waarheidswaarde (*boolean*)	`True`, `False`	uitkomst van een vergelijking: `vocht < 30`

Python beslist het type zelf, uit de schrijfwijze: `16` is een int, `16.0` een float, `"16"` tekst. Een decimaal getal schrijf je in Python **altijd met een punt**, zoals in Scratch. Omzetten doe je met een functie die de naam van het type draagt:

```
>>> int("16") + 4
20
>>> float("1.25") * 1400
1750.0
>>> str(16) + " lichturen"
'16 lichturen'
>>> int(7.9)
7
>>> round(7.9)
8
>>> float(3)
3.0
```

Let op het verschil tussen `int(7.9)` en `round(7.9)`: int() kapt de decimalen af, round() rondt af. Tekst die geen getal voorstelt, kan je niet omzetten: `float("3,5")` geeft een foutmelding, want Python kent de decimale komma niet. Vervang eerst de komma door een punt.

> **type() als detective**
> Doet een berekening iets vreemds, vraag dan het type op: `type(x)`. Een getal dat stiekem tekst is (`'25'` in plaats van `25`), verklaart de helft van alle beginnersfouten met input().



int(), float() en str() zetten een waarde om; een decimale komma moet eerst een punt worden.

UITLEG · 27.8

Rekenen: van + tot **

8/26

Python rekent zoals een rekenmachine, met een paar tekens die je moet kennen:

teken	bewerking	voorbeeld	uitkomst	opmerking
`+`	optellen	`17 + 5`	`22`	
`-`	aftrekken	`17 - 5`	`12`	
`*`	vermenigvuldigen	`17 * 5`	`85`	geen × of ·
`/`	delen	`17 / 5`	`3.4`	altijd een float, ook bij `10 / 5`
`//`	gehele deling	`17 // 5`	`3`	afgerond naar beneden
`%`	rest (modulo)	`17 % 5`	`2`	wat overblijft na de gehele deling

teken	bewerking	voorbeeld	uitkomst	opmerking
`**`	macht	`2 ** 10`	`1024`	niet ^ (dat is iets anders in Python)

Met `//` en `%` reken je met hele stukken en resten: minuten naar uren en minuten, plantplaatsen over goten verdelen, testen of een getal even is (`n % 2 == 0`). De **volgorde** is die van de wiskunde: eerst machten, dan vermenigvuldigen en delen, dan optellen en aftrekken, van links naar rechts; haakjes gaan voor alles.

uitdrukking	uitkomst	waarom
`2 + 3 * 4`	`14`	eerst *, dan +
`(2 + 3) * 4`	`20`	haakjes eerst
`2 * 3 ** 2`	`18`	eerst de macht: 3 ** 2 = 9
`-2 ** 2`	`-4`	de macht gaat vóór het minteken: −(2²)
`(-2) ** 2`	`4`	met haakjes: (−2)²
`7 - 4 - 1`	`2`	van links naar rechts
`-7 // 2`	`-4`	naar beneden afgerond: −3,5 wordt −4

Voor wortels, sinussen en π laad je de bibliotheek `math`:

```
>>> import math
>>> math.sqrt(1.65 ** 2 + 1.00 ** 2)
1.9293781381574737
>>> round(math.pi, 4)
3.1416
>>> math.sin(math.radians(30))
0.49999999999999994
```

De eerste regel is de diagonaal van een zonnepaneel van 1,65 m op 1,00 m (stelling van Pythagoras). Let op de laatste: math.sin rekent in radialen, dus eerst omzetten; en ook hier is het resultaat een kommagetal met een klein foutje in de laatste cijfers.

Gehele deling en rest: 17 = 3 · 5 + 2.

UITGEWERKT VOORBEELD · 27.9

Seconden omrekenen naar een kloktijd

9/26

De datalogger van de klimaatmodule telt de seconden sinds middernacht. Een meting heeft het tijdstip 49 317. Schrijf een programma dat dat omzet naar uren, minuten en seconden.

GEGEVEN

- t = 49 317 s sinds middernacht
- 1 h = 3600 s, 1 min = 60 s

GEVRAAGD

- uren, minuten en seconden, getoond als 13:41:57

OPLOSSING

1. **Uren.** Hoeveel volle uren zitten er in t? Dat is de gehele deling: `uren = t // 3600` geeft 13.
2. **Wat overblijft.** De rest na die volle uren: `rest = t % 3600` geeft 2517 s.
3. **Minuten en seconden.** Hetzelfde met 60: `rest // 60` is 41 minuten, `rest % 60` is 57 seconden.
4. **Tonen.** Met een f-string en `:02d` (twee cijfers, aanvullen met een nul) ziet het eruit als een klok:

```
t = 49317
uren = t // 3600
rest = t % 3600
minuten = rest // 60
seconden = rest % 60
print(f"{uren:02d}:{minuten:02d}:{seconden:02d}")
```

```
13:41:57
```

ANTWOORD

Tijdstip 49 317 s is 13:41:57.

Klopt dit? Terugrekenen: $13 \cdot 3600 + 41 \cdot 60 + 57 = 46\,800 + 2460 + 57 = 49\,317$. Klopt. En een test op de randen: t = 0 moet 00:00:00 geven en t = 86 399 (de laatste seconde van de dag) 23:59:59.

CODE · 27.10

Uitvoer: print() en f-strings

10/26

`print()` toont al zijn argumenten, met een spatie ertussen, en begint daarna een nieuwe regel. Voor nette uitvoer gebruik je een **f-string**: een tekst met een `f` vóór het aanhalingsteken, waarin je waarden tussen accolades zet. Achter een dubbelepunt in de accolades staat de **opmaak**: `{zon:.0f}` betekent 'zon als kommagetal met 0 decimalen'.

Python **uitvoer.py** 13 regels

```
# uitvoer.py - getallen netjes tonen
zon = 1400 * 2.99       # Wh per dag in maart (4 panelen van 350 Wp)
verbruik = 3269.5       # Wh per dag
water = 30 - 0.24 - 0.24 - 0.24
print(water)
print("water:", round(water, 2), "L")
print(f"zon: {zon:.0f} Wh, verbruik: {verbruik:.1f} Wh")
print(f"saldo: {zon - verbruik:+.1f} Wh")
print(f"dekking: {zon / verbruik:.1%}")
kwh = f"{verbruik / 1000:.2f}".replace(".", ",")
print("verbruik:", kwh, "kWh per dag")
print(f"{'lampen':<10}{3040:>6} Wh")
print(f"{'pomp':<10}{48:>6} Wh")
```

Uitvoer:

```
29.280000000000005
water: 29.28 L
```

```
zon: 4186 Wh, verbruik: 3269.5 Wh
saldo: +916.5 Wh
dekking: 128.0%
verbruik: 3,27 kWh per dag
lampen      3040 Wh
pomp          48 Wh
```

De eerste regel toont waarom je afrondt: drie keer 0,24 aftrekken geeft een getal met een foutje in de vijftiende decimaal. `round(water, 2)` rekent verder met de afgeronde waarde; een f-string rondt alleen af bij het tonen en laat de variabele zelf ongemoeid.

waarde	f-string	resultaat	betekenis
`x = 3269.5`	`f"{x:.2f}"`	`'3269.50'`	2 decimalen
`x = 3269.5`	`f"{x:.0f}"`	`'3270'`	afgerond op een geheel getal
`x = 3269.5`	`f"{x:10.1f}"`	`'    3269.5'`	10 tekens breed, 1 decimaal (rechts uitgelijnd)
`x = 916.5`	`f"{x:+.1f}"`	`'+916.5'`	altijd een teken, ook +
`x = 0.535`	`f"{x:.0%}"`	`'54%'`	als procent (maal 100)
`s = "pomp"`	`f"{s:<10}|"`	`'pomp     |'`	10 tekens breed, links uitgelijnd
`n = 48`	`f"{n:>5}"`	`'   48'`	5 tekens breed, rechts uitgelijnd
`n = 48`	`f"{n:05d}"`	`'00048'`	aanvullen met nullen

Een decimale komma tonen

Python schrijft getallen altijd met een punt. Voor een Nederlandstalig verslag maak je eerst de tekst en vervang je dan de punt: `f"{x:.2f}".replace(".", ",")`. Doe dat alleen bij het tonen: met de tekst `"3,27"` kan je niet meer rekenen.

Afronden op ,5

Python rondt een getal dat precies op ,5 eindigt af naar het **even** getal: `round(2.5)` geeft 2, `round(3.5)` geeft 4. Zo vallen afrondingsfouten bij veel getallen tegen elkaar weg. Bij metingen merk je er bijna nooit iets van.

UITLEG · 27.11

Invoer: input() geeft altijd tekst

11/26

Met `input()` vraagt een programma iets aan de gebruiker. Het toont de tekst tussen de haakjes, wacht tot de gebruiker iets typt en op Enter drukt, en geeft wat getypt werd terug als **tekst**. Ook als de gebruiker `30` typt, krijg je `"30"`. Om ermee te rekenen, zet je het om met `int()` of `float()`. Wie een decimale komma typt, vang je op met `.replace(",", ".")`:

```python
# reservoir.py - hoe lang doet het reservoir van de meetkas het?
inhoud = float(input("Inhoud van het reservoir (L)? ").replace(",", "."))
verbruik = float(input("Verbruik per dag (L)? ").replace(",", "."))
dagen = inhoud // verbruik
print(f"Het reservoir houdt {dagen:.0f} volle dagen.")
```

Zo ziet een sessie eruit (het programma is gedraaid; de gebruiker typte 30 en 1,8):

```
Inhoud van het reservoir (L)? 30
Verbruik per dag (L)? 1,8
Het reservoir houdt 16 volle dagen.
```

Wat gebeurt er met de tekst voor ze een getal wordt? In de shell kan je het stap voor stap nagaan, met een variabele in de plaats van input():

```
>>> tekst = "1,8"
>>> type(tekst)
<class 'str'>
>>> tekst * 2
'1,81,8'
>>> tekst.replace(",", ".")
'1.8'
>>> float(tekst.replace(",", "."))
1.8
>>> 30 // float(tekst.replace(",", "."))
16.0
```

> ⚠ **Programma's met input() in dit boek**
>
> De programma's in dit boek worden gedraaid terwijl de pagina's gemaakt worden, en dan zit er niemand aan het toetsenbord. Waar een programma invoer nodig heeft, staat de invoer daarom in een variabele, met de echte input()-regel als commentaar ernaast. Wil je het zelf interactief maken, dan wissel je die twee om.

Wat de gebruiker typt, komt als tekst binnen: eerst omzetten, dan rekenen.

CODE · 27.12

Voorwaarden: if, elif en else

12/26

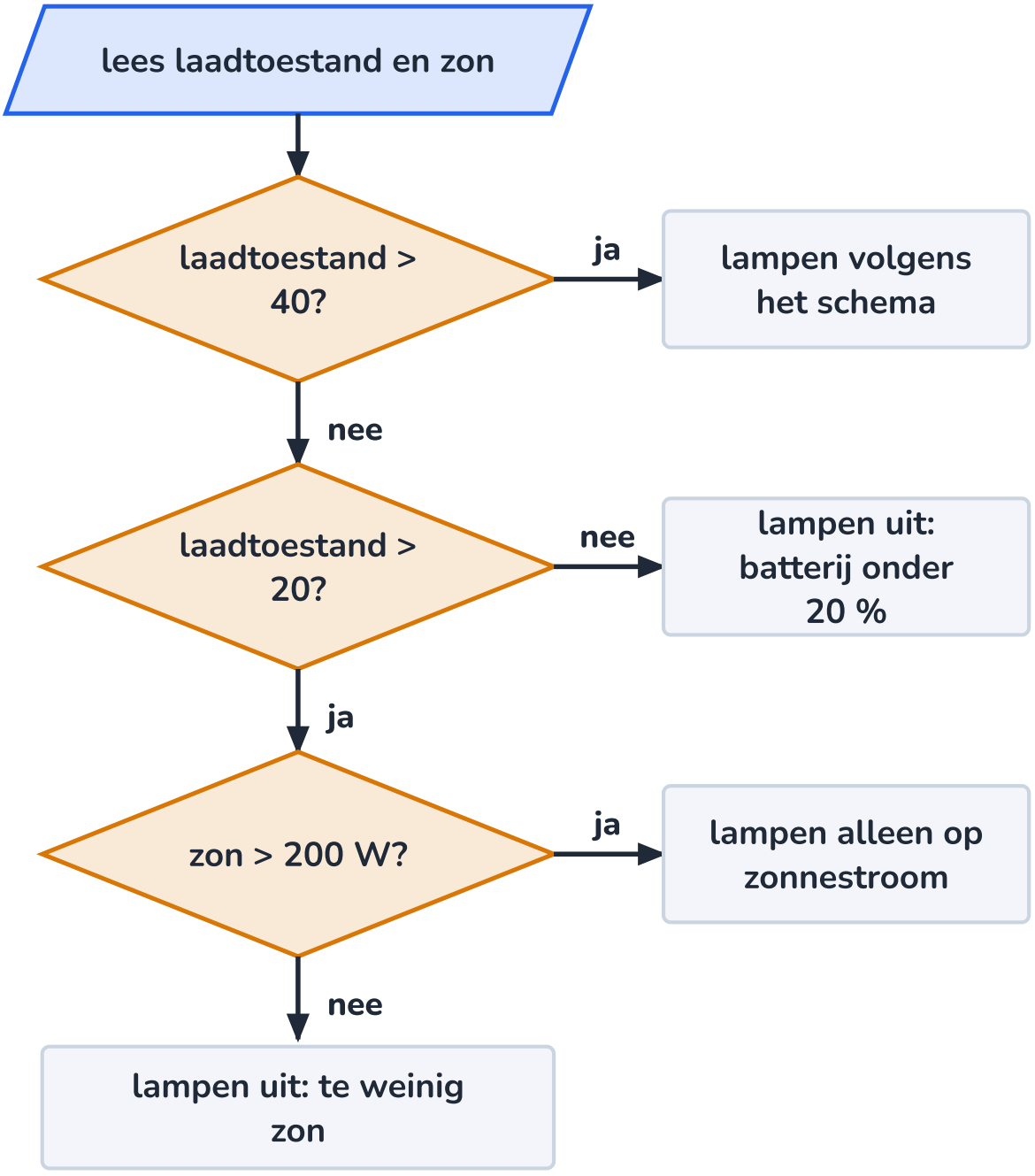
Een keuze begint met `if`, een voorwaarde en een dubbelepunt. Het ingesprongen blok eronder draait alleen als de voorwaarde waar is. Met `elif` ('else if') test je een tweede voorwaarde als de eerste niet waar was, met `else` vang je alles op wat overblijft. Python test van boven naar onder en voert **alleen het eerste blok** uit waarvan de voorwaarde waar is.

Vergelijken doe je met `==` (gelijk), `!=` (niet gelijk), `<`, `<=`, `>` en `>=`. Het programma is het energiebeheer van de klimaatmodule uit hoofdstuk 20: de groeilampen zijn de last die je afschakelt als de batterij leegloopt.

Python **energiebeheer.py** 13 regels

```python
# energiebeheer.py - de groeilampen als afschakelbare last (hoofdstuk 20)
laadtoestand = 30       # procent
zon = 350               # vermogen van de zonnepanelen op dit moment (W)

if laadtoestand > 40:
    print("lampen volgens het schema")
elif laadtoestand > 20:
    if zon > 200:
        print("lampen alleen op zonnestroom")
    else:
        print("lampen uit: te weinig zon")
else:
```

```
13      print("lampen uit: batterij onder 20 %")
```



Elke ruit wordt een if of elif; de pijl naar rechts is het ingesprongen blok.

Uitvoer voor vier toestanden (telkens de eerste twee regels aangepast):

laadtoestand	zon	uitvoer
55 %	350 W	`lampen volgens het schema`
30 %	350 W	`lampen alleen op zonnestroom`
30 %	120 W	`lampen uit: te weinig zon`
12 %	400 W	`lampen uit: batterij onder 20 %`

Er staat een `if` ín een `elif`: een **geneste** keuze, met acht spaties. Het werkt, maar het is omslachtig. Met `and` kan het korter (volgende blad).

De volgorde van elif

Zet de strengste voorwaarde bovenaan. Stond `laadtoestand > 20` vóór `laadtoestand > 40`, dan zou een volle batterij ook in de tweede tak belanden: bij 55 % is `55 > 20` al waar, en de rest wordt nooit meer getest.

UITLEG · 27.13

and, or en not: voorwaarden combineren

13/26

Met `and` is een samengestelde voorwaarde alleen waar als **beide** delen waar zijn; met `or` als **minstens één** deel waar is; `not` keert om. Het is dezelfde logica als schakelaars in serie en parallel (F18). Het energiebeheer wordt er korter mee:

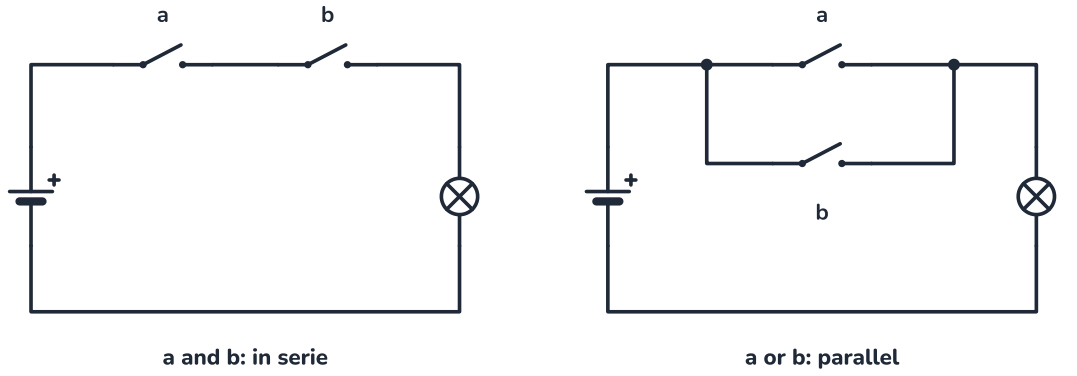
```
if laadtoestand > 40:
    print("lampen volgens het schema")
elif laadtoestand > 20 and zon > 200:
    print("lampen alleen op zonnestroom")
else:
    print("lampen uit")
```

a	b	a and b	a or b	not a
True	True	True	True	False
True	False	False	True	False
False	True	False	True	False
False	False	False	False	True

Python kent ook een **kettingvergelijking**, zoals in de wiskunde: `5.5 <= ph <= 6.5` betekent `5.5 <= ph and ph <= 6.5`. De volgorde van voorrang is: eerst vergelijken, dan `not`, dan `and`, dan `or`. Twijfel je, zet haakjes: `(a or b) and c` is iets anders dan `a or (b and c)`.

"x == 1 or 2" test of x 1 of 2 is

Python leest `x == 1 or 2` als `(x == 1) or 2`. Als x niet 1 is, wordt de uitkomst 2, en een getal dat geen nul is, telt als waar. De test is dus altijd waar, ook voor x = 5. Schrijf de vergelijking twee keer: `x == 1 or x == 2`.



a and b: in serie

a or b: parallel

Een gesloten schakelaar is True: in serie brandt de lamp alleen met a én b, parallel met a of b.

UITGEWERKT VOORBEELD · 27.14

pH en EC van de voedingsoplossing bewaken

De voedingsoplossing van de klimaatmodule moet een pH hebben tussen 5,5 en 6,5 (hoofdstuk 15) en een EC tussen 1,2 en 1,8 mS/cm (hoofdstuk 13). Schrijf een programma dat bij elke meting zegt wat er moet gebeuren. Kunnen twee problemen tegelijk voorkomen?

GEGEVEN

- pH: in orde van 5,5 tot 6,5; te hoog: pH-min doseren; te laag: dosering controleren
- EC: in orde van 1,2 tot 1,8 mS/cm; te laag: stamoplossing A en B bijdoseren; te hoog: aanlengen met regenwater

GEVRAAGD

- een programma dat per meting de juiste berichten toont

OPLOSSING

1. **Eerst het normale geval.** Alles in orde betekent: pH in het bereik *en* EC in het bereik. Met kettingvergelijkingen: `5.5 <= ph <= 6.5 and 1.2 <= ec <= 1.8`.

2. **Anders: twee aparte vragen.** pH en EC hangen niet van elkaar af: een oplossing kan tegelijk een te hoge pH én een te lage EC hebben. Daarom twee losse keuzes na elkaar (`if` … `if`), geen elif tussen pH en EC. Binnen elke meting sluiten 'te hoog' en 'te laag' elkaar wel uit: daar past `elif`.

3. **Het programma:**

```
# bewaking.py - pH en EC van de voedingsoplossing (hoofdstukken 13 en 15)
ph = 6.7
ec = 1.1                # mS/cm bij 25 graden

if 5.5 <= ph <= 6.5 and 1.2 <= ec <= 1.8:
    print("voedingsoplossing in orde")
else:
    if ph > 6.5:
        print(f"pH {ph}: te hoog, pH-min doseren")
    elif ph < 5.5:
        print(f"pH {ph}: te laag, dosering controleren")
    if ec < 1.2:
        print(f"EC {ec}: te laag, stamoplossing A en B bijdoseren")
    elif ec > 1.8:
        print(f"EC {ec}: te hoog, aanlengen met regenwater")
```

4. **Testen met vier metingen**, gekozen zodat elke tak minstens één keer draait:

PH	EC	UITVOER
6,7	1,1	`pH 6.7: te hoog, pH-min doseren` `EC 1.1: te laag, stamoplossing A en B bijdoseren`
5,9	1,4	`voedingsoplossing in orde`
6,2	1,9	`EC 1.9: te hoog, aanlengen met regenwater`
5,3	1,5	`pH 5.3: te laag, dosering controleren`

ANTWOORD

Het programma toont 'voedingsoplossing in orde' of een bericht per probleem. Bij pH 6,7 en EC 1,1 verschijnen twee berichten, omdat pH en EC met twee losse if-blokken getest worden.

Klopt dit? Test altijd de grenzen: bij pH 6,5 moet 'in orde' komen (6.5 <= 6.5 is waar), bij 6,51 niet. En test een geval per tak. Een tak die je nooit ziet draaien, kan een fout bevatten die pas maanden later opduikt.

for-lussen en range()

Een `for`-lus herhaalt een blok voor elke waarde uit een reeks. De reeks komt meestal van `range()`:

- `range(5)`: 0, 1, 2, 3, 4. Vijf waarden, beginnend bij 0: de stopwaarde doet niet mee.
- `range(2, 11, 3)`: start 2, stop 11, stap 3: 2, 5, 8.
- `range(10, 0, -3)`: met een negatieve stap tel je af: 10, 7, 4, 1.

De **lusvariabele** (hieronder `uur`) krijgt elke ronde de volgende waarde. Een for-lus kan ook over een vaste reeks waarden lopen, tussen haakjes; in hoofdstuk 28 wordt dat een lijst. En een lus mag in een lus staan: de binnenste loopt dan helemaal rond voor elke ronde van de buitenste.

```
for uur in range(6, 22, 3):
    print(uur, end=" ")
print()
for temp in (18, 22.5, 27):
    print(temp, "graden")
for dag in range(1, 3):
    for uur in range(0, 24, 8):
        print("dag", dag, "uur", uur)
```

```
6 9 12 15 18 21
18 graden
22.5 graden
27 graden
dag 1 uur 0
dag 1 uur 8
dag 1 uur 16
dag 2 uur 0
dag 2 uur 8
dag 2 uur 16
```

`end=" "` zet een spatie in plaats van een nieuwe regel achter de uitvoer; de lege `print()` daarna sluit de regel af.

Blokken en lussen

herhaal (4) wordt `for i in range(4):`. **herhaal tot** een voorwaarde wordt een `while` met de omgekeerde voorwaarde (volgende bladen). Het eindeloze **herhaal** wordt `while True:`, zoals de `loop()` van een Arduino.

Gevulde bolletjes doen mee; het open bolletje is de stopwaarde, die er net niet meer bij hoort.

De drie lussen van Scratch, met erboven hun vorm in Python.

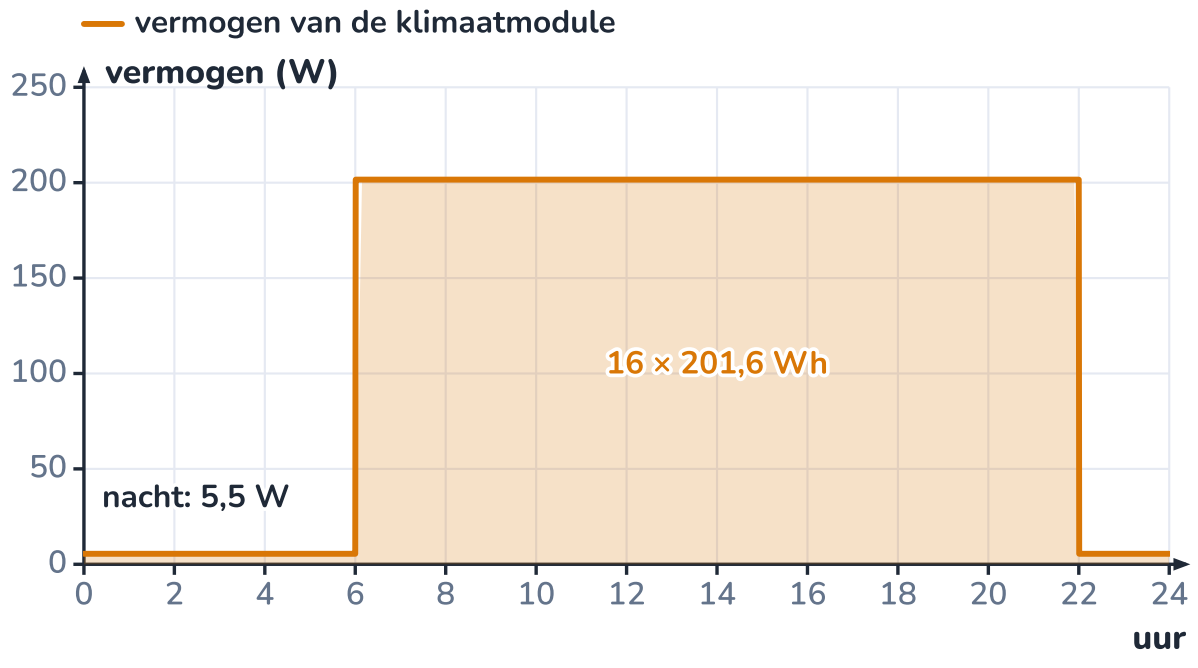
CODE · 27.16

Optellen in een lus: het verbruik van een dag

Een heel nuttig patroon: een **ophoper**. Je begint met `totaal = 0` vóór de lus, telt er in de lus telkens iets bij, en gebruikt het resultaat ná de lus. Hier telt de lus het verbruik van de klimaatmodule uur per uur op: 16 lichturen met gemiddeld 201,6 W en 8 nachturen met 5,5 W (hoofdstuk 20). Elk uur levert zijn vermogen maal één uur, dus zoveel wattuur.

Python **dagverbruik.py** 12 regels

```python
# dagverbruik.py - het verbruik van de klimaatmodule uur per uur optellen
p_licht = 201.6          # W, gemiddeld tijdens de 16 lichturen (lampen, pomp, sensoren, ventilatie)
p_nacht = 5.5            # W, 's nachts (sensoren, ventilator, batterijbeheer)
totaal = 0               # Wh: hier telt de lus alles bij op

for uur in range(24):
    if 6 <= uur < 22:    # lampen van 6 tot 22 uur (zomertijd)
        totaal = totaal + p_licht
    else:
        totaal = totaal + p_nacht

print(f"verbruik per dag: {totaal:.1f} Wh")
```



Elke ronde van de lus is één rechthoekje van één uur breed; de lus telt de oppervlakte op.

Uitvoer:

```
verbruik per dag: 3269.6 Wh
```

Hoofdstuk 20 kwam op 3269,5 Wh: het verschil van 0,1 Wh komt doordat 201,6 W zelf een afgerond gemiddelde is. De voorwaarde `6 <= uur < 22` is een kettingvergelijking: uur 6 telt mee, uur 22 niet. Zo zijn het precies 16 uren (6 tot en met 21).

De beginwaarde hoort vóór de lus

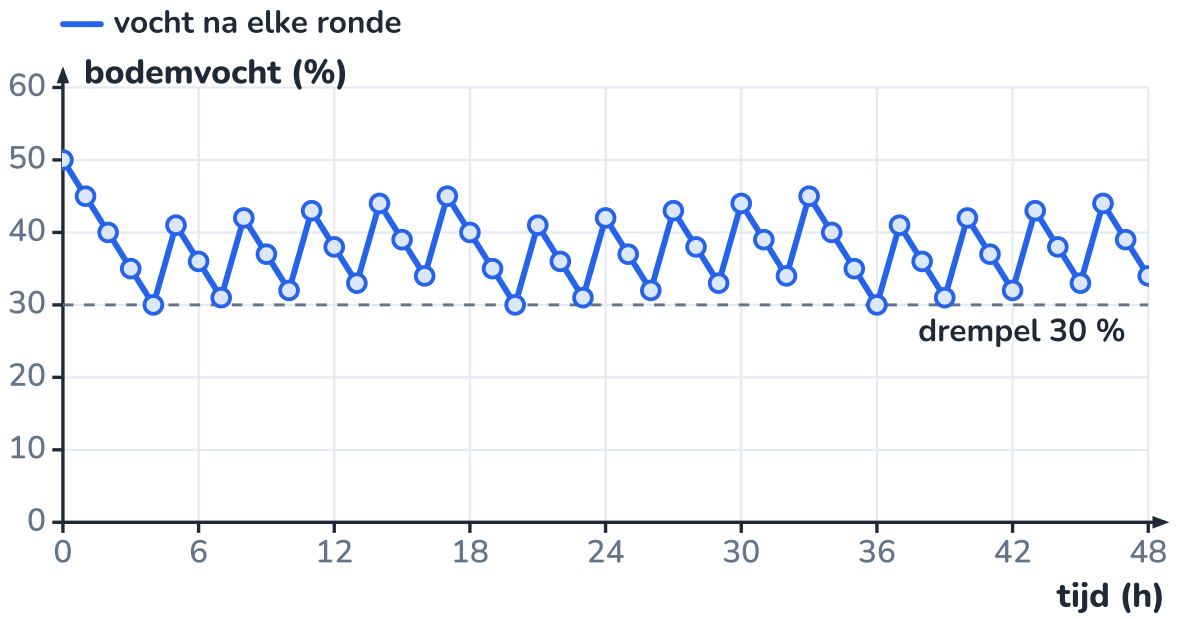
Staat `totaal = 0` ín de lus, dan begin je elke ronde opnieuw bij nul en houd je alleen het laatste uur over (5,5 Wh). Het is dezelfde fout als 'maak score 0' in de lus van een Scratch-script.

UITGEWERKT VOORBEELD · 27.17

Een lus naspelen met een trace-tabel

17/26

Speel de kas-simulatie van het begin van dit hoofdstuk na voor de eerste elf uren. Na hoeveel uur komt de eerste pompbeurt, en om de hoeveel uur volgen de andere?



De zaagtand van de simulatie: telkens 5 omlaag, en een sprong van 16 bij elke beurt.

GEGEVEN

- het programma kas_simulatie.py
- start: vocht 50, water 30, beurten 0

GEVRAAGD

- een trace-tabel met de waarden na elke ronde
- het ritme van de pompbeurten

OPLOSSING

1 **Kolommen kiezen.** Eén kolom per variabele die verandert (vocht, water, beurten), plus de lusvariabele en de uitkomst van de voorwaarde. Zo zie je niet alleen de waarden, maar ook *waarom* ze veranderen.

2 **Rij per ronde invullen**, precies in de volgorde van het programma: eerst −5, dan de test, dan eventueel de beurt.

UUR	VOCHT BIJ DE START	NA VOCHT − 5	VOCHT < 30?	VOCHT NA DE RONDE	WATER (L)	BEURTEN
0	50	45	nee	45	30,00	0
1	45	40	nee	40	30,00	0
2	40	35	nee	35	30,00	0
3	35	30	nee	30	30,00	0
4	30	25	**ja**	41	29,76	1
5	41	36	nee	36	29,76	1
6	36	31	nee	31	29,76	1
7	31	26	**ja**	42	29,52	2
8	42	37	nee	37	29,52	2
9	37	32	nee	32	29,52	2
10	32	27	**ja**	43	29,28	3
…						
47	39	34	nee	34	26,64	14

3 **De grens.** In de rij uur = 3 is het vocht precies 30. `30 < 30` is False: nog geen beurt. Pas in de rij uur = 4 zakt het naar 25 en volgt de eerste beurt.

4 **Het ritme.** Na een beurt staat het vocht op 41. Na drie rondes is het 26: opnieuw een beurt. Dat geeft 42, dan 27, dan 43: elke beurt komt het vocht 1 procentpunt hoger uit, tot het op een keer na vier rondes pas onder 30 zakt.

ANTWOORD

De eerste beurt komt bij uur = 4, de vijfde ronde; daarna ongeveer om de drie uur. Na 48 rondes zijn het 14 beurten en is er 26,64 L water over, zoals het programma zelf toonde.

Klopt dit? Het laatste getal in de tabel moet gelijk zijn aan de uitvoer van het programma. Is het niet zo, dan heb je ergens de volgorde van de regels verkeerd nagespeeld: dat is precies de fout die een trace-tabel zichtbaar maakt.

UITLEG · 27.18

while: herhalen zolang …

18/26

Een `while`-lus herhaalt haar blok **zolang** de voorwaarde waar is. De voorwaarde wordt vóór elke ronde getest; is ze al bij de start niet waar, dan draait het blok nul keer. Je gebruikt while als je vooraf niet weet hoeveel rondes nodig zijn: hoeveel dagen tot het reservoir op het alarmniveau staat, hoeveel pogingen iemand nodig heeft.

```
# reservoir.py - hoeveel warme dagen tot het alarmniveau?
inhoud = 60             # L, vol voedingsreservoir
verbruik = 6            # L per warme dag
dagen = 0
while inhoud > 24:      # alarm bij 10 cm water, 24 L
    inhoud = inhoud - verbruik
    dagen = dagen + 1
print("na", dagen, "dagen:", inhoud, "L")
```

```
na 6 dagen: 24 L
```

Het voedingsreservoir van 60 L verliest op een warme dag tot 6 L (hoofdstuk 13); bij 10 cm water, 24 L, gaat het alarm af (hoofdstuk 11). Het Scratch-blok 'herhaal tot inhoud ≤ 24' wordt dus

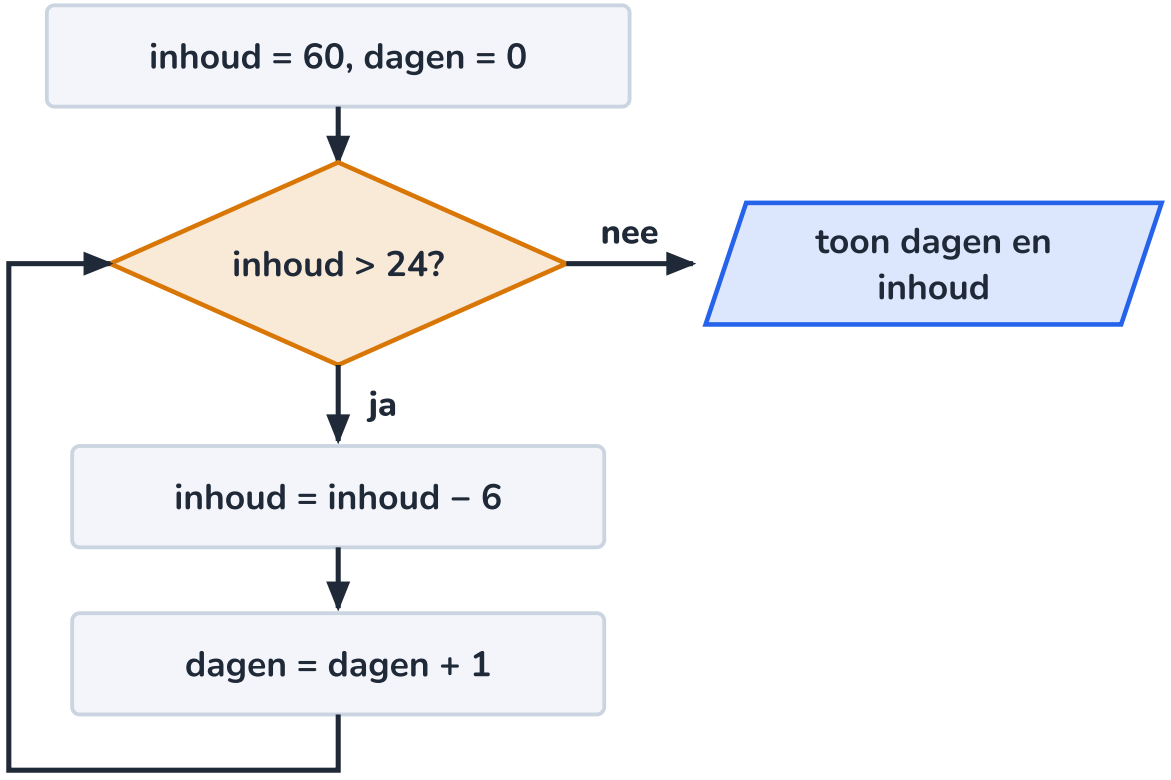
`while inhoud > 24`: de voorwaarde draait om, want Scratch zegt wanneer je stopt, Python zegt wanneer je doorgaat.

Een lus die nooit stopt

Verandert er in het blok niets aan de voorwaarde, dan stopt de lus nooit: een oneindige lus. Vergeet je `inhoud = inhoud - verbruik`, dan blijft de inhoud 60. Het programma lijkt te hangen; met **Ctrl+C** (of de stopknop van de editor) breek je het af.

while True en break

`while True:` loopt eindeloos, zoals de `loop()` van een Arduino. Met `break` spring je uit een lus, meestal in een if: `if inhoud <= 24: break`. Handig als de reden om te stoppen midden in het blok valt, maar gebruik het spaarzaam: een lus met drie break-punten is moeilijk te lezen.



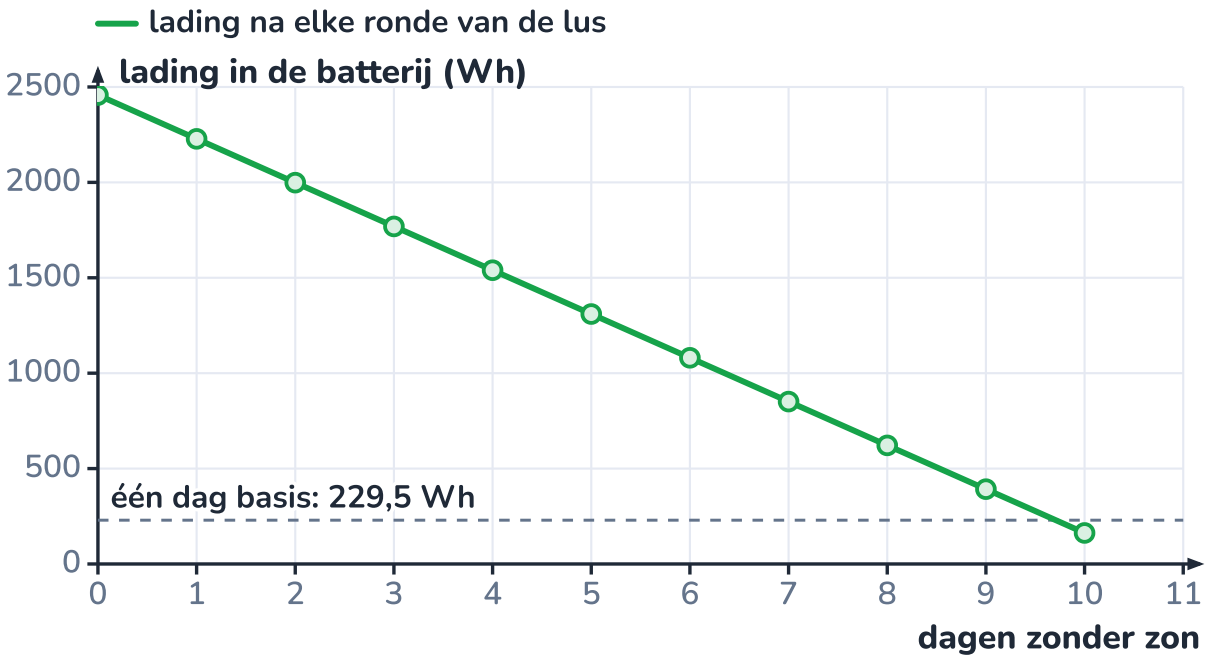
De ruit staat bovenaan: eerst testen, dan pas het blok. De pijl terug maakt er een lus van.

UITGEWERKT VOORBEELD · 27.19

Hoe lang houdt de batterij het vol zonder zon?

19/26

In een donkere winterweek levert het dak van de klimaatmodule bijna niets. De lampen gaan uit; de basis (pomp, sensoren, ventilatie, batterijbeheer) verbruikt 229,5 Wh per dag. De twee batterijen hebben samen 2457,6 Wh bruikbare energie (hoofdstuk 20). Hoeveel volle dagen houdt de module het vol?



Na tien rondes zit er minder dan één dag in de batterij: de voorwaarde wordt False.

GEGEVEN

- bruikbare energie 2457,6 Wh (2 × 12,8 V × 120 Ah × 80 %)
- basis 229,5 Wh per dag
- geen zonnestroom

GEVRAAGD

- het aantal volle dagen, met een while-lus
- de energie die daarna overblijft

OPLOSSING

1. **Lus of formule?** Met een formule: $2457{,}6 / 229{,}5 \approx 10{,}7$, dus 10 volle dagen. De lus doet het stap voor stap, en is makkelijk uit te breiden (een dag met wat zon, een dag met lampen).

2. **Voorwaarde.** Een dag kan alleen als er nog minstens één dagverbruik in de batterij zit: `while lading >= basis`.

3. **Het blok.** Eén dag aftrekken en één dag tellen:

```
# autonomie.py - hoeveel dagen zonder zon houdt de batterij het vol?
lading = 2457.6          # Wh bruikbaar: twee batterijen van 12,8 V en 120 Ah, 80 %
basis = 229.5            # Wh per dag zonder lampen
dagen = 0
while lading >= basis:
    lading = lading - basis
    dagen = dagen + 1
print(dagen, "volle dagen, rest", round(lading, 1), "Wh")
```

4. **Uitvoer:**

```
10 volle dagen, rest 162.6 Wh
```

ANTWOORD

10 volle dagen; daarna blijft er 162,6 Wh over, te weinig voor een elfde dag.

Klopt dit? De gehele deling geeft hetzelfde: `2457.6 // 229.5` is 10.0. Zonder `round()` toont Python voor de rest 162.5999999999999: tien keer een kommagetal aftrekken laat een klein foutje achter. Met de lampen aan (3269,5 Wh per dag) kom je zelfs niet door één dag.

ONTLEED · 27.20

Een foutmelding lezen

20/26

Een foutmelding is geen straf maar een aanwijzing. Python zegt waar het misliep, in welke regel, en welk soort fout het is. Lees ze **van onder naar boven**: de laatste regel is de belangrijkste. Het volledige pad naar je bestand is hier ingekort tot de bestandsnaam.

budget.py:

```
1  verbruik = 3269.5
2  print(verbruik)
```

uitvoer:

```
Traceback (most recent call last):                    (1)
  File "budget.py", line 2, in <module>               (2)
    print(verbruik)                                   (3)
          ^^^^^^^                                     (4)
NameError: name 'verbruik' is not defined. Did you mean: 'verbruik'?
(5)                                                   (6)
```

Een tikfout in een naam: Python stopt op regel 2 en stelt de juiste naam voor.

1 Traceback
Het programma is gestart en tijdens het draaien vastgelopen. Alles wat vóór deze regel gebeurde (hier niets), is wel uitgevoerd.

2 Waar
Bestand en regelnummer. `<module>` betekent: in het hoofdprogramma, niet in een functie (hoofdstuk 28).

3 De regel zelf
Python herhaalt de regel waar het misliep. Vergelijk ze teken voor teken met wat je bedoelde.

4 De plek
De dakjes ^ onderstrepen het deel dat de fout veroorzaakte, hier de naam `verbruik`.

5 Het soort fout
NameError
: Python kent deze naam niet. Een tikfout, of een variabele die nog geen waarde kreeg.

6 De tip
Sinds Python 3.10 stelt Python soms zelf een oplossing voor: *Did you mean: 'verbruik'?*

UITLEG · 27.21

Syntaxfouten: Python begrijpt de tekst niet

21/26

Bij een **syntaxfout** kan Python je programma niet eens lezen. Dan draait er **niets**, ook niet de regels vóór de fout, en er staat geen 'Traceback' boven de melding. Vier klassiekers, met de echte meldingen van Python 3.13:

De dubbelepunt ontbreekt achter de voorwaarde. Na `if`, `elif`, `else`, `for` en `while` komt altijd een dubbelepunt.

```
verbruik = 3269.5
if verbruik > 3000
    print("veel verbruik")
```

```
  File "budget.py", line 2
    if verbruik > 3000
                       ^
SyntaxError: expected ':'
```

Na de dubbelepunt verwacht Python een ingesprongen blok. Zet vier spaties voor `print`.

```
zon = 1750
if zon < 3269.5:
print("tekort")
```

```
  File "budget.py", line 3
    print("tekort")
    ^^^^^
IndentationError: expected an indented block after 'if' statement on line 2
```

Er zijn twee haakjes geopend en maar één gesloten. Het pijltje wijst het haakje aan dat nooit gesloten werd.

```
zon = 1400 * 1.25
print("zon:", round(zon, 1)
```

```
  File "budget.py", line 2
    print("zon:", round(zon, 1)
          ^
SyntaxError: '(' was never closed
```

De tekst (string) begint met een aanhalingsteken maar eindigt er niet mee.

```
print("verbruik per dag)
```

```
  File "budget.py", line 1
    print("verbruik per dag)
          ^
SyntaxError: unterminated string literal (detected at line 1)
```

De fout zit soms een regel hoger

Een vergeten haakje of aanhalingsteken merkt Python pas als het verder leest. Staat de fout op een regel die er goed uitziet, kijk dan naar de regel erboven. Een goede editor kleurt een tekst die niet sluit meteen verkeerd.

Tabs en spaties

Spring altijd in met vier spaties. De meeste editors zetten een druk op Tab om in vier spaties. Een regel die één spatie te veel inspringt, geeft `IndentationError: unexpected indent`.

Syntaxfouten stoppen alles vooraf; andere fouten stoppen op de regel waar ze gebeuren; een logische fout stopt niets.

Fouten tijdens het draaien

Is de schrijfwijze in orde, dan begint Python uit te voeren. Loopt het vast, dan verschijnt een **Traceback** en stopt het programma op die regel. Wat ervoor stond, is uitgevoerd (en getoond). De meest voorkomende:

TypeError: een bewerking die niet past bij het type. Tekst en een getal aan elkaar plakken:

```
lichturen = 16
print("lichturen: " + lichturen)
```

```
Traceback (most recent call last):
  File "budget.py", line 2, in <module>
    print("lichturen: " + lichturen)
          ~~~~~~~~~~~~~~^~~~~~~~~~~
TypeError: can only concatenate str (not "int") to str
```

Oplossing: `print("lichturen:", lichturen)` of een f-string. Na `input()` zie je vaak een tweede soort TypeError, omdat de invoer nog tekst is:

```
antwoord = "25"
if antwoord > 20:
    print("warm")
```

```
Traceback (most recent call last):
  File "budget.py", line 2, in <module>
    if antwoord > 20:
       ^^^^^^^^^^^^
TypeError: '>' not supported between instances of 'str' and 'int'
```

ValueError: het type klopt, de waarde niet. Tekst met een komma omzetten:

```
tekst = "3,5"
liter = float(tekst)
```

```
Traceback (most recent call last):
  File "budget.py", line 2, in <module>
    liter = float(tekst)
ValueError: could not convert string to float: '3,5'
```

ZeroDivisionError: delen door nul, bijvoorbeeld een gemiddelde over nul dagen:

```
dagen = 0
gemiddeld = 25 / dagen
```

```
Traceback (most recent call last):
  File "budget.py", line 2, in <module>
    gemiddeld = 25 / dagen
                ~~~^~~~~~~
ZeroDivisionError: division by zero
```

Deel je een kommagetal door nul, dan zegt Python `float division by zero`: dezelfde fout. Voorkom ze met een test vooraf: `if dagen > 0:`.

fout	wat Python zegt	wat er misliep
NameError	deze naam ken ik niet	tikfout, of de variabele kreeg nog geen waarde
TypeError	dit kan niet met dit type	tekst en getal gemengd, of een functie met te weinig argumenten
ValueError	het type klopt, de waarde niet	tekst die geen getal is: 'maart', '3,5', ''
ZeroDivisionError	delen door nul	een noemer die (soms) nul is

UITGEWERKT VOORBEELD · 27.23

Debuggen: een fout zonder foutmelding

23/26

Een leerling berekent de gemiddelde temperatuur uit vier metingen in de klimaatmodule. Het programma draait zonder melding, maar geeft een gemiddelde van 72,25 °C. Vind en verbeter de fout.

```
# gemiddelde.py - gemiddelde temperatuur uit vier metingen
t1 = 18.5
t2 = 22.0
t3 = 26.5
t4 = 21.0
gemiddelde = t1 + t2 + t3 + t4 / 4
print("gemiddelde:", gemiddelde)
```

```
gemiddelde: 72.25
```

GEGEVEN

- vier metingen: 18,5; 22,0; 26,5 en 21,0 °C
- de uitvoer: 72.25

GEVRAAGD

- de oorzaak
- het verbeterde programma

OPLOSSING

1. **Schat eerst.** Een gemiddelde ligt altijd tussen de kleinste en de grootste meting: tussen 18,5 en 26,5 °C. De uitkomst 72,25 is dus fout, ook al klaagt Python niet. Dat is een **logische fout**.

2. **Maak tussenstappen zichtbaar.** Zet tijdelijk print()-regels in het programma, vlak voor de verdachte regel:

```
print("som:", t1 + t2 + t3 + t4)
print("t4 / 4:", t4 / 4)
```

```
som: 88.0
t4 / 4: 5.25
gemiddelde: 72.25
```

3. **Vergelijk met je verwachting.** De som (88,0) klopt. Maar 72,25 is precies $18{,}5 + 22{,}0 + 26{,}5 + 5{,}25$: alleen t4 werd door 4 gedeeld. Delen gaat vóór optellen, net zoals in de wiskunde.

4. **Verbeter en test opnieuw.** Haakjes rond de som, en de hulpregels weer weg:

```
gemiddelde = (t1 + t2 + t3 + t4) / 4
```

```
gemiddelde: 22.0
```

ANTWOORD

De haakjes ontbraken: alleen t4 werd gedeeld. Met (t1 + t2 + t3 + t4) / 4 is het gemiddelde 22,0 °C.

Klopt dit? 22,0 ligt tussen 18,5 en 26,5, en ook de schatting 'rond 22' klopt. Andere manieren om zo'n fout te vinden: het programma hardop uitleggen aan iemand (of aan een badeendje: het werkt echt), en het testen met getallen waarvan je de uitkomst kent, zoals vier keer 20.

CODE · 27.24

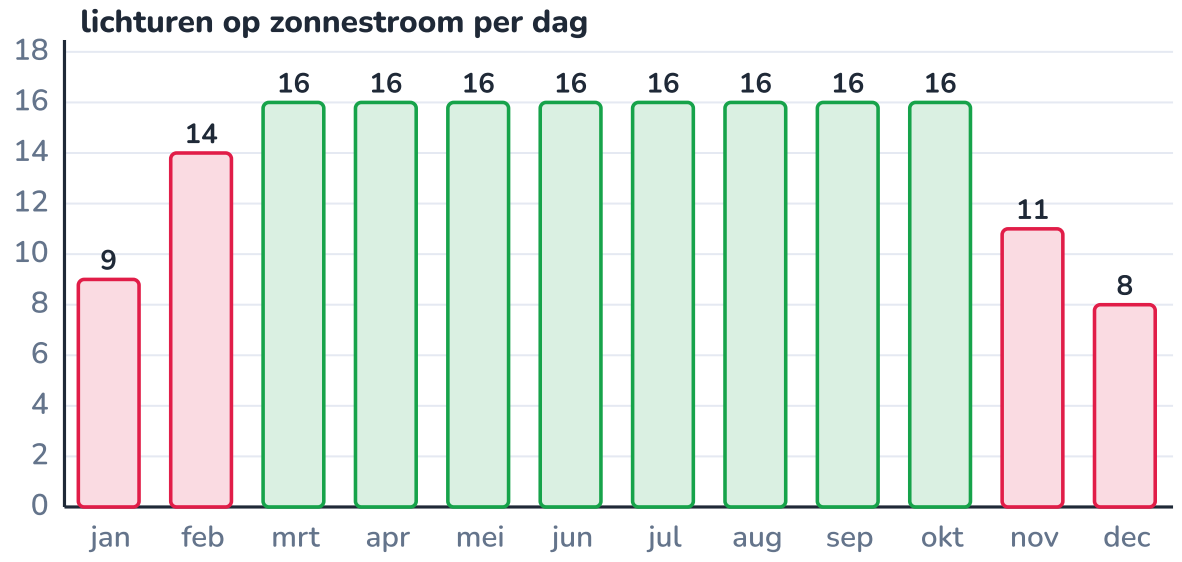
Habitat: een rekenhulp voor het energiebudget

24/26

In hoofdstuk 20 stelde je het energiebudget van de klimaatmodule op en koos je panelen en batterijen. Met een rekenhulp in Python beantwoord je snel de vragen die daarna komen: klopt het budget in een bepaalde maand? Hoeveel lichturen kan de zon alleen betalen? Hoe lang houdt de batterij de basis draaiende? Alles wat je in dit hoofdstuk leerde, zit erin: variabelen, rekenen, f-strings, if/else en twee while-lussen.

Python **energiebudget.py** 39 regels

```python
# energiebudget.py - rekenhulp voor de klimaatmodule (gegevens uit hoofdstuk 20)

# 1. verbruik per dag in Wh: vermogen (W) maal uren per dag
lampen = 190 * 16                 # twee groeilampen van 95 W, 16 lichturen
pomp = 12 * 4                     # 10 minuten per uur is 4 uur per dag
sensoren = 3.0 * 24               # microcontroller, sensoren en datalogger
ventilatie = 5 * 16 + 2 * 8       # 5 W overdag, 2 W 's nachts
co2_klep = 6 * 0.25
bms = 0.5 * 24                    # laadregelaar en batterijbeheer
basis = pomp + sensoren + ventilatie + co2_klep + bms
verbruik = basis + lampen

# 2. zonnestroom in de gekozen maand
maand = "december"                # in het echt: maand = input("Maand? ")
wh_per_wp = 1.25                  # PVGIS: Wh per Wp per dag in die maand
piek = 4 * 350                    # vier panelen van 350 Wp
zon = piek * wh_per_wp
saldo = zon - verbruik

print(f"verbruik: {verbruik:.1f} Wh per dag, waarvan {lampen} Wh voor de lampen")
print(f"zonnestroom in {maand}: {zon:.0f} Wh per dag")
if saldo >= 0:
    print(f"overschot: {saldo:.0f} Wh per dag")
else:
    print(f"tekort: {-saldo:.0f} Wh per dag")

# 3. hoeveel lichturen kan de zon alleen betalen?
lichturen = 16
while lichturen > 0 and basis + 190 * lichturen > zon:
    lichturen = lichturen - 1
print("lichturen op zonnestroom:", lichturen, "van de 16")

# 4. hoeveel dagen houdt de batterij de basis draaiende zonder zon?
lading = 2 * 0.80 * 120 * 12.8    # twee LiFePO4-batterijen, 80 % bruikbaar (Wh)
dagen = 0
while lading >= basis:
    lading = lading - basis
    dagen = dagen + 1
print(f"batterij: {dagen} dagen zonder zon (en zonder lampen)")
```



Het programma twaalf keer gedraaid, één keer per maand: van november tot februari betaalt de zon geen 16 lichturen.

Uitvoer voor december:

```
verbruik: 3269.5 Wh per dag, waarvan 3040 Wh voor de lampen
zonnestroom in december: 1750 Wh per dag
tekort: 1520 Wh per dag
lichturen op zonnestroom: 8 van de 16
batterij: 10 dagen zonder zon (en zonder lampen)
```

En voor maart (enkel `maand` en `wh_per_wp` aangepast, 2,99 Wh per Wp):

```
verbruik: 3269.5 Wh per dag, waarvan 3040 Wh voor de lampen
zonnestroom in maart: 4186 Wh per dag
overschot: 916 Wh per dag
lichturen op zonnestroom: 16 van de 16
batterij: 10 dagen zonder zon (en zonder lampen)
```

De lichturen-lus begint bij 16 en trekt er één af zolang basis plus lampen meer vragen dan de zon levert. In december blijven er 8 over. Hoofdstuk 20 kwam op ongeveer 7,5 uur, omdat daar ook het verlies in de batterij meetelde: de lampen branden deels 's avonds, op stroom die eerst in de batterij moest. De rekenhulp is dus een snelle eerste schatting, geen vervanging van het uurmodel.

Waarom geen lijst met twaalf maanden?

Om een andere maand te kiezen, pas je nu twee regels aan. Met twaalf elif-takken zou het programma de waarde zelf kunnen kiezen, maar dat wordt lang en foutgevoelig. In hoofdstuk 28 zet je de twaalf PVGIS-waarden in één **lijst** en rekent een **functie** elke maand in één lus uit.

NASLAG · 27.25

Python in één oogopslag

25/26

De bouwstenen uit dit hoofdstuk, met hun blokken uit Fundamental en hun vorm in C++, zoals je ze in hoofdstuk 29 tegenkomt.

Scratch	Python	C++ (Arduino, hoofdstuk 29)
maak [x] (5)	`x = 5`	`int x = 5;`

Scratch	Python	C++ (Arduino, hoofdstuk 29)
verander [x] met (1)	`x = x + 1` of `x += 1`	`x = x + 1;` of `x++;`
herhaal (10)	`for i in range(10):`	`for (int i = 0; i < 10; i++) { ... }`
herhaal tot <v>	`while not v:`	`while (!v) { ... }`
herhaal (eindeloos)	`while True:`	`void loop() { ... }`
als <v> dan ... anders	`if v:` ... `else:`	`if (v) { ... } else { ... }`
zeg (x)	`print(x)`	`Serial.println(x);`
vraag [...] en wacht	`input("...")`	(tekst lezen via Serial)
<x = 5>	`x == 5`	`x == 5`
en, of, niet	`and`, `or`, `not`	`&&`, `||`, `!`

teken	bewerking	voorbeeld	uitkomst	opmerking
`+`	optellen	`17 + 5`	`22`	
`-`	aftrekken	`17 - 5`	`12`	
`*`	vermenigvuldigen	`17 * 5`	`85`	geen × of ·
`/`	delen	`17 / 5`	`3.4`	altijd een float, ook bij `10 / 5`
`//`	gehele deling	`17 // 5`	`3`	afgerond naar beneden
`%`	rest (modulo)	`17 % 5`	`2`	wat overblijft na de gehele deling
`**`	macht	`2 ** 10`	`1024`	niet ^ (dat is iets anders in Python)

waarde	f-string	resultaat	betekenis
`x = 3269.5`	`f"{x:.2f}"`	`'3269.50'`	2 decimalen
`x = 3269.5`	`f"{x:.0f}"`	`'3270'`	afgerond op een geheel getal
`x = 3269.5`	`f"{x:10.1f}"`	`'    3269.5'`	10 tekens breed, 1 decimaal (rechts uitgelijnd)
`x = 916.5`	`f"{x:+.1f}"`	`'+916.5'`	altijd een teken, ook +
`x = 0.535`	`f"{x:.0%}"`	`'54%'`	als procent (maal 100)
`s = "pomp"`	`f"{s:<10}|"`	`'pomp     |'`	10 tekens breed, links uitgelijnd
`n = 48`	`f"{n:>5}"`	`'   48'`	5 tekens breed, rechts uitgelijnd
`n = 48`	`f"{n:05d}"`	`'00048'`	aanvullen met nullen

melding	typische oorzaak	wat je doet
`SyntaxError: expected ':'`	dubbelepunt vergeten na if, elif, else, for, while	dubbelepunt toevoegen
`SyntaxError: '(' was never closed`	haakje niet gesloten	haakjes tellen; de fout zit vaak op de regel ervoor
`SyntaxError: unterminated string literal`	aanhalingsteken vergeten	tekst afsluiten met hetzelfde teken
`IndentationError`	blok niet (of te veel) ingesprongen	vier spaties per niveau, geen tabs mengen
`NameError: name '...' is not defined`	tikfout in een naam, of variabele nog niet gemaakt	naam nakijken (hoofdletters tellen)
`TypeError: can only concatenate str ...`	tekst + getal	f-string gebruiken, of `str()`

melding	typische oorzaak	wat je doet
`TypeError: '>' not supported ...`	tekst vergelijken met een getal (vaak na input())	eerst `int()` of `float()`
`ValueError: could not convert ...`	tekst die geen getal is, of een decimale komma	`.replace(",", ".")` of invoer controleren
`ZeroDivisionError`	delen door nul	eerst testen of de noemer nul is

HET GEHEEL · 27.26

Waar Python vandaan komt en waar het heen gaat

26/26

Dit hoofdstuk is de brug tussen de blokken van Fundamental en alles wat daarna met code gebeurt. De denkstappen zijn dezelfde gebleven (decompositie, een algoritme, naspelen, debuggen), alleen de schrijfwijze is nieuw. Programmeren raakt bijna elke leerlijn: je rekent energiebudgetten uit (E), simuleert een proces (W), verwerkt metingen (O, A) en stuurt een regeling aan (R).

F22 Computationeel denken
Algoritme, stroomdiagram en trace-tabel: dezelfde denkstappen, nu in tekst.

F23 Programmeren met blokken
De blokken die je hier naar Python vertaalde, en de kas-simulatie.

F25 Sensoren en automatisch regelen
Een drempel en een aan/uit-regeling: if/else op een microcontroller.

I20 Zonne-energie dimensioneren
Het energiebudget waarvoor je de rekenhulp schreef.

I25 Van analoog naar digitaal
Het tweetallig stelsel verklaart waarom 0,1 + 0,2 niet precies 0,3 is.

I28 Functies, lijsten en data in Python
Functies, lijsten en woordenboeken, CSV-bestanden en grafieken: de volgende stap.

I29 Microcontrollers programmeren in C++
Dezelfde structuren in C++ op de Arduino: accolades in plaats van inspringen.

I30 Datalogging en communicatie
Python leest de metingen die de microcontroller doorstuurt.

I39 Data visualiseren en eerste stappen in AI
Een beslisboom is een reeks if-else-vragen die de computer zelf leert.

E34 Algoritmen en complexiteit
Hoe snel een lus is, en waarom sommige algoritmen te traag worden.

E36 Numerieke methoden en wetenschappelijk rekenen
Lussen die een differentiaalvergelijking stap voor stap uitrekenen.

Habitat
De klimaatmodule heeft nu een rekenhulp voor haar energiebudget: ze toont per maand het saldo, het aantal lichturen dat de zon alleen kan betalen (december 8, januari 9, november 11, februari 14, de rest van het jaar 16) en de autonomie van de batterij voor de basis (10 dagen). In opdracht 27.14 breid je ze uit; in hoofdstuk 28 wordt ze korter met lijsten en functies.

SAMENVATTING

In het kort

- Een Python-programma is tekst; de interpreter voert het regel per regel uit. In de shell (>>>) probeer je één regel uit.
- Een C-blok wordt een regel met een dubbelepunt en een ingesprongen blok (vier spaties); alles na # is commentaar.
- `naam = waarde` hangt een naam aan een waarde; `==` vergelijkt. Types: int, float, str, bool; omzetten met int(), float(), str().
- / geeft altijd een float, // de gehele deling, % de rest, ** de macht; haakjes en de gewone voorrangsregels bepalen de volgorde.

- print() en f-strings tonen resultaten: `{x:.1f}` rondt af; een decimale komma maak je met `.replace(".", ",")`.
- input() geeft altijd tekst: eerst omzetten (en een komma vervangen), dan rekenen.
- if, elif, else: alleen het eerste ware blok draait. Combineer voorwaarden met and, or en not.
- for met range() herhaalt een vast aantal keer; while herhaalt zolang een voorwaarde waar is en kan oneindig lopen.
- Lees een foutmelding van onder naar boven: soort fout, dan de regel. Syntaxfouten stoppen alles vooraf; een logische fout vind je door te schatten, tussenwaarden te tonen en na te spelen.

BEGRIPPEN

Wat je nu kent

Python
Een programmeertaal die je leest bijna als Engelse tekst; een interpreter voert het programma regel per regel uit.

interpreter
Het programma dat Python-code leest en meteen uitvoert (tegenover een compiler, die eerst alles vertaalt).

interactieve shell
Het venster met de prompt >>> waarin je één regel Python typt en meteen de uitkomst ziet.

toekenning
Een regel als `vocht = 50`: de naam links wijst voortaan naar de waarde rechts.

type
De soort van een waarde: int (geheel getal), float (kommagetal), str (tekst) of bool (True of False).

f-string
Tekst met f ervoor waarin waarden tussen accolades komen, met een opmaak zoals `{x:.1f}`.

voorwaarde
Een uitdrukking die True of False oplevert, zoals `vocht < 30`; ze stuurt if en while.

inspringen
Spaties aan het begin van een regel (vier per niveau): ze bepalen welke regels bij een if, for of while horen.

for-lus
Een lus die een vast aantal keer loopt, of één keer voor elke waarde uit een reeks, zoals range(48).

while-lus
Een lus die herhaalt zolang een voorwaarde waar is; de voorwaarde wordt vóór elke ronde getest.

trace-tabel
Een tabel waarin je de waarden van de variabelen na elke stap noteert, om een programma na te spelen.

traceback
De foutmelding van Python: waar het misliep (bestand, regel) en welk soort fout, onderaan het belangrijkst.

syntaxfout
Een fout in de schrijfwijze: Python begrijpt de tekst niet en voert niets uit.

logische fout
Het programma draait zonder melding, maar rekent iets anders uit dan je bedoelde.

I INTERMEDIATE · DEEL I7 · PROGRAMMEREN

HANDBOEK · HOOFDSTUK 28

Functies, lijsten en data in Python

Eigen functies, lijsten en woordenboeken, en een week meetdata van de klimaatmodule tot een besluit

Een datalogger schrijft elke paar minuten een regel met metingen. Na een week zijn dat er honderden: te veel om met de hand te bekijken, maar voor Python een paar regels code. In dit hoofdstuk leer je je eigen functies schrijven, veel waarden bewaren in lijsten en woordenboeken, een CSV-bestand inlezen en er gemiddelden, minima, maxima en grafieken uit halen. Aan het eind analyseer je een week in de klimaatmodule.

9× uitleg · 3× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 7× code · ± 9 lestijden

P Programmeren en computationeel denken · A Data en AI · O Onderzoeken en meten

Een week meten levert honderden getallen op; een programma maakt er een besluit van.

Na dit hoofdstuk

- Je kan een functie schrijven met parameters, een standaardwaarde, een returnwaarde en een docstring, en ze aanroepen.
- Je kan het verschil tussen print en return uitleggen en een functie testen met gekozen invoer.
- Je kan lijsten maken, indexeren (ook met een negatieve index), in stukken knippen en aanvullen met append.
- Je kan met len, sum, min, max en sorted rekenen op een lijst, en een gemiddelde en een maximum ook zelf programmeren.
- Je kan over een lijst lopen om op te hopen, te tellen, te filteren en een nieuwe lijst te maken, ook met zip.
- Je kan een woordenboek gebruiken om waarden op te zoeken en om metingen per groep te verzamelen.
- Je kan een CSV-bestand inlezen met de csv-module, de teksten omzetten en ontbrekende waarden opvangen.
- Je kan een week meetdata per dag samenvatten en er een onderbouwd besluit uit trekken.
- Je kan met matplotlib een lijngrafiek en een staafdiagram maken met asopschriften, eenheden en een legende.

Vijf landen te weinig

In 2010 publiceerden twee bekende Amerikaanse economen een studie over staatsschulden. Hun besluit: landen met een schuld boven 90 % van hun jaarlijkse productie groeiden gemiddeld niet meer, zelfs licht achteruit. Politici overal ter wereld haalden het cijfer aan.

In 2013 kreeg Thomas Herndon, een student aan de universiteit van Massachusetts, als oefening de opdracht om een bekende studie opnieuw uit te rekenen. Hij koos deze, kwam niet op dezelfde getallen uit, en vroeg het rekenblad van de auteurs op. Daarin vond hij onder meer een formule voor een gemiddelde die over een te kort bereik liep: vijf landen telden niet mee, Australië, Oostenrijk, **België**, Canada en Denemarken. Samen met andere keuzes in de analyse (welke jaren meetelden, hoe landen gewogen werden) werd de gemiddelde groei bij een hoge schuld ongeveer +2,2 % per jaar in plaats van −0,1 %. De auteurs erkenden de rekenbladfout, maar verdedigden hun besluit.

Een rekenblad is niet slecht, maar een formule in cel F67 zie je niet staan. In een programma staat elke stap zichtbaar als tekst, test je met `len()` hoeveel waarden er echt meetellen, en kan iemand anders dezelfde berekening op dezelfde data opnieuw draaien. Dat heet **reproduceerbaar** werken, en het is de kern van goed onderzoek (hoofdstuk 2).

> **Waarom dit hoofdstuk ertoe doet**
>
> De klimaatmodule meet temperatuur, vochtigheid, verbruik en zonnestroom. Die metingen zijn alleen iets waard als je ze betrouwbaar samenvat: per dag, per dagdeel, met gemiddelden en grafieken. Dat doe je hier met functies, lijsten en woordenboeken, en straks met de echte datalogger van hoofdstuk 30.

Een berekening opnieuw uitvoeren is de beste controle: zo werd de fout in het rekenblad gevonden.

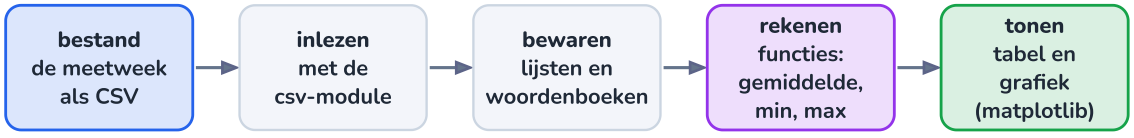
UITLEG · 28.2

2/24

Wat je al kan, en waar dit hoofdstuk naartoe gaat

wat je al kan	wat er nu bijkomt
variabelen, if, for en while; ingebouwde functies als print(), round() en len() (hoofdstuk 27)	zelf functies schrijven, met parameters en een returnwaarde
een rij waarden tussen haakjes in een for-lus (hoofdstuk 27)	lijsten die je aanvult, doorzoekt, sorteert en in stukken knipt; woordenboeken
gemiddelde, bereik en uitschieters (hoofdstuk 1); een rekenblad en een trendlijn (hoofdstuk 3)	dezelfde statistiek in code, voor honderden metingen in één keer
een grafiek met assen, grootheid, eenheid en schaal (F4)	grafieken laten tekenen met matplotlib

In hoofdstuk 27 schreef je programma's van boven naar onder, met een variabele voor elke waarde. Voor een week meetdata werkt dat niet meer: 56 rijen met elk vijf waarden zijn 280 getallen. Drie nieuwe bouwstenen lossen dat op. Een **functie** geeft een berekening een naam, zodat je ze telkens opnieuw kan gebruiken. Een **lijst** bewaart veel waarden onder één naam. Een **woordenboek** koppelt een naam (een sleutel) aan een waarde. Samen met de **csv-module** om bestanden te lezen en **matplotlib** om te tekenen, heb je alles om van meetdata een besluit te maken.



elke stap is een paar regels code, die je voor de volgende week gewoon opnieuw draait

Vijf stappen van bestand tot besluit; dit hoofdstuk doet ze alle vijf.

UITLEG · 28.3

Functies: een berekening met een naam

3/24

Je gebruikte al veel functies: `print()`, `round()`, `len()`. Een eigen functie maak je met `def`:

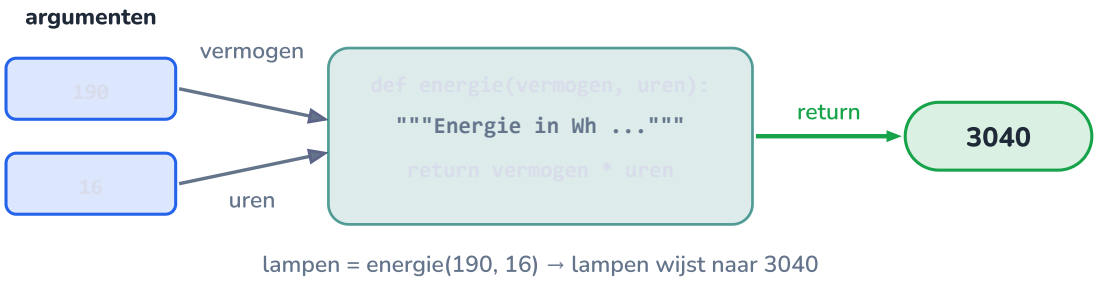
```
def energie(vermogen, uren):
    """Energie in Wh van een toestel met dit vermogen (W) dat zoveel uren aanstaat."""
    return vermogen * uren
```

De eerste regel is de **kop**: het woord `def`, de naam, de **parameters** tussen haakjes en een dubbelepunt. Daaronder het ingesprongen **lichaam**. De tekst tussen drie aanhalingstekens is de **docstring**: wat de functie doet, met eenheden. Met `return` geeft de functie een waarde terug en stopt ze.

Een `def` voert nog niets uit; het maakt de functie alleen klaar. Pas bij een **aanroep** zoals `energie(190, 16)` draait het lichaam: de **argumenten** 190 en 16 worden de waarden van de parameters `vermogen` en `uren`, en de aanroep zelf krijgt de waarde 3040.

Wanneer maak je een functie?

Als je dezelfde berekening twee keer typt, of als een stuk code een duidelijke eigen taak heeft (een ADC-waarde omrekenen, een gemiddelde nemen). Een goede functie doet één ding, heeft een naam die zegt wat (een werkwoord of wat ze teruggeeft), en een docstring. Test ze met invoer waarvan je de uitkomst kent.



lampen = energie(190, 16) → lampen wijst naar 3040

De argumenten gaan erin, return geeft het resultaat terug aan wie de functie aanriep.

CODE · 28.4

Je eerste eigen functies

4/24

Twee functies voor het energiebudget: `energie()` rekent, `komma()` maakt van een getal een tekst met een decimale komma (de truc uit hoofdstuk 27, nu met een naam). `{x:.{decimalen}f}` is een f-string in een f-string: het aantal decimalen komt zelf uit een variabele.

Python **functies.py** 17 regels

```python
# functies.py - eigen functies met parameters en een returnwaarde
def energie(vermogen, uren):
    """Energie in Wh van een toestel met dit vermogen (W) dat zoveel uren aanstaat."""
    return vermogen * uren


def komma(x, decimalen=1):
    """Getal als tekst met een decimale komma: komma(3.14159, 2) geeft '3,14'."""
    return f"{x:.{decimalen}f}".replace(".", ",")


lampen = energie(190, 16)
pomp = energie(12, 4)
print(lampen, pomp)
print("samen:", komma((lampen + pomp) / 1000, 2), "kWh")
print("pomp per uur:", komma(energie(12, 10 / 60)), "Wh")
print(energie.__doc__)
```

Uitvoer:

```
3040 48
samen: 3,09 kWh
pomp per uur: 2,0 Wh
Energie in Wh van een toestel met dit vermogen (W) dat zoveel uren aanstaat.
```

Een functie mag een andere functie aanroepen, en een aanroep mag in een berekening staan: `komma(energie(12, 10 / 60))` rekent eerst de energie van 10 minuten pompen uit en maakt er dan een tekst van. De docstring is op te vragen met `energie.__doc__`, of netter met `help(energie)` in de shell.

UITLEG · 28.5

return is geen print

5/24

Het verschil tussen `print` en `return` is de klassieke valkuil bij functies. `print` *toont* iets op het scherm; `return` *geeft* een waarde terug aan de code die de functie aanriep. Een functie zonder return

geeft de speciale waarde `None` ('niets') terug.

```
def gemiddelde_print(lijst):
    print(sum(lijst) / len(lijst))


def gemiddelde(lijst):
    return sum(lijst) / len(lijst)


a = gemiddelde_print([20, 22, 24])
b = gemiddelde([20, 22, 24])
print("a:", a, "- b:", b)
```

```
22.0
a: None - b: 22.0
```

De eerste functie toont 22.0, maar `a` is None: met a kan je niet verder rekenen. Rekenfuncties geven dus terug; alleen het hoofdprogramma toont.

Standaardwaarden. Een parameter mag een standaardwaarde krijgen, zoals `decimalen=1`. Laat je dat argument weg, dan geldt de standaardwaarde. Argumenten mag je ook met hun naam meegeven, in elke volgorde:

```
def komma(x, decimalen=1):
    return f"{x:.{decimalen}f}".replace(".", ",")


print(komma(3269.5))
print(komma(3269.5, 3))
print(komma(decimalen=0, x=3269.5))
```

```
3269,5
3269,500
3270
```

Lokale variabelen. Een variabele die je in een functie maakt, bestaat alleen tijdens die aanroep. Zo kunnen twee functies elk hun eigen `totaal` hebben zonder elkaar te storen. Wat een functie nodig heeft, geef je mee als argument; wat ze oplevert, geeft ze terug.

> **"Mijn functie werkt, ik zie het resultaat toch?"**
>
> Dat je een getal op het scherm ziet, betekent niet dat de functie het teruggeeft. Probeer altijd `x = mijn_functie(...)` en `print(x)`. Staat er None, dan ontbreekt return.

Links komt het resultaat alleen op het scherm, rechts komt het terug in een variabele.

UITGEWERKT VOORBEELD · 28.6

Een functie voor de vochtsensor

De capacitieve vochtsensor van de meetkas geeft een ruwe ADC-waarde. Bij de ijking (F25) gaf ze 600 in droge potgrond en 300 in natte, en daartussen is het verband lineair: $\text{vocht} = (600 - \text{ruw})/3$ %. Schrijf een functie die een ruwe waarde omzet in een vochtpercentage, ook voor waarden buiten de ijking.

GEGEVEN

- ijkpunten: ruw 600 → 0 %, ruw 300 → 100 %
- verband: vocht = (600 − ruw) / 3

GEVRAAGD

- een functie vocht_procent(ruw) met docstring
- een test met gekozen waarden

OPLOSSING

1 **Kop en docstring.** Eén parameter (de ruwe waarde), één returnwaarde (het vocht in %). De docstring vermeldt de ijking: wie de sensor later opnieuw ijkt, weet waar de getallen vandaan komen.

2 **Buiten de ijking.** Een ruwe waarde van 640 (grond droger dan bij de ijking, of een los draadje) geeft −13 %, en 250 geeft 117 %. Die bestaan niet. Begrenzen doe je met `max(0.0, ...)` (niet onder 0) en `min(100.0, ...)` (niet boven 100).

3 **Testen.** Kies waarden waarvan je de uitkomst kent: de twee ijkpunten, de drempel van 30 % (ruw 510), een waarde in het midden, en twee buiten de ijking.

```python
# vochtsensor.py - van ruwe ADC-waarde naar bodemvocht in procent
def vocht_procent(ruw):
    """Bodemvocht (%) uit de ruwe ADC-waarde van de capacitieve sensor.

    IJking (F25): 600 in droge potgrond, 300 in natte. Begrensd tot 0-100 %.
    """
    vocht = (600 - ruw) / 3
    return min(100.0, max(0.0, vocht))


for ruw in (600, 510, 405, 300, 250, 640):
    print(ruw, "->", round(vocht_procent(ruw), 1), "%")
```

```
600 -> 0.0 %
510 -> 30.0 %
405 -> 65.0 %
300 -> 100.0 %
250 -> 100.0 %
640 -> 0.0 %
```

ANTWOORD

De functie geeft (600 − ruw) / 3, begrensd tot 0-100 %. De drempel van 30 % hoort bij ruw 510, zoals in F25.

Klopt dit? De uitkomst voor de ijkpunten klopt (0 en 100 %), 405 ligt precies tussen 510 en 300 en geeft 65 %. Eén kanttekening: door het begrenzen zie je een losse sensor niet meer. In een echte regeling test je daarom ook apart of de ruwe waarde plausibel is (bijvoorbeeld tussen 250 en 650).

UITLEG · 28.7

Lijsten: veel waarden onder één naam

Een **lijst** zet je tussen vierkante haken, met komma's tussen de elementen. Elk element heeft een **index**: zijn plaats, **vanaf 0**. Met een negatieve index tel je van achteren: `[-1]` is het laatste element. Hier de

hoogste temperatuur van elke dag van de meetweek:

```
>>> maxima = [28.1, 25.1, 21.1, 22.8, 27.4, 28.6, 25.1]
>>> maxima[0]
28.1
>>> maxima[5]
28.6
>>> maxima[-1]
25.1
>>> len(maxima)
7
>>> 28.6 in maxima
True
>>> maxima.index(28.6)
5
```

Een lijst kan groeien en veranderen. Je begint vaak met een lege lijst en voegt met `append` telkens een meting toe; een element vervang je via zijn index:

```
>>> metingen = []
>>> metingen.append(19.4)
>>> metingen.append(21.0)
>>> metingen
[19.4, 21.0]
>>> metingen[0] = 19.6
>>> metingen
[19.6, 21.0]
>>> len(metingen)
2
```

Een lijst mag waarden van elk type bevatten, ook teksten: `dagen = ["ma", "di", ...]`. Twee lijsten met evenveel elementen (dagen en maxima) horen vaak samen: het element met index 5 in de ene lijst hoort bij index 5 in de andere.

> ⚠ **Index 0 is het eerste element**
>
> Een lijst met 7 elementen heeft de indexen 0 tot en met 6. `maxima[7]` bestaat niet en geeft een IndexError. Dit is dezelfde afspraak als bij `range(7)`.

index	0	1	2	3	4	5	6
maxima	28.1	25.1	21.1	22.8	27.4	28.6	25.1
dag	ma	di	wo	do	vr	za	zo
negatief	-7	-6	-5	-4	-3	-2	-1

Boven de index vanaf 0, onder de negatieve index vanaf het einde.

UITLEG · 28.8

Slicing: een stuk uit een lijst of een tekst

8/24

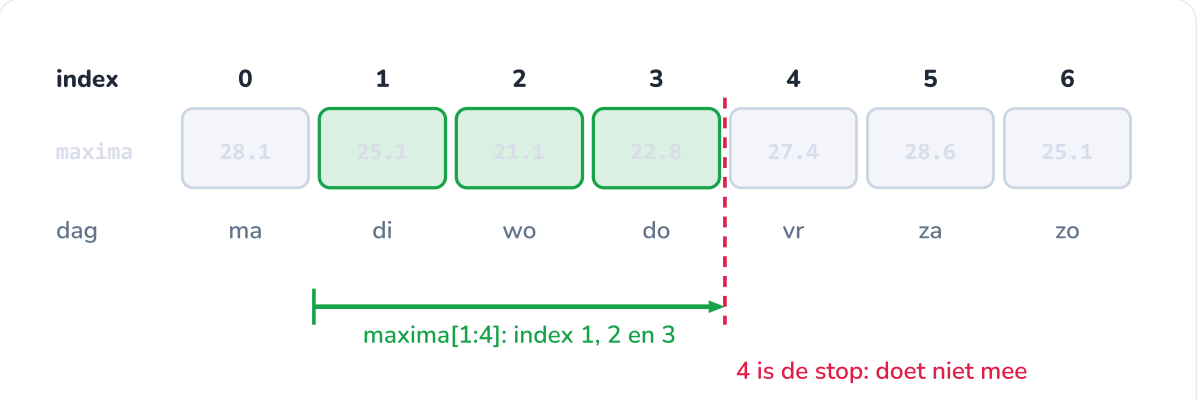
Met `lijst[a:b]` neem je een stuk (een **slice**): van index a tot, maar niet met, index b. Laat je a weg, dan begin je vooraan; laat je b weg, dan loop je tot het einde. Een derde getal is de stap, zoals bij range(). Het resultaat is een nieuwe lijst; de oude blijft ongewijzigd.

```
>>> maxima = [28.1, 25.1, 21.1, 22.8, 27.4, 28.6, 25.1]
>>> maxima[1:4]
[25.1, 21.1, 22.8]
>>> maxima[:3]
[28.1, 25.1, 21.1]
>>> maxima[5:]
[28.6, 25.1]
>>> maxima[-2:]
[28.6, 25.1]
>>> maxima[::2]
[28.1, 21.1, 27.4, 25.1]
```

Een tekst (str) is ook een reeks, van tekens. Daarom werkt slicing ook op tekst, en dat is handig bij datalogbestanden, waar de datum en het uur samen in één tekst staan:

```
>>> tijdstip = "2026-04-20 06:00"
>>> tijdstip[:10]
'2026-04-20'
>>> tijdstip[11:13]
'06'
>>> int(tijdstip[11:13])
6
>>> len(tijdstip)
16
```

Met `tijdstip[:10]` groepeer je straks de metingen per dag, met `int(tijdstip[11:13])` per uur.



maxima[1:4] neemt de elementen met index 1, 2 en 3; de stopindex 4 hoort er net niet meer bij.

CODE · 28.9

Rekenen met lijsten: len, sum, min, max en sorted

9/24

Voor lijsten met getallen heeft Python ingebouwde functies die de statistiek van hoofdstuk 1 in één regel doen. Het gemiddelde is `sum(l) / len(l)`. `sorted()` geeft een nieuwe, gesorteerde lijst (met `reverse=True` van groot naar klein); `l.index(x)` geeft de

plaats van x, en met die index zoek je in een tweede lijst op welke dag het was.

Python **week.py** 11 regels

```python
# week.py - rekenen met een lijst: de hoogste temperatuur van elke dag
dagen = ["ma", "di", "wo", "do", "vr", "za", "zo"]
maxima = [28.1, 25.1, 21.1, 22.8, 27.4, 28.6, 25.1]        # graden Celsius

print("aantal dagen:", len(maxima))
print("gemiddeld maximum:", round(sum(maxima) / len(maxima), 1))
print("laagste:", min(maxima), "- hoogste:", max(maxima))
i = maxima.index(max(maxima))
print("warmste dag van de week:", dagen[i])
print("gesorteerd:", sorted(maxima))
print("drie warmste:", sorted(maxima, reverse=True)[:3])
```

Uitvoer:

```
aantal dagen: 7
gemiddeld maximum: 25.5
laagste: 21.1 - hoogste: 28.6
warmste dag van de week: za
gesorteerd: [21.1, 22.8, 25.1, 25.1, 27.4, 28.1, 28.6]
drie warmste: [28.6, 28.1, 27.4]
```

schrijfwijze	betekenis	voorbeeld met maxima	uitkomst
`l[i]`	element met index i (vanaf 0)	`maxima[2]`	`21.1`
`l[-1]`	laatste element	`maxima[-1]`	`25.1`
`l[a:b]`	stuk van index a tot (niet met) b	`maxima[1:4]`	`[25.1, 21.1, 22.8]`
`len(l)`	aantal elementen	`len(maxima)`	`7`
`sum(l)`, `min(l)`, `max(l)`	som, kleinste, grootste	`max(maxima)`	`28.6`
`sorted(l)`	nieuwe, gesorteerde lijst	`sorted(maxima)[:3]`	`[21.1, 22.8, 25.1]`
`l.index(x)`	index van de eerste x	`maxima.index(25.1)`	`1`
`l.count(x)`	hoe vaak x voorkomt	`maxima.count(25.1)`	`2`
`x in l`	zit x in de lijst?	`30.0 in maxima`	`False`
`[f(x) for x in l if ...]`	nieuwe lijst uit een oude	`[t for t in maxima if t > 26]`	`[28.1, 27.4, 28.6]`
`l.append(x)`	x achteraan toevoegen	verandert de lijst zelf	
`l.copy()`	een echte kopie	zonder copy() krijg je een tweede naam voor dezelfde lijst	

⚠ **sorted() of .sort()**

`sorted(maxima)` maakt een nieuwe lijst en laat maxima ongewijzigd. `maxima.sort()` sorteert de lijst zelf en geeft None terug. Schrijf dus nooit `maxima = maxima.sort()`: dan is je lijst weg.

UITLEG · 28.10

Lussen over lijsten: ophopen, tellen, filteren

10/24

Een for-lus loopt niet alleen over range(), maar over elke lijst: de lusvariabele krijgt elke ronde het volgende element. Vier patronen komen steeds terug. **Ophopen**: een totaal vóór de lus, in de lus erbij tellen. **Tellen of filteren**: in de lus een if. **Twee lijsten tegelijk**: met `zip()`, en met `enumerate()` krijg je

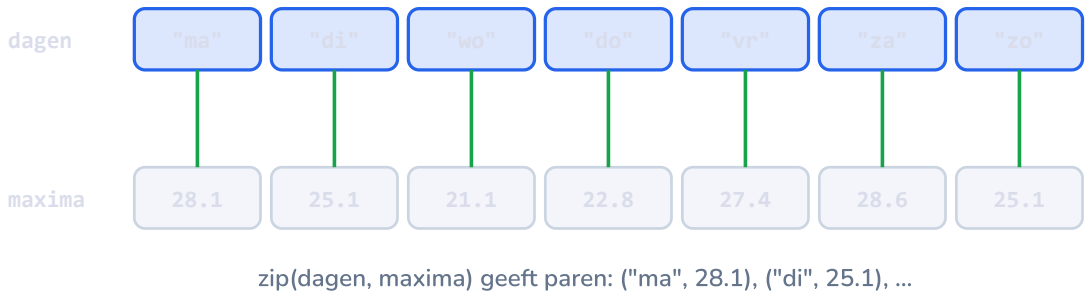
er een volgnummer bij. **Een nieuwe lijst** maken uit een oude: met append in een lus, of korter met een **lijstomschrijving** (*list comprehension*) tussen vierkante haken.

```
# patronen.py - vier manieren om over een lijst te lopen
dagen = ["ma", "di", "wo", "do", "vr", "za", "zo"]
maxima = [28.1, 25.1, 21.1, 22.8, 27.4, 28.6, 25.1]

totaal = 0                              # 1. ophopen
for t in maxima:
    totaal = totaal + t
print("som:", round(totaal, 1))

aantal = 0                              # 2. tellen met een voorwaarde
for t in maxima:
    if t > 26:
        aantal = aantal + 1
print("dagen boven 26 graden:", aantal)

for dag, t in zip(dagen, maxima):       # 3. twee lijsten tegelijk
    if t < 23:
        print("koele dag:", dag, t)

for nummer, dag in enumerate(dagen[:3], start=1):
    print(nummer, dag)

kelvin = [round(t + 273.15, 2) for t in maxima]     # 4. een nieuwe lijst
print(kelvin)
```

```
som: 178.2
dagen boven 26 graden: 3
koele dag: wo 21.1
koele dag: do 22.8
1 ma
2 di
3 wo
[301.25, 298.25, 294.25, 295.95, 300.55, 301.75, 298.25]
```

De lijstomschrijving `[round(t + 273.15, 2) for t in maxima]` leest als 'voor elke t in maxima: t in kelvin'. Met een if erachter filter je: `[t for t in maxima if t > 26]` geeft alleen de warme dagen.

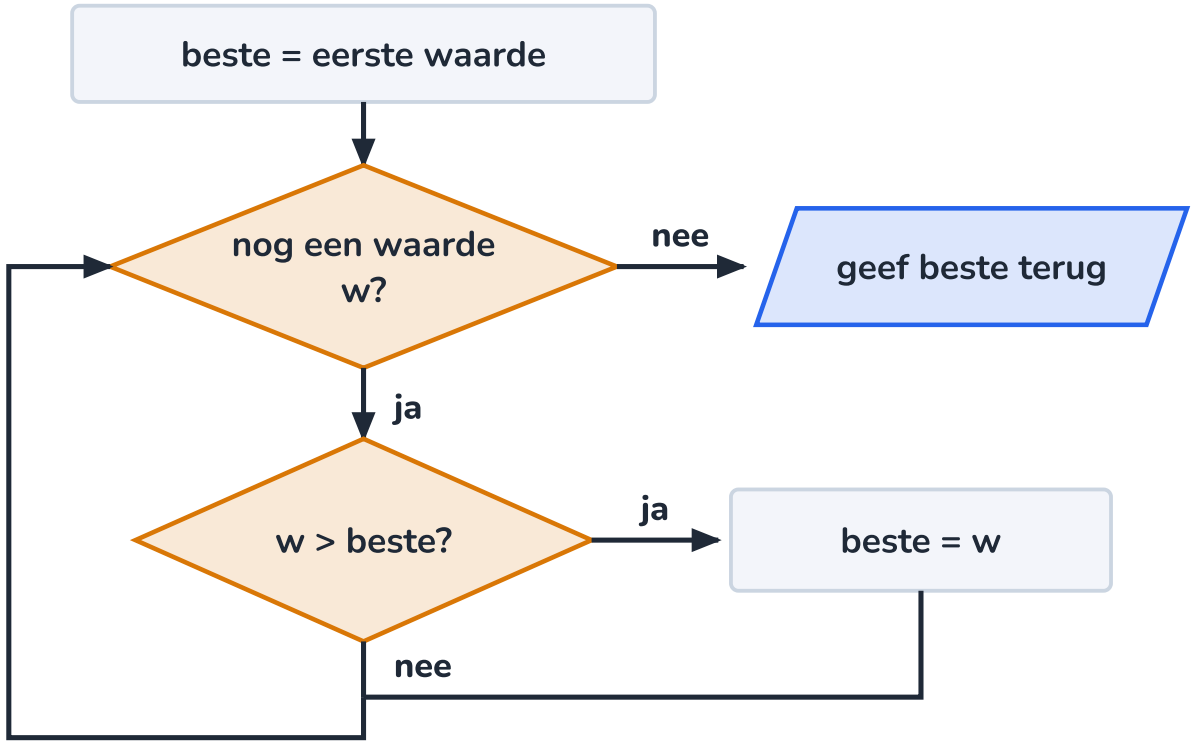
> **Kies de eenvoudigste vorm**
> Een gewone lus met append is altijd goed. Een lijstomschrijving is korter, maar schrijf ze alleen als ze op één regel leesbaar blijft.



zip koppelt het eerste element van de ene lijst aan het eerste van de andere, enzovoort.

Gemiddelde en maximum zelf programmeren

sum(), len() en max() doen het werk voor je, maar hoe werken ze? Schrijf zelf een functie voor het gemiddelde en een voor de grootste waarde, test ze op de zonnestroom per dag van de meetweek, en vergelijk met de ingebouwde functies.



Het algoritme van max(): een voorlopige beste, die bij elke grotere waarde vervangen wordt.

GEGEVEN

- zonnestroom per dag (Wh), maandag tot zondag: 7465, 5105, 1547, 3077, 6949, 7717, 4676

GEVRAAGD

- functies gemiddelde(waarden) en grootste(waarden)
- een trace van grootste()

OPLOSSING

1. **Gemiddelde.** Het patroon ophopen: `totaal = 0`, in de lus elke waarde erbij, na de lus delen door het aantal. Een lege lijst zou een deling door nul geven; de docstring zegt dat ze niet leeg mag zijn.

2. **Grootste.** Hetzelfde idee als bij het sorteren van kaarten in F22, waar je telkens de kleinste zocht: neem de eerste waarde als voorlopig beste en vergelijk elke volgende waarde ermee; is ze groter, dan wordt zij de nieuwe beste. `waarden[1:]` slaat de eerste over, want die is al de beste.

3 **Het programma:**

```
# statistiek.py - gemiddelde en maximum zelf uitrekenen
def gemiddelde(waarden):
    """Rekenkundig gemiddelde van een lijst getallen (de lijst mag niet leeg zijn)."""
    totaal = 0
    for w in waarden:
        totaal = totaal + w
    return totaal / len(waarden)


def grootste(waarden):
    """Grootste waarde van een lijst, zonder max()."""
    beste = waarden[0]
    for w in waarden[1:]:
        if w > beste:
            beste = w
    return beste


zon = [7465, 5105, 1547, 3077, 6949, 7717, 4676]        # Wh per dag, maandag tot zondag
print("gemiddeld:", round(gemiddelde(zon)), "Wh - met sum/len:", round(sum(zon) / len(zon)))
print("grootste:", grootste(zon), "Wh - met max():", max(zon))
```

```
gemiddeld: 5219 Wh - met sum/len: 5219
grootste: 7717 Wh - met max(): 7717
```

4 **Trace van grootste().** Beste begint op 7465.

W	W > BESTE?	BESTE DAARNA
5105	nee	7465
1547	nee	7465
3077	nee	7465
6949	nee	7465
7717	ja	7717
4676	nee	7717

ANTWOORD

Beide functies geven dezelfde uitkomst als de ingebouwde: gemiddeld 5219 Wh per dag, hoogstens 7717 Wh (zaterdag).

Klopt dit? Test ook de randgevallen: een lijst met één element (dan is dat het maximum), een lijst waarin het maximum vooraan of helemaal achteraan staat, en negatieve getallen. Wie `beste = 0` als begin kiest in plaats van de eerste waarde, krijgt bij een lijst met alleen negatieve temperaturen 0 als maximum: een klassieke fout.

UITLEG · 28.12

Woordenboeken: opzoeken met een sleutel

12/24

In een lijst zoek je een waarde op met een getal (de index). In een **woordenboek** (*dictionary*, type `dict`) zoek je op met een **sleutel**, meestal een tekst. Je schrijft het tussen accolades, met paren `sleutel: waarde`. Een voorbeeld is de pintabel van de Arduino Mega van de klimaatmodule:

```
>>> pinnen = {"A0": "NTC reservoir", "A1": "LDR", "A2": "pH-module", "A3":
"batterijdeler"}
>>> pinnen["A2"]
```

```
'pH-module'
>>> pinnen["D9"] = "pomp (MOSFET)"
>>> len(pinnen)
5
>>> "A5" in pinnen
False
>>> pinnen.get("A5", "vrij")
'vrij'
```

Over alle paren loop je met `items()`:

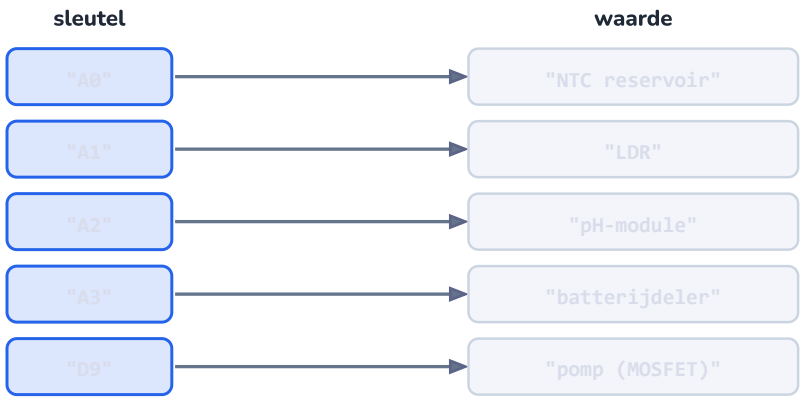
```
for pin, onderdeel in pinnen.items():
    print(pin, "->", onderdeel)
```

```
A0 -> NTC reservoir
A1 -> LDR
A2 -> pH-module
A3 -> batterijdeler
D9 -> pomp (MOSFET)
```

schrijfwijze	betekenis
`d = {"A0": "NTC", "A1": "LDR"}`	woordenboek met sleutels en waarden
`d["A0"]`	waarde opzoeken (KeyError als de sleutel ontbreekt)
`d["D9"] = "pomp"`	toevoegen of vervangen
`d.get("A5", "vrij")`	opzoeken met een standaardwaarde
`"A5" in d`	bestaat de sleutel?
`for k, w in d.items():`	over alle paren lopen
`d.keys()`, `d.values()`, `len(d)`	sleutels, waarden, aantal paren

Een woordenboek is ook de handigste manier om metingen **per groep** te verzamelen: de sleutel is de datum (of het dagdeel), de waarde een lijst met de metingen van die groep. Dat doe je verderop met de meetweek.

> **Lijst of woordenboek?**
> Een lijst als de volgorde telt en de waarden van dezelfde soort zijn (de temperatuur elk uur). Een woordenboek als je iets opzoekt met een naam (welke sensor op A2 zit, hoeveel watt de pomp verbruikt).



Elke sleutel komt één keer voor en wijst naar één waarde.

CODE · 28.13

Habitat: het energiebudget met een woordenboek en functies

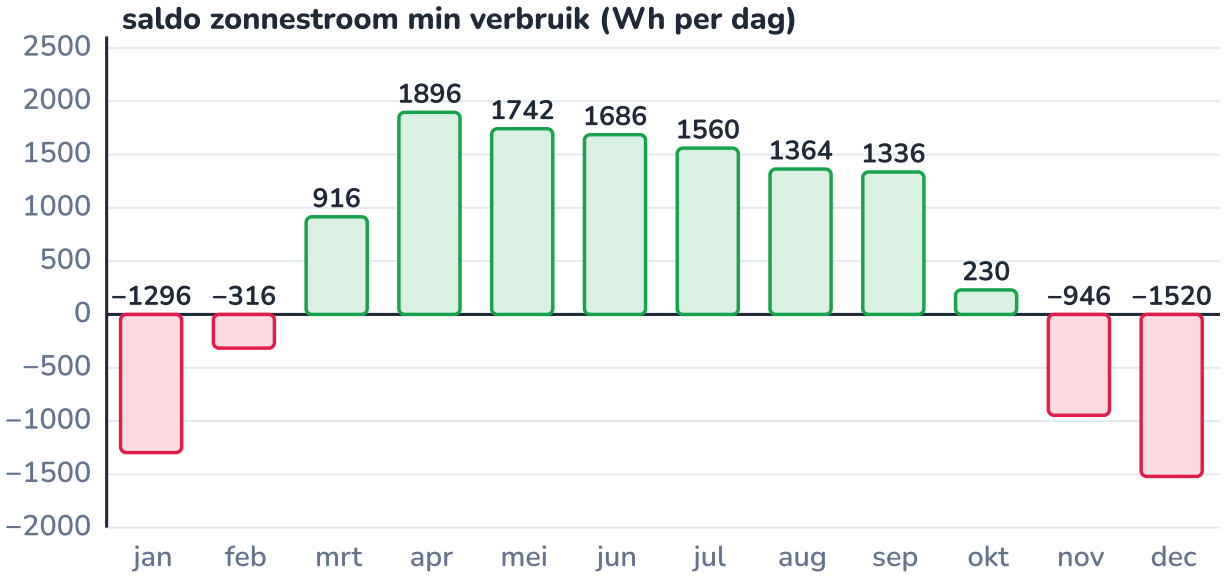
13/24

De rekenhulp van hoofdstuk 27 rekende één maand tegelijk; voor een andere maand moest je twee regels aanpassen. Met een woordenboek voor de verbruikers, een lijst met de twaalf PVGIS-waarden en drie functies rekent hetzelfde programma het hele jaar door. In de for-lus over `items()` pakt `naam, (vermogen, uren)` elk paar meteen uit.

Python **budget_maanden.py** 42 regels

```python
# budget_maanden.py - de rekenhulp van hoofdstuk 27 met een woordenboek, lijsten en functies
VERBRUIKERS = {                          # naam: (vermogen in W, uren per dag)
    "groeilampen": (190, 16),
    "pomp": (12, 4),
    "sensoren": (3.0, 24),
    "ventilatie dag": (5, 16),
    "ventilatie nacht": (2, 8),
    "CO2-klep": (6, 0.25),
    "BMS": (0.5, 24),
}
MAANDEN = ['jan', 'feb', 'mrt', 'apr', 'mei', 'jun', 'jul', 'aug', 'sep', 'okt', 'nov', 'dec']
WH_PER_WP = [1.41, 2.11, 2.99, 3.69, 3.58, 3.54, 3.45, 3.31, 3.29, 2.5, 1.66, 1.25]
PIEK = 4 * 350                           # Wp


def verbruik_per_dag(lichturen=16):
    """Verbruik van de module in Wh per dag, als de groeilampen 'lichturen' uur branden."""
    totaal = 0
    for naam, (vermogen, uren) in VERBRUIKERS.items():
        if naam == "groeilampen":
            uren = lichturen
        totaal = totaal + vermogen * uren
    return totaal


def zon_per_dag(maand):
    """Zonnestroom in Wh per dag in maand 1 tot 12 (PVGIS, Brussel, 60 graden zuid)."""
    return PIEK * WH_PER_WP[maand - 1]


def max_lichturen(maand):
    """Hoeveel lichturen de zonnestroom alleen kan betalen (hoogstens 16)."""
    for uren in range(16, -1, -1):
        if verbruik_per_dag(uren) <= zon_per_dag(maand):
            return uren
    return 0

```

```
38
39  print("maand   zon  saldo  lichturen")
40  for m in range(1, 13):
41      saldo = zon_per_dag(m) - verbruik_per_dag()
42      print(f"{MAANDEN[m - 1]:5} {zon_per_dag(m):5.0f} {saldo:+6.0f} {max_lichturen(m):6d}")
```



Van november tot februari is er een tekort bij 16 lichturen, de rest van het jaar een overschot.

Uitvoer:

```
maand   zon  saldo  lichturen
jan    1974  -1296      9
feb    2954   -316     14
mrt    4186   +916     16
apr    5166  +1896     16
mei    5012  +1742     16
jun    4956  +1686     16
jul    4830  +1560     16
aug    4634  +1364     16
sep    4606  +1336     16
okt    3500   +230     16
nov    2324   -946     11
dec    1750  -1520      8
```

De kolom lichturen is precies wat de rekenhulp van hoofdstuk 27 één maand per keer gaf. Het verschil: een verbruiker toevoegen is nu één regel in het woordenboek, en elke functie is apart te testen. `max_lichturen` gebruikt `return` midden in een lus: zodra de voorwaarde klopt, stopt de functie.

UITLEG · 28.14

CSV: een tabel als tekst

14/24

Een **CSV-bestand** (*comma-separated values*) is een tabel als gewone tekst: één rij per regel, de kolommen gescheiden door komma's. De eerste regel is meestal de **kop** met de namen van de

kolommen. Elk rekenblad kan CSV openen en bewaren, elke datalogger kan het schrijven, en je leest het met een teksteditor. Daarom is het het meest gebruikte formaat voor meetdata.

Het bestand van de klimaatmodule heeft vijf kolommen: het **tijdstip** (datum en uur, zoals '2026-04-20 06:00'), de **temperatuur** in °C en de **vochtigheid** in % (gemiddelden over de drie uur die op dat tijdstip beginnen), en het **verbruik** en de **zonnestroom** in Wh in diezelfde drie uur. De eenheid staat in de naam van de kolom: `temperatuur_C`, `verbruik_Wh`. Een lege cel betekent: geen meting.

> ⚠ **Een CSV uit een Nederlandstalig rekenblad**
>
> In het Nederlands is de komma het decimaalteken. Een rekenblad met Belgische of Nederlandse instellingen bewaart 'CSV' daarom vaak met een **puntkomma** tussen de kolommen en een **komma** in de getallen: `2026-04-20 06:00;19,9;78;571;126`, en soms schrijft het de datum anders (`20/04/2026 6:00`). Kijk zo'n bestand eerst na in een teksteditor. Lees het met `csv.DictReader(bestand, delimiter=";")` en zet elk getal om met `float(tekst.replace(",", "."))`. Bewaart het rekenblad als 'CSV UTF-8', dan staat er vaak een onzichtbaar teken vooraan het bestand; open het dan met `encoding="utf-8-sig"`, anders heet je eerste kolom niet 'tijdstip'.

> 💡 **Noem je programma nooit csv.py**
>
> Heet je eigen bestand `csv.py`, dan laadt `import csv` jouw bestand in plaats van de module, en krijg je rare foutmeldingen.

als tekst (in een teksteditor)

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh
2026-04-20 00:00,18.6,84,22,0
2026-04-20 03:00,18.5,83,22,0
2026-04-20 06:00,19.9,78,571,126
2026-04-20 09:00,23.6,71,172,1758
```

→

als tabel (in een rekenblad)

tijdstip	T	vocht	verbr.	zon
04-20 00:00	18.6	84	22	0
04-20 03:00	18.5	83	22	0
04-20 06:00	19.9	78	571	126
04-20 09:00	23.6	71	172	1758

de kop wordt de sleutels van DictReader

Dezelfde vijf regels: links zoals ze in het bestand staan, rechts zoals een rekenblad ze toont.

CODE · 28.15

Het bestand: een week in de klimaatmodule

15/24

Dit is het volledige bestand: van maandag 20 tot zondag 26 april 2026, om de drie uur een regel, 56 rijen onder de kop. Met de knop *Download* bewaar je het als `klimaatmodule_week.csv`, in dezelfde map als je programma's. Kijk er eerst zelf naar: welke dagen waren zonnig? En waar ontbreekt een meting?

tekst **klimaatmodule_week.csv** 57 regels

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh
2026-04-20 00:00,18.6,84,22,0
2026-04-20 03:00,18.5,83,22,0
2026-04-20 06:00,19.9,78,571,126
2026-04-20 09:00,23.6,71,172,1758
2026-04-20 12:00,28.0,63,90,2799
2026-04-20 15:00,28.1,63,88,2291
2026-04-20 18:00,23.7,69,482,491
2026-04-20 21:00,20.1,79,216,0
2026-04-21 00:00,18.5,84,22,0
2026-04-21 03:00,17.8,84,22,0
2026-04-21 06:00,19.4,79,578,95
2026-04-21 09:00,22.3,77,283,1305
2026-04-21 12:00,25.1,66,169,1776
2026-04-21 15:00,25.1,66,225,1540
2026-04-21 18:00,22.1,75,506,389
```

```
2026-04-21 21:00,19.7,81,216,0
2026-04-22 00:00,18.2,83,22,0
2026-04-22 03:00,17.7,82,22,0
2026-04-22 06:00,19.2,83,595,25
2026-04-22 09:00,19.7,78,525,314
2026-04-22 12:00,,78,454,611
2026-04-22 15:00,21.1,76,483,484
2026-04-22 18:00,20.3,80,574,113
2026-04-22 21:00,19.1,83,216,0
2026-04-23 00:00,18.5,83,22,0
2026-04-23 03:00,17.8,83,22,0
2026-04-23 06:00,19.6,81,587,58
2026-04-23 09:00,20.9,78,431,698
2026-04-23 12:00,22.8,73,317,1172
2026-04-23 15:00,22.7,76,371,942
2026-04-23 18:00,21.1,77,551,207
2026-04-23 21:00,19.1,84,216,0
2026-04-24 00:00,18.9,80,22,0
2026-04-24 03:00,18.4,83,22,0
2026-04-24 06:00,19.9,81,571,125
2026-04-24 09:00,23.4,71,189,1690
2026-04-24 12:00,27.4,65,90,2664
2026-04-24 15:00,27.1,64,113,2000
2026-04-24 18:00,23.4,73,487,470
2026-04-24 21:00,19.5,83,216,0
2026-04-25 00:00,19.1,83,22,0
2026-04-25 03:00,18.5,84,22,0
2026-04-25 06:00,20.4,80,569,132
2026-04-25 09:00,24.0,71,138,1896
2026-04-25 12:00,28.6,61,90,2998
2026-04-25 15:00,28.2,63,88,2186
2026-04-25 18:00,24.1,72,478,505
2026-04-25 21:00,20.4,81,216,0
2026-04-26 00:00,18.7,82,22,0
2026-04-26 03:00,18.2,85,22,0
2026-04-26 06:00,19.6,81,582,81
2026-04-26 09:00,22.6,72,289,1280
2026-04-26 12:00,25.1,68,156,1829
2026-04-26 15:00,24.6,71,310,1194
2026-04-26 18:00,22.1,74,530,292
2026-04-26 21:00,19.2,84,216,0
```

Wat je al ziet zonder te rekenen: 's nachts is het verbruik 22 Wh per blok (sensoren, pomp, ventilator, batterijbeheer) en de zonnestroom 0. Overdag bepalen de groeilampen het verbruik, en die dimmen als er daglicht is: op een zonnige middag zakt het verbruik tot 90 Wh per blok. Woensdag 22 april om 12 uur is de temperatuurcel leeg: de sensor gaf die keer geen geldige waarde. Een programma moet dat opvangen, en mag die cel zeker niet als 0 °C meetellen.

CODE · 28.16

CSV inlezen met de csv-module

16/24

Met `with open(...) as bestand:` open je het bestand; na het ingesprongen blok sluit Python het vanzelf. `csv.DictReader` leest de kop en geeft daarna elke rij als een woordenboek, met de kolomnamen als sleutels: `rij["zon_Wh"]`. Alle waarden zijn **tekst**, zoals bij input(): zet ze om met `int()` of `float()`. `continue` springt meteen naar de volgende ronde van de lus, zodat een lege temperatuur niet omgezet wordt.

Python **inlezen.py** 25 regels

```python
# inlezen.py - de metingen van een week inlezen met de csv-module
import csv

tijden = []
temperaturen = []
verbruik = []
```

```
zon = []
ontbrekend = 0

with open("klimaatmodule_week.csv", newline="", encoding="utf-8") as bestand:
    for rij in csv.DictReader(bestand):
        verbruik.append(int(rij["verbruik_Wh"]))
        zon.append(int(rij["zon_Wh"]))
        if rij["temperatuur_C"] == "":          # de sensor gaf geen waarde
            ontbrekend = ontbrekend + 1
            continue
        tijden.append(rij["tijdstip"])
        temperaturen.append(float(rij["temperatuur_C"]))

print("rijen:", len(verbruik), "- met temperatuur:", len(temperaturen), "- ontbrekend:", ontbrekend)
print("eerste meting:", tijden[0], temperaturen[0])
print("gemiddelde temperatuur:", round(sum(temperaturen) / len(temperaturen), 1))
print("laagste:", min(temperaturen), "- hoogste:", max(temperaturen))
print("het warmst op", tijden[temperaturen.index(max(temperaturen))])
print("in de hele week: verbruik", sum(verbruik), "Wh, zonnestroom", sum(zon), "Wh")
```

Uitvoer (met het bestand in dezelfde map):

```
rijen: 56 - met temperatuur: 55 - ontbrekend: 1
eerste meting: 2026-04-20 00:00 18.6
gemiddelde temperatuur: 21.4
laagste: 17.7 - hoogste: 28.6
het warmst op 2026-04-25 12:00
in de hele week: verbruik 14552 Wh, zonnestroom 36536 Wh
```

De lijsten `tijden` en `temperaturen` blijven even lang, omdat ze samen gevuld worden, ná de test op een lege cel. Zo hoort het tijdstip met index i altijd bij de temperatuur met index i, en vind je met `temperaturen.index(max(temperaturen))` het moment van het maximum.

csv.reader: elke rij als lijst

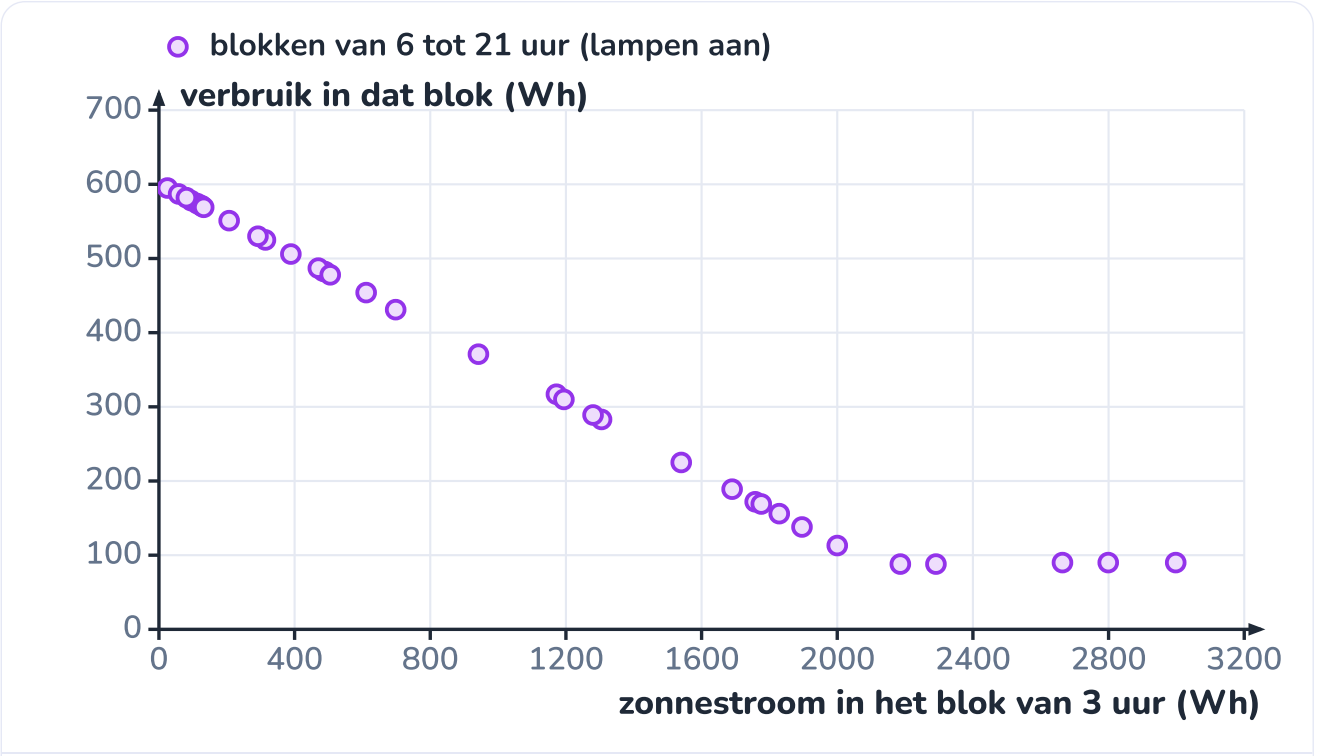
Met `csv.reader(bestand)` krijg je elke rij als lijst van teksten, `['2026-04-20 00:00', '18.6', '84', '22', '0']`, en is de kop gewoon de eerste rij. DictReader is leesbaarder (`rij["zon_Wh"]` in plaats van `rij[4]`) en blijft werken als er later een kolom bijkomt.

UITGEWERKT VOORBEELD · 28.17

Wanneer was het te warm, en wat deed de zon met het verbruik?

17/24

Sla verdraagt warmte slecht; boven ongeveer 26 °C wil de beheerder van de klimaatmodule dat opvolgen. Zoek in de meetweek alle blokken boven 26 °C. Onderzoek ook of de module minder verbruikt als de zon schijnt, zoals je zou verwachten van lampen die dimmen.



Elk punt is een blok van drie uur met de lampen aan: hoe meer zon, hoe lager het verbruik.

GEGEVEN

- klimaatmodule_week.csv
- lampen aan van 6 tot 22 uur: de blokken van 6, 9, 12, 15 en 18 uur hebben drie volle lichturen

GEVRAAGD

- de warme blokken en hun aantal uren
- het gemiddelde verbruik per blok bij veel en bij weinig zon

OPLOSSING

1. **Filteren.** Een lege lijst `warm`, en in de lus een if met twee voorwaarden: de cel is niet leeg *en* de temperatuur is hoger dan 26. De volgorde telt: door `and` wordt float() niet meer uitgevoerd als de cel leeg is.

2. **Het uur uit de tekst.** `int(rij["tijdstip"][11:13])` geeft het uur. Alleen de blokken van 6 tot en met 18 uur tellen mee voor de vergelijking: 's nachts is de zon 0 én zijn de lampen uit, dat zou de vergelijking scheeftrekken.

3 **Twee groepen.** Veel zon: meer dan 1500 Wh in drie uur (gemiddeld 500 W op 1400 Wp). Weinig zon: minder dan 500 Wh. De blokken daartussen laat je weg, zodat de groepen duidelijk verschillen.

```python
# te_warm.py - wanneer was het warmer dan 26 graden, en wat deed de zon met het verbruik?
import csv

warm = []
veel_zon = []
weinig_zon = []
with open("klimaatmodule_week.csv", newline="", encoding="utf-8") as bestand:
    for rij in csv.DictReader(bestand):
        uur = int(rij["tijdstip"][11:13])
        if rij["temperatuur_C"] != "" and float(rij["temperatuur_C"]) > 26:
            warm.append(rij["tijdstip"])
        if 6 <= uur <= 18:                        # blokken waarin de lampen drie uur mogen branden
            if int(rij["zon_Wh"]) > 1500:
                veel_zon.append(int(rij["verbruik_Wh"]))
            elif int(rij["zon_Wh"]) < 500:
                weinig_zon.append(int(rij["verbruik_Wh"]))

print("boven 26 graden:", len(warm), "blokken, of", 3 * len(warm), "uur:")
for t in warm:
    print("  ", t)
print(f"verbruik per blok bij veel zon: {sum(veel_zon) / len(veel_zon):.0f} Wh ({len(veel_zon)} blokken)")
print(f"verbruik per blok bij weinig zon: {sum(weinig_zon) / len(weinig_zon):.0f} Wh ({len(weinig_zon)} blokken)")
```

```
boven 26 graden: 6 blokken, of 18 uur:
   2026-04-20 12:00
   2026-04-20 15:00
   2026-04-24 12:00
   2026-04-24 15:00
   2026-04-25 12:00
   2026-04-25 15:00
verbruik per blok bij veel zon: 134 Wh (12 blokken)
verbruik per blok bij weinig zon: 546 Wh (15 blokken)
```

ANTWOORD

6 blokken, dus 18 uur boven 26 °C, telkens van 12 tot 18 uur op de zonnige dagen (maandag, vrijdag en zaterdag). Bij veel zon verbruikt de module per blok veel minder dan bij weinig zon: de lampen dimmen.

Klopt dit? Vergelijk met de grafiek: de punten met veel zonnestroom liggen onderaan (laag verbruik). Toch is een verband in data nog geen bewijs van een oorzaak (hoofdstuk 39). Hier kennen we de oorzaak wel: de regeling dimt de lampen op het daglicht (hoofdstuk 25). Let ook op de maat: een blok is een gemiddelde over drie uur; een korte piek van 30 °C binnen dat blok zie je in deze data niet.

CODE · 28.18

De week per dag samenvatten

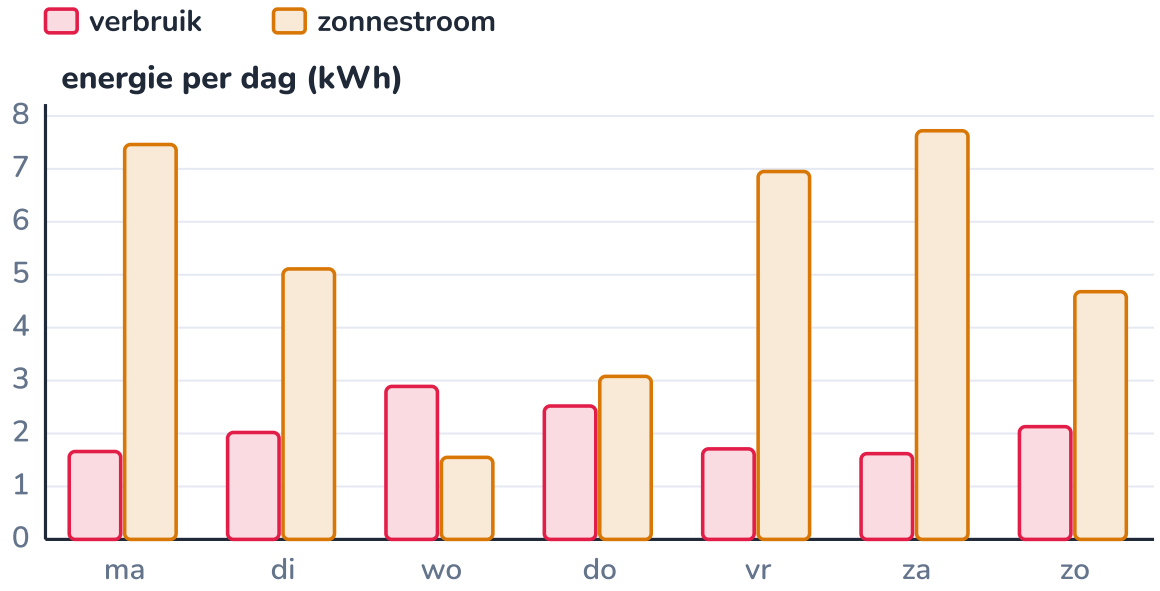
18/24

Een woordenboek verzamelt de rijen per dag: de sleutel is de datum (`rij["tijdstip"][:10]`), de waarde een lijst met de rijen van die dag. Bestaat de sleutel nog niet, dan maak je eerst een lege lijst. Een functie vat één dag samen en geeft vijf waarden tegelijk terug; bij de aanroep pak je ze uit in vijf variabelen.

Python **per_dag.py** 24 regels

```python
# per_dag.py - de week per dag samenvatten
import csv

per_dag = {}                                  # datum -> lijst met de rijen van die dag
with open("klimaatmodule_week.csv", newline="", encoding="utf-8") as bestand:
    for rij in csv.DictReader(bestand):
```

```
        datum = rij["tijdstip"][:10]          # "2026-04-20 06:00" -> "2026-04-20"
        if datum not in per_dag:
            per_dag[datum] = []
        per_dag[datum].append(rij)


def samenvatting(rijen):
    """Gemiddelde, laagste en hoogste temperatuur, verbruik en zonnestroom van een dag."""
    temp = [float(r["temperatuur_C"]) for r in rijen if r["temperatuur_C"] != ""]
    verbruik = sum([int(r["verbruik_Wh"]) for r in rijen])
    zon = sum([int(r["zon_Wh"]) for r in rijen])
    return sum(temp) / len(temp), min(temp), max(temp), verbruik, zon


print("datum       T gem  T min  T max  verbruik    zon   saldo")
for datum, rijen in per_dag.items():
    gem, laag, hoog, verbruik, zon = samenvatting(rijen)
    print(f"{datum} {gem:5.1f} {laag:6.1f} {hoog:6.1f} {verbruik:9d} {zon:6d} {zon - verbruik:+7d}")
```



Zonnige dagen leveren tot vier keer meer dan de module verbruikt; op de regendag is er een tekort.

Uitvoer:

```
datum       T gem  T min  T max  verbruik    zon   saldo
2026-04-20  22.6   18.5   28.1      1663   7465   +5802
2026-04-21  21.2   17.8   25.1      2021   5105   +3084
2026-04-22  19.3   17.7   21.1      2891   1547   -1344
2026-04-23  20.3   17.8   22.8      2517   3077    +560
2026-04-24  22.2   18.4   27.4      1710   6949   +5239
2026-04-25  22.9   18.5   28.6      1623   7717   +6094
2026-04-26  21.3   18.2   25.1      2127   4676   +2549
```

Wat de tabel vertelt: op zonnige dagen (maandag, vrijdag, zaterdag) is het warmst, verbruikt de module het minst (de lampen dimmen) en blijft er het meest zonnestroom over. Alleen op de regendag (woensdag) is het saldo negatief: dan moet de batterij bijspringen. De temperatuur van woensdag is een gemiddelde over zeven blokken, omdat er één ontbreekt; ze telt dus iets minder zwaar mee dan de andere dagen.

PROEF · 28.19

Van meting tot bestand: het klaslokaal een lesuur lang

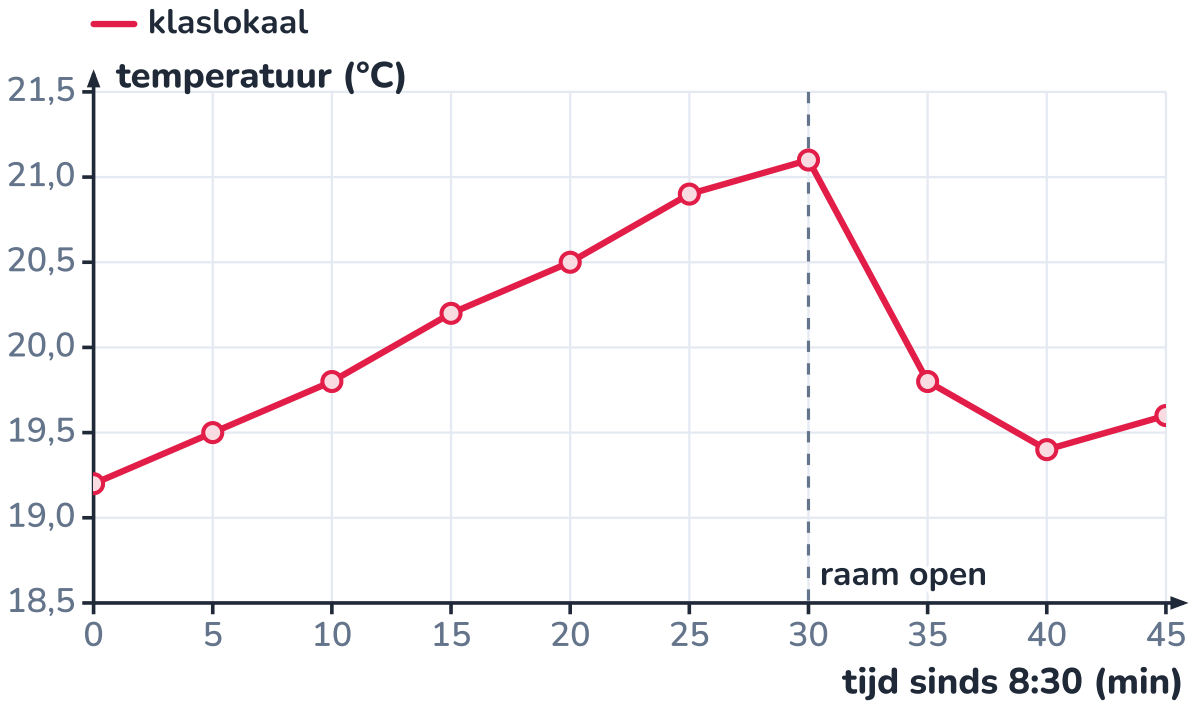
19/24

ONDERZOEKSVRAAG

Hoe verandert de temperatuur in het klaslokaal tijdens een lesuur, en wat doet een open raam?

MATERIAAL

- een thermometer met een resolutie van 0,1 °C (of een temperatuursensor op een microcontroller)
- een horloge of de klok van de computer
- een teksteditor (Kladblok, of de editor van Thonny)
- Python met het programma klaslokaal.py



De temperatuur stijgt met de klas, en zakt snel als het raam opengaat.

WERKWIJZE

1. Hang de thermometer op 1,5 m hoogte, uit de zon en weg van de verwarming.
2. Noteer om de 5 minuten de temperatuur, een lesuur lang. Zet na 30 minuten een raam wijd open en noteer het tijdstip.
3. Typ de metingen in een teksteditor als CSV: eerst de kop `tijd,temperatuur_C`, dan één regel per meting, met een decimale **punt**. Bewaar als `klaslokaal.csv` in UTF-8.
4. Draai het programma: het leest de kolom met een eigen functie en toont gemiddelde, minimum, maximum en de verandering per 5 minuten.
5. Maak met matplotlib een grafiek (pas grafiek_week.py aan) en zet een verticale lijn op het moment dat het raam openging (`plt.axvline(30)`).

WAARNEMING

Een voorbeeld van één klas (klaslokaal.csv):

```
tijd,temperatuur_C
08:30,19.2
08:35,19.5
08:40,19.8
08:45,20.2
08:50,20.5
08:55,20.9
09:00,21.1
09:05,19.8
09:10,19.4
09:15,19.6
```

```
# klaslokaal.py - de temperatuur in het klaslokaal, een lesuur lang
import csv


def lees_kolom(naam, kolom):
    """Alle niet-lege waarden van een kolom uit een CSV-bestand, als kommagetallen."""
    waarden = []
    with open(naam, newline="", encoding="utf-8") as bestand:
        for rij in csv.DictReader(bestand):
            if rij[kolom] != "":
                waarden.append(float(rij[kolom]))
    return waarden


temp = lees_kolom("klaslokaal.csv", "temperatuur_C")
print("metingen:", len(temp))
print(f"gemiddeld {sum(temp) / len(temp):.1f} graden, laagst {min(temp)}, hoogst {max(temp)}")
verschil = [round(temp[i + 1] - temp[i], 1) for i in range(len(temp) - 1)]
print("verandering per 5 minuten:", verschil)
print("grootste daling:", min(verschil))
```

Uitvoer:

```
metingen: 10
gemiddeld 20.0 graden, laagst 19.2, hoogst 21.1
verandering per 5 minuten: [0.3, 0.3, 0.4, 0.3, 0.4, 0.2, -1.3, -0.4, 0.2]
grootste daling: -1.3
```

VERKLARING

Een klas van twintig leerlingen verwarmt het lokaal: elke persoon geeft ongeveer 100 W warmte af (hoofdstuk 14 rekende met ±116 W voor een bewoner). Dat is 2 kW, zoveel als een elektrische kachel. Met het raam open komt koude lucht binnen en zakt de temperatuur in vijf minuten meer dan een graad. Daarna stijgt ze weer licht: de muren en meubels hebben warmte opgeslagen en geven die af (hoofdstuk 18). De functie `lees_kolom` is bruikbaar voor elk CSV-bestand: voor de klimaatmodule roep je ze aan met een andere bestandsnaam en kolomnaam. Dat is de winst van een goede functie.

Uit de zon, weg van de verwarming, op 1,5 m hoogte: zo meet je de temperatuur van het lokaal.

CODE · 28.20

Grafieken met matplotlib

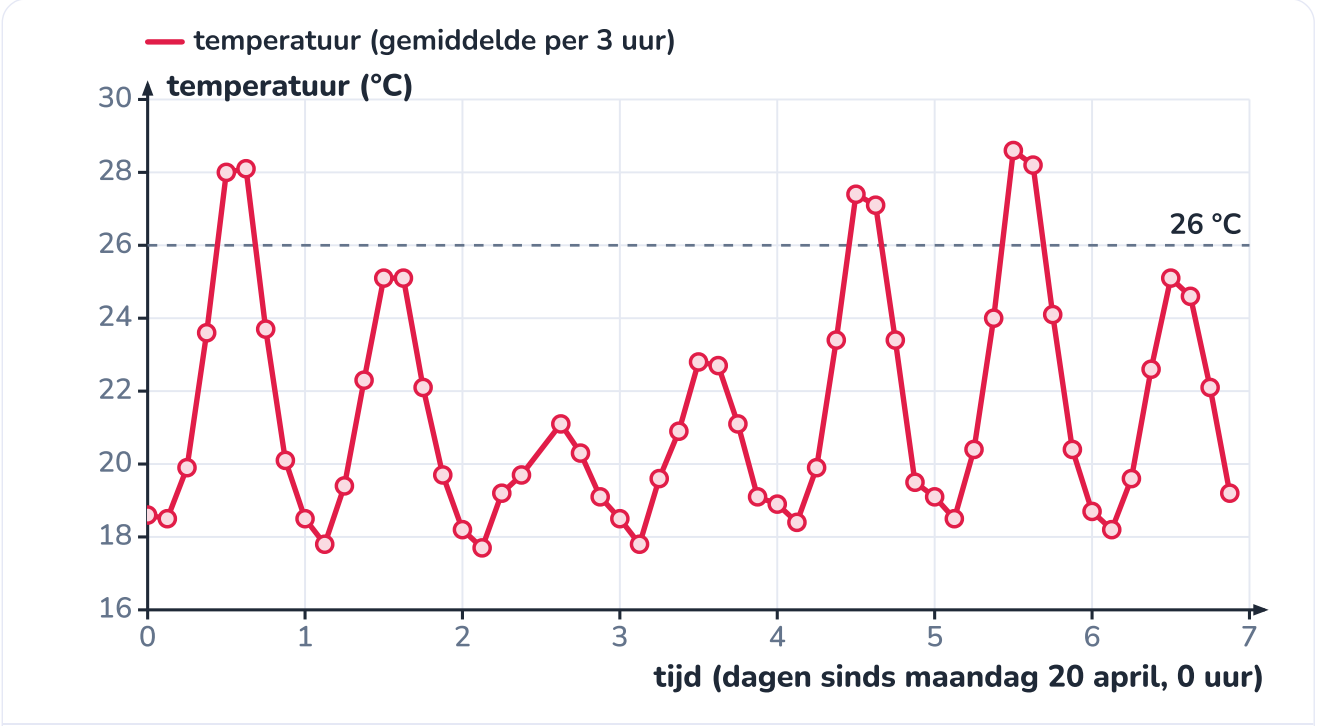
20/24

matplotlib is de bekendste Python-bibliotheek voor grafieken. Je installeert ze één keer (in een terminal: `pip install matplotlib`, in Thonny via *Tools, Manage packages*). De module `pyplot` importeer je met de korte naam `plt`. Het programma leest de temperaturen en de tijd in dagen, en tekent ze als lijn met punten. `plt.savefig()` bewaart de grafiek als afbeelding voor je verslag; `plt.show()` opent ze in een venster.

Python **grafiek_week.py** 21 regels

```python
# grafiek_week.py - de temperatuur van een week in een grafiek
import csv
import matplotlib.pyplot as plt

dagen = []              # tijd in dagen sinds maandag 0 uur
temperaturen = []
with open("klimaatmodule_week.csv", newline="", encoding="utf-8") as bestand:
    for i, rij in enumerate(csv.DictReader(bestand)):
        if rij["temperatuur_C"] != "":
            dagen.append(i * 3 / 24)          # rij i begint i * 3 uur na maandag 0 uur
            temperaturen.append(float(rij["temperatuur_C"]))

plt.plot(dagen, temperaturen, marker="o", label="temperatuur (gemiddelde per 3 uur)")
plt.axhline(26, linestyle="--", color="gray")
plt.xlabel("tijd (dagen sinds maandag 20 april, 0 uur)")
plt.ylabel("temperatuur (°C)")
plt.title("Klimaatmodule, week van 20 april 2026")
plt.grid(True)
```

```
plt.legend()
plt.savefig("temperatuur_week.png", dpi=150)
plt.show()
```



Dezelfde gegevens als in grafiek_week.py, hier getekend in de stijl van dit boek.

De code geeft dezelfde grafiek als hiernaast: dezelfde punten, dezelfde assen, de hulplijn op 26 °C (`plt.axhline`). matplotlib gebruikt haar eigen kleuren en schrijft getallen met een decimale punt. De grafieken in dit boek zijn met dezelfde gegevens getekend in de huisstijl, zodat ze ook op papier en in het donkere thema leesbaar zijn. Het ontbrekende punt van woensdagmiddag is gewoon overgeslagen: de lijn loopt van 9 naar 15 uur.

Een staafdiagram maak je met `plt.bar`. Twee reeksen naast elkaar schuif je elk 0,2 opzij, met een breedte van 0,4; `plt.xticks` zet de namen van de dagen onder de staven. De getallen komen uit per_dag.py (in kWh):

```
# energie_week.py - verbruik en zonnestroom per dag als staafdiagram
import matplotlib.pyplot as plt

dagen = ["ma", "di", "wo", "do", "vr", "za", "zo"]
verbruik = [1.66, 2.02, 2.89, 2.52, 1.71, 1.62, 2.13]        # kWh per dag, uit per_dag.py
zon = [7.46, 5.11, 1.55, 3.08, 6.95, 7.72, 4.68]
x = range(len(dagen))
plt.bar([i - 0.2 for i in x], verbruik, width=0.4, label="verbruik")
plt.bar([i + 0.2 for i in x], zon, width=0.4, label="zonnestroom")
plt.xticks(x, dagen)
plt.ylabel("energie per dag (kWh)")
plt.legend()
plt.show()
```

Die code tekent het staafdiagram van het blad 'De week per dag samenvatten'.

Een grafiek die zegt wat hij toont

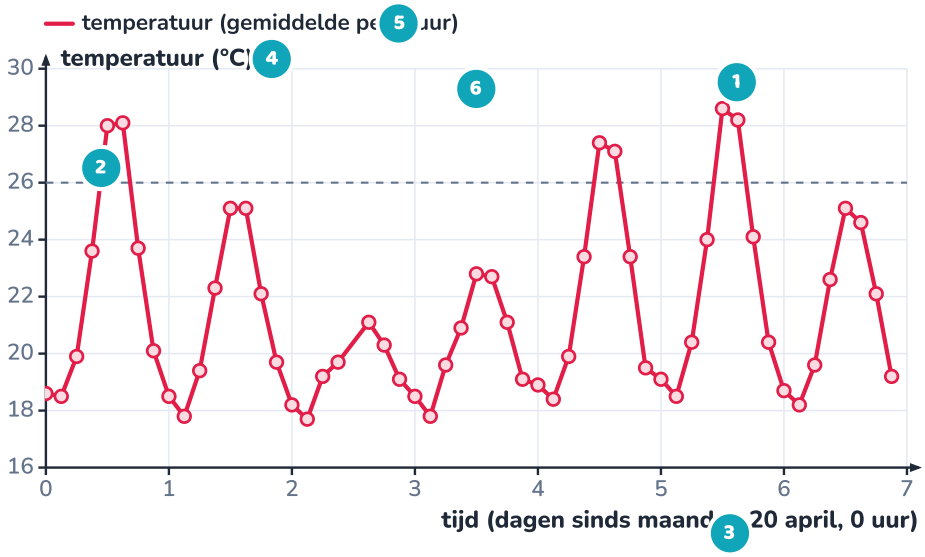
Elke as krijgt een grootheid en een eenheid, elke reeks een label in de legende. Een titel is handig op het scherm; in een verslag komt de uitleg in het onderschrift.

ONTLEED · 28.21

Een goede grafiek ontleed

21/24

Dezelfde grafiek, met bij elk onderdeel de regel matplotlib die het tekent. Een grafiek zonder één van deze onderdelen is moeilijker te lezen, of zegt zelfs iets anders dan de data (hoofdstuk 39).



Elk onderdeel van een grafiek heeft een eigen regel code.

1 Datapunten en lijn

`plt.plot(dagen, temperaturen, marker="o")`: punten op de gemeten waarden, verbonden door een lijn omdat de tijd doorloopt. Het hoogste punt: zaterdag 12 uur, 28,6 °C.

2 Hulplijn

`plt.axhline(26, linestyle="--")`: een grens die de lezer helpt, hier 26 °C.

3 x-as met grootheid en eenheid

`plt.xlabel(...)`: wat staat er horizontaal, en vanaf wanneer telt de tijd?

4 y-as met grootheid en eenheid

`plt.ylabel("temperatuur (°C)")`. De as begint hier bij 16 °C, niet bij 0: dat mag, als je het ziet aan de getallen.

5 Legende

`label="..."` in plt.plot en daarna `plt.legend()`: nodig zodra er meer dan één reeks is.

6 Raster

`plt.grid(True)`: lichte lijnen om waarden af te lezen. De stap (1 dag, 2 °C) kies je zo dat de getallen rond zijn.

UITLEG · 28.22

Foutmeldingen en valkuilen bij lijsten, woordenboeken en functies

22/24

Met lijsten, woordenboeken en functies komen er nieuwe foutmeldingen bij. Lees ze zoals in hoofdstuk 27: van onder naar boven. Nieuw is dat een traceback door een functie heen loopt: dan staan er twee plaatsen, de aanroep en de regel in de functie. De echte meldingen van Python 3.13:

IndexError. Drie elementen hebben de indexen 0, 1 en 2. De laatste is `temperaturen[-1]` of `temperaturen[len(temperaturen) - 1]`.

```
temperaturen = [19.4, 21.0, 23.5]
print(temperaturen[3])
```

```
Traceback (most recent call last):
  File "fouten.py", line 2, in <module>
    print(temperaturen[3])
```

```
        ~~~~~~~~~~~~^^^
IndexError: list index out of range
```

KeyError. De sleutel bestaat niet. Test eerst met `in`, of gebruik `pinnen.get("A2", "vrij")`. Bij een CSV-bestand is het vaak een tikfout in de kolomnaam.

```
pinnen = {"A0": "NTC", "A1": "LDR"}
print(pinnen["A2"])
```

```
Traceback (most recent call last):
  File "fouten.py", line 2, in <module>
    print(pinnen["A2"])
          ~~~~~~^^^^^^
KeyError: 'A2'
```

AttributeError. Een lijst kent geen push (dat is JavaScript); in Python is het `append`.

```
metingen = [19.4, 21.0]
metingen.push(23.5)
```

```
Traceback (most recent call last):
  File "fouten.py", line 2, in <module>
    metingen.push(23.5)
    ^^^^^^^^^^^^
AttributeError: 'list' object has no attribute 'push'
```

TypeError. De functie heeft twee parameters en krijgt maar één argument. Geef ze allebei mee, of geef de parameter een standaardwaarde.

```
def energie(vermogen, uren):
    return vermogen * uren


print(energie(190))
```

```
Traceback (most recent call last):
  File "fouten.py", line 5, in <module>
    print(energie(190))
          ~~~~~~~^^^^^
TypeError: energie() missing 1 required positional argument: 'uren'
```

ZeroDivisionError. De traceback heeft nu twee niveaus: regel 5 riep de functie aan, en in de functie (regel 2) liep het mis. Een lege lijst heeft lengte 0. Zeg in de docstring dat de lijst niet leeg mag zijn, of test het: `if len(lijst) == 0: return None`.

```
def gemiddelde(lijst):
    return sum(lijst) / len(lijst)


print(gemiddelde([]))
```

```
Traceback (most recent call last):
  File "fouten.py", line 5, in <module>
    print(gemiddelde([]))
          ~~~~~~~~~~^^^^
  File "fouten.py", line 2, in gemiddelde
    return sum(lijst) / len(lijst)
```

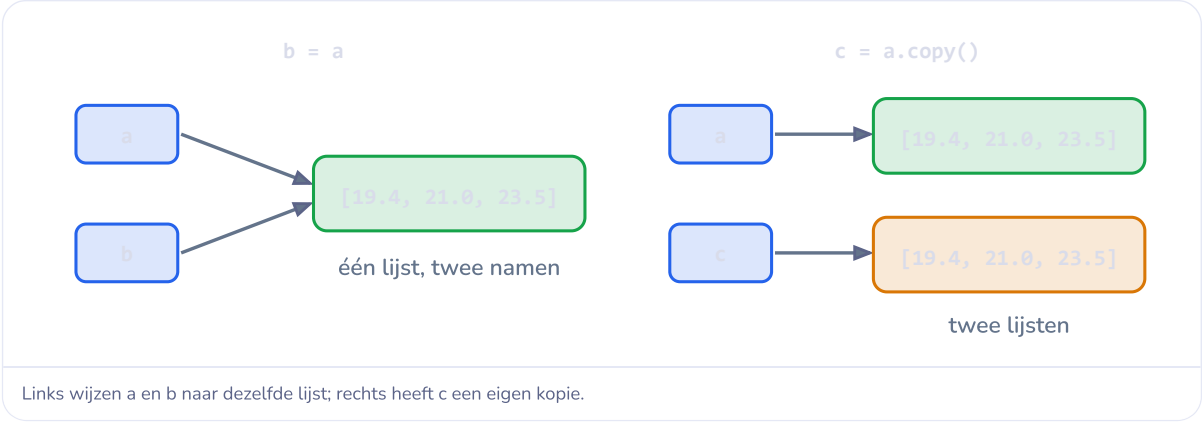
```
            ~~~~~~~~~~~^~~~~~~~~~~~
ZeroDivisionError: division by zero
```

"b = a maakt een kopie van de lijst"

Een naam is een naamkaartje (hoofdstuk 27). Na `b = a` hangen twee kaartjes aan **dezelfde** lijst: wie via b iets toevoegt, verandert ook a. Wil je een echte kopie, schrijf dan `b = a.copy()`.

```
>>> a = [19.4, 21.0]
>>> b = a
>>> b.append(23.5)
>>> a
[19.4, 21.0, 23.5]
>>> c = a.copy()
>>> c.append(99)
>>> a
[19.4, 21.0, 23.5]
```

Bij getallen en teksten merk je dit nooit, omdat je die niet kan veranderen: `x = x + 1` maakt een nieuwe waarde.



Links wijzen a en b naar dezelfde lijst; rechts heeft c een eigen kopie.

NASLAG · 28.23

Functies, lijsten, woordenboeken, CSV en matplotlib

23/24

De schrijfwijzen uit dit hoofdstuk op één blad.

schrijfwijze	betekenis
`def naam(a, b=2):`	functie met een verplichte parameter a en een parameter b met standaardwaarde 2
`"""Uitleg."""`	docstring: eerste regel in de functie, zegt wat ze doet
`return waarde`	geeft de waarde terug en stopt de functie; zonder return krijg je None
`x = naam(5)`, `naam(5, b=3)`	aanroepen, met argumenten op volgorde of met hun naam
`return a, b` en `x, y = naam(...)`	twee waarden teruggeven en uitpakken

schrijfwijze	betekenis	voorbeeld met maxima	uitkomst
`l[i]`	element met index i (vanaf 0)	`maxima[2]`	`21.1`
`l[-1]`	laatste element	`maxima[-1]`	`25.1`
`l[a:b]`	stuk van index a tot (niet met) b	`maxima[1:4]`	`[25.1, 21.1, 22.8]`
`len(l)`	aantal elementen	`len(maxima)`	`7`

schrijfwijze	betekenis	voorbeeld met maxima	uitkomst
`sum(l)`, `min(l)`, `max(l)`	som, kleinste, grootste	`max(maxima)`	`28.6`
`sorted(l)`	nieuwe, gesorteerde lijst	`sorted(maxima)[:3]`	`[21.1, 22.8, 25.1]`
`l.index(x)`	index van de eerste x	`maxima.index(25.1)`	`1`
`l.count(x)`	hoe vaak x voorkomt	`maxima.count(25.1)`	`2`
`x in l`	zit x in de lijst?	`30.0 in maxima`	`False`
`[f(x) for x in l if ...]`	nieuwe lijst uit een oude	`[t for t in maxima if t > 26]`	`[28.1, 27.4, 28.6]`
`l.append(x)`	x achteraan toevoegen	verandert de lijst zelf	
`l.copy()`	een echte kopie	zonder copy() krijg je een tweede naam voor dezelfde lijst	

schrijfwijze	betekenis
`d = {"A0": "NTC", "A1": "LDR"}`	woordenboek met sleutels en waarden
`d["A0"]`	waarde opzoeken (KeyError als de sleutel ontbreekt)
`d["D9"] = "pomp"`	toevoegen of vervangen
`d.get("A5", "vrij")`	opzoeken met een standaardwaarde
`"A5" in d`	bestaat de sleutel?
`for k, w in d.items():`	over alle paren lopen
`d.keys()`, `d.values()`, `len(d)`	sleutels, waarden, aantal paren

schrijfwijze	betekenis
`import csv`	de csv-module laden
`with open(naam, newline="", encoding="utf-8") as f:`	bestand openen; het sluit vanzelf na het blok
`csv.DictReader(f)`	elke rij als woordenboek met de kop als sleutels
`csv.reader(f)`	elke rij als lijst van teksten
`delimiter=";"`, `encoding="utf-8-sig"`	voor een CSV uit een Nederlandstalig rekenblad
`import matplotlib.pyplot as plt`	matplotlib laden (eenmalig installeren: `pip install matplotlib`)
`plt.plot(x, y, marker="o", label="...")`	lijngrafiek met punten
`plt.bar(x, hoogte, width=0.4)`, `plt.scatter(x, y)`	staafdiagram, puntenwolk
`plt.xlabel()`, `plt.ylabel()`, `plt.title()`	asopschriften (grootheid en eenheid) en titel
`plt.legend()`, `plt.grid(True)`, `plt.axhline(26)`	legende, raster, horizontale hulplijn
`plt.savefig("naam.png", dpi=150)`, `plt.show()`	bewaren als afbeelding, tonen in een venster

HET GEHEEL · 28.24

Waar data in Python vandaan komt en waar ze heen gaat

24/24

Met functies, lijsten en bestanden kan je programma omgaan met echte hoeveelheden data. Dat verbindt programmeren met onderzoeken en meten (een meting is pas iets waard als je ze betrouwbaar samenvat) en met data en AI (een model leert uit precies zulke tabellen). Wiskundig oefen je gemiddelden, sommen en verbanden; technisch lees je het energiebudget en de klimaatregeling af uit echte cijfers.

F4 Tabellen, grafieken en verbanden

Tabel en grafiek: grootheid, eenheid, schaal. matplotlib volgt dezelfde regels.

I1 Meetfouten en nauwkeurigheid

Gemiddelde, bereik en uitschieters; in code voor honderden metingen tegelijk.

I3 Verbanden in data

Een verband in data zoeken; het rekenblad dat je hier door code vervangt.

I20 Zonne-energie dimensioneren

Het energiebudget en de PVGIS-waarden achter budget_maanden.py.

I25 Van analoog naar digitaal

De ruwe ADC-waarden die vocht_procent() omrekent, en het dimmen van de lampen.

I27 Python: de basis

Variabelen, keuzes, lussen en foutmeldingen: de basis onder dit hoofdstuk.

I30 Datalogging en communicatie

De datalogger die zulke CSV-bestanden schrijft, en ze in Python visualiseren.

I39 Data visualiseren en eerste stappen in AI

De juiste grafiek kiezen, misleidende grafieken, en correlatie tegenover oorzaak.

E6 Statistiek en meetonzekerheid

Standaardafwijking, betrouwbaarheidsintervallen en toetsen op zulke data.

E35 Objectgericht programmeren en softwareontwerp

Functies en data samen in klassen, en functies testen met unit tests.

E36 Numerieke methoden en wetenschappelijk rekenen

numpy rekent met hele lijsten tegelijk: sneller en korter dan een lus.

Habitat

De klimaatmodule heeft nu een analyseprogramma voor haar metingen: inlezen, per dag samenvatten, warme uren opsporen en grafieken maken. In de voorbeeldweek van april verbruikte ze 14,6 kWh en leverden de panelen 36,5 kWh; alleen de regendag gaf een tekort (1344 Wh). Het dimmen van de lampen scheelt op een zonnige dag ruim een kilowattuur. In hoofdstuk 30 komt de echte datalogger die dit bestand schrijft; opdracht 28.12 is de analyse voor je portfolio.

SAMENVATTING

In het kort

- `def naam(parameters):` maakt een functie; de docstring zegt wat ze doet, `return` geeft een waarde terug. Zonder return krijg je None.
- print toont, return geeft terug: rekenfuncties geven terug, het hoofdprogramma toont. Standaardwaarden maken een argument optioneel.
- Een lijst `[...]` heeft indexen vanaf 0; `[-1]` is het laatste element, `l[a:b]` een stuk zonder index b. append voegt toe.
- len, sum, min, max en sorted rekenen op een lijst; het gemiddelde is sum / len. Een maximum zoek je zelf met een voorlopige beste.
- Een lus over een lijst hoopt op, telt, filtert of maakt een nieuwe lijst; zip loopt over twee lijsten tegelijk.
- Een woordenboek `{sleutel: waarde}` zoekt op met een naam en verzamelt metingen per groep.
- CSV is een tabel als tekst. csv.DictReader geeft elke rij als woordenboek met teksten: omzetten, en lege cellen overslaan.
- matplotlib tekent met plot, bar en scatter; elke as krijgt een grootheid en een eenheid, elke reeks een label.
- b = a kopieert een lijst niet: gebruik a.copy().

BEGRIPPEN

Wat je nu kent

functie
Een stuk code met een naam dat je kan aanroepen; het krijgt argumenten mee en geeft met return een waarde terug.

parameter
De naam in de functiedefinitie die bij een aanroep de waarde van een argument krijgt, zoals vermogen in energie(vermogen, uren).

returnwaarde
Wat een functie met return teruggeeft aan wie haar aanriep; zonder return is dat None.

docstring
De tekst tussen drie aanhalingstekens op de eerste regel van een functie, die zegt wat ze doet.

lijst
Een geordende reeks waarden onder één naam, tussen vierkante haken; het eerste element heeft index 0.

slicing
Een stuk uit een lijst of tekst nemen met l[a:b]: van index a tot, maar niet met, index b.

woordenboek (dictionary)
Een verzameling paren sleutel: waarde, waarin je een waarde opzoekt met haar sleutel.

CSV
Comma-separated values: een tabel als tekstbestand, één rij per regel, de kolommen gescheiden door een komma.

ontbrekende waarde
Een lege cel in een databestand: de meting ontbreekt en mag niet als nul meetellen.

matplotlib
Een Python-bibliotheek om grafieken te tekenen: lijnen, staven, puntenwolken.

I INTERMEDIATE · DEEL I7 · PROGRAMMEREN

HANDBOEK · HOOFDSTUK 29

Microcontrollers programmeren in C++

Van losse schetsen naar één programma dat nooit stilstaat: de sturing van de klimaatmodule

In Fundamental liet je een led knipperen en gaf de meetkas water met een paar regels code. In hoofdstuk 22 tot 26 schreef je voor elk onderdeel van de klimaatmodule een eigen schets: de pomp, de knop, de lampen, het dak. Nu moeten ze samen op één Arduino Mega draaien, dag en nacht, maandenlang. Daarvoor moet je weten hoe groot een getal op een microcontroller mag worden, hoe je meerdere taken tegelijk laat lopen zonder `delay()`, hoe je met de seriële monitor in je programma kijkt, en hoe je code over bestanden en bibliotheken verdeelt. Aan het eind staat de volledige sturing van de klimaatmodule, stap voor stap opgebouwd en getest.

7× uitleg · 3× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 3× naslag · 1× verhaal · 1× het geheel · 6× code · ± 9 lestijden

P Programmeren en computationeel denken · L Elektriciteit en elektronica · R Regelen en robotica

De Mega in de kast van de klimaatmodule: één programma voor sensoren, pomp, lampen en dak.

Na dit hoofdstuk

- Je kan uitleggen wat setup() en loop() doen, wat er bij het compileren en uploaden gebeurt en welk geheugen (flash, SRAM, EEPROM) waarvoor dient.
- Je kan voor elke grootheid het juiste C++-type kiezen (bool, byte, int, unsigned int, long, unsigned long, float) en weet dat een int op de Mega 16 bit is.
- Je kan overloop, gehele deling en te kleine types herkennen, voorspellen wat er gebeurt en de code verbeteren (L, UL, 5.0).
- Je kan functies schrijven met parameters en een returnwaarde, en het verschil uitleggen tussen globale, lokale en static variabelen.
- Je kan de in- en uitgangen van de klimaatmodule instellen en gebruiken met pinMode, digitalRead, digitalWrite, analogRead en analogWrite.
- Je kan met Serial.print en de seriële plotter tussenresultaten zichtbaar maken en zo een fout vinden.
- Je kan meerdere taken tegelijk laten lopen met millis() in plaats van delay(), en uitleggen waarom nu − vorige ≥ interval ook na de overloop van 49,7 dagen werkt.
- Je kan een bibliotheek installeren en gebruiken, en een eigen .h-bestand maken.
- Je kan een groter programma in stappen opbouwen en elke stap testen.

VERHAAL · 29.1

248 dagen: een teller in een vliegtuig

Op 1 mei 2015 publiceerde de Amerikaanse luchtvaartautoriteit FAA een dwingend voorschrift voor de Boeing 787. Een toestel dat 248 dagen onafgebroken onder stroom stond, kon **alle** wisselstroom verliezen: de regeleenheden van de generatoren zouden allemaal tegelijk in hun veilige stand springen. De oorzaak volgens het voorschrift: een softwareteller in die regelaars die na 248 dagen **overloopt**. Tot er nieuwe software was, moesten maatschappijen elk toestel geregeld volledig uitschakelen.

248 dagen is een verdacht getal. Een teller van 32 bits met een teken telt tot $2^{31} - 1 = 2\,147\,483\,647$. Telt hij honderdsten van een seconde, dan is hij vol na $2^{31}/100\,\text{s} = 21\,474\,836\,\text{s}$, en dat is 248,55 dagen. De FAA gaf geen details over de code, maar het patroon herkent elke programmeur: een getal in een vak dat te klein is voor de tijd dat het systeem draait.

De klimaatmodule heeft precies zo'n teller. De functie `millis()` telt de milliseconden sinds de Mega aanging, in een vak van 32 bits zonder teken. Na $2^{32}\,\text{ms}$, ongeveer **49,7 dagen**, begint hij weer bij 0. Een kasmodule draait veel langer dan 49,7 dagen. In dit hoofdstuk leer je waarom de code van de module die overgang toch zonder haperen doorstaat, en waarom een kleine vergissing met types hetzelfde probleem al na 33 seconden zou geven.

Waarom dit verhaal ertoe doet

Een microcontroller is geen computer met een scherm die je af en toe herstart. Hij draait maandenlang alleen, en een fout die pas na dagen opduikt, vind je niet met één test. Daarom leer je hier rekenen met de grenzen van elk type, vóór je het programma schrijft.

Een vliegtuig aan de grondstroom: ook als het niet vliegt, draaien de computers aan boord door.

UITLEG · 29.2

Wat je al weet, en wat er nu verandert

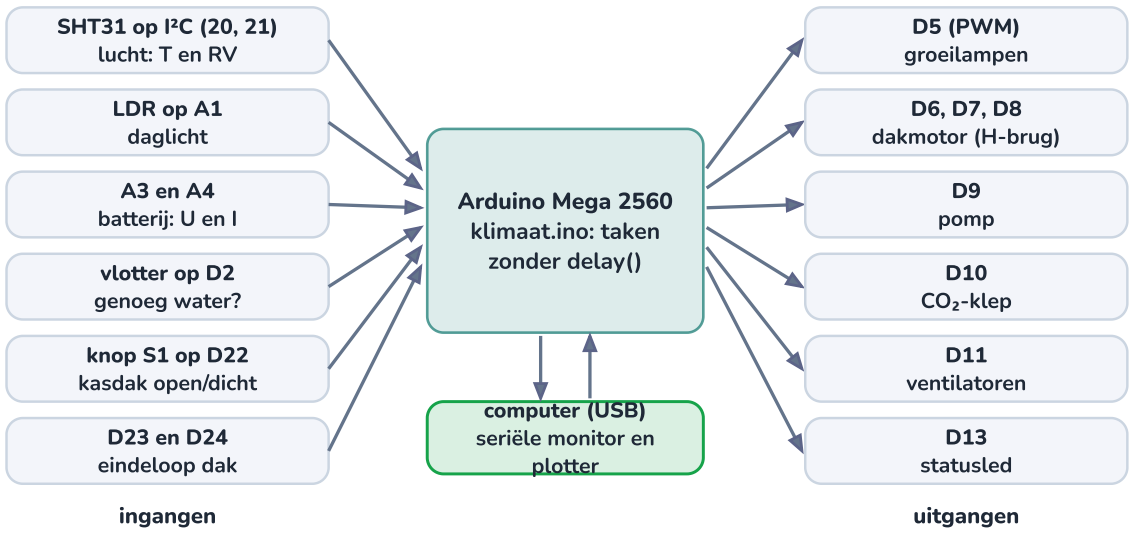
Je hebt al heel wat C++ geschreven, alleen nooit alles tegelijk:

wat je al kan	waar	wat er nu bij komt
setup() en loop(), pinMode, digitalWrite, digitalRead, delay()	F24	wat er onder setup() en loop() gebeurt, en waarom delay() in een groot programma niet meer kan
een sensor lezen met analogRead en een drempel	F25, hoofdstuk 22	alle sensoren van de module in één meettaak
losse schetsen: pomp (23), knop (24), lampen (25), dak (26), laadtoestand (16)	hoofdstuk 16, 22-26	één programma waarin ze tegelijk werken
variabelen, types, functies en lijsten in Python	hoofdstuk 27-28	types met een vaste grootte, accolades en puntkomma's
binaire getallen, bits en bytes	hoofdstuk 25	waarom een int op de Mega na 32 767 omslaat

Elke schets uit hoofdstuk 22 tot 26 werkte op zich. Maar zet je ze achter elkaar in één `loop()`, dan gaat het mis: `water_geven.ino` uit F25 wacht met `delay(900000UL)` een kwartier. In dat kwartier leest de Mega geen knop, geen vlotter en geen eindeloopschakelaar. Het dak zou tegen zijn eindpunt blijven duwen tot de zekering smelt.

Het blokschema toont wat de Mega van de klimaatmodule allemaal tegelijk moet doen. Links de **ingangen** (sensoren en schakelaars), rechts de **uitgangen** (actuatoren). In het midden draait één programma dat elke taak op tijd een beurt geeft. Onderaan de USB-kabel naar de computer: daarlangs laad je het programma op en kijk je mee met de seriële monitor.

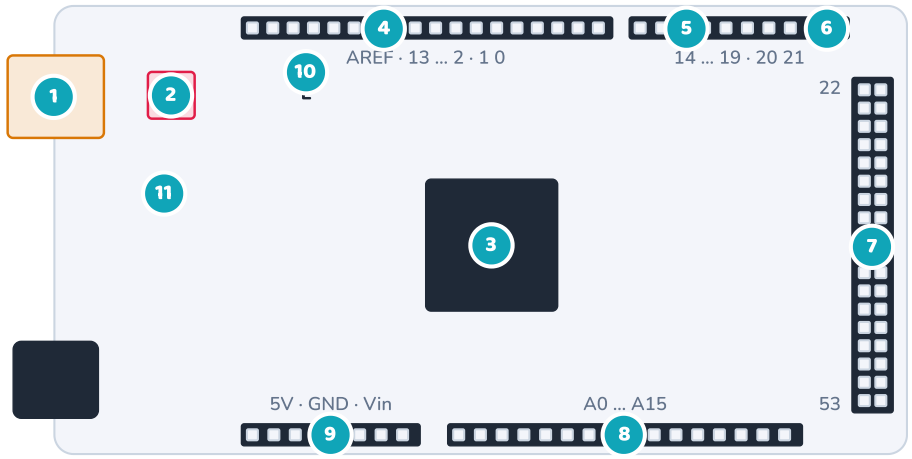
> **Leesadvies**
> Heb je hoofdstuk 22 tot 26 niet gedaan, dan volstaat de pintabel verderop. De natuurkunde achter elke sensor en actuator staat in die hoofdstukken; hier gaat het om het programma.



Zes groepen ingangen en zes groepen uitgangen, één programma. De pijlen tonen de richting van de informatie.

De Arduino Mega 2560

De meetkas heeft genoeg aan een Arduino Uno. De klimaatmodule heeft meer pinnen nodig: zeven analoge sensoren, een knop, eindeloopschakelaars, zes actuatoren en een I²C-bus. De **Mega 2560** heeft er 70 (54 digitale, waarvan 15 met PWM, en 16 analoge), vier seriële poorten en vier keer zoveel werkgeheugen als de Uno. Programmeren gaat precies zoals op de Uno.



Vereenvoudigde bovenkant van de Mega 2560 (101,6 × 53,3 mm): de pinrijen en de belangrijkste onderdelen.

1 USB-aansluiting
Voeding (5 V) en de verbinding met de computer: hierlangs laad je programma's op en praat de Mega met de seriële monitor.

2 USB-chip (ATmega16U2)
Zet USB om naar een seriële verbinding (TX/RX) voor de hoofdchip. Op de computer verschijnt de Mega als COM-poort.

3 ATmega2560
De microcontroller: 16 MHz, 256 kB flash voor het programma, 8 kB SRAM voor variabelen, 4 kB EEPROM. Alles wat je schrijft, draait hier.

4 Pinnen 2 tot 13: PWM
Digitale pinnen die ook analogWrite kunnen. De module gebruikt D2 (vlotter), D5 (lampen), D6 tot D11 (dak, pomp, CO_2, ventilatoren) en D13.

5 14 tot 19: seriële poorten 1 tot 3
Drie extra seriële poorten (TX/RX), naast poort 0 op pin 0 en 1 die met de USB-chip verbonden is.

6 20 en 21: I²C
SDA en SCL voor de I²C-bus, met 10 kΩ pull-ups op het bordje. Hier hangt de SHT31 voor temperatuur en vocht (hoofdstuk 30).

7 22 tot 53
Extra digitale pinnen: D22 (knop S1), D23 en D24 (eindeloop). Pin 50 tot 53 zijn ook de SPI-bus voor de SD-kaart (hoofdstuk 30).

8 A0 tot A15: analoge ingangen
Zestien ingangen voor de 10-bits ADC. A0 tot A6 zijn bezet: NTC, LDR, pH, batterij, stroom, druk, EC.

9 Voedingspinnen
5V en GND voor de sensoren, Vin voor een voeding van 7 tot 12 V. In de module komt de 5 V van een omvormer (hoofdstuk 21).

10 Ledje L
Hangt aan pin 13. De statusled van de module zit op dezelfde pin: je ziet hem ook op het bordje knipperen.

11 Resetknop
Start het programma opnieuw: setup() loopt weer en millis() begint bij 0. Globale variabelen krijgen hun beginwaarde terug.

UITLEG · 29.4

Hoe een schets draait: van tekst tot stroompjes

Wat je in de Arduino-IDE typt, is C++: tekst voor mensen. Bij **Verifiëren** vertaalt de **compiler** (avr-gcc) die tekst naar machinecode voor de ATmega2560. Bij **Uploaden** gaat die code via USB naar het **flashgeheugen**. Daar zorgt een klein programmaatje, de **bootloader**, dat de nieuwe code binnenkomt; daarna springt de Mega naar jouw programma.

Een C++-programma begint altijd in een functie `main()`. Die schrijf je bij Arduino niet zelf: de IDE voegt ze toe, en ze doet ongeveer dit:

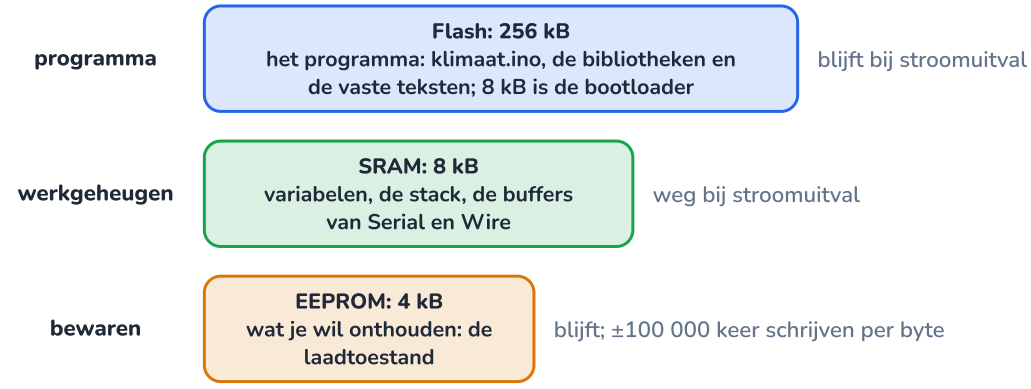
```
int main() {
  init();           // klokken, timers en de ADC klaarzetten
  setup();          // jouw setup(): één keer
  for (;;) {        // een lus zonder einde
    loop();         // jouw loop(): opnieuw en opnieuw
  }
}
```

Na `loop()` komt dus meteen de volgende `loop()`, duizenden keren per seconde als er niets te doen is. Wat tussen twee rondes bewaard moet blijven, zet je in een **globale** variabele; een variabele die je binnen `loop()` maakt, is na elke ronde weg.

De ATmega2560 heeft drie soorten geheugen. Het programma staat in **flash** en blijft daar zonder stroom. De variabelen staan in het **SRAM**, 8 kB, en zijn weg bij een stroomonderbreking of een reset. Wat de module moet onthouden, zoals de laadtoestand van de batterij, gaat in het **EEPROM**: dat houdt zijn inhoud, maar elke byte verslijt na ongeveer 100 000 keer schrijven. Na het compileren meldt de IDE hoeveel flash en SRAM je programma gebruikt; hou minstens een paar honderd bytes SRAM vrij voor de stack, waar de lokale variabelen van functies komen.

Teksten in flash met F()

Elke tekst tussen aanhalingstekens, zoals `"pomp aan"`, wordt bij de start naar het SRAM gekopieerd. Met `Serial.println(F("pomp aan"))` blijft ze in het flashgeheugen. Op de Mega met 8 kB valt dat in dit programma nog mee; op een Uno met 2 kB is het vaak het verschil tussen werken en vastlopen.



Flash voor het programma, SRAM voor wat verandert, EEPROM voor wat moet blijven. De balken zijn niet op schaal.

Types in C++ op de Mega

In Python maakt een variabele zelf uit hoe groot ze wordt. In C++ kies jij het **type**, en dat legt vast hoeveel bytes ze krijgt. Met n bits kan je 2^n verschillende waarden maken (hoofdstuk 25). Een type met teken gebruikt de helft voor negatieve getallen. Let op: een `int` is op de Mega **16 bit**, op je computer 32 bit, en `double` is op de Mega gewoon een `float`.

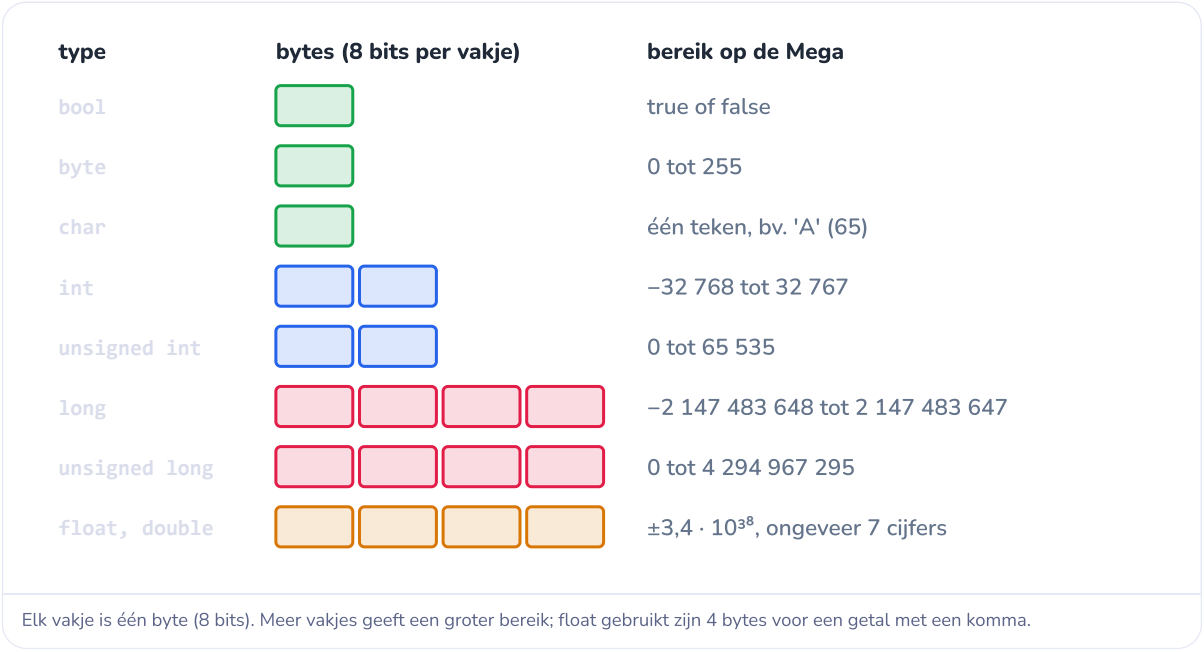
bereik van een geheel type met n bits

$$\text{zonder teken: } 0 \dots 2^n - 1 \qquad \text{met teken: } -2^{n-1} \dots 2^{n-1} - 1$$

symbool	betekenis	eenheid
n	aantal bits (8 per byte)	

type	bytes	bereik	gebruik in de module
`bool`	1	true, false	pompAan, dakOpen, genoegWater
`byte` (= `uint8_t`)	1	0 tot 255	een I²C-adres, een byte van een sensor
`int`	2	−32 768 tot 32 767	ADC-waarden (0-1023), PWM (0-255), pinnummers
`unsigned int`	2	0 tot 65 535	tellers die niet negatief worden
`long`	4	±2 147 483 647	een som van 100 ADC-waarden
`unsigned long`	4	0 tot 4 294 967 295	elke tijd in ms: millis(), vorige, intervallen
`float` (`double`)	4	±3,4 · 10³⁸, ±7 cijfers	spanning, temperatuur, laadtoestand

Constanten. `const unsigned long MEET_MS = 2000;` is een variabele die niet mag veranderen: de compiler kent haar type en controleert het. `#define MEET_MS 2000` is een tekstvervanging vóór het compileren, zonder type. Kies `const`, behalve voor dingen die alleen met `#define` kunnen (zoals de beveiliging van een headerbestand). Getallen kunnen een achtervoegsel krijgen: `60000UL` is een unsigned long, `100L` een long, `5.0` een kommagetal.



Elk vakje is één byte (8 bits). Meer vakjes geeft een groter bereik; float gebruikt zijn 4 bytes voor een getal met een komma.

CODE · 29.6

Wat past er in welk type?

6/23

Dit programma draait één keer in `setup()` en toont wat er gebeurt aan de grenzen van de types. `sizeof` geeft het aantal bytes van een type.

C++ (Arduino) **types.ino** 45 regels

```cpp
// types.ino - wat past er in welk type? Draait één keer in setup().
void setup() {
  Serial.begin(115200);

  Serial.print("bytes in int: ");
  Serial.println(sizeof(int));          // 2 op de Mega (4 op je computer)
  Serial.print("bytes in long: ");
  Serial.println(sizeof(long));
  Serial.print("bytes in float: ");
  Serial.println(sizeof(float));
  Serial.print("bytes in double: ");
  Serial.println(sizeof(double));       // op de Mega is double gewoon een float

  int a = 32767;                        // de grootste int
  a = a + 1;
  Serial.print("32767 + 1 = ");
  Serial.println(a);                    // overloop: de teller springt naar de kleinste int

  byte b = 250;
  b = b + 10;
  Serial.print("byte 250 + 10 = ");
  Serial.println(b);                    // 260 past niet in 8 bits: 260 - 256

  int wacht = 60 * 1000;                // bedoeld: een minuut in ms
  Serial.print("int wacht = ");
  Serial.println(wacht);
  long goed = 60L * 1000;               // L: reken in long
  Serial.print("long goed = ");
  Serial.println(goed);

  Serial.print("7 / 2 = ");
  Serial.println(7 / 2);                // twee gehele getallen: gehele deling
  Serial.print("7 / 2.0 = ");
  Serial.println(7 / 2.0);
  Serial.print("7 % 2 = ");
  Serial.println(7 % 2);                // de rest

  float f = 16777216.0;                 // 2 tot de 24e
  f = f + 1.0;
  Serial.print("float 16777216 + 1 = ");
  Serial.println(f, 1);                 // een float onthoudt maar ongeveer 7 cijfers
}

void loop() {
}
```

Uitvoer op een Mega (seriële monitor op 115200 baud):

```
bytes in int: 2
bytes in long: 4
bytes in float: 4
bytes in double: 4
32767 + 1 = -32768
byte 250 + 10 = 4
int wacht = -5536
long goed = 60000
```

```
7 / 2 = 3
7 / 2.0 = 3.50
7 % 2 = 1
float 16777216 + 1 = 16777216.0
```

Regel voor regel: `32767 + 1` past niet meer in 16 bits met teken; de bits lopen over en het getal springt naar de kleinste int, −32 768. Een `byte` gaat van 255 naar 0: $260 - 256 = 4$. `60 * 1000` rekent de Mega in int, want beide getallen zijn int: $60\,000 - 65\,536 = -5536$. Met `60L` rekent hij in long. `7 / 2` is een **gehele deling**: de rest (`7 % 2` = 1) valt weg. En een float onthoudt ongeveer zeven cijfers: bij 16 777 216 (2^{24}) is een stap van 1 al te fijn.

> **"De compiler zou me wel waarschuwen"**
>
> Voor `60 * 1000` geeft de compiler de waarschuwing *integer overflow in expression*, maar de Arduino-IDE toont standaard geen waarschuwingen. Zet in de voorkeuren *Compilerwaarschuwingen* op *Alle*. Voor `a = a + 1` komt er nooit een waarschuwing: de compiler weet niet welke waarde a later krijgt.

UITGEWERKT VOORBEELD · 29.7

Een uur in milliseconden

7/23

De pomp draait 10 minuten per uur. Een leerling schrijft `const unsigned long CYCLUS_MS = 60 * 60 * 1000;` en ziet de pomp nooit meer aanslaan na de eerste keer. Wat staat er in CYCLUS_MS, en hoe schrijf je het juist?

GEGEVEN

- `int` op de Mega: 16 bit, van −32 768 tot 32 767
- `unsigned long`: 32 bit, van 0 tot 4 294 967 295

GEVRAAGD

- de waarde van CYCLUS_MS
- een juiste schrijfwijze

OPLOSSING

1. **In welk type rekent de Mega?** Het type links van `=` telt pas bij het bewaren. De berekening zelf gebeurt in het type van de getallen: 60, 60 en 1000 zijn allemaal int, dus ook de tussenresultaten.

2. **Stap voor stap.** $60 \cdot 60 = 3600$ past in een int. $3600 \cdot 1000 = 3\,600\,000$ niet: van dat getal blijven alleen de laagste 16 bits over, $3\,600\,000 - 54 \cdot 65\,536 = 61\,056$, en die bits lees je als int met teken: $61\,056 - 65\,536 = -4480$.

3. **Bewaren in unsigned long.** −4480 zonder teken in 32 bits wordt $2^{32} - 4480 = 4\,294\,962\,816$. Dat is 49,7 dagen: de pomp wacht dus zeven weken op haar volgende beurt.

4. **Narekenen met Python** (dat zelf nooit overloopt, dus je bootst de 16 bits na):

```python
# uur_in_ms.py - 60 * 60 * 1000 met gehele getallen van 16 bit, zoals een int op de Mega
def als_int16(x):
    """Houd alleen de laagste 16 bits over en lees ze als getal met teken."""
    x = x % 2**16
    return x - 2**16 if x >= 2**15 else x

stap1 = als_int16(60 * 60)          # 3600: past nog
stap2 = als_int16(stap1 * 1000)     # 3 600 000 past niet in 16 bits
print(stap1, stap2)
print("in unsigned long:", stap2 % 2**32)
```

```
3600 -4480
in unsigned long: 4294962816
```

5. **Juist schrijven.** Maak het eerste getal al een unsigned long, dan rekent de Mega alles in 32 bits: `60UL * 60 * 1000` = 3 600 000.

ANTWOORD

CYCLUS_MS = 4 294 962 816 in plaats van 3 600 000. Schrijf `60UL * 60 * 1000`.

Klopt dit? Past het antwoord? $3\,600\,000 < 4\,294\,967\,295$, dus in een unsigned long is er ruimte genoeg. Vuistregel: zodra een tijd in ms boven 32 767 kan komen (33 seconden), zet je UL achter het eerste getal.

UITLEG · 29.8

Functies en variabelen: waar leeft een waarde?

Een groot programma schrijf je als een reeks **functies**, elk met één taak. Een functie in C++ vermeldt eerst het type van wat ze teruggeeft, dan haar naam en haar parameters met hun type:

```
float batterijspanning() {                          // geeft een float terug, geen parameters
  return gemiddeld(BATTERIJ) * 5.0 / 1024.0 * (133.0 / 33.0);
}

int gemiddeld(int pin) {                            // krijgt een pin, geeft een int terug
  long som = 0;
  for (int i = 0; i < 16; i++) som += analogRead(pin);
  return som / 16;
}

void printTwee(unsigned long x) {                   // void: geeft niets terug
  if (x < 10) Serial.print('0');
  Serial.print(x);
}
```

Een parameter is een **kopie**: verandert de functie `pin`, dan merkt de aanroeper daar niets van. Waar een variabele gemaakt wordt, bepaalt hoe lang ze bestaat:

- **Globaal**, bovenaan buiten elke functie: bestaat zolang de Mega draait, elke functie kan erbij. Tijden als `vorigeMeting` en toestanden als `pompAan` moeten globaal zijn, want ze moeten van de ene ronde van `loop()` naar de volgende blijven.
- **Lokaal**, binnen een functie: bestaat alleen tijdens die aanroep. `som` hierboven begint elke keer opnieuw bij 0, en dat is precies wat je wil.
- **static lokaal**: alleen in die functie zichtbaar, maar de waarde blijft bewaard. Handig voor een taak die haar eigen `vorige` bijhoudt, zonder dat de rest van het programma eraan kan.

Globale variabelen zijn handig, maar gevaarlijk in een groot programma: elke functie kan ze per ongeluk veranderen. Hou ze beperkt tot wat echt gedeeld moet worden, en geef ze duidelijke namen.

globaal: bovenaan, buiten elke functie

bestaat zolang de Mega draait; elke functie kan erbij

```
unsigned long vorigeMeting; float soc; int stand;
```

`taakMeten(unsigned long nu)`

parameter nu: een kopie

lokaal: int n (weg na de return)

`loop()`

lokaal: unsigned long nu

(elke ronde opnieuw gemaakt)

`taakNiveau(unsigned long nu)`

```
static unsigned long vorige = 0;
```

static: alleen hier zichtbaar, maar de waarde blijft tussen aanroepen

Een globale variabele leeft zo lang als het programma, een lokale zo lang als één aanroep.

UITLEG · 29.9

Ingangen en uitgangen van de module

9/23

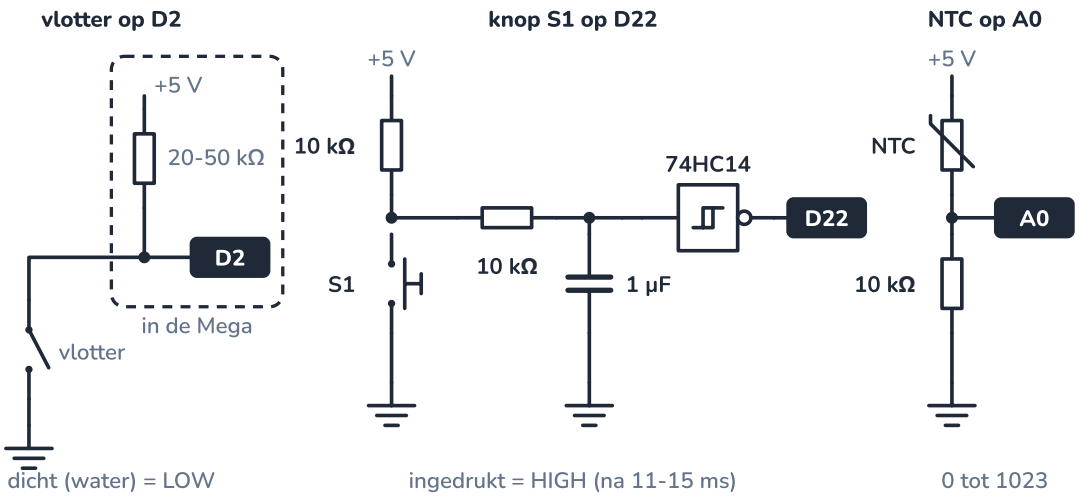
Elke pin stel je in `setup()` in met `pinMode`. Een uitgang (OUTPUT) kan stroom leveren; een ingang (INPUT) meet alleen. **INPUT_PULLUP** zet binnen in de Mega een weerstand van 20 tot 50 kΩ naar 5 V: een schakelaar naar GND geeft dan LOW als hij dicht is en HIGH als hij open is. Zonder pull-up 'zweeft' een open ingang en leest ze willekeurig HIGH of LOW.

pin	onderdeel	pinMode	lezen of schrijven	waarden
D2	vlotter (naar GND)	INPUT_PULLUP	digitalRead	LOW = genoeg water
D22	knop S1 via RC en 74HC14	INPUT	digitalRead	HIGH = ingedrukt
D23, D24	eindeloop open, dicht (verbreekcontact)	INPUT_PULLUP	digitalRead	HIGH = eindpunt (of draad los)
A0-A6	NTC, LDR, pH, batterij, stroom, druk, EC	(niet nodig)	analogRead	0 tot 1023
D5	groeilampen (MOSFET)	OUTPUT	analogWrite	0 tot 255 (PWM, 490 Hz)
D6, D7	dakmotor sluiten, openen (H-brug)	OUTPUT	analogWrite	0 tot 255
D8, D9, D10, D11	H-brug aan, pomp, CO_2-klep, ventilatoren	OUTPUT	digitalWrite	LOW of HIGH
D13	statusled (en ledje L)	OUTPUT	digitalWrite	LOW of HIGH
D20, D21	I²C: SDA en SCL	(Wire regelt het)	Wire	hoofdstuk 30

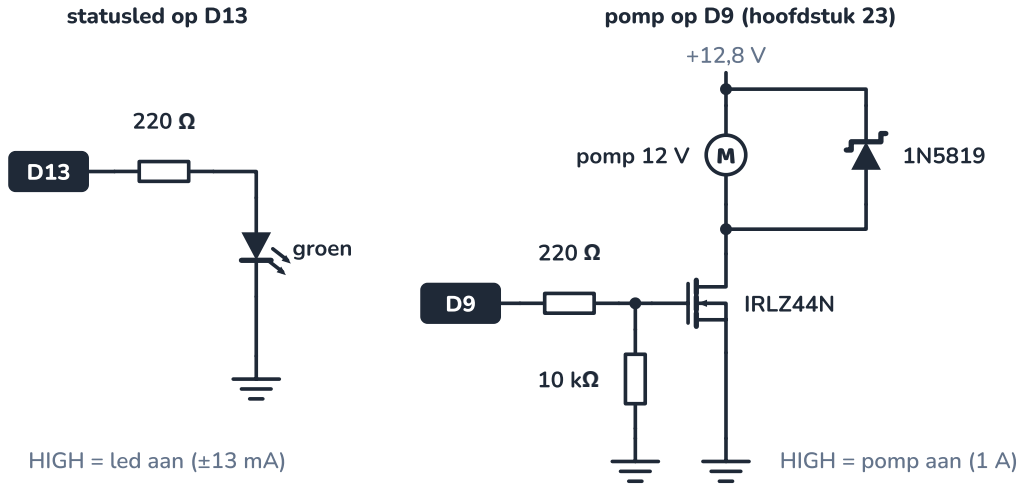
`analogRead` duurt ongeveer 0,1 ms en geeft 0 tot 1023 voor 0 tot 5 V (hoofdstuk 25). `analogWrite` werkt alleen op PWM-pinnen en maakt geen echte spanning, maar een blokgolf: op D5 490 keer per seconde, met een aandeel 'aan' van waarde/255. Voor de knop D22 staat de pin op INPUT zonder pull-up: de 74HC14 levert zelf een harde 0 of 5 V (hoofdstuk 24).

⚠ Een pin is geen voeding

Een pin van de Mega mag hoogstens 40 mA leveren (20 mA is een veilige werkwaarde). Een led met 220 Ω lukt; een pomp van 1 A niet: die schakel je met een MOSFET (hoofdstuk 23), en de pin stuurt alleen de gate.



Links een schakelaar met de interne pull-up, midden de knop met RC-filter en Schmitt-trigger, rechts een sensor als spanningsdeler.



Een led kan rechtstreeks aan een pin; voor de pomp stuurt de pin de gate van een MOSFET, de diode vangt de piek bij het uitschakelen op.

CODE · 29.10

Serial: meekijken in je programma

10/23

Een microcontroller heeft geen scherm. Met `Serial.begin(115200)` opent hij een verbinding over de USB-kabel; alles wat je met `Serial.print` stuurt, verschijnt in de **seriële monitor** van de IDE. `println` zet er een nieuwe regel achter, `print(x, 2)` toont een kommagetal met 2 decimalen. Zet de monitor rechtsonder op dezelfde snelheid (115200 baud), anders zie je rare tekens.

Het nuttigste gebruik: **debuggen**, fouten zoeken. Een leerling ziet op de monitor altijd `12.00 V` voor de batterij, ook als die volgeladen is. De code ziet er juist uit. Druk dan de tussenresultaten af:

C++ (Arduino) **batterij_debug.ino** 23 regels

```
1  // batterij_debug.ino - waarom toont de batterij altijd 12.00 V? Tussenresultaten afdrukken.
2  #include "pinnen.h"
3
```

```ino
void setup() {
  Serial.begin(115200);
}

void loop() {
  int n = analogRead(BATTERIJ);
  float fout = n * 5 / 1024 * 133 / 33;            // de oorspronkelijke regel
  float goed = n * 5.0 / 1024.0 * 133.0 / 33.0;    // met kommagetallen

  Serial.print("n = ");
  Serial.print(n);
  Serial.print("   n * 5 / 1024 = ");
  Serial.print(n * 5 / 1024);                      // debug: het tussenresultaat
  Serial.print("   fout: ");
  Serial.print(fout);
  Serial.print(" V   goed: ");
  Serial.print(goed);
  Serial.println(" V");
  delay(1000);                                     // in een testschets mag dat
}
```

Uitvoer bij vijf batterijspanningen tussen 12,05 en 14,10 V:

```
n = 612   n * 5 / 1024 = 2   fout: 8.00 V   goed: 12.04 V
n = 650   n * 5 / 1024 = 3   fout: 12.00 V   goed: 12.79 V
n = 670   n * 5 / 1024 = 3   fout: 12.00 V   goed: 13.19 V
n = 693   n * 5 / 1024 = 3   fout: 12.00 V   goed: 13.64 V
n = 716   n * 5 / 1024 = 3   fout: 12.00 V   goed: 14.09 V
```

Het tussenresultaat verraadt de fout: `n * 5 / 1024` is 3, terwijl $670 \cdot 5/1024$ ongeveer 3,27 is. Alle getallen in die berekening zijn int, dus elke deling is een gehele deling: 3,27 wordt 3, en $3 \cdot 133/33 = 12$. Pas daarna wordt het resultaat een float. Een enkele `5.0` maakt de hele berekening een kommaberekening. Deze fout is verraderlijk omdat 12,00 V *plausibel* lijkt: alleen doordat je de tussenstap ziet, valt ze op.

> **Debuggen met Serial, in drie regels**
> (1) Druk de invoer af (hier n), (2) druk elk tussenresultaat af, (3) reken één geval met de hand na. Waar hand en monitor verschillen, zit de fout. Haal de extra regels daarna weg: Serial kost tijd (hoofdstuk 30 rekent uit hoeveel).

CODE · 29.11

De seriële plotter

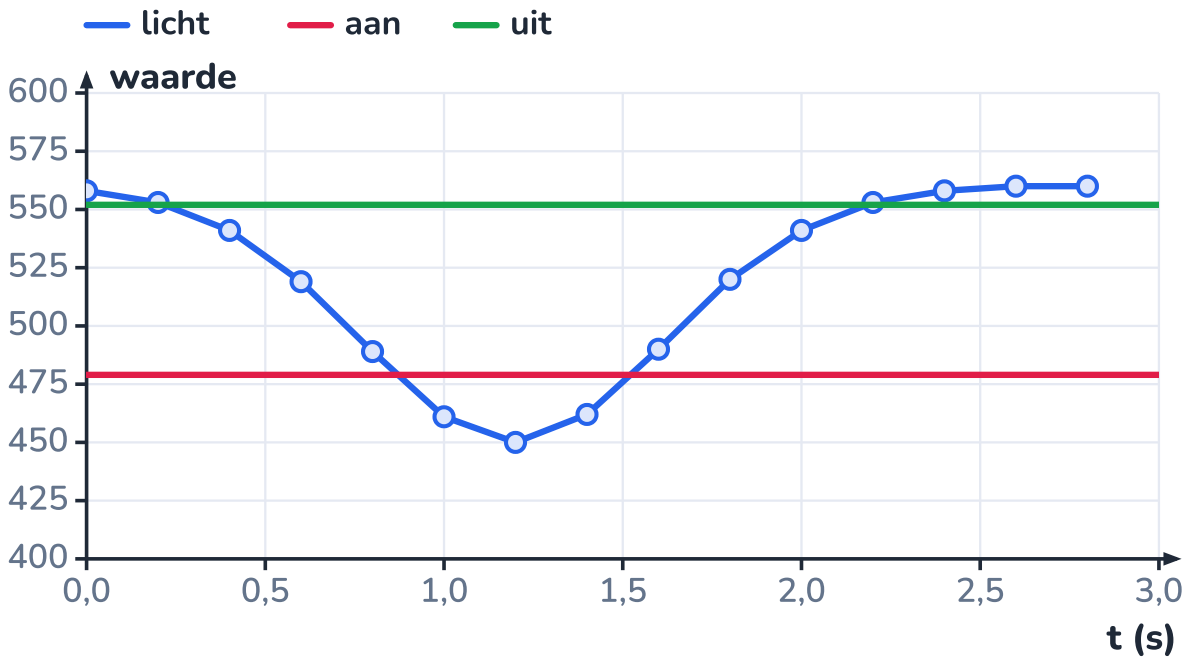
11/23

Een lange kolom getallen is moeilijk te lezen. De **seriële plotter** (menu *Tools*) tekent ze als grafiek. Stuur per regel een of meer waarden als `naam:waarde`, gescheiden door komma's; de plotter maakt voor elke naam een lijn. Hier de lichtsensor samen met de twee drempels van de lampen (hoofdstuk 22): je ziet meteen of een wolk de lampen zou aanzetten.

C++ (Arduino) **plotter.ino** 23 regels

```ino
// plotter.ino - de lichtsensor en zijn twee drempels in de seriële plotter (Tools > Serial Plotter)
#include "pinnen.h"

const int AAN = 479;       // lampen aan onder 4000 lx (hoofdstuk 22)
const int UIT = 552;       // lampen uit boven 6000 lx
unsigned long vorige = 0;

void setup() {
  Serial.begin(115200);
}

void loop() {
```

```
  unsigned long nu = millis();
  if (nu - vorige >= 200) {           // vijf punten per seconde
    vorige += 200;
    Serial.print("licht:");           // naam:waarde, gescheiden door komma's
    Serial.print(analogRead(LDR));
    Serial.print(",aan:");
    Serial.print(AAN);
    Serial.print(",uit:");
    Serial.println(UIT);
  }
}
```



De eerste drie seconden uit de simulatie: een wolk trekt voor de zon en de waarde duikt onder 'aan'.

Wat de monitor ontvangt (de plotter maakt er de grafiek hiernaast van):

```
licht:558,aan:479,uit:552
licht:553,aan:479,uit:552
licht:541,aan:479,uit:552
licht:519,aan:479,uit:552
licht:489,aan:479,uit:552
licht:461,aan:479,uit:552
```

…

Voor de monitor zijn het gewoon regels tekst; het formaat maakt er een grafiek van. Dat idee, een afgesproken formaat per regel, is de basis van de datalogger in hoofdstuk 30.

PROEF · 29.12

delay() tegenover millis(): hoeveel drukken mis je?

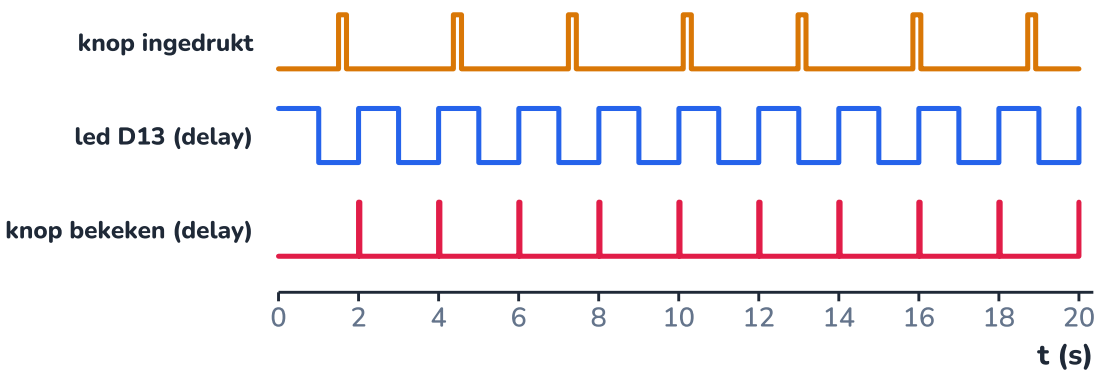
12/23

ONDERZOEKSVRAAG

Een led knippert (1 s aan, 1 s uit) en het programma telt hoe vaak je op een knop drukt. Hoeveel van twintig korte drukken worden geteld als het knipperen met `delay()` gebeurt, en hoeveel met `millis()`?

MATERIAAL

Arduino (Mega of Uno) met USB-kabel
drukknop tussen pin 22 en GND (op een Uno: pin 2)
led op pin 13 (of het ledje L op het bordje)
de schetsen knop_delay.ino en knop_millis.ino (hieronder)



De eerste 20 s van proef A: alleen een druk die over een kijkmoment valt, telt.

WERKWIJZE

1. Laad knop_delay.ino op en open de seriële monitor op 115200 baud.
2. Druk twintig keer kort op de knop (ongeveer 0,2 s), op willekeurige momenten, niet in het ritme van de led. Noteer hoeveel drukken de monitor telt.
3. Laad knop_millis.ino op en herhaal met twintig drukken.
4. Voorspel vooraf: welk deel van de tijd kijkt knop_delay.ino naar de knop?

WAARNEMING

In een simulatie met twintig drukken van 0,2 s, om de 2,87 s, telt knop_delay.ino er **2**, knop_millis.ino alle **20**:

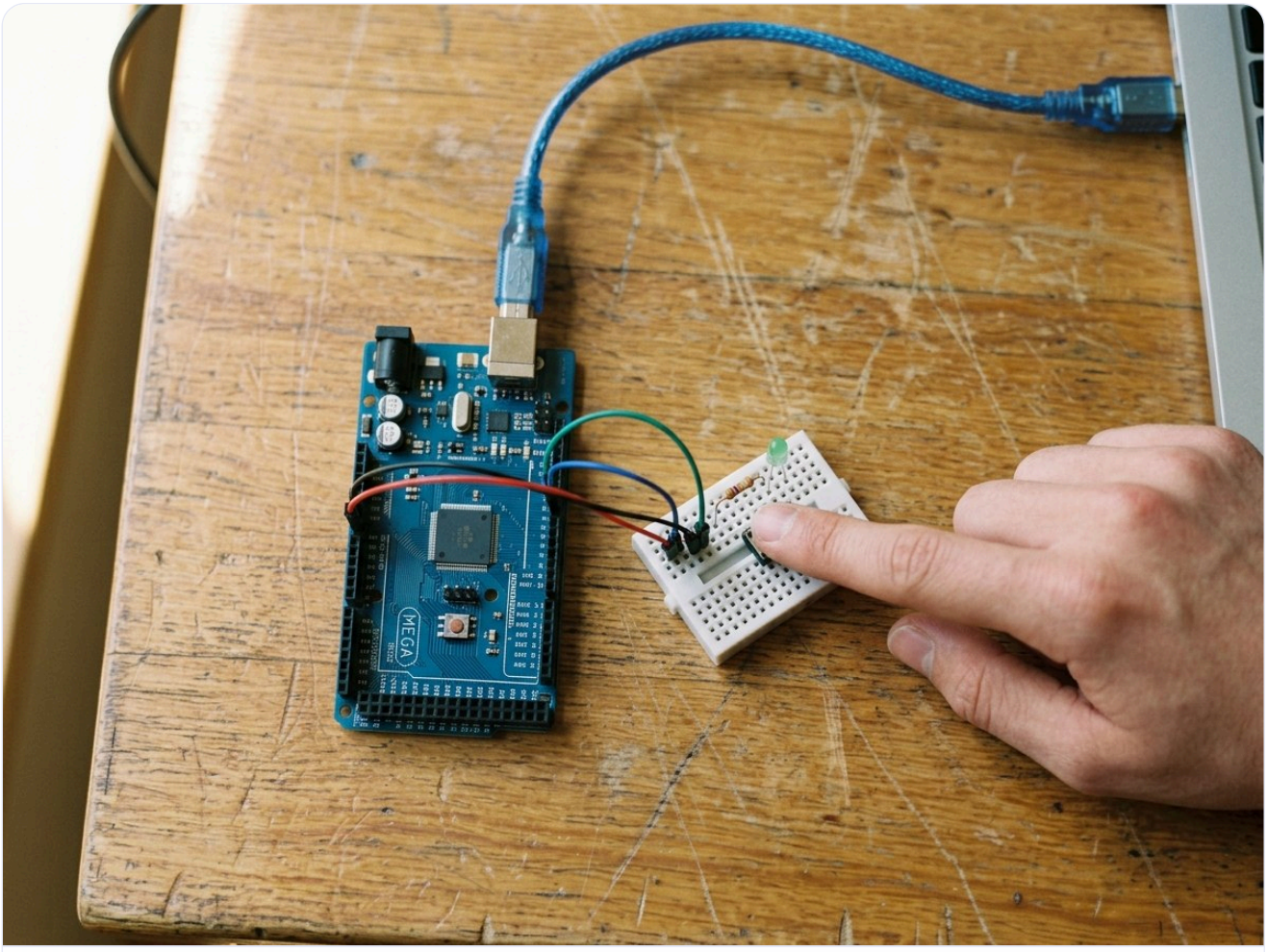
```
// knop_delay.ino - proef A: knipperen met delay() en o
ndertussen drukken tellen
const int KNOP = 22;           // drukknop naar GND (pr
oefopstelling: interne pull-up)
const int LED = 13;
int teller = 0;
int vorige = HIGH;

void setup() {
  pinMode(KNOP, INPUT_PULLUP);
  pinMode(LED, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
  int lezing = digitalRead(KNOP);         // de knop wo
rdt maar om de 2 s bekeken
  if (lezing == LOW && vorige == HIGH) {
    teller++;
    Serial.print("gedrukt: ");
    Serial.println(teller);
  }
  vorige = lezing;
}
```

```
gedrukt: 1
gedrukt: 2
-- 20 keer gedrukt
```

```
void loop() {
  unsigned long nu = millis();
  if (nu - vorigeKnipper >= 1000) {       // 1 s aan, 1
s uit
    vorigeKnipper += 1000;
    ledAan = !ledAan;
    digitalWrite(LED, ledAan ? HIGH : LOW);
  }
  int lezing = digitalRead(KNOP);         // elke rond
e, duizenden keren per seconde
  if (lezing != vorigeLezing) {
    laatsteWissel = nu;
    vorigeLezing = lezing;
  }
  if (nu - laatsteWissel >= 20 && lezing != stabiel) {
    stabiel = lezing;
    if (stabiel == LOW) {
      teller++;
      Serial.print("gedrukt: ");
      Serial.println(teller);
    }
  }
}
```

VERKLARING

knop_delay.ino kijkt maar één keer per 2 s naar de knop, na de tweede `delay(1000)`. Een druk van 0,2 s wordt alleen geteld als hij net over dat ene moment valt: de kans is $0{,}2/2 = 10\,\%$, dus gemiddeld 2 van de 20. knop_millis.ino kijkt duizenden keren per seconde en telt elke druk. Het knipperen gebeurt er tussendoor, alleen als de tijd om is. Het tijdsdiagram toont waarom proef A zo weinig ziet: de korte streepjes zijn de enige momenten waarop de knop gelezen wordt.



De opstelling van de proef: een knop, een led en de seriële monitor.

UITLEG · 29.13

millis(): kijken hoe laat het is, in plaats van wachten

13/23

`millis()` geeft het aantal milliseconden sinds de start, als `unsigned long`. Het idee is eenvoudig: wacht niet, maar kijk telkens of het tijd is. Voor elke taak onthoud je in een globale variabele wanneer ze het laatst liep:

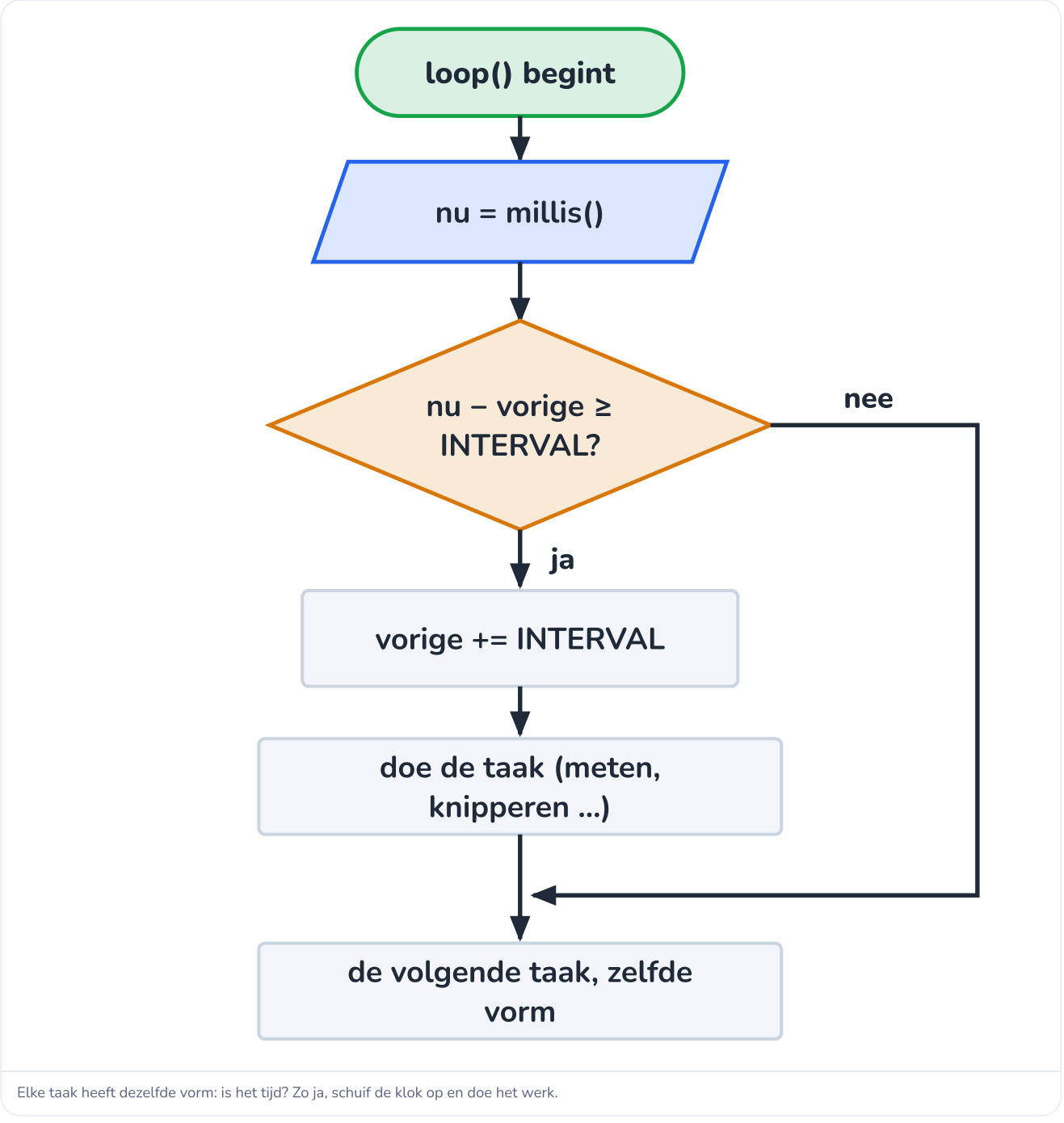
```
const unsigned long MEET_MS = 2000;
unsigned long vorigeMeting = 0;

void taakMeten(unsigned long nu) {
  if (nu - vorigeMeting < MEET_MS) return;   // nog niet aan de beurt: meteen terug
  vorigeMeting += MEET_MS;                   // de volgende beurt is precies 2 s later
  // ... meten ...
}
```

Een taak die niet aan de beurt is, kost maar een paar microseconden. Zo krijgt elke taak duizenden keren per seconde de kans, en merkt de knop een druk van een fractie van een seconde. Drie details maken het verschil:

- **Eén keer `millis()` per ronde.** `loop()` leest de klok en geeft `nu` door aan alle taken, zodat ze met dezelfde tijd rekenen.
- `vorige += INTERVAL`, **niet** `vorige = nu`**.** Komt de lus 3 ms te laat, dan blijft de volgende beurt toch op het vaste ritme; met `= nu` schuift elke kleine vertraging mee en loopt de taak langzaam achter. Voor een led maakt dat niets uit, voor een meting om de 2 s wel.

- **Altijd** `nu - vorige` **vergelijken met het interval**, nooit `nu >= vorige + INTERVAL`. Waarom, zie je bij de overloop.

Dit heet **niet-blokkerende** code: geen enkele taak houdt de lus lang bezig. Het is geen echt multitasking, want de Mega doet nog altijd één ding tegelijk, maar zo snel na elkaar dat het lijkt of alles tegelijk gebeurt.



Elke taak heeft dezelfde vorm: is het tijd? Zo ja, schuif de klok op en doe het werk.

CODE · 29.14

Vier taken tegelijk

14/23

Vier taken uit de module in één programma: meten om de 2 s, de statusled die knippert, de knop die ontdenderd wordt (hoofdstuk 24), en de pomp die 10 minuten per uur draait als er water is (hoofdstuk 23). Geen enkele `delay()`.

C++ (Arduino) **taken.ino** 76 regels

```
1  // taken.ino - vier taken tegelijk, zonder delay(): meten, statusled, knop en pomp
```

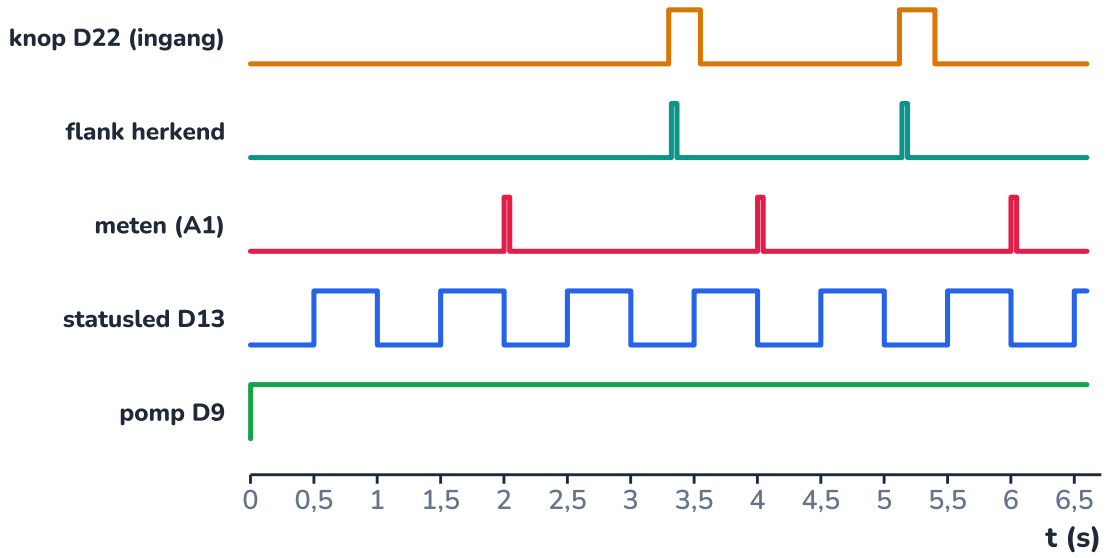
```ino
#include "pinnen.h"

const unsigned long MEET_MS = 2000;                 // elke 2 s meten
const unsigned long KNIPPER_MS = 500;               // statusled: 0,5 s aan, 0,5 s uit
const unsigned long DENDER_MS = 20;                 // zo lang moet de knop stil blijven
const unsigned long POMP_AAN_MS = 10UL * 60 * 1000;  // 10 minuten pompen ...
const unsigned long CYCLUS_MS = 60UL * 60 * 1000;    // ... per uur

unsigned long vorigeMeting = 0;      // globale variabelen: ze onthouden hun waarde
unsigned long vorigeKnipper = 0;     // tussen twee rondes van loop()
unsigned long startCyclus = 0;
unsigned long laatsteWissel = 0;
int vorigeLezing = LOW;
int stabiel = LOW;
bool ledAan = false;
int keerGedrukt = 0;

void taakMeten(unsigned long nu) {
  if (nu - vorigeMeting < MEET_MS) return;          // nog niet aan de beurt: meteen terug
  vorigeMeting += MEET_MS;
  int n = analogRead(LDR);                          // lokale variabele: bestaat alleen hier
  Serial.print(nu);
  Serial.print(" ms  LDR ");
  Serial.println(n);
}

void taakStatusled(unsigned long nu) {
  if (nu - vorigeKnipper < KNIPPER_MS) return;
  vorigeKnipper += KNIPPER_MS;
  ledAan = !ledAan;
  digitalWrite(STATUSLED, ledAan ? HIGH : LOW);
}

void taakKnop(unsigned long nu) {                   // ontdenderen in software (hoofdstuk 24)
  int lezing = digitalRead(KNOP);
  if (lezing != vorigeLezing) {                     // er verandert iets: echt of dender
    laatsteWissel = nu;
    vorigeLezing = lezing;
  }
  if (nu - laatsteWissel >= DENDER_MS && lezing != stabiel) {
    stabiel = lezing;
    if (stabiel == HIGH) {                          // de flank 'net ingedrukt'
      keerGedrukt++;
      Serial.print(nu);
      Serial.print(" ms  knop ingedrukt, keer ");
      Serial.println(keerGedrukt);
    }
  }
}

void taakPomp(unsigned long nu) {                   // 10 min per uur, niet zonder water (hoofdstuk 23)
  if (nu - startCyclus >= CYCLUS_MS) {
    startCyclus += CYCLUS_MS;                       // een nieuw uur begint
  }
  bool tijdOmTePompen = (nu - startCyclus) < POMP_AAN_MS;
  bool genoegWater = (digitalRead(VLOTTER) == LOW);
  digitalWrite(POMP, (tijdOmTePompen && genoegWater) ? HIGH : LOW);
}

void setup() {
  pinMode(KNOP, INPUT);             // de 74HC14 levert zelf een harde 0 of 5 V
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(STATUSLED, OUTPUT);
  pinMode(POMP, OUTPUT);
  Serial.begin(115200);
  Serial.println("taken.ino gestart");
}

```

```
70  void loop() {
71    unsigned long nu = millis();       // één keer de klok lezen per ronde
72    taakMeten(nu);
73    taakStatusled(nu);
74    taakKnop(nu);
75    taakPomp(nu);
76  }
```

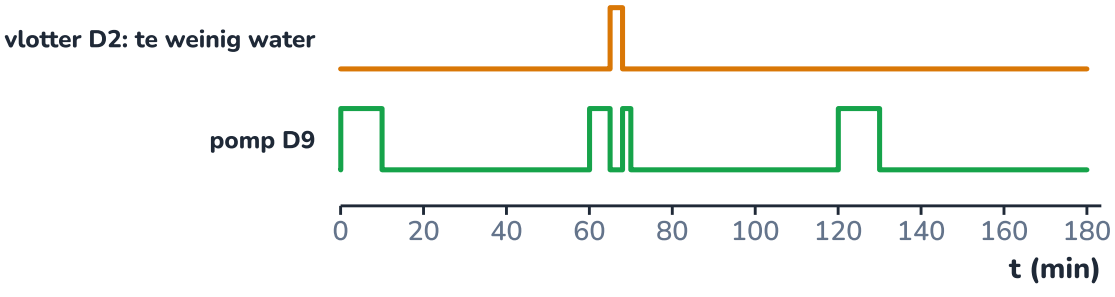
Uitvoer van de eerste 6,6 s, met twee drukken op de knop (van 3,30 tot 3,55 s en van 5,12 tot 5,40 s):

```
taken.ino gestart
2000 ms  LDR 421
3320 ms  knop ingedrukt, keer 1
4000 ms  LDR 413
5140 ms  knop ingedrukt, keer 2
6000 ms  LDR 403
```

De metingen komen precies om de 2000 ms, de knop wordt telkens 20 ms na het indrukken herkend (de ontdenderingstijd), en ondertussen knippert de led zonder haperen. Het tijdsdiagram toont de vier taken naast elkaar. Over een langere tijd zie je de pomptaak: 10 minuten aan per uur, en uit zodra de vlotter te weinig water meldt.



De vier taken lopen door elkaar zonder op elkaar te wachten. Getekend uit de pinwissels van de simulatie.



De pomp volgt haar uurritme; tijdens het watertekort blijft ze uit en daarna gaat ze verder tot het einde van de 10 minuten.

UITGEWERKT VOORBEELD · 29.15

15/23

Na 49,7 dagen: de overloop van millis()

Na 2^{32} ms = 4 294 967 296 ms, of 49,7 dagen, springt `millis()` van 4 294 967 295 terug naar 0. Stel: de laatste meting was bij `vorige` = 4 294 966 000, het interval is 2000 ms. Werkt `if (nu - vorige >= INTERVAL)` nog? En `if (nu >= vorige + INTERVAL)`?

GEGEVEN

- `unsigned long`: 32 bit, alle bewerkingen modulo 2^{32}
- vorige = 4 294 966 000, INTERVAL = 2000

GEVRAAGD

- wanneer elke versie de volgende meting doet

OPLOSSING

1. **Wanneer zou de meting moeten komen?** 2000 ms na vorige, dus 704 ms na de overloop: $4\,294\,966\,000 + 2000 - 4\,294\,967\,296 = 704$.

2. **Versie 1:** `nu - vorige`. Bij nu = 704 is $704 - 4\,294\,966\,000$ negatief, maar een unsigned long kent geen negatieve getallen: het resultaat is modulo 2^{32}, $704 - 4\,294\,966\,000 + 4\,294\,967\,296 = 2000$. Precies het verstreken aantal ms, dus de meting komt op tijd. Het verschil van twee tijden is altijd juist, zolang er minder dan 49,7 dagen tussen liggen.

3. **Versie 2:** `vorige + INTERVAL`. $4\,294\,966\,000 + 2000$ loopt zelf over en wordt 704. Vlak vóór de overloop is nu = 4 294 967 000, en $4\,294\,967\,000 \geq 704$ is waar: de meting komt 1,3 s te vroeg. Erger: vorige wordt 704, 2704, 4704 ..., allemaal kleiner dan nu, dus de taak loopt bij élke ronde opnieuw, tot vorige na ruim twee miljoen rondes weer bij nu is.

4. **Narekenen met Python.** Een Python-int loopt nooit over; met `% 2**32` reken je zoals de Mega:

```python
# overloop.py - de Mega nabootsen: unsigned long rekent modulo 2**32
M = 2**32
INTERVAL = 2000
vorige = 4_294_966_000                    # de vorige meting, 1296 ms voor de overloop

for nu in (4_294_967_000, 4_294_967_295, 0, 704):
    verschil = (nu - vorige) % M          # zoals nu - vorige in unsigned long
    fout = nu >= (vorige + INTERVAL) % M   # zoals nu >= vorige + INTERVAL
    print(f"nu = {nu:>10}   nu - vorige = {verschil:>10}   goed: {verschil >= INTERVAL!s:5}   fout: {fout}")
```

```
nu = 4294967000   nu - vorige =       1000   goed: False   fout: True
nu = 4294967295   nu - vorige =       1295   goed: False   fout: True
nu =          0   nu - vorige =       1296   goed: False   fout: False
nu =        704   nu - vorige =       2000   goed: True    fout: True
```

5 **De test op de 'Mega'.** In de simulatie start de klok 7,3 s voor de overloop. `overloop.ino` (hieronder) meet gewoon door, telkens na 2000 ms:

```
// overloop.ino - elke 2 s een regel, ook op het moment dat millis() weer bij 0 begint
const unsigned long INTERVAL = 2000;
unsigned long vorige = 0;

void setup() {
  Serial.begin(115200);
  vorige = millis();
}

void loop() {
  unsigned long nu = millis();
  if (nu - vorige >= INTERVAL) {
    Serial.print("nu = ");
    Serial.print(nu);
    Serial.print("   vorige = ");
    Serial.print(vorige);
    Serial.print("   nu - vorige = ");
    Serial.println(nu - vorige);
    vorige += INTERVAL;
  }
}
```

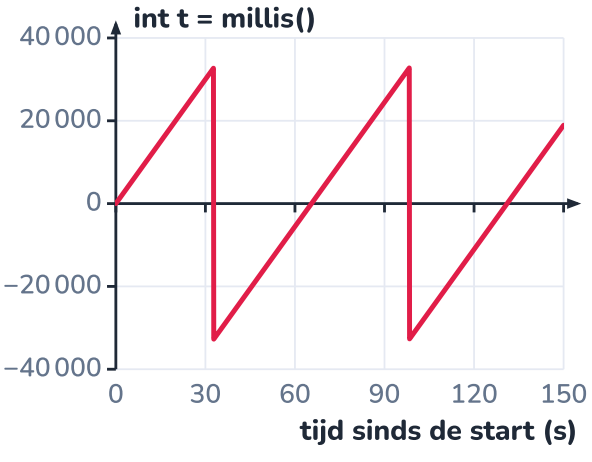
```
nu = 4294962000   vorige = 4294960000   nu - vorige = 2000
nu = 4294964000   vorige = 4294962000   nu - vorige = 2000
nu = 4294966000   vorige = 4294964000   nu - vorige = 2000
nu = 704   vorige = 4294966000   nu - vorige = 2000
nu = 2704   vorige = 704   nu - vorige = 2000
nu = 4704   vorige = 2704   nu - vorige = 2000
```

ANTWOORD

`nu - vorige >= INTERVAL` meet op het juiste moment (nu = 704), ook over de overloop heen. `nu >= vorige + INTERVAL` meet te vroeg en daarna bij elke ronde.

Klopt dit? Werkt het ook als je tijden in een `int` bewaart? Nee: een int loopt al na 32,8 s over (grafiek rechts), en dan is ook het verschil fout, omdat `millis()` en de int een ander bereik hebben. Tijden altijd in unsigned long.

millis() klimt 49,7 dagen lang en begint dan weer bij 0.



Dezelfde tijd in een int: na 32,8 s negatief, na 65,5 s weer bij 0.

UITLEG · 29.16

Bibliotheken en eigen bestanden

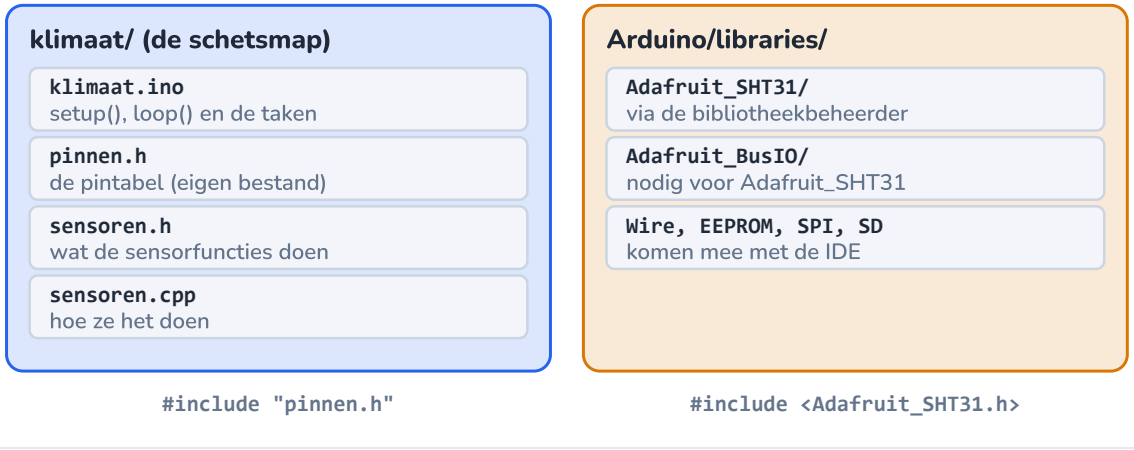
Niemand schrijft alles zelf. Een **bibliotheek** is code die iemand anders geschreven en getest heeft, voor een onderdeel (een sensor, een display) of een taak (I²C, bestanden op een SD-kaart). Je haalt ze in je programma met `#include`. Er zijn drie soorten:

- **Meegeleverd met de IDE**: `Wire` (I²C), `EEPROM`, `SPI`, `SD`, `Servo` (F25).
- **Te installeren** via de bibliotheekbeheerder (*Tools, Manage Libraries*). De kaslucht van de module krijgt een SHT31: een sensor van een paar millimeter die temperatuur (±0,2 °C) en relatieve vochtigheid (±2 %) meet en ze digitaal doorgeeft over de I²C-bus (in hoofdstuk 21 rekenden we nog met een DHT22; de SHT31 is nauwkeuriger en deelt de bus met de klok van hoofdstuk 30). Zoek 'SHT31' en installeer *Adafruit SHT31 Library*; de beheerder stelt voor om ook *Adafruit BusIO* te installeren, die ze nodig heeft.
- **Eigen bestanden** in de map van je schets, bv. `pinnen.h`. De IDE toont ze als tabbladen.

`#include <naam.h>` zoekt bij de bibliotheken, `#include "naam.h"` eerst in de map van je schets. Een bibliotheek gebruik je bijna altijd in drie stappen: een **object** maken, het in `setup()` starten, en dan zijn functies aanroepen:

```
#include <Wire.h>
#include <Adafruit_SHT31.h>

Adafruit_SHT31 sht = Adafruit_SHT31();    // 1. het object

void setup() {
  if (!sht.begin(0x44)) {                 // 2. starten op I2C-adres 0x44
    Serial.println("geen SHT31 gevonden");
  }
}

void loop() {
  float t, rv;
  sht.readBoth(&t, &rv);                  // 3. gebruiken: meet beide in één keer
}
```

Wat er achter `begin(0x44)` en `readBoth` gebeurt (een adres, een commando, zes bytes en een controlegetal over twee draadjes), leer je in hoofdstuk 30. Dat is precies het nut van een bibliotheek: je kan de sensor gebruiken vóór je het protocol kent. Het `&` voor t en rv geeft het *adres* van de variabelen door, zodat de functie er twee waarden in kan zetten.



Eigen bestanden zitten in de schetsmap; bibliotheken in de map libraries. Tussen punthaken zoekt de compiler bij de bibliotheken.

CODE · 29.17

Een eigen headerbestand: pinnen.h

17/23

Elke schets van de module begon met dezelfde rij `const int`'s. Die horen in één bestand, dan pas je een pin maar op één plaats aan. Maak in de IDE een nieuw tabblad (het pijltje rechtsboven) met de naam `pinnen.h`, en zet bovenaan elke schets `#include "pinnen.h"`.

C++ (Arduino) **pinnen.h** 33 regels

```cpp
// pinnen.h - de pintabel van de klimaatmodule (Arduino Mega 2560)
// Eén bestand voor alle schetsen van de module: verandert er een pin, dan pas je het maar op één plaats aan.
#ifndef PINNEN_H
#define PINNEN_H

// analoge ingangen (0 tot 5 V, 10 bit)
const int NTC = A0;             // NTC 10 kOhm in het voedingsreservoir (hoofdstuk 22)
const int LDR = A1;             // lichtsensor onder een kapje, 1,0 kOhm naar GND (hoofdstuk 22)
const int PH = A2;              // pH-module (hoofdstuk 15)
const int BATTERIJ = A3;        // deler 100 kOhm / 33 kOhm (hoofdstuk 22)
const int STROOM = A4;          // ACS712-20A in de batterijleiding (hoofdstuk 16)
const int DRUK = A5;            // MPX5010: het niveau van het reservoir (hoofdstuk 11)
const int EC = A6;              // geleidbaarheid (hoofdstuk 13)

// digitale ingangen
const int VLOTTER = 2;          // naar GND, INPUT_PULLUP: LOW = genoeg water (hoofdstuk 23)
const int KNOP = 22;            // S1 'kasdak' via RC + 74HC14, INPUT: HIGH = ingedrukt (hoofdstuk 24)
const int EIND_OPEN = 23;       // verbreekcontact, INPUT_PULLUP: HIGH = dak open (hoofdstuk 26)
const int EIND_DICHT = 24;      // verbreekcontact, INPUT_PULLUP: HIGH = dak dicht

// uitgangen
const int LAMPEN = 5;           // PWM (490 Hz) naar de IRLZ44N van de groeilampen (hoofdstuk 25)
const int PWM_DICHT = 6;        // H-brug: sluiten (hoofdstuk 26)
const int PWM_OPEN = 7;         // H-brug: openen
const int DAK_ENABLE = 8;       // H-brug: beide helften vrijgeven
const int POMP = 9;             // IRLZ44N van de pomp (hoofdstuk 23)
const int CO2_KLEP = 10;        // IRLZ44N van de CO2-klep
const int VENTILATOREN = 11;    // IRLZ44N van de ventilatoren
const int STATUSLED = 13;       // groene statusled (en het ledje L op het bordje)

// I2C: SDA = 20, SCL = 21 (vast op de Mega)

#endif
```

Drie dingen vallen op. De regels `#ifndef PINNEN_H`, `#define PINNEN_H` en `#endif` zijn een **include-bewaking**: wordt het bestand twee keer ingevoegd (via twee andere bestanden), dan slaat de compiler het de tweede keer over. De pinnen zijn `const int`: een `const` mag je in elk bestand invoegen, een gewone `int LDR = A1;` zou bij twee `.cpp`-bestanden een fout *multiple definition* geven. En de commentaren vermelden het hoofdstuk waar elk onderdeel uitgelegd is: zo blijft het bestand ook over een jaar leesbaar.

Een .h en een .cpp

Een echte bibliotheek bestaat uit twee bestanden: in het `.h`-bestand staat *wat* er is (de declaraties, zoals `float daglichtLux();`), in het `.cpp`-bestand *hoe* het werkt. In opdracht 29.13 splits je zo de sensorfuncties af.

UITLEG · 29.18

Het plan: taken en intervallen van de klimaatmodule

18/23

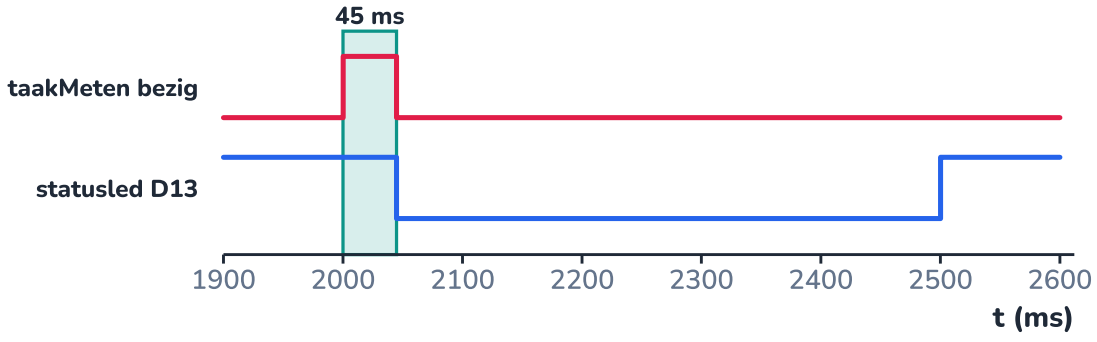
Voor je het volledige programma schrijft, maak je een plan: welke taken, hoe vaak, welke pinnen, en wat ze met elkaar delen. Dat plan is de klimaatmodule in tabelvorm:

taak	hoe vaak	leest	schrijft	uit
taakMeten	om de 2 s	SHT31, A1, A3, A4	luchtT, luchtRV, lux, uBat, iBat, soc, grens	16, 22, 25, 20
taakLampen	om de 200 ms	lux, grens, klok	D5 (PWM)	25
taakPomp	elke ronde	D2, klok	D9	23
taakDak	elke ronde	D22, D23, D24	D6, D7	24, 26
taakStatusled	om de 500 ms (alarm: 100 ms)	genoegWater, soc	D13	dit hoofdstuk
taakMelden	om de minuut	alles	Serial	dit hoofdstuk
taakBewaren	om het uur	soc	EEPROM	dit hoofdstuk

De taken praten niet rechtstreeks met elkaar, maar via globale variabelen: taakMeten schrijft `lux` en `grens`, taakLampen leest ze. Zo kan je elke taak apart testen.

Het energiebeheer komt uit hoofdstuk 20: boven 40 % laadtoestand mogen de lampen volgens het schema branden, onder 20 % blijven ze uit, en daartussen alleen 'op zon'. Hoe weet de Mega of de zon ze kan voeden? Aan de batterijstroom: laadt de batterij (stroom negatief), dan is er zonne-overschot en mag de grens voor de lampen 5 stappen omhoog; levert de batterij stroom, dan 5 stappen omlaag. Zo krijgen de lampen precies het overschot. In hoofdstuk 32 wordt dit een toestandsmachine met de modi van de module.

Het tijdbudget. Een taak mag de lus niet lang ophouden. De meettaak is de traagste: de SHT31 heeft 20 ms nodig voor een meting, de stroommeting neemt 100 metingen in 20,4 ms (tien PWM-perioden, hoofdstuk 25), en er zijn nog 32 gewone ADC-metingen van 0,11 ms. In de simulatie duurde de langste ronde van `loop()` 44,8 ms. Het tijdsdiagram toont het gevolg: de statusled moest om 2000 ms uitgaan, maar de meettaak was toen net begonnen. Voor een led is dat onzichtbaar; voor de knop betekent het hoogstens 45 ms extra reactietijd, en de eindeloopschakelaar van het dak wordt in die tijd bij volle snelheid (0,10 m/s) 4,5 mm te laat gezien. Aanvaardbaar, maar het is de grens: een taak die langer blokkeert, moet je opsplitsen.



Uit de simulatie van klimaat.ino: zolang taakMeten loopt, krijgt geen enkele andere taak een beurt.

UITGEWERKT VOORBEELD · 29.19

De sturing opbouwen in vijf stappen

19/23

Schrijf de sturing van de klimaatmodule zo dat je na elke stap iets kan testen, en dat de module nooit in een gevaarlijke toestand komt als een stap nog niet af is.

GEGEVEN

- de pintabel (pinnen.h)
- de losse schetsen uit hoofdstuk 16 en 22 tot 26
- het plan met zeven taken

GEVRAAGD

- een volgorde van stappen, met per stap een test

OPLOSSING

1 **Stap 1: pinnen en een veilige start.** Alle uitgangen eerst op OUTPUT en LOW: lampen, pomp en motor uit, wat er ook gebeurt. Een lijst van pinnen met een for-lus is korter en je vergeet er geen:

```
void setup() {
  for (int i = 0; i < AANTAL_UIT; i++) {
    pinMode(UITGANGEN[i], OUTPUT);
    digitalWrite(UITGANGEN[i], LOW);                // veilige begintoestand: alles uit
  }
  digitalWrite(DAK_ENABLE, HIGH);
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(KNOP, INPUT);
  pinMode(EIND_OPEN, INPUT_PULLUP);
  pinMode(EIND_DICHT, INPUT_PULLUP);
  Serial.begin(115200);
  Serial.println("klimaatmodule v1 (hoofdstuk 29)");
  Wire.begin();
  if (!sht.begin(0x44)) Serial.println("ALARM: geen SHT31 op adres 0x44");
  EEPROM.get(ADRES_SOC, soc);
  if (isnan(soc) || soc < 0.0 || soc > 100.0) soc = 50.0;   // leeg EEPROM: een ruwe schatting
  Serial.print("laadtoestand uit het EEPROM: ");
  Serial.print(soc, 1);
  Serial.println(" %");
  uBat = batterijspanning();
  dakOpen = (digitalRead(EIND_OPEN) == HIGH);
}
```

Test: na het opladen blijft alles uit en de monitor toont de laadtoestand uit het EEPROM.

2 **Stap 2: meten en melden.** taakMeten en taakMelden, nog zonder actuatoren. *Test:* houd je hand op de SHT31 (T en RV stijgen), dek de LDR af (licht daalt), vergelijk de batterijspanning met een multimeter.

3 **Stap 3: de pomp.** taakPomp uit `taken.ino`, met een melding bij elke wissel. *Test:* til de vlotter op (de pomp stopt en het alarm verschijnt). Om niet een uur te wachten, zet je tijdelijk `CYCLUS_MS` op 60 s en `POMP_AAN_MS` op 10 s.

4 **Stap 4: knop, dak en statusled.** De dakcode uit hoofdstuk 26, nu als taak met `nu`. *Test:* eerst met de motor los van het dak: druk en kijk of de juiste PWM-pin reageert; houd een eindeloopschakelaar met de hand in; laat de time-out van 12 s eens aflopen.

5 **Stap 5: lampen en energiebeheer.** taakLampen met de grens uit taakMeten, en taakBewaren. *Test:* zet in het EEPROM tijdelijk een laadtoestand van 35 % en kijk of de lampen het zonne-overschot volgen.

ANTWOORD

Vijf stappen, elk met een eigen test; het volledige programma staat op het volgende blad.

Klopt dit? Waarom deze volgorde? Wat meet, kan niets kapotmaken; wat beweegt (pomp, dak), komt pas als de metingen kloppen; en het energiebeheer, dat alles aan elkaar knoopt, komt als laatste.

CODE · 29.20

klimaat.ino: de volledige sturing

Het volledige programma, met `pinnen.h` in dezelfde map en de bibliotheek Adafruit SHT31 geïnstalleerd. Het is lang, maar elke taak is een korte functie die je herkent uit een eerder hoofdstuk. Lees eerst `loop()` onderaan: zeven regels, zeven taken.

C++ (Arduino) **klimaat.ino** 268 regels

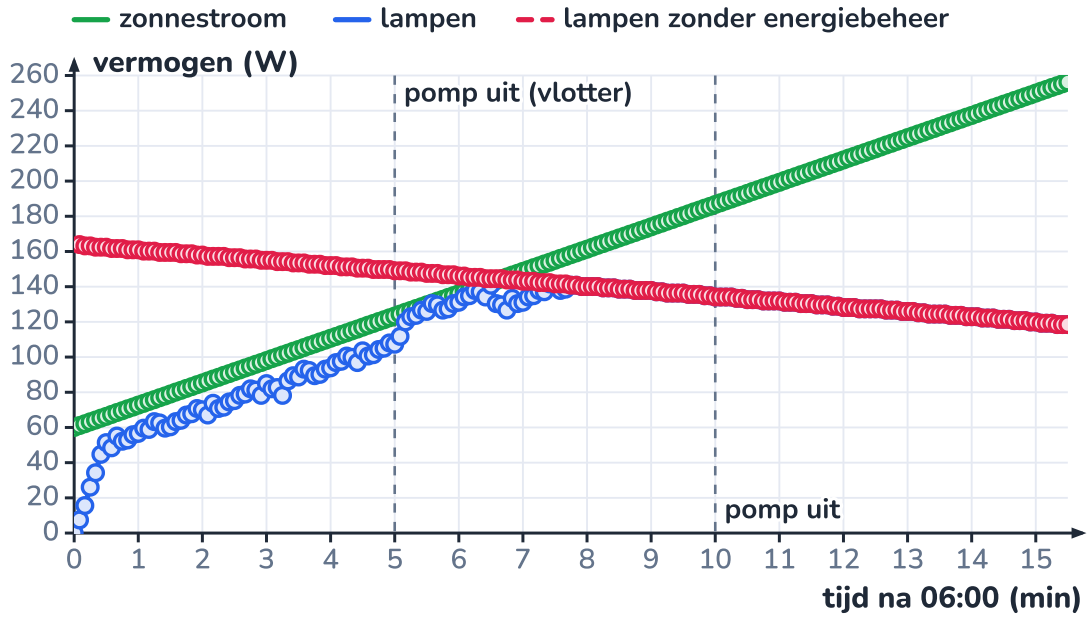
```cpp
// klimaat.ino - de sturing van de klimaatmodule, versie 1 (Arduino Mega 2560, hoofdstuk 29)
// Zeven taken zonder delay(): meten (2 s), lampen (200 ms), pomp, kasdak, statusled, melden (1 min) en bewaren (1 h).
// Ventilatoren en CO2-klep krijgen hun regeling in hoofdstuk 31; de echte klok en de logger komen in hoofdstuk 30.
#include <Wire.h>
#include <EEPROM.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"

// ------------------------------------------------------------------ instellingen
const unsigned long MEET_MS = 2000;
const unsigned long LAMP_MS = 200;
const unsigned long MELD_MS = 60000UL;
const unsigned long BEWAAR_MS = 3600000UL;          // de laadtoestand elk uur in het EEPROM
const unsigned long POMP_AAN_MS = 10UL * 60 * 1000; // 10 minuten pompen ...
const unsigned long CYCLUS_MS = 60UL * 60 * 1000;   // ... per uur
const unsigned long DENDER_MS = 20;
const int START_UUR = 6;                            // de Mega wordt om 6 uur aangezet (echte klok: hoofdstuk 30)
const float CAPACITEIT_AH = 240.0;                  // twee LiFePO4-batterijen van 120 Ah (hoofdstuk 20)
const int ADRES_SOC = 0;                            // waar de laadtoestand in het EEPROM staat
const float DOEL = 200.0;                           // umol/(m2 s) op het kweekbed (hoofdstuk 25)
const float LAMPEN_VOL = 200.0;                     // wat de lampen op 100 % geven
const float LUX_PER_UMOL = 54.0;
const float U_NODIG = 11.63, U_START = 2.21;        // dakmotor (hoofdstuk 26)
const unsigned long AANLOOP_MS = 1000, REMMEN_VANAF_MS = 6000, TIMEOUT_MS = 12000;
const int UITGANGEN[] = {LAMPEN, PWM_DICHT, PWM_OPEN, DAK_ENABLE, POMP, CO2_KLEP, VENTILATOREN, STATUSLED};
const int AANTAL_UIT = sizeof(UITGANGEN) / sizeof(UITGANGEN[0]);

// ------------------------------------------------------------------ toestand (globaal: alle taken gebruiken ze)
Adafruit_SHT31 sht = Adafruit_SHT31();
float luchtT = NAN, luchtRV = NAN;                  // kaslucht, van de SHT31 op I2C
float lux = 0, uBat = 0, iBat = 0, soc = 50.0;
int grens = 0;                                      // energiebeheer: de hoogste PWM die de lampen mogen
int stand = 0;                                      // de PWM die de lampen nu krijgen
bool pompAan = false, genoegWater = true;
enum Dak { STIL, SLUITEN, OPENEN };
Dak dak = STIL;
bool dakOpen = false;
unsigned long vorigeMeting = 0, vorigeLamp = 0, vorigeMelding = 0, vorigeBewaring = 0;
unsigned long startCyclus = 0, startDak = 0, laatsteWissel = 0, vorigeKnipper = 0;
int vorigeLezing = LOW, stabiel = LOW;
bool ledAan = false;

// ------------------------------------------------------------------ sensoren
int gemiddeld(int pin) {                            // 16 metingen middelen tegen ruis (hoofdstuk 22)
  long som = 0;
  for (int i = 0; i < 16; i++) som += analogRead(pin);
  return som / 16;
}

float daglichtLux() {                               // LDR onder een kapje (hoofdstuk 22 en 25)
  int n = gemiddeld(LDR);
  if (n < 1) n = 1;
  float r = 1000.0 * (1024.0 - n) / n;
  return 10.0 * pow(r / 15000.0, -1.0 / 0.7) / 0.10;
}

float batterijspanning() {                          // deler 100 kOhm / 33 kOhm (hoofdstuk 22)
  return gemiddeld(BATTERIJ) * 5.0 / 1024.0 * (133.0 / 33.0);
```

```
}

float batterijstroom() {                          // ACS712-20A: 100 metingen in 20,4 ms (hoofdstuk 16 en 25)
  long som = 0;
  unsigned long t0 = micros();
  for (int i = 0; i < 100; i++) {
    while (micros() - t0 < i * 204UL) {
    }
    som += analogRead(STROOM);
  }
  return (som / 100.0 * 5.0 / 1024.0 - 2.50) / 0.100;  // A, positief = ontladen
}

// ---------------------------------------------------------------- tijd
unsigned long secondenVanDeDag(unsigned long nu) {
  return (START_UUR * 3600UL + nu / 1000) % 86400UL;
}

void printTwee(unsigned long x) {                 // altijd twee cijfers: 7 -> 07
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void printTijd(unsigned long nu) {
  unsigned long s = secondenVanDeDag(nu);
  printTwee(s / 3600);
  Serial.print(':');
  printTwee(s / 60 % 60);
  Serial.print(':');
  printTwee(s % 60);
  Serial.print("  ");
}

// ---------------------------------------------------------------- taken
void taakMeten(unsigned long nu) {
  if (nu - vorigeMeting < MEET_MS) return;
  vorigeMeting += MEET_MS;
  sht.readBoth(&luchtT, &luchtRV);                // nan als de sensor niet antwoordt
  lux = daglichtLux();
  uBat = batterijspanning();
  iBat = batterijstroom();
  soc -= iBat * (MEET_MS / 3600000.0) / CAPACITEIT_AH * 100.0;    // lading tellen (hoofdstuk 16)
  if (uBat >= 14.2 && iBat < 0.0 && iBat > -2.0) soc = 100.0;     // vol: opnieuw ijken
  soc = constrain(soc, 0.0, 100.0);
  // energiebeheer (hoofdstuk 20): hoeveel licht mag er hoogstens branden?
  if (soc >= 40.0) grens = 255;
  else if (soc < 20.0) grens = 0;
  else if (iBat < 0.0) grens = min(grens + 5, 255);   // de batterij laadt: er is zon over
  else grens = max(grens - 5, 0);                     // de batterij levert: minder licht
}

void taakLampen(unsigned long nu) {               // dimmen met het daglicht (hoofdstuk 25)
  if (nu - vorigeLamp < LAMP_MS) return;
  vorigeLamp += LAMP_MS;
  int wens = 0;
  unsigned long uur = secondenVanDeDag(nu) / 3600;
  if (uur >= 6 && uur < 22) {                     // lichtperiode van 16 uur
    float f = constrain((DOEL - lux / LUX_PER_UMOL) / LAMPEN_VOL, 0.0, 1.0);
    wens = (int)(f * 255 + 0.5);
  }
  int doelStand = min(wens, grens);
  if (doelStand > stand) stand++;                 // zacht: hoogstens 1 stap per 200 ms
  else if (doelStand < stand) stand--;
  analogWrite(LAMPEN, stand);
}

void taakPomp(unsigned long nu) {                 // 10 min per uur, nooit zonder water (hoofdstuk 23)
  if (nu - startCyclus >= CYCLUS_MS) startCyclus += CYCLUS_MS;
```

```
  bool tijd = (nu - startCyclus) < POMP_AAN_MS;
  bool water = (digitalRead(VLOTTER) == LOW);
  bool aan = tijd && water;
  if (water != genoegWater) {                        // alleen melden als er iets verandert
    printTijd(nu);
    Serial.println(water ? "vlotter: weer genoeg water" : "ALARM vlotter: te weinig water");
    genoegWater = water;
  }
  if (aan != pompAan) {
    printTijd(nu);
    Serial.println(aan ? "pomp aan" : "pomp uit");
    pompAan = aan;
  }
  digitalWrite(POMP, aan ? HIGH : LOW);
}

bool knopGedrukt(unsigned long nu) {                 // true bij de flank 'net ingedrukt' (hoofdstuk 24)
  int lezing = digitalRead(KNOP);
  if (lezing != vorigeLezing) {
    laatsteWissel = nu;
    vorigeLezing = lezing;
  }
  if (nu - laatsteWissel >= DENDER_MS && lezing != stabiel) {
    stabiel = lezing;
    return stabiel == HIGH;
  }
  return false;
}

void motor(int richting, int waarde) {               // +1 sluiten, -1 openen, 0 stil (hoofdstuk 26)
  analogWrite(PWM_DICHT, richting > 0 ? waarde : 0);
  analogWrite(PWM_OPEN, richting < 0 ? waarde : 0);  // nooit beide tegelijk
}

void stopDak(unsigned long nu, const char* reden) {
  motor(0, 0);
  dak = STIL;
  printTijd(nu);
  Serial.println(reden);
}

void taakDak(unsigned long nu) {
  if (knopGedrukt(nu)) {
    if (dak != STIL) {
      stopDak(nu, "dak gestopt met de knop");        // een druk tijdens de beweging = stoppen
    } else {
      dak = dakOpen ? SLUITEN : OPENEN;
      startDak = nu;
      printTijd(nu);
      Serial.println(dak == SLUITEN ? "dak sluiten" : "dak openen");
    }
  }
  if (dak == STIL) return;
  unsigned long t = nu - startDak;
  if (dak == SLUITEN && digitalRead(EIND_DICHT) == HIGH) {
    dakOpen = false;
    stopDak(nu, "dak dicht");
  } else if (dak == OPENEN && digitalRead(EIND_OPEN) == HIGH) {
    dakOpen = true;
    stopDak(nu, "dak open");
  } else if (t > TIMEOUT_MS) {
    stopDak(nu, "ALARM dak: eindpunt niet bereikt");
  } else {                                           // aanlopen, constant, afremmen tot 30 %
    float s = 1.0;
    if (t < AANLOOP_MS) s = t / (float)AANLOOP_MS;
    else if (t > REMMEN_VANAF_MS) s = max(0.3, 1.0 - (t - REMMEN_VANAF_MS) / (float)AANLOOP_MS);
    float u = U_START + (U_NODIG - U_START) * s;
    motor(dak == SLUITEN ? +1 : -1, constrain((int)(u / uBat * 255 + 0.5), 0, 255));
```

```
  }
}

void taakStatusled(unsigned long nu) {                // rustig knipperen = alles goed, snel = alarm
  bool alarm = !genoegWater || soc < 10.0;
  unsigned long periode = alarm ? 100 : 500;
  if (nu - vorigeKnipper < periode) return;
  vorigeKnipper = nu;                                 // hier mag het: een led mag wat verlopen
  ledAan = !ledAan;
  digitalWrite(STATUSLED, ledAan ? HIGH : LOW);
}

void taakMelden(unsigned long nu) {                   // elke minuut één regel voor de seriële monitor
  if (nu - vorigeMelding < MELD_MS) return;
  vorigeMelding += MELD_MS;
  printTijd(nu);
  Serial.print("T ");
  Serial.print(luchtT, 1);
  Serial.print(" C  RV ");
  Serial.print(luchtRV, 0);
  Serial.print(" %  licht ");
  Serial.print(lux, 0);
  Serial.print(" lx  bat ");
  Serial.print(uBat, 2);
  Serial.print(" V ");
  Serial.print(iBat, 1);
  Serial.print(" A  SoC ");
  Serial.print(soc, 1);
  Serial.print(" %  lampen ");
  Serial.print(stand);
  Serial.print("/");
  Serial.println(grens);
}

void taakBewaren(unsigned long nu) {                  // de laadtoestand overleeft een stroomonderbreking
  if (nu - vorigeBewaring < BEWAAR_MS) return;
  vorigeBewaring += BEWAAR_MS;
  EEPROM.put(ADRES_SOC, soc);                         // schrijft alleen de bytes die veranderen
}

// ---------------------------------------------------------------- setup en loop
void setup() {
  for (int i = 0; i < AANTAL_UIT; i++) {
    pinMode(UITGANGEN[i], OUTPUT);
    digitalWrite(UITGANGEN[i], LOW);                  // veilige begintoestand: alles uit
  }
  digitalWrite(DAK_ENABLE, HIGH);
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(KNOP, INPUT);
  pinMode(EIND_OPEN, INPUT_PULLUP);
  pinMode(EIND_DICHT, INPUT_PULLUP);
  Serial.begin(115200);
  Serial.println("klimaatmodule v1 (hoofdstuk 29)");
  Wire.begin();
  if (!sht.begin(0x44)) Serial.println("ALARM: geen SHT31 op adres 0x44");
  EEPROM.get(ADRES_SOC, soc);
  if (isnan(soc) || soc < 0.0 || soc > 100.0) soc = 50.0;   // leeg EEPROM: een ruwe schatting
  Serial.print("laadtoestand uit het EEPROM: ");
  Serial.print(soc, 1);
  Serial.println(" %");
  uBat = batterijspanning();
  dakOpen = (digitalRead(EIND_OPEN) == HIGH);
}

void loop() {
  unsigned long nu = millis();                        // één keer de klok lezen per ronde
  taakMeten(nu);
  taakLampen(nu);
```

```
  taakPomp(nu);
  taakDak(nu);
  taakStatusled(nu);
  taakMelden(nu);
  taakBewaren(nu);
}
```



Uit dezelfde simulatie: tot 06:08 volgen de lampen de zonnestroom min het basisverbruik; daarna begrenst het daglicht ze.

Een test in de simulatie: een lentemorgen vanaf 06:00, laadtoestand 35 % in het EEPROM, het daglicht stijgt van 1500 naar 4000 lx en de zonnepanelen leveren van 60 naar 250 W. Om 06:03:10 drukt iemand op de knop van het dak, van 06:05:00 tot 06:06:30 meldt de vlotter te weinig water. De seriële monitor:

```
klimaatmodule v1 (hoofdstuk 29)
laadtoestand uit het EEPROM: 35.0 %
06:00:00  pomp aan
06:01:00  T 17.9 C  RV 84 %  licht 1666 lx  bat 13.05 V -0.1 A  SoC 35.0 %  lampen 76/80
06:02:00  T 18.0 C  RV 84 %  licht 1831 lx  bat 13.05 V 0.0 A  SoC 35.0 %  lampen 94/90
06:03:00  T 18.0 C  RV 83 %  licht 1997 lx  bat 13.03 V 0.2 A  SoC 35.0 %  lampen 114/110
06:03:10  dak openen
06:03:15  dak open
06:04:00  T 18.1 C  RV 83 %  licht 2161 lx  bat 13.05 V -0.2 A  SoC 35.0 %  lampen 126/130
06:05:00  ALARM vlotter: te weinig water
06:05:00  pomp uit
06:05:00  T 18.2 C  RV 83 %  licht 2322 lx  bat 13.05 V 0.0 A  SoC 35.0 %  lampen 144/140
06:06:00  T 18.3 C  RV 83 %  licht 2492 lx  bat 13.05 V -0.2 A  SoC 35.0 %  lampen 176/180
06:06:30  vlotter: weer genoeg water
06:06:30  pomp aan
06:07:00  T 18.3 C  RV 83 %  licht 2658 lx  bat 13.05 V -0.2 A  SoC 35.0 %  lampen 176/180
06:08:00  T 18.4 C  RV 82 %  licht 2833 lx  bat 13.05 V -0.5 A  SoC 35.0 %  lampen 188/240
06:09:00  T 18.5 C  RV 82 %  licht 2983 lx  bat 13.07 V -1.6 A  SoC 35.0 %  lampen 185/255
06:10:00  pomp uit
06:10:00  T 18.6 C  RV 82 %  licht 3159 lx  bat 13.11 V -2.8 A  SoC 35.0 %  lampen 180/255
06:11:00  T 18.7 C  RV 82 %  licht 3325 lx  bat 13.15 V -4.9 A  SoC 35.1 %  lampen 177/255
06:12:00  T 18.7 C  RV 82 %  licht 3498 lx  bat 13.17 V -6.1 A  SoC 35.1 %  lampen 172/255
06:13:00  T 18.8 C  RV 81 %  licht 3660 lx  bat 13.19 V -7.2 A  SoC 35.1 %  lampen 169/255
```

```
06:14:00  T 18.9 C  RV 81 %  licht 3828 lx  bat 13.20 V -8.4 A  SoC 35.2 %  lampen 165/255
06:15:00  T 19.0 C  RV 81 %  licht 3981 lx  bat 13.22 V -9.6 A  SoC 35.3 %  lampen 161/255
```

Wat je ziet: bij 35 % zit de module tussen 20 en 40 %, dus de lampen krijgen alleen het zonne-overschot. De batterijstroom blijft rond 0 A, en de laadtoestand blijft 35,0 %. Als de vlotter alarm slaat, stopt de pomp en gaat de statusled snel knipperen; er komen 12 W vrij, en de lampen nemen dat over. Vanaf ongeveer 06:08 is er meer zon dan de lampen willen (bij meer daglicht dimmen ze, hoofdstuk 25): de grens klimt naar 255 en het overschot laadt de batterij, 35,3 % om 06:15. Het dak opent in 5 s. Het EEPROM is in deze 15 minuten niet beschreven: dat gebeurt maar één keer per uur.

NASLAG · 29.21

Foutzoektabel

21/23

De fouten die iedereen maakt, met wat je ziet en wat de compiler erover zegt. De eerste vijf houdt de compiler tegen; de rest compileert zonder klagen en moet je zelf vinden, met Serial en een tijdsdiagram in je hoofd. Een compilerfout staat vaak op de regel **na** de echte fout.

fout	wat je ziet	wat de compiler zegt	oplossing
puntkomma vergeten	compileert niet	`expected ',' or ';' before 'if'`: op de **volgende** regel	kijk één regel hoger
hoofdletter fout: `digitalwrite`	compileert niet	`'digitalwrite' was not declared in this scope`	C++ onderscheidt hoofd- en kleine letters
accolade `}` vergeten	compileert niet	`a function-definition is not allowed here`, `expected '}' at end of input`	laat de IDE inspringen (Ctrl+T) en kijk waar het scheef loopt
variabele buiten haar functie gebruikt	compileert niet	`'n' was not declared in this scope`	maak ze globaal of geef ze terug met return
bibliotheek niet geïnstalleerd of typfout in de naam	compileert niet	`Adafruit_SHT3x.h: No such file or directory`	bibliotheekbeheerder; naam van het .h-bestand nakijken
`if (x = 5)` in plaats van `==`	de if is altijd waar, x wordt 5	alleen een waarschuwing: `suggest parentheses around assignment`	`==` vergelijkt, `=` kent toe
puntkomma na de if: `if (x > 3);`	het blok eronder loopt altijd	waarschuwing: `empty body in an 'if' statement`	geen puntkomma na `if (…)`
gehele deling: `5 / 9 * (t - 32)`	altijd 0	niets	schrijf `5.0 / 9`
overloop: `int wacht = 60 * 1000;`	−5536 en rare wachttijden	waarschuwing `integer overflow in expression` (als waarschuwingen aanstaan)	`60000UL` of `60UL * 1000`, en unsigned long
tijd in een `int` bewaren	na 32,8 s gaat alles mis	niets	tijd altijd in `unsigned long`
`delay()` in de lus	knoppen en eindeloopschakelaars reageren niet	niets	taken met millis()
pinMode(..., OUTPUT) vergeten	led brandt zwak of niet, MOSFET half open	niets	alle uitgangen instellen in setup()
ingang zonder pull-up	de waarde springt willekeurig	niets	INPUT_PULLUP of een weerstand
baudrate verschilt van de monitor	rare tekens op het scherm	niets	115200 in Serial.begin en rechtsonder in de monitor
kommagetallen vergelijken met `==`	de vergelijking lukt bijna nooit	niets	vergelijk met `<` of `>`, of met een marge

NASLAG · 29.22

C++ voor de Arduino in één oogopslag

overloop van millis()

$$t_{\text{overloop}} = \frac{2^{32}\ \text{ms}}{86\,400\,000\ \text{ms/dag}} \approx 49{,}7\ \text{dagen}$$

symbool	betekenis	eenheid
2^{32}	aantal waarden van een unsigned long	

functie	wat ze doet	geeft terug
`pinMode(pin, OUTPUT)`	INPUT, INPUT_PULLUP of OUTPUT kiezen	niets
`digitalWrite(pin, HIGH)`	pin op 5 V (HIGH) of 0 V (LOW)	niets
`digitalRead(pin)`	de spanning op de pin lezen	HIGH of LOW
`analogRead(A1)`	spanning 0-5 V meten (±0,1 ms)	int 0 tot 1023
`analogWrite(5, 128)`	PWM op een PWM-pin (2-13, 44-46)	niets
`millis()`, `micros()`	tijd sinds de start	unsigned long (ms, µs)
`delay(ms)`	wachten, en verder niets doen	niets
`Serial.begin(115200)`	seriële poort openen	niets
`Serial.print(x, 2)`, `println`	tekst of getal sturen (2 decimalen), println met nieuwe regel	aantal verstuurde tekens
`constrain(x, a, b)`, `map(…)`	begrenzen, omrekenen van bereik	de nieuwe waarde
`EEPROM.put(adres, x)`, `EEPROM.get(adres, x)`	een variabele bewaren of terughalen	(get vult x)

	Python (hoofdstuk 27-28)	C++ op de Mega
variabele	`soc = 35.0`	`float soc = 35.0;` (type vooraf, puntkomma)
blok	inspringen na `:`	accolades `{ }`; inspringen is voor de lezer
keuze	`if a > b:` … `elif` … `else:`	`if (a > b) { } else if (…) { } else { }`
lus	`for i in range(16):`	`for (int i = 0; i < 16; i++) { }`
functie	`def lux(n):` … `return`	`float lux(int n) { … return …; }`
en, of, niet	`and`, `or`, `not`	`&&`, `||`, `!`
gehele deling	`7 // 2` (en `7 / 2` = 3.5)	`7 / 2` = 3; `7 / 2.0` = 3.5
overloop	bestaat niet: een int groeit mee	int 16 bit, long 32 bit: pas op
commentaar	`# …`	`// …` of `/* … */`

HET GEHEEL · 29.23

Waar dit hoofdstuk in het geheel past

Dit hoofdstuk rijgt de losse stukken van deel I6 aan elkaar tot één programma. Het is ook een brug tussen twee manieren van programmeren: Python op een computer die rekent en tekent (hoofdstuk 27-28), en C++ op een microcontroller die meet en schakelt, zonder scherm en zonder pauze.

F24 De microcontroller

F25 Sensoren en automatisch regelen

I16 Redox en batterijen

setup(), loop(), pinMode en digitalWrite: de eerste stappen, op de Uno van de meetkas.

water_geven.ino met delay(): het programma dat je hier zonder delay() herschrijft.

De laadtoestand tellen met de stroom op A4: het hart van het energiebeheer.

I20 Zonne-energie dimensioneren
De drempels van 40 en 20 % voor de lampen.

I24 Condensatoren en de RC-tijd
De knop ontdenderen: in hardware en in software, met millis().

I25 Van analoog naar digitaal
Bits, bereik en PWM: waarom een int overloopt en hoe analogWrite dimt.

I26 Motoren en actuatoren
De dakmotor met H-brug, eindeloop en time-out, hier een taak.

I27 Python: de basis
Dezelfde bouwstenen (variabelen, keuzes, lussen) in een andere taal.

I30 Datalogging en communicatie
Wat er achter Serial en de SHT31-bibliotheek zit, een echte klok en een logboek op een SD-kaart.

I31 Regelsystemen
Ventilatoren en verwarming met hysterese, als nieuwe taken in klimaat.ino.

I32 Toestandsmachines
De modi dag, nacht, alarm en onderhoud met switch-case; het dak en het energiebeheer worden toestanden.

E37 Embedded systemen en IoT
Interrupts en timers: taken die de hardware zelf op tijd start, en slaapstanden.

E39 PID-regeling
Een regelaar die met millis() op een vaste bemonsteringstijd rekent.

Habitat

De klimaatmodule heeft nu één programma: klimaat.ino, versie 1, met pinnen.h. Zeven taken lopen naast elkaar: meten om de 2 s (SHT31 op I²C 0x44, licht, batterij), lampen om de 200 ms met energiebeheer, de pomp 10 minuten per uur met vlotterbeveiliging, het dak via de knop, de statusled op D13, een statusregel per minuut en de laadtoestand elk uur in het EEPROM. De langste ronde van loop() duurt ongeveer 45 ms. In hoofdstuk 30 komen er een echte klok en een logboek bij, in 31 en 32 de regelingen en de modi. Opdracht 29.14 voegt een niveaualarm voor het reservoir toe.

SAMENVATTING

In het kort

- Een schets heeft setup() (één keer) en loop() (eindeloos); de IDE voegt main() toe. Het programma staat in flash, de variabelen in SRAM, wat moet blijven in EEPROM.
- Op de Mega is een int 16 bit (–32 768 tot 32 767), een long en unsigned long 32 bit, en double = float (4 bytes, ±7 cijfers).
- Rekenen gebeurt in het type van de getallen: 60 * 60 * 1000 loopt over in int. Schrijf 60UL * 60 * 1000; schrijf 5.0 om een gehele deling te vermijden.
- Globale variabelen onthouden iets tussen twee rondes van loop(); lokale bestaan alleen tijdens één aanroep; static lokaal combineert beide.
- pinMode kiest INPUT, INPUT_PULLUP of OUTPUT; digitalRead/digitalWrite werken met HIGH en LOW, analogRead geeft 0-1023, analogWrite 0-255 op een PWM-pin.
- Serial.print en de seriële plotter laten zien wat je programma doet: druk tussenresultaten af om een fout te vinden.
- delay() legt alles stil. Met millis() kijkt elke taak of ze aan de beurt is: if (nu – vorige ≥ INTERVAL) { vorige += INTERVAL; … }.
- millis() loopt na 49,7 dagen over; het verschil nu – vorige in unsigned long blijft toch juist. Vergelijk nooit nu ≥ vorige + INTERVAL.

- Bibliotheken haal je met #include <...>, eigen bestanden met #include "..."; een .h-bestand krijgt een include-bewaking.
- Bouw een groot programma in stappen en test elke stap; begin met een veilige toestand van alle uitgangen.

BEGRIPPEN

Wat je nu kent

compileren
Een programma in C++ vertalen naar machinecode die de microcontroller kan uitvoeren.

setup() en loop()
De twee functies van elke Arduino-schets: setup() loopt één keer na de start, loop() daarna eindeloos.

type (C++)
Wat voor waarde een variabele bevat en hoeveel geheugen ze krijgt, bv. int (2 bytes op de Mega) of float (4 bytes).

overloop (overflow)
Een getal dat groter wordt dan zijn type kan bevatten, springt naar het andere uiterste: 32 767 + 1 wordt −32 768 in een int.

gehele deling
Deling van twee gehele getallen in C++: de rest valt weg, 7 / 2 = 3.

globale variabele
Variabele buiten elke functie: elke functie kan ze gebruiken en ze houdt haar waarde zolang het programma draait.

lokale variabele
Variabele binnen een functie: ze bestaat alleen tijdens die aanroep.

millis()
Het aantal milliseconden sinds de start van de microcontroller, als unsigned long; na 49,7 dagen begint het weer bij 0.

niet-blokkerende code
Code die nooit wacht, maar telkens kijkt of het tijd is voor een taak, zodat meerdere taken tegelijk kunnen lopen.

bibliotheek (library)
Kant-en-klare code voor een taak of een onderdeel, die je met #include in je programma haalt.

headerbestand (.h)
Bestand met declaraties en constanten dat je met #include in een ander bestand invoegt.

EEPROM
Klein geheugen in de microcontroller dat zijn inhoud houdt zonder stroom; per byte maar ±100 000 keer te beschrijven.

seriële plotter
Venster in de Arduino-IDE dat getallen van de seriële poort als grafiek tekent.

HANDBOEK · HOOFDSTUK 30

Datalogging en communicatie

Een logboek dat zichzelf schrijft: van bits op een draad tot een grafiek in Python

De klimaatmodule meet om de twee seconden temperatuur, vochtigheid, licht en energie. Zonder geheugen is elke meting na twee seconden vergeten. In dit hoofdstuk bouw je de datalogger: de Mega praat met de computer over een seriële verbinding, leest sensoren over de I²C-bus, kent de datum dankzij een klokchip en schrijft elke minuut een regel op een SD-kaart, in het CSV-formaat van hoofdstuk 28. Daarna lees je een hele week in met Python, schoon je ze op en maak je er grafieken van die vertellen wat de module deed.

7× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 2× naslag · 1× verhaal · 1× het geheel · 11× code · ± 10 lestijden

P Programmeren en computationeel denken · O Onderzoeken en meten · A Data en AI

Elke minuut een regel op de kaart: het logboek van de klimaatmodule.

Na dit hoofdstuk

- Je kan uitleggen hoe een teken over een seriële verbinding gaat (startbit, databits, stopbit) en rekenen met de baudrate.
- Je kan een eenvoudig protocol ontwerpen (één CSV-regel per meting, commentaarregels, een kop) en de regels in Python inlezen met pyserial.
- Je kan I²C-sensoren aansluiten en uitlezen: adressen, SDA en SCL, pull-ups, en je kan de stijgtijd bij een lange kabel narekenen.
- Je kan de bytes van een sensor omrekenen met de datasheet en controleren met een CRC.
- Je kan uitleggen waarom millis() geen klok is, en een RTC (DS3231) lezen en zetten.
- Je kan metingen met een tijdstempel op een SD-kaart bewaren, weet waarom flush() of close() nodig is, en kan uitrekenen hoe lang een kaart volstaat.
- Je kan logbestanden in Python inlezen, opschonen (lege cellen, onmogelijke waarden, ontbrekende minuten), samenvatten per dag en in een grafiek zetten.
- Je kan het geheugengebruik en het eigen verbruik van een datalogger schatten.

VERHAAL · 30.1

Zestig jaar meten op een vulkaan

In maart 1958 begon de Amerikaanse scheikundige Charles David Keeling met metingen van koolstofdioxide in de lucht, op de flank van de vulkaan Mauna Loa op Hawaï: ver van steden en bossen, zodat hij de lucht van de hele planeet meet en niet die van één fabriek. Na een paar jaar zag hij twee dingen. Elk jaar zakt het CO_2-gehalte in de lente en de zomer, als de planten op het noordelijk halfrond groeien, en stijgt het weer in de herfst. En van jaar tot jaar gaat het omhoog: van ongeveer 315 ppm in 1958 naar meer dan 420 ppm vandaag. Die grafiek heet de **Keeling-curve**.

Het geheim van de curve is niet één slimme meting, maar een **logboek** dat nooit stopt: elke meting met een datum en een uur, elke ijking genoteerd, elke onderbreking verklaard. Op 27 november 2022 barstte de Mauna Loa zelf uit; de lava sneed de weg en de stroomkabel naar het observatorium af. Vanaf 8 december werd er tijdelijk gemeten op de naburige Maunakea, en in de gegevens staat precies wanneer en waar. Zonder die notities zou niemand de meetreeksen van voor en na de uitbarsting kunnen vergelijken.

De datalogger van de klimaatmodule is veel kleiner, maar hij moet hetzelfde kunnen: meten op vaste tijden, elke meting een tijdstempel geven, alles bewaren in een vorm die een programma kan lezen, en eerlijk tonen wat er ontbreekt.

Waarom dit hoofdstuk ertoe doet

In hoofdstuk 28 kreeg je een CSV-bestand en analyseerde je het. Hier maak je het bestand zelf, van de sensor tot de kaart, en zie je waarom lege cellen, tijdstempels en een vaste vorm zo belangrijk zijn.

Een meetstation hoog op een vulkaan: ver van alles, om de lucht van de hele aarde te meten.

UITLEG · 30.2

Wat je al weet, en het plan voor de logger

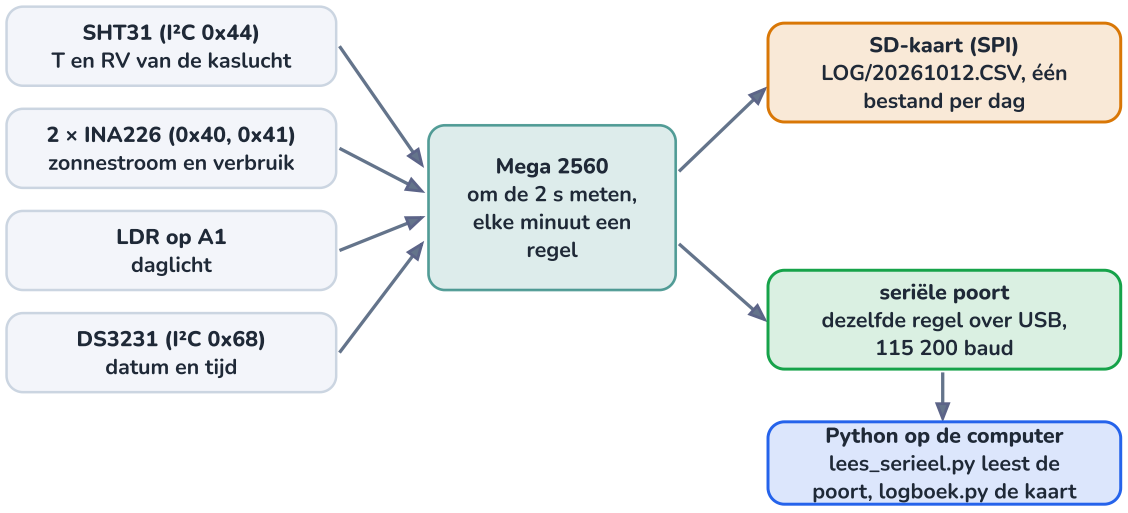
wat je al kan	waar	wat er nu bij komt
Serial.print voor debuggen, de seriële plotter	hoofdstuk 29	wat er op de draad gebeurt, en een protocol waar een programma mee overweg kan
de SHT31 met een bibliotheek	hoofdstuk 29	I²C zelf: adressen, pull-ups, bytes en een controlegetal
millis() en taken	hoofdstuk 29	een echte klok met datum
CSV inlezen met csv.DictReader, gemiddelden, matplotlib	hoofdstuk 28	het bestand zelf maken, en duizenden regels opschonen
binair en hexadecimaal, bits en bytes	hoofdstuk 25	BCD, de bits van een teken, twee bytes tot één getal
RC-tijd $\tau = R \cdot C$	hoofdstuk 24	waarom een lange I²C-kabel sterkere pull-ups vraagt

De logger van de klimaatmodule heeft vier soorten ingangen en drie uitgangen. De sensoren voor temperatuur, vocht, energie en tijd hangen aan de **I²C-bus**: twee draden voor allemaal samen. De lichtsensor blijft op A1. De Mega meet om de 2 s, telt de metingen van één minuut samen en schrijft dan één regel: op de **SD-kaart**, en tegelijk over de **seriële poort** naar de computer, waar Python ze kan lezen.

De regel heeft het formaat van hoofdstuk 28, met twee kolommen extra achteraan:

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
```

Omdat je programma's uit hoofdstuk 28 de kolommen bij hun naam opvragen (`rij["zon_Wh"]`), werken ze zonder aanpassing op de nieuwe bestanden. Alleen staat er nu één regel per minuut, in plaats van één per drie uur.



Vier sensorgroepen in, twee kopieën van elke regel uit: één op de kaart, één naar de computer.

UITLEG · 30.3

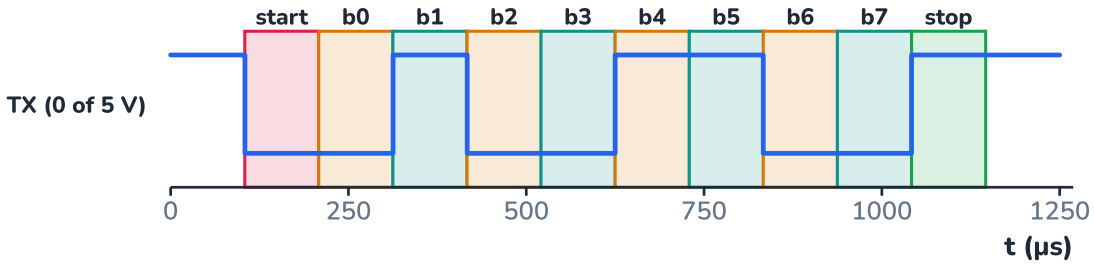
Seriële communicatie: bits op één draad

Een **seriële** verbinding stuurt de bits na elkaar over één draad: de uitgang TX (transmit) van de ene kant hangt aan de ingang RX (receive) van de andere, en omgekeerd; de massa (GND) is gedeeld. Er is geen klokdraad. Beide kanten spreken vooraf een snelheid af, de **baudrate**: bij 9600 baud duurt één bit $1/9600\,\text{s} = 104\,\mu\text{s}$. Het blok in de Mega dat dit doet, heet een **UART**. De Mega heeft er vier: Serial (pin 0 en 1, via de USB-chip naar de computer) en Serial1 tot Serial3 (pin 14 tot 19).

In rust is de lijn **hoog** (5 V). Een teken begint met een **startbit**: de lijn gaat één bittijd laag, zodat de ontvanger weet dat er iets komt en zijn klok gelijkzet. Dan volgen de acht databits, **het minst belangrijke bit eerst** (b0), en ten slotte een **stopbit** (hoog). Dat heet 8N1: 8 databits, geen pariteitsbit, 1 stopbit. Eén teken kost dus 10 bittijden.

Het teken '2' is in ASCII 50, of 0x32 = 0011 0010. Van b0 naar b7 is dat 0, 1, 0, 0, 1, 1, 0, 0: zo zie je het in het tijdsdiagram.

> ⚠ **Alleen gelijke snelheden verstaan elkaar**
>
> Staat de seriële monitor op 9600 en de Mega op 115 200, dan meet de monitor elf keer te trage bits: je ziet onzin. Er is geen melding, alleen rare tekens. Kijk dan eerst naar de baudrate.



Het teken '2' zoals het op de TX-lijn staat: een startbit, b0 tot b7 en een stopbit, elk 104 µs lang.

UITGEWERKT VOORBEELD · 30.4

9600 of 115 200 baud: hoeveel tijd kost een regel?

De logger stuurt elke minuut een regel van 48 tekens, plus `\r\n`. klimaat.ino stuurt elke minuut ook een statusregel van ongeveer 110 tekens. Hoe lang duurt het versturen bij 9600 en bij 115 200 baud? Wat doet `Serial.print` ondertussen?

GEGEVEN

- 8N1: 10 bits per teken
- zendbuffer van Serial: 64 tekens

GEVRAAGD

- tijd per regel
- hoe lang print() wacht

OPLOSSING

1. **Tekens per seconde.** 9600 baud / 10 bits = 960 tekens per seconde; bij 115 200 baud 11 520.
2. **De logregel.** 50 tekens: $50/960 = 52\,\text{ms}$ bij 9600, $50/11\,520 = 4{,}3\,\text{ms}$ bij 115 200.
3. **De buffer.** `Serial.print` zet tekens in een buffer van 64 en keert meteen terug; de UART stuurt ze op de achtergrond. Past een regel in de buffer, dan wacht print() niet. Van de statusregel van 110 tekens moeten er 46 wachten tot er plaats is: bij 9600 baud $46/960 = 48\,\text{ms}$, bij 115 200 $4{,}0\,\text{ms}$.

4 Overzicht:

BAUDRATE	TIJD PER BIT	TEKENS PER SECONDE	REGEL VAN 48 TEKENS + \R\N
9600	104,2 µs	960	52,08 ms
19 200	52,1 µs	1920	26,04 ms
57 600	17,4 µs	5760	8,68 ms
115 200	8,7 µs	11 520	4,34 ms

ANTWOORD

Bij 115 200 baud kost een regel ongeveer 4 ms; bij 9600 baud 52 ms, en dan houdt een lange regel de lus 48 ms op.

Klopt dit? Past het in een minuut? Ruimschoots: zelfs bij 9600 baud is de lijn maar 0,1 % van de tijd bezig. Het probleem is niet de hoeveelheid, maar dat print() de lus kan ophouden. Kies daarom 115 200 baud.

CODE · 30.5

Een eigen protocol: één regel per meting

5/29

Een **protocol** is een afspraak over de vorm van de berichten. Voor de logger kiezen we het eenvoudigste dat een programma aan de andere kant zeker begrijpt:

- elke meting is **één regel**, afgesloten met `\r\n` (println); de regelovergang is het einde van het bericht;
- de velden staan gescheiden door **komma's**, met een decimale **punt**;
- de eerste gewone regel is de **kop** met de namen van de kolommen;
- een regel die met # begint, is commentaar of een melding: een programma slaat hem over;
- geen geldige meting: de cel blijft **leeg**, nooit 0 of –999.

C++ (Arduino) **serieel_csv.ino** 30 regels

```cpp
// serieel_csv.ino - om de 2 s een meting als CSV-regel naar de computer
#include <Wire.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"

Adafruit_SHT31 sht = Adafruit_SHT31();
unsigned long vorige = 0;

void setup() {
  Serial.begin(115200);
  sht.begin(0x44);
  Serial.println("# klimaatmodule: een meting om de 2 s");   // regels met # zijn commentaar
  Serial.println("tijd_s,temperatuur_C,vocht_pct,licht_ruw");  // de kop: namen van de kolommen
}

void loop() {
  unsigned long nu = millis();
  if (nu - vorige >= 2000) {
    vorige += 2000;
    float t, rv;
    bool ok = sht.readBoth(&t, &rv);   // false als de sensor niet (goed) antwoordde
    Serial.print(nu / 1000);
    Serial.print(',');
    if (ok) Serial.print(t, 2);        // geen geldige meting: de cel blijft leeg
    Serial.print(',');
    if (ok) Serial.print(rv, 1);
    Serial.print(',');
    Serial.println(analogRead(LDR));   // println: de regel eindigt op \r\n
  }
```

```
}
```

Wat de computer ontvangt (in de simulatie zit het stekkertje van de SHT31 tussen 7 en 9 s even los):

```
# klimaatmodule: een meting om de 2 s
tijd_s,temperatuur_C,vocht_pct,licht_ruw
2,23.50,64.0,477
4,23.61,63.8,478
6,23.70,63.6,479
8,,,480
10,23.90,63.2,481
12,24.01,63.0,482
```

De regel van 8 s heeft twee lege cellen: de meting mislukte, en dat staat er eerlijk. Een 0 zou een programma als een echte temperatuur van 0 °C lezen.

CODE · 30.6

De regels inlezen in Python

6/29

Aan de kant van de computer leest Python de regels met de bibliotheek **pyserial** (installeren met `pip install pyserial`; je importeert ze als `serial`). Welke poort de Mega heeft, zie je in de Arduino-IDE (*Tools, Port*) of met `python -m serial.tools.list_ports`. Sluit eerst de seriële monitor: een poort kan maar door één programma tegelijk open zijn.

Python **lees_serieel.py** 25 regels

```python
# lees_serieel.py - metingen van de Mega over de seriële poort inlezen en bewaren
import csv
from datetime import datetime

import serial                                   # eenmalig installeren: pip install pyserial

poort = serial.Serial("COM3", 115200, timeout=5)     # Linux: "/dev/ttyACM0", macOS: "/dev/cu.usbmodem…"
kop = None
with open("metingen.csv", "w", newline="", encoding="utf-8") as bestand:
    schrijver = csv.writer(bestand)
    for _ in range(9):                          # voor de test negen regels; echt meten: while True
        regel = poort.readline().decode("utf-8", errors="replace").strip()
        velden = regel.split(",")
        if regel.startswith("#"):
            print("commentaar:", regel)
        elif velden[0] == "tijd_s":             # de kop: de namen van de kolommen
            kop = velden
            schrijver.writerow(["pc_tijd"] + kop)
        elif kop is None or len(velden) != len(kop):
            print("overgeslagen:", regel)
        else:
            nu = datetime.now().strftime("%Y-%m-%d %H:%M:%S")   # de computer kent de datum wel
            schrijver.writerow([nu] + velden)
            print(nu, velden)
poort.close()
```

Uitvoer:

```
overgeslagen: 63.8,478
commentaar: # klimaatmodule: een meting om de 2 s
2026-10-12 10:00:00 ['2', '23.50', '64.0', '477']
2026-10-12 10:00:02 ['4', '23.61', '63.8', '478']
2026-10-12 10:00:04 ['6', '23.70', '63.6', '479']
2026-10-12 10:00:06 ['8', '', '', '480']
```

```
2026-10-12 10:00:08 ['10', '23.90', '63.2', '481']
2026-10-12 10:00:10 ['12', '24.01', '63.0', '482']
```

`readline()` wacht tot er een hele regel binnen is (tot `\n`) en geeft **bytes**; `decode` maakt er tekst van, `strip` haalt `\r\n` weg. De eerste regel is een restje van vóór het openen van de poort: hij heeft te weinig velden en er is nog geen kop, dus hij wordt overgeslagen. De computer zet er zijn eigen datum en tijd voor, want serieel_csv.ino kent alleen de seconden sinds de start.

> **Zo draait dit programma hier**
>
> In dit boek praat het programma met een nagebootste poort die de regels teruggeeft van de simulatie op het vorige blad, en een klok die om de 2 s verspringt. Op je computer gebruik je de echte poort, precies zoals in de code. Let op: het openen van de poort herstart de Mega (de USB-chip trekt de reset even laag); de eerste 2 s komt er niets.

UITLEG · 30.7

I²C: twee draden, veel sensoren

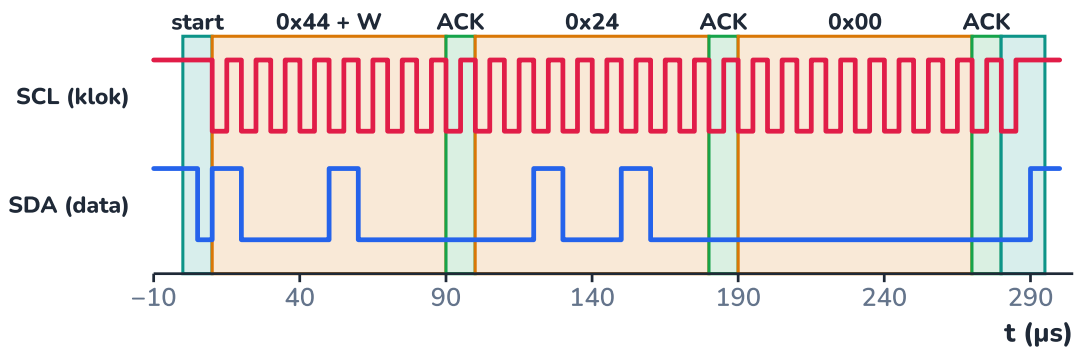
7/29

De Mega heeft maar een handvol seriële poorten, en elke sensor een eigen poort geven lukt niet. **I²C** (spreek uit: i-kwadraat-c) is een **bus**: tot ongeveer honderd chips hangen aan dezelfde twee draden, **SDA** (data) en **SCL** (klok). De Mega is de **master**: hij maakt de klok en begint elk gesprek. De sensoren zijn **slaves**: ze antwoorden alleen als ze aangesproken worden.

Elke chip heeft een **adres** van 7 bits. Een gesprek gaat zo: een **START** (SDA zakt terwijl SCL hoog is), dan het adres met één bit erachter (0 = schrijven, 1 = lezen), dan geeft de aangesproken chip een **ACK** (hij trekt SDA laag bij de negende klokpuls). Daarna volgen de databytes, elk met een ACK, en een **STOP** (SDA stijgt terwijl SCL hoog is). Antwoordt er niemand op een adres, dan blijft SDA hoog: een **NACK**, en `Wire.endTransmission()` geeft 2 terug in plaats van 0.

De chips maken een lijn nooit actief hoog: ze kunnen ze alleen laag trekken of loslaten (**open drain**). Een **pull-upweerstand** naar 5 V trekt de losgelaten lijn weer hoog. Zo kunnen twee chips die tegelijk iets zeggen, nooit kortsluiting maken. De Mega heeft 10 kΩ pull-ups op pin 20 en 21.

adres	chip	wat	hoe
0x40	INA226	stroom en spanning van de zonnepanelen	A0 en A1 aan GND
0x41	INA226	stroom en spanning naar de verbruikers	A0 aan VS
0x44	SHT31	temperatuur en vochtigheid van de lucht	ADDR aan GND (0x45 aan VDD)
0x57	AT24C32	geheugenchipje op het RTC-moduletje (niet gebruikt)	vast op het moduletje
0x68	DS3231	klok met datum	vast



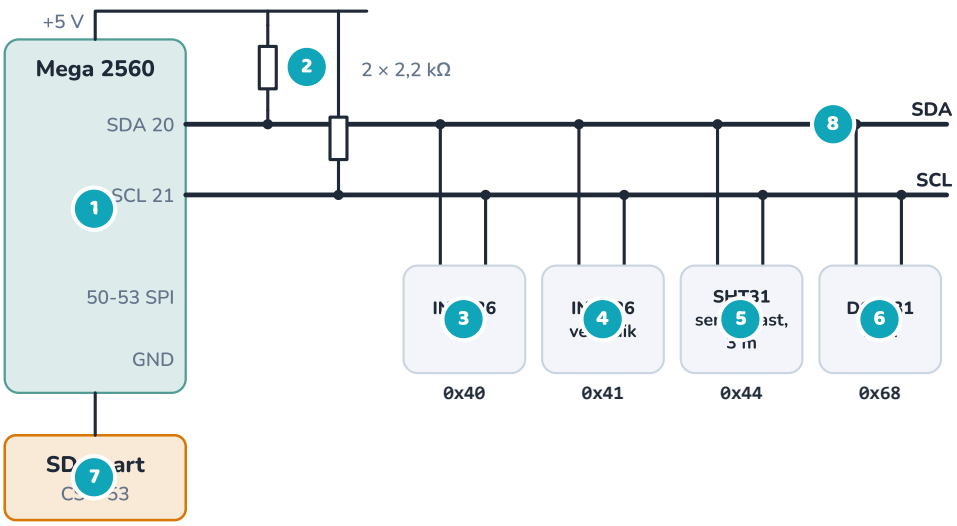
Drie bytes, elk gevolgd door een ACK van de SHT31; het strookje achteraan is de STOP. De data verandert alleen als SCL laag is.

ONTLEED · 30.8

De I²C-bus van de klimaatmodule

8/29

Vier chips op twee draden, en de SD-kaart apart op de SPI-bus. De INA226's en de klok zitten in de elektronicakast naast de Mega; de SHT31 zit in de sensorkast in de kweekruimte, 3 m verder.



Alle chips parallel op SDA en SCL; de adressen maken het verschil.

1 De Mega: master
Maakt de klok op SCL (100 kHz) en begint elk gesprek. De bibliotheek Wire doet het werk; SDA is pin 20, SCL pin 21.

2 Pull-ups
Houden SDA en SCL hoog als niemand ze laag trekt. Door de lange kabel naar de sensorkast komen er 2,2 kΩ bij de 10 kΩ van de Mega (volgend voorbeeld).

3 INA226 zon (0x40)
Meet de spanning over een shunt in de leiding van de laadregelaar: de zonnestroom.

4 INA226 verbruik (0x41)
Dezelfde chip, een ander adres: pin A0 aan de voeding in plaats van aan GND.

5 SHT31 (0x44)
Temperatuur en vocht van de kaslucht, in de sensorkast op 3 m kabel.

6 DS3231 (0x68)
De klok. Op het moduletje zit ook een geheugenchipje op 0x57, dat de scanner ook vindt.

7 SD-kaart op SPI
Niet op I²C: de kaart praat SPI, via pin 50 tot 53. CS (53) kiest de kaart.

8 Eén bus, vier chips
Wat er op de bus gebeurt, horen alle chips; alleen de chip met het juiste adres antwoordt.

CODE · 30.9

Wie zit er op de bus? De I²C-scanner

9/29

Het eerste programma dat je op een nieuwe I²C-bus draait: het probeert elk adres van 1 tot 126 en meldt waar er een ACK komt. Zo zie je meteen of een sensor goed aangesloten is, en welk adres hij echt heeft (dat staat niet altijd juist op het bordje).

C++ (Arduino) **i2c_scanner.ino** 22 regels

```
// i2c_scanner.ino - welke adressen antwoorden op de I2C-bus?
#include <Wire.h>

void setup() {
  Serial.begin(115200);
  Wire.begin();
  int gevonden = 0;
```

```
  for (byte adres = 1; adres < 127; adres++) {
    Wire.beginTransmission(adres);         // alleen het adres sturen, geen data
    if (Wire.endTransmission() == 0) {     // 0 = er kwam een ACK: hier zit een apparaat
      Serial.print("apparaat op 0x");
      if (adres < 16) Serial.print('0');
      Serial.println(adres, HEX);
      gevonden++;
    }
  }
  Serial.print(gevonden);
  Serial.println(" apparaten gevonden");
}

void loop() {
}
```

Uitvoer met de bus van de klimaatmodule:

```
apparaat op 0x40
apparaat op 0x41
apparaat op 0x44
apparaat op 0x57
apparaat op 0x68
5 apparaten gevonden
```

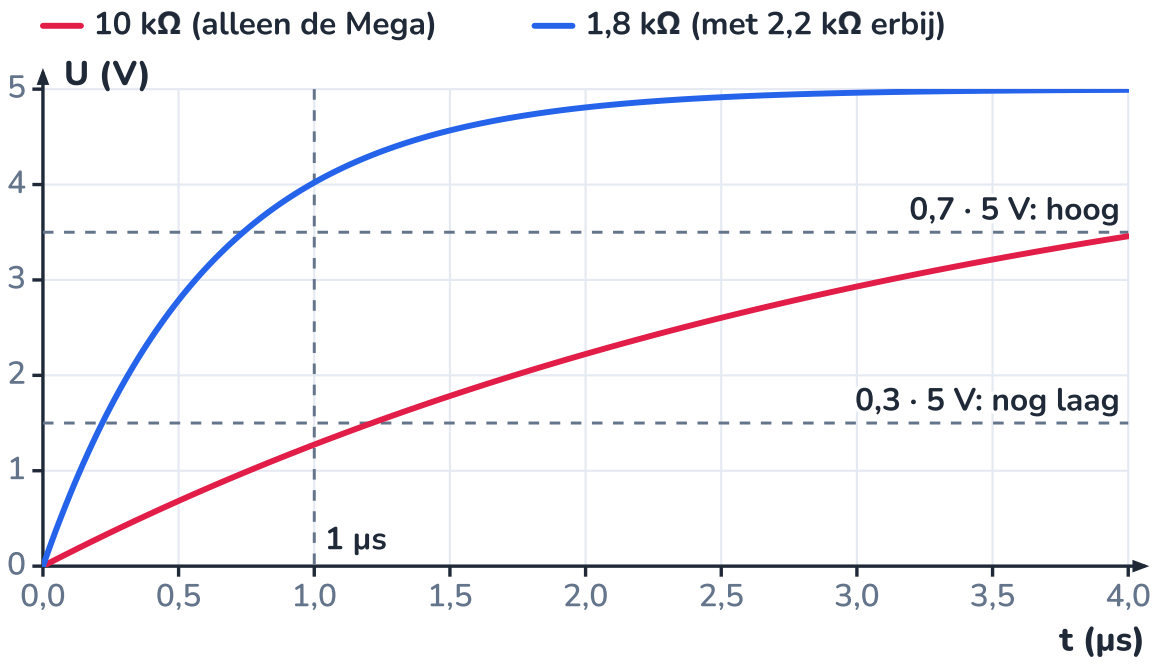
Vijf antwoorden voor vier moduletjes: het klokmoduletje heeft ook een geheugenchip (0x57). Vindt de scanner niets, dan zijn het bijna altijd de draden: SDA en SCL verwisseld, geen gemeenschappelijke GND, of geen voeding op het bordje. Vindt hij een chip op een onverwacht adres, pas dan je code aan of verander de adrespin.

UITGEWERKT VOORBEELD · 30.10

Pull-ups voor een kabel van 3 m

10/29

De SHT31 zit in de sensorkast, 3 m van de Mega. De kabel heeft tussen twee aders een capaciteit van ongeveer 100 pF per meter; de vier chips en de Mega voegen elk ongeveer 10 pF toe. Werkt de bus op 100 kHz met alleen de 10 kΩ van de Mega? Zo niet: welke extra pull-up kies je?



Met 10 kΩ komt SDA na 1 µs niet eens tot 3,5 V; met 1,8 kΩ ruim op tijd.

GEGEVEN

- I²C-norm bij 100 kHz: stijgtijd (van 30 % tot 70 % van 5 V) hoogstens 1000 ns
- een chip moet 3 mA kunnen wegtrekken bij 0,4 V
- 10 kΩ op de Mega

GEVRAAGD

- de stijgtijd met 10 kΩ
- een geschikte extra weerstand

OPLOSSING

1. **Capaciteit.** $C \approx 3 \cdot 100 + 5 \cdot 10 = 350\,\text{pF}$; reken met 340 pF.
2. **Stijgen is een RC-laadcurve** (hoofdstuk 24): $U = 5\,\text{V} \cdot (1 - \text{e}^{-t/\tau})$. Van 30 % naar 70 % duurt $t_r = \tau \ln(0{,}7/0{,}3) = 0{,}847\,\tau$.
3. **Met 10 kΩ:** $\tau = 10\,\text{k}\Omega \cdot 340\,\text{pF} = 3{,}4\,\mu\text{s}$, $t_r = 2{,}9\,\mu\text{s}$: bijna drie keer te traag. De bus haalt dan geen 100 kHz; een bit wordt verkeerd gelezen of de chip antwoordt niet.
4. **Grenzen voor R.** Hoogstens $R = 1\,\mu\text{s}/(0{,}847 \cdot 340\,\text{pF}) = 3{,}5\,\text{k}\Omega$. Minstens $R = (5 - 0{,}4)\,\text{V}/3\,\text{mA} = 1{,}5\,\text{k}\Omega$, anders kan een chip de lijn niet laag genoeg trekken.
5. **Keuze.** 2,2 kΩ extra, parallel met de 10 kΩ: $R = 1/(1/10 + 1/2{,}2) = 1{,}8\,\text{k}\Omega$. Dan $\tau = 0{,}61\,\mu\text{s}$ en $t_r = 0{,}52\,\mu\text{s}$.

ANTWOORD

Met alleen 10 kΩ is de stijgtijd 2,9 µs (te traag). Met 2,2 kΩ extra op SDA en op SCL: 1,8 kΩ en 0,52 µs.

Klopt dit? Tel alle pull-ups op de bus mee: veel breakoutbordjes (SHT31, INA226) hebben er zelf 10 kΩ op. Met twee zulke bordjes erbij zou 10 ∥ 10 ∥ 10 ∥ 2,2 kΩ = 1,4 kΩ te klein worden; dan kies je 3,3 kΩ extra of haal je de pull-ups van een bordje af. Op nog langere kabels gebruik je een I²C-busversterker.

CODE · 30.11

De SHT31 zonder bibliotheek

In hoofdstuk 29 las je de SHT31 met `readBoth`. Zo doet de bibliotheek het, stap voor stap volgens de datasheet:

uit de datasheet (SHT3x-DIS)	waarde
adres	0x44 (ADDR aan GND) of 0x45 (ADDR aan VDD)
commando één meting, hoge herhaalbaarheid, zonder clock stretching	0x24 0x00
duur van die meting	typisch 12,5 ms, hoogstens 15 ms
antwoord	6 bytes: T (2), CRC, RV (2), CRC; eerst de hoge byte
omrekening	$T = -45 + 175 \cdot S_T / 65\,535$ (°C), $RV = 100 \cdot S_{RV} / 65\,535$ (%)
CRC	CRC-8, veelterm 0x31, beginwaarde 0xFF; CRC(0xBE 0xEF) = 0x92
nauwkeurigheid SHT31 (typisch)	±0,2 °C, ±2 % RV
voeding, stroom	2,15 tot 5,5 V; 0,6 mA tijdens een meting, 0,2 µA in rust

Het programma stuurt het commando, wacht 20 ms, leest zes bytes, controleert beide **CRC**'s en rekent om. Een CRC is een controlegetal dat de sensor uit de twee databytes berekent; de Mega rekent het na. Valt er onderweg een bit om (storing op de lange kabel), dan klopt het niet meer en gooit het programma de meting weg.

C++ (Arduino) **sht31_zelf.ino** 73 regels

```cpp
// sht31_zelf.ino - de SHT31 uitlezen zonder bibliotheek, stap voor stap volgens de datasheet
#include <Wire.h>

const byte ADRES = 0x44;            // pin ADDR aan GND (aan VDD: 0x45)
float luchtT = NAN, luchtRV = NAN;

byte crc8(byte a, byte b) {         // CRC-8 over twee bytes: veelterm 0x31, beginwaarde 0xFF
  byte crc = 0xFF;
  byte data[2] = {a, b};
  for (int i = 0; i < 2; i++) {
    crc ^= data[i];
    for (int bit = 0; bit < 8; bit++) {
      if (crc & 0x80) crc = (crc << 1) ^ 0x31;
      else crc = crc << 1;
    }
  }
  return crc;
}

void printHex(byte b) {
  if (b < 16) Serial.print('0');
  Serial.print(b, HEX);
  Serial.print(' ');
}

bool leesSht31() {
  Wire.beginTransmission(ADRES);
  Wire.write(0x24);                 // commando 0x2400: één meting, hoge herhaalbaarheid,
  Wire.write(0x00);                 // zonder 'clock stretching'
  if (Wire.endTransmission() != 0) { // geen ACK: sensor los of verkeerd adres
    Serial.println("geen antwoord op 0x44");
    return false;
  }
  delay(20);                        // de meting duurt tot 15 ms (datasheet)
  if (Wire.requestFrom(ADRES, (byte)6) != 6) {
    Serial.println("minder dan 6 bytes ontvangen");
```

```
    return false;
  }
  byte b[6];
  for (int i = 0; i < 6; i++) b[i] = Wire.read();
  Serial.print("bytes: ");
  for (int i = 0; i < 6; i++) printHex(b[i]);
  if (crc8(b[0], b[1]) != b[2] || crc8(b[3], b[4]) != b[5]) {
    Serial.println(" CRC fout: meting weggegooid");
    return false;
  }
  unsigned int ruwT = ((unsigned int)b[0] << 8) | b[1];      // twee bytes tot één getal van 16 bit
  unsigned int ruwRV = ((unsigned int)b[3] << 8) | b[4];
  luchtT = -45.0 + 175.0 * ruwT / 65535.0;                   // formules uit de datasheet
  luchtRV = 100.0 * ruwRV / 65535.0;
  Serial.print(" -> ");
  Serial.print(ruwT);
  Serial.print(" en ");
  Serial.print(ruwRV);
  Serial.print(" -> ");
  Serial.print(luchtT, 2);
  Serial.print(" C, ");
  Serial.print(luchtRV, 1);
  Serial.println(" %");
  return true;
}

void setup() {
  Serial.begin(115200);
  Wire.begin();
  Serial.print("test CRC(BE EF) = ");           // het voorbeeld uit de datasheet: 92
  Serial.println(crc8(0xBE, 0xEF), HEX);
}

void loop() {
  leesSht31();
  delay(2000);                                  // testschets: hier mag het
}
```

Uitvoer: eerst de test met het voorbeeld uit de datasheet, dan drie metingen; bij de derde valt een bit om in de tweede byte, bij de vierde is de sensor losgekoppeld:

```
test CRC(BE EF) = 92
bytes: 64 29 59 9C ED F5  -> 25641 en 40173 -> 23.47 C, 61.3 %
bytes: 64 31 A3 9C AB 6E  -> 25649 en 40107 -> 23.49 C, 61.2 %
bytes: 64 3C 2B 9C 6A 18  CRC fout: meting weggegooid
geen antwoord op 0x44
```

Narekenen: 0x6429 = 25 641, en $-45 + 175 \cdot 25\,641 / 65\,535 = 23{,}47$ °C. Twee bytes worden één getal met `(hoog << 8) | laag`: de hoge byte schuift acht plaatsen naar links, zoals 0x64 · 256. Wie de fout negeert, krijgt bij een losse sensor 0xFF 0xFF (Wire.read() geeft −1): dat is precies 130,0 °C en 100,0 %. Je vindt die waarden verderop terug in echte logbestanden.

UITLEG · 30.12

Tijd: waarom millis() geen klok is

12/29

Een meting zonder tijdstip is weinig waard. `millis()` lijkt een klok, maar volstaat niet:

- ze kent **geen datum**, alleen de tijd sinds de start;
- na een **stroomonderbreking of reset** begint ze weer bij 0;
- ze is **onnauwkeurig**: de 16 MHz van de Mega komt van een keramische resonator die tot ±0,5 % mag afwijken, in het slechtste geval ruim 7 minuten per dag, en ze verloopt met de temperatuur;

- ze loopt na 49,7 dagen over.

Een **RTC** (real-time clock) lost dat op. De **DS3231** heeft een eigen kwartskristal met temperatuurcorrectie: ±2 ppm tussen 0 en 40 °C, hoogstens ongeveer een minuut per jaar. Een knoopcel houdt hem jaren aan de gang als de module geen stroom heeft (hij verbruikt dan ongeveer een microampère). De Mega leest hem via I²C op adres 0x68. De tijd staat in registers 0x00 tot 0x06 in **BCD** (binair gecodeerd decimaal): elk cijfer apart in 4 bits, zodat 34 minuten als 0x34 = 0011 0100 staat.

Welke tijd? In België springt de klok twee keer per jaar een uur. In het najaar komt 2:30 dan twee keer voor, en in de lente ontbreekt een uur: een logboek met gewone kloktijd heeft dubbele en ontbrekende regels. De logger loopt daarom het hele jaar op **wintertijd** (UTC+1). In de zomer loopt hij dus een uur achter op je horloge: 12:00 in het logboek is 13:00 zomertijd.

Eén keer zetten, dan jaren niet meer

RTClib kan de klok zetten op het moment van compileren: `rtc.adjust(DateTime(F(__DATE__), F(__TIME__)))`. Dat is de tijd van je computer, met zomertijd en een paar seconden te laat. logger.ino laat je de klok zetten met een opdracht vanaf de computer; opdracht 30.14 werkt dat uit.

12:34:56 op maandag 12-10-2026 in de registers van de DS3231

Elk vakje is een register van één byte; in hexadecimaal lees je de cijfers gewoon af.

CODE · 30.13

De klok lezen: DS3231 en BCD

13/29

Twee manieren: met de bibliotheek **RTClib** (Adafruit, via de bibliotheekbeheerder), en zelf, met de registers. Het patroon 'registerwijzer zetten, dan bytes lezen' gebruiken ook de INA226 en veel andere I²C-chips.

C++ (Arduino) **klok.ino** 69 regels

```cpp
// klok.ino - de DS3231 lezen: met RTClib, en daarna de registers zelf (BCD)
#include <Wire.h>
#include <RTClib.h>

RTC_DS3231 rtc;

byte bcdNaarGetal(byte b) {            // 0x34 -> 3 * 10 + 4 = 34
  return (b >> 4) * 10 + (b & 0x0F);
}

void printTwee(int x) {
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void setup() {
  Serial.begin(115200);
  Wire.begin();
  if (!rtc.begin()) {
    Serial.println("# geen DS3231 op adres 0x68");
```

```
    return;
  }
  if (rtc.lostPower()) Serial.println("# de klok heeft stilgestaan: stel ze opnieuw in");

  DateTime nu = rtc.now();            // 1. met de bibliotheek
  Serial.print("RTClib:    ");
  Serial.print(nu.year());
  Serial.print('-');
  printTwee(nu.month());
  Serial.print('-');
  printTwee(nu.day());
  Serial.print(' ');
  printTwee(nu.hour());
  Serial.print(':');
  printTwee(nu.minute());
  Serial.print(':');
  printTwee(nu.second());
  Serial.println();

  Wire.beginTransmission(0x68);       // 2. zelf: registerwijzer op 0x00 ...
  Wire.write(0x00);
  Wire.endTransmission();
  Wire.requestFrom(0x68, 7);          // ... en zeven registers lezen
  byte r[7];
  Serial.print("registers:");
  for (int i = 0; i < 7; i++) {
    r[i] = Wire.read();
    Serial.print(" ");
    if (r[i] < 16) Serial.print('0');
    Serial.print(r[i], HEX);
  }
  Serial.println();
  Serial.print("zelf:      20");
  printTwee(bcdNaarGetal(r[6]));                // jaar
  Serial.print('-');
  printTwee(bcdNaarGetal(r[5] & 0x1F));         // maand (bit 7 is de eeuw)
  Serial.print('-');
  printTwee(bcdNaarGetal(r[4]));                // dag
  Serial.print(' ');
  printTwee(bcdNaarGetal(r[2] & 0x3F));         // uur (24-uursmodus)
  Serial.print(':');
  printTwee(bcdNaarGetal(r[1]));
  Serial.print(':');
  printTwee(bcdNaarGetal(r[0] & 0x7F));
  Serial.println();
}

void loop() {
}
```

Uitvoer:

```
RTClib:    2026-10-12 12:34:56
registers: 56 34 12 01 12 10 26
zelf:      2026-10-12 12:34:56
```

`bcdNaarGetal` neemt de hoge helft (`b >> 4`, de tientallen) en de lage helft (`b & 0x0F`, de eenheden). In twee registers staan extra bits die je eerst wegmaskert: bit 7 van de maand is de eeuw, bit 6 van de uren kiest de 12-uursmodus. Register 3 is de dag van de week (RTClib: maandag = 1). RTClib doet dit allemaal voor je en geeft een `DateTime`.

UITLEG · 30.14

De SD-kaart: SPI, bestanden en flush

Een SD-kaart praat **SPI**: een snelle bus met een klokdraad (SCK), een draad van de Mega naar de kaart (MOSI), een terug (MISO) en een selectiedraad (CS) per apparaat. Op de Mega zijn dat pin 52, 51, 50 en 53. Een kaart werkt op 3,3 V: gebruik een moduletje met een spanningsregelaar en een niveauomzetter. SPI werk je in Expert (hoofdstuk E37) verder uit; de bibliotheek **SD** regelt het.

De bibliotheek leest kaarten met een **FAT16**- of **FAT32**-bestandssysteem (formatteer ze met de SD Card Formatter van de SD Association) en kent alleen korte **8.3-namen**: hoogstens acht tekens, een punt en drie. `2026-10-12.CSV` mislukt, `20261012.CSV` lukt. De logger maakt één bestand per dag, in een map LOG: een map omdat de hoofdmap van een FAT16-kaart maar 512 namen kan bevatten, en per dag omdat een klein bestand makkelijk te kopiëren is en bij een fout niet alles meesleurt.

Een kaart schrijft in **blokken van 512 bytes**. `println` vult eerst een buffer in het SRAM; pas bij `flush()` of `close()` gaat alles echt naar de kaart en komt de nieuwe bestandslengte in de map. De logger opent het bestand elke minuut, schrijft één regel en sluit het: zo verlies je bij een stroomonderbreking hoogstens de lopende minuut.

Een kaart uittrekken terwijl de logger schrijft

Trek de kaart er alleen uit als de logger niet schrijft (vlak na een volle minuut is veilig), of schakel eerst de voeding uit. Hoofdstuk 24 berekende dat een supercondensator de Mega 3,1 s in leven houdt: genoeg om bij een stroomonderbreking het bestand netjes te sluiten.

Valt de stroom uit tussen 1 en 3, dan staat de nieuwe bestandslengte nog niet in de map: alles sinds de vorige flush() is weg, ook wat al in volle blokken op de kaart stond.

Een regel gaat eerst naar een buffer; pas flush() of close() legt hem echt vast.

UITGEWERKT VOORBEELD · 30.15

Hoeveel dagen past er op 1 GB?

De logger schrijft elke minuut een regel. Hoe groot wordt een dagbestand, hoe lang doet een kaart van 1 GB ermee, en wat gebeurt er als ze toch vol is?

GEGEVEN

- een regel is 44 tot 49 tekens plus \r\n (één byte per teken)
- de kop: 71 tekens plus \r\n
- 1440 minuten per dag

GEVRAAGD

- bytes per dag
- aantal dagen op 1 GB

OPLOSSING

1 **Per dag, geschat.** Met gemiddeld 47,5 bytes per regel: $1440 \cdot 47{,}5 + 73 \approx 68\,500$ bytes.

2 **Nagemeten** op de zeven dagbestanden van de testweek (verderop in dit hoofdstuk): gemiddeld 68 514 bytes per dag.

3 **1 GB.** $10^9 / 68\,514 \approx 14\,596$ dagen, of ongeveer 40 jaar. Een bestand neemt op de kaart wel een geheel aantal clusters in (enkele kB tot 32 kB); per bestand gaat hoogstens één cluster verloren, dat verandert weinig.

4 **Kaart vol.** Dan schrijft `println` minder tekens dan gevraagd. logger.ino vergelijkt het aantal met dat van dezelfde regel naar Serial, meldt het en zet de statusled aan. De simulatie met een 'kaart' van 200 bytes:

```
# SD-kaart klaar
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 06:00,21.0,55.0,0.09,0.00,4004,13.12
2026-10-12 06:01,21.0,55.0,0.09,0.00,4004,13.12
2026-10-12 06:02,21.0,55.0,0.09,0.00,4004,13.12
# SD: kaart vol, regel niet bewaard
```

ANTWOORD

Ongeveer 68,5 kB per dag; 1 GB volstaat voor ruim 40 jaar. De kaart is niet de beperking.

Klopt dit? Wat wel beperkt: een kaart kan stukgaan en elke schrijfbeurt slijt de cellen een beetje (de kaart spreidt dat zelf over de hele kaart). Kopieer de bestanden geregeld naar de computer, en kies voor een logger die jaren draait een kaart van een bekend merk, liefst een 'high endurance'-type.

UITLEG · 30.16

Energie meten met de INA226

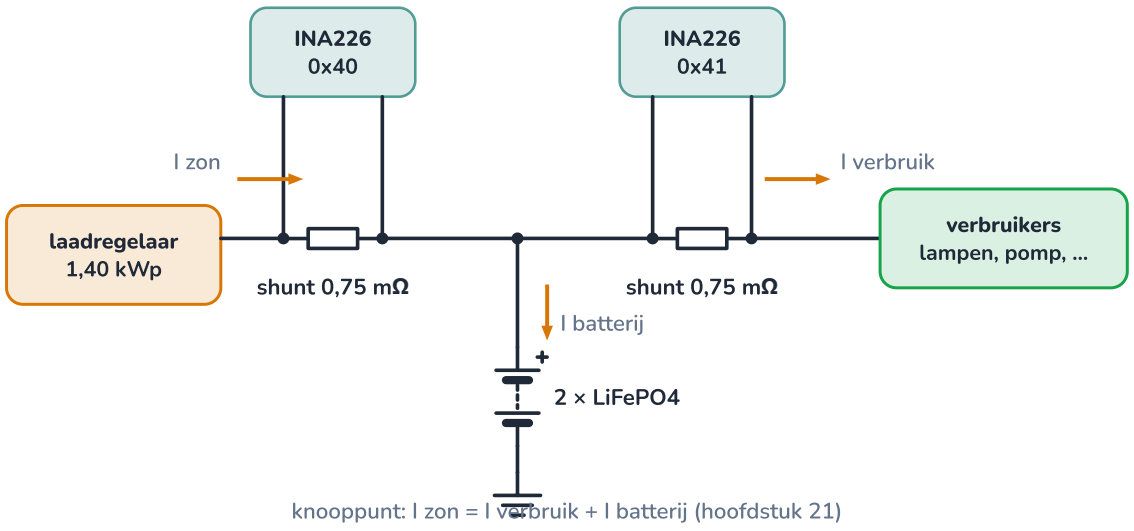
16/29

Voor verbruik_Wh en zon_Wh moet de logger stroom meten. De ACS712-20A op A4 (hoofdstuk 16) meet tot 20 A, maar op een heldere middag laden de panelen de batterij met tot $(1000 - 20)\,\mathrm{W} / 13{,}3\,\mathrm{V} \approx 74\,\mathrm{A}$: daar loopt hij tegen zijn grens. De logger gebruikt daarom twee **INA226**-chips, elk met een **shunt** van 100 A / 75 mV (0,75 mΩ): één in de leiding van de laadregelaar, één in de leiding naar de verbruikers. De batterijstroom volgt uit de knooppuntwet (hoofdstuk 21).

De INA226 meet de kleine spanning over de shunt (tot 81,92 mV, in stappen van 2,5 µV) en de spanning van de rail (in stappen van 1,25 mV), en geeft ze via I²C. Twee dezelfde chips op één bus kan, omdat je met de pinnen A0 en A1 zestien adressen kan kiezen.

register	naam	per stap	in logger.ino
0x00	configuratie		0x4927: 128 metingen middelen, 1,1 ms per conversie, continu
0x01	shuntspanning (met teken)	2,5 µV	stroom = stappen · 2,5 µV / 0,75 mΩ = stappen · 3,33 mA
0x02	busspanning	1,25 mV	de spanning van de 12,8 V-rail
0xFE	fabrikant		0x5449 ('TI'): handig om te testen of je de juiste chip leest

De groeilampen worden met PWM gedimd (490 Hz, hoofdstuk 25): de stroom springt 490 keer per seconde tussen 0 en 15 A. Eén meting van 1,1 ms zou op een willekeurig moment vallen (aliasing). Daarom middelt de INA226 128 metingen van spanning en stroom: $128 \cdot 2 \cdot 1{,}1\,\mathrm{ms} = 282\,\mathrm{ms}$, bijna 140 PWM-perioden. De logger rekent zo: $P = U \cdot I$ om de 2 s, het gemiddelde over een minuut, maal 1/60 h = Wh.



Twee metingen, en de derde stroom volgt uit de knooppuntwet. De minleidingen zijn niet getekend.

CODE · 30.17

logger.ino: de datalogger van de klimaatmodule

17/29

Alles samen: om de 2 s meten, per minuut optellen, bij een nieuwe minuut één regel naar de computer en naar de kaart. De regel wordt geschreven door één functie, `schrijfRegel(Print &uit)`, die zowel `Serial` als een `File` aankan: beide zijn een `Print`, iets waar je print() op kan doen. Zo zijn de twee kopieën gegarandeerd gelijk. Zet `pinnen.h` uit hoofdstuk 29 in dezelfde map, en installeer RTClib en Adafruit SHT31 met de bibliotheekbeheerder.

C++ (Arduino) **logger.ino** 198 regels

```cpp
// logger.ino - de datalogger van de klimaatmodule (Arduino Mega 2560, hoofdstuk 30)
// Om de 2 s meten; elke minuut één CSV-regel op de SD-kaart (map LOG, één bestand per dag) en naar de computer.
// I2C (SDA 20, SCL 21): INA226 zon 0x40, INA226 verbruik 0x41, SHT31 0x44, DS3231 0x68.
// SD-kaart op SPI: 50 MISO, 51 MOSI, 52 SCK, 53 CS. De klok loopt het hele jaar op wintertijd (UTC+1).
// De klok zetten vanaf de computer: stuur  T2026-06-01 06:00:00  en Enter.
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <RTClib.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"

const unsigned long MEET_MS = 2000;
const int SD_CS = 53;
const byte INA_ZON = 0x40;            // INA226 in de leiding van de laadregelaar (A0 en A1 aan GND)
const byte INA_VERBRUIK = 0x41;       // INA226 in de leiding naar de verbruikers (A0 aan VS)
const float R_SHUNT = 0.00075;        // ohm: shunt van 100 A / 75 mV
const char KOP[] = "tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V";

RTC_DS3231 rtc;
Adafruit_SHT31 sht = Adafruit_SHT31();
bool kaartOk = false;
unsigned long vorigeMeting = 0;
DateTime minuut;                      // de minuut waarvan we nu metingen verzamelen
bool bezig = false;

float somT = 0, somRV = 0, somLux = 0, somZon = 0, somVerbruik = 0, somU = 0;   // sommen over één minuut
int nT = 0, nMeting = 0, nVermogen = 0;

char opdracht[32];                    // wat de computer stuurt
int lengte = 0;
```

```
32
33  // ------------------------------------------------------------- sensoren
34  float daglichtLux() {                // LDR onder een kapje (hoofdstuk 22 en 25)
35    long som = 0;
36    for (int i = 0; i < 16; i++) som += analogRead(LDR);
37    int n = som / 16;
38    if (n < 1) n = 1;
39    float r = 1000.0 * (1024.0 - n) / n;
40    return 10.0 * pow(r / 15000.0, -1.0 / 0.7) / 0.10;
41  }
42
43  void inaSchrijf(byte adres, byte reg, unsigned int waarde) {
44    Wire.beginTransmission(adres);
45    Wire.write(reg);                   // eerst het register ...
46    Wire.write(waarde >> 8);           // ... dan de hoge byte ...
47    Wire.write(waarde & 0xFF);         // ... en de lage byte
48    Wire.endTransmission();
49  }
50
51  long inaLees(byte adres, byte reg) {  // 16 bits uit een register, of -1 bij een fout
52    Wire.beginTransmission(adres);
53    Wire.write(reg);                   // de registerwijzer zetten
54    if (Wire.endTransmission() != 0) return -1;
55    if (Wire.requestFrom(adres, (byte)2) != 2) return -1;
56    byte hoog = Wire.read();           // eerst de hoge byte; twee aparte regels,
57    byte laag = Wire.read();           // zodat de volgorde vastligt
58    return ((long)hoog << 8) | laag;
59  }
60
61  float inaSpanning(byte adres) {      // busspanning: 1,25 mV per stap
62    long bus = inaLees(adres, 0x02);
63    if (bus < 0) return NAN;
64    return bus * 0.00125;
65  }
66
67  float inaVermogen(byte adres) {      // W, of NAN bij een fout
68    long shunt = inaLees(adres, 0x01);  // shuntspanning: 2,5 uV per stap, met teken
69    float u = inaSpanning(adres);
70    if (shunt < 0 || isnan(u)) return NAN;
71    float stroom = (int16_t)shunt * 0.0000025 / R_SHUNT;    // A
72    return u * stroom;
73  }
74
75  // ------------------------------------------------------------- meten en schrijven
76  void meten() {
77    float t, rv;
78    if (sht.readBoth(&t, &rv)) {       // alleen geldige metingen tellen mee
79      somT += t;
80      somRV += rv;
81      nT++;
82    }
83    somLux += daglichtLux();
84    float pZon = inaVermogen(INA_ZON);
85    float pVerbruik = inaVermogen(INA_VERBRUIK);
86    float u = inaSpanning(INA_VERBRUIK);
87    if (!isnan(pZon) && !isnan(pVerbruik) && !isnan(u)) {
88      somZon += pZon;
89      somVerbruik += pVerbruik;
90      somU += u;
91      nVermogen++;
92    }
93    nMeting++;
94  }
95
96  void wisSommen() {
97    somT = somRV = somLux = somZon = somVerbruik = somU = 0;
98    nT = nMeting = nVermogen = 0;
99  }
```

```
100
101 size_t schrijfRegel(Print &uit) {     // één CSV-regel, naar de kaart of naar de computer
102   size_t n = 0;                       // hoeveel tekens er echt geschreven zijn
103   char tijd[17];
104   snprintf(tijd, sizeof(tijd), "%04d-%02d-%02d %02d:%02d", minuut.year(), minuut.month(), minuut.day(),
105            minuut.hour(), minuut.minute());
106   n += uit.print(tijd);
107   n += uit.print(',');
108   if (nT > 0) n += uit.print(somT / nT, 1);               // geen geldige meting: lege cel
109   n += uit.print(',');
110   if (nT > 0) n += uit.print(somRV / nT, 1);
111   n += uit.print(',');
112   if (nVermogen > 0) n += uit.print(somVerbruik / nVermogen / 60.0, 2);   // gemiddeld vermogen x 1/60 h = Wh
113   n += uit.print(',');
114   if (nVermogen > 0) n += uit.print(somZon / nVermogen / 60.0, 2);
115   n += uit.print(',');
116   if (nMeting > 0) n += uit.print(somLux / nMeting, 0);
117   n += uit.print(',');
118   if (nVermogen > 0) n += uit.print(somU / nVermogen, 2);
119   n += uit.println();
120   return n;
121 }
122
123 void opslaan() {
124   size_t lang = schrijfRegel(Serial);                    // altijd naar de computer
125   if (!kaartOk) return;
126   char naam[20];
127   snprintf(naam, sizeof(naam), "LOG/%04d%02d%02d.CSV", minuut.year(), minuut.month(), minuut.day());
128   bool nieuw = !SD.exists(naam);
129   File f = SD.open(naam, FILE_WRITE);                    // FILE_WRITE schrijft achteraan bij
130   if (!f) {
131     Serial.println("# SD: kan het bestand niet openen");
132     digitalWrite(STATUSLED, HIGH);
133     return;
134   }
135   if (nieuw) f.println(KOP);                             // een nieuwe dag: eerst de kop
136   if (schrijfRegel(f) < lang) {                          // niet alles paste nog op de kaart
137     Serial.println("# SD: kaart vol, regel niet bewaard");
138     digitalWrite(STATUSLED, HIGH);
139   }
140   f.close();                                             // close() schrijft alles echt weg
141 }
142
143 void leesOpdracht() {                // bv. "T2026-06-01 06:00:00": de klok zetten
144   while (Serial.available() > 0) {
145     char c = Serial.read();
146     if (c == '\n' || c == '\r') {
147       opdracht[lengte] = '\0';
148       int j, mo, d, u, mi, s;
149       if (lengte > 0 && sscanf(opdracht, "T%d-%d-%d %d:%d:%d", &j, &mo, &d, &u, &mi, &s) == 6) {
150         rtc.adjust(DateTime(j, mo, d, u, mi, s));
151         Serial.println("# klok gezet");
152         wisSommen();                                     // de lopende minuut hoort bij de oude tijd:
153         bezig = false;                                   // niet bewaren, opnieuw beginnen
154       } else if (lengte > 0) {
155         Serial.println("# onbekende opdracht");
156       }
157       lengte = 0;
158     } else if (lengte < (int)sizeof(opdracht) - 1) {
159       opdracht[lengte] = c;
160       lengte++;
161     }
162   }
163 }
164
165 // ------------------------------------------------------------- setup en loop
166 void setup() {
167   Serial.begin(115200);
```

```
  pinMode(STATUSLED, OUTPUT);
  Wire.begin();
  if (!rtc.begin()) Serial.println("# ALARM: geen klok (DS3231) op 0x68");
  if (rtc.lostPower()) Serial.println("# de klok heeft stilgestaan: stuur T jjjj-mm-dd uu:mm:ss");
  if (!sht.begin(0x44)) Serial.println("# ALARM: geen SHT31 op 0x44");
  inaSchrijf(INA_ZON, 0x00, 0x4927);                      // 128 metingen middelen: het gemiddelde over de PWM
  inaSchrijf(INA_VERBRUIK, 0x00, 0x4927);
  kaartOk = SD.begin(SD_CS);
  if (kaartOk && !SD.exists("LOG")) SD.mkdir("LOG");
  Serial.println(kaartOk ? "# SD-kaart klaar" : "# ALARM: geen SD-kaart, alleen naar de computer");
  digitalWrite(STATUSLED, kaartOk ? LOW : HIGH);          // led aan = probleem met de kaart
  Serial.println(KOP);
}

void loop() {
  unsigned long nu = millis();
  leesOpdracht();
  if (nu - vorigeMeting < MEET_MS) return;
  vorigeMeting += MEET_MS;
  DateTime t = rtc.now();
  DateTime begin(t.year(), t.month(), t.day(), t.hour(), t.minute(), 0);   // het begin van deze minuut
  if (!bezig) {
    minuut = begin;
    bezig = true;
  } else if (begin.unixtime() != minuut.unixtime()) {     // een nieuwe minuut: de vorige is af
    opslaan();
    wisSommen();
    minuut = begin;
  }
  meten();
}
```

Een test in de simulatie: zondag 11 oktober 2026 van 23:58 tot 00:02, met in de minuut 23:59 een losse SHT31 en om 00:00 de pomp die aanslaat. De seriële monitor:

```
# SD-kaart klaar
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-11 23:58,18.1,83.8,0.09,0.00,0,13.25
2026-10-11 23:59,,,0.09,0.00,0,13.25
2026-10-12 00:00,18.0,83.9,0.29,0.00,0,13.24
2026-10-12 00:01,18.0,84.0,0.29,0.00,0,13.24
```

En wat er daarna op de kaart staat, twee dagbestanden met elk een kop:

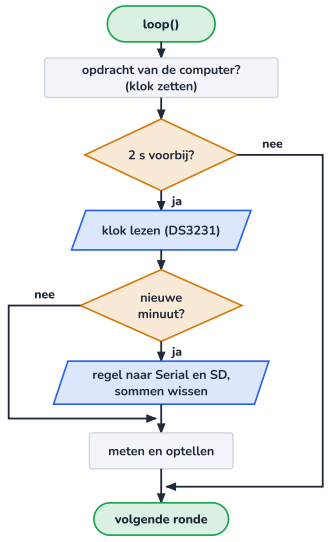
```
== LOG/20261011.CSV (157 bytes)
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-11 23:58,18.1,83.8,0.09,0.00,0,13.25
2026-10-11 23:59,,,0.09,0.00,0,13.25
== LOG/20261012.CSV (165 bytes)
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 00:00,18.0,83.9,0.29,0.00,0,13.24
2026-10-12 00:01,18.0,84.0,0.29,0.00,0,13.24
```

Het verbruik springt om 00:00 van 0,09 naar 0,29 Wh per minuut: 5,5 W wordt 17,5 W als de pomp (12 W) aanslaat. De minuut 23:59 heeft lege cellen voor temperatuur en vocht, maar wel energie: die komt van andere chips. De eerste regel komt bij de eerste minuutwissel; start de logger midden in een minuut, dan is die eerste regel een gemiddelde over minder metingen.

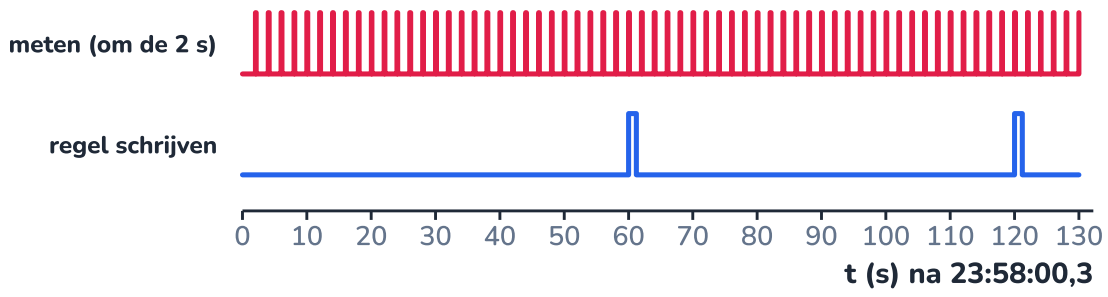
De klok zetten vanaf de computer

Stuur in de seriële monitor `T2026-10-11 20:00:00` (met Enter). logger.ino leest de tekens met `Serial.read()` tot de regelovergang en ontleedt ze met `sscanf`. Zonder knoopcel, of na een lege knoopcel, meldt de logger dat de klok stilstond:

```
# de klok heeft stilgestaan: stuur T jjjj-mm-dd uu:mm:ss
# SD-kaart klaar
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
# klok gezet
# onbekende opdracht
2026-10-11 20:00,21.0,55.0,0.09,0.00,4004,13.12
```



loop(): elke 2 s de klok lezen; bij een nieuwe minuut eerst de vorige wegschrijven.



De meettaak om de 2 s en het schrijven bij het begin van elke minuut.

NASLAG · 30.18

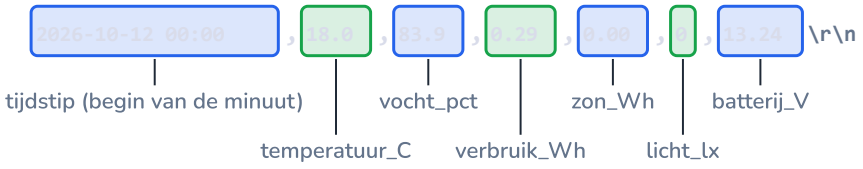
Het CSV-formaat van de klimaatmodule

18/29

Het formaat van hoofdstuk 28, met twee kolommen achteraan. Deze afspraken maken het bestand bruikbaar voor elk programma, ook over tien jaar:

afspraak	waarom
de eerste regel is de kop, met de eenheid in de naam (temperatuur_C)	een programma vraagt kolommen op bij naam; de eenheid gaat nooit verloren

afspraak	waarom
tijdstip als jjjj-mm-dd uu:mm, in wintertijd (UTC+1)	sorteert vanzelf juist; geen dubbele of ontbrekende uren bij de overgang naar zomertijd
het tijdstip is het begin van de minuut	zoals in hoofdstuk 28 (een blok heet naar zijn begin)
komma tussen de velden, punt als decimaalteken	Python leest het zonder omzetten (float("23.5"))
lege cel = geen geldige meting	een 0 of –999 zou meetellen in een gemiddelde
nieuwe kolommen alleen achteraan	oude programma's met csv.reader en een kolomnummer blijven werken
energie als Wh per minuut, niet als vermogen	optellen geeft meteen Wh per uur, per dag, per week
één bestand per dag: LOG/JJJJMMDD.CSV	8.3-namen; kleine bestanden, makkelijk te kopiëren



Zeven velden, zes komma's en een regelovergang: meer is een logregel niet.

CODE · 30.19

Testdata: een week in het formaat van de logger

19/29

Om de analyse te schrijven, hoef je niet een week te wachten. Ingenieurs maken eerst **testdata**: een model dat bestanden maakt in precies het formaat van het echte toestel, met de fouten die je verwacht. Dit programma maakt zeven dagbestanden voor 12 tot 18 oktober 2026. Het model gebruikt wat je al weet: een zonneprofiel per dag (van helder tot grijs), de pomp 10 minuten per uur, lampen die dimmen met het daglicht en het energiebeheer van hoofdstuk 20 (onder 40 % alleen op zon, onder 20 % uit). En het zet er drie soorten fouten in, zoals ze in echte logbestanden voorkomen.

Python **maak_testdata.py** 50 regels

```python
# maak_testdata.py - een week testdata in het formaat van de logger (een model, geen echte metingen)
import math
import os
import random

rng = random.Random(2026)                       # vast zaadje: telkens dezelfde 'metingen'
os.makedirs("LOG", exist_ok=True)
KOP = "tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V"
WEER = [0.95, 0.55, 0.2, 0.4, 0.8, 0.65, 0.3]    # 1 = helder; maandag 12 tot zondag 18 oktober 2026
soc = 0.65                                      # laadtoestand van de batterij (0 tot 1)
ruis = 0.0

for dag in range(7):
    w = WEER[dag]
    with open(f"LOG/202610{dag + 12}.CSV", "w", newline="") as f:
        f.write(KOP + "\r\n")
        for m in range(1440):
            u = m / 60                                          # uur, wintertijd (UTC+1)
            hoogte = max(0.0, math.sin(math.pi * (u - 7.1) / 10.7))    # zon op 7:06, onder 17:48
            lux = 40000 * w * hoogte ** 1.3 * (1 + (1 - w) * (rng.random() - 0.5))
            paneel = math.cos(math.pi * (u - 12.45) / 10.8) ** 1.3 if abs(u - 12.45) < 5.4 else 0.0
            zon = 1000 * w * paneel * (1 + 0.6 * (1 - w) * (rng.random() - 0.5))   # W die de panelen kunnen geven
            basis = 3.5 + (12 if m % 60 < 10 else 0) + (5 if 8 <= u < 20 else 2)   # sensoren, pomp, ventilatoren
            wens = 190 * min(max((200 - lux / 54) / 200, 0), 1) if 6 <= u < 22 else 0.0
            if soc >= 0.40:
```

```
                lampen = wens                                       # energiebeheer (hoofdstuk 20)
            elif soc >= 0.20:
                lampen = min(wens, max(zon - basis, 0))             # alleen op zon
            else:
                lampen = 0.0
            verbruik = basis + lampen
            if soc >= 1.0 and zon > verbruik:
                zon = verbruik                                      # batterij vol: de regelaar knijpt af
            stroom = (zon - verbruik) / 13.2                        # A, positief = laden
            soc = min(1.0, soc + stroom * 13.2 * (0.95 if stroom > 0 else 1) / 60 / 3072)
            bat = 13.0 + 0.4 * soc + 0.015 * stroom
            ruis = 0.9 * ruis + rng.gauss(0, 0.05)
            temp = 18.0 + (6.0 * w + 0.8) * max(0.0, math.sin(math.pi * (u - 9) / 9.5)) ** 1.2 + 1.5 * lampen / 190 +
ruis
            vocht = 84 - 1.6 * (temp - 18) + rng.gauss(0, 0.4)
            tijd = f"2026-10-{dag + 12} {m // 60:02d}:{m % 60:02d}"
            t_tekst, v_tekst = f"{temp:.1f}", f"{vocht:.1f}"
            # fouten die in echte logbestanden voorkomen:
            if (dag, m) in ((0, 552), (0, 553), (1, 605)):
                t_tekst, v_tekst = "130.0", "100.0"                 # versie 1 las 0xFF 0xFF: sensor antwoordde
niet
            if dag == 1 and m == 840:
                continue                                            # 14:00: nieuwe software opgeladen
            if dag == 2 and 707 <= m <= 718:
                t_tekst, v_tekst = "", ""                           # stekkertje los: versie 2 laat de cel leeg
            f.write(f"{tijd},{t_tekst},{v_tekst},{verbruik / 60:.2f},{zon / 60:.2f},{lux:.0f},{bat:.2f}\r\n")
print("zeven bestanden in de map LOG")
```

Uitvoer: zeven bestanden in de map LOG. Het begin van het eerste bestand:

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 00:00,17.9,84.1,0.29,0.00,0,13.24
2026-10-12 00:01,17.9,83.9,0.29,0.00,0,13.24
2026-10-12 00:02,17.9,83.5,0.29,0.00,0,13.24
2026-10-12 00:03,17.9,84.7,0.29,0.00,0,13.24
2026-10-12 00:04,17.8,84.5,0.29,0.00,0,13.24
…
```

Het zijn **geen metingen**, maar een model met toevalsruis. Met dezelfde startwaarde van de toevalsgenerator (2026) krijg je precies dezelfde bestanden: download het programma en je werkt met dezelfde getallen als dit boek. De fouten: op maandag om 9:12 en 9:13 en dinsdag om 10:05 staat 130,0 °C en 100,0 % (de eerste versie van de logger controleerde niet of de SHT31 antwoordde); dinsdag om 14:00 ontbreekt een regel (toen werd versie 2 opgeladen); woensdag van 11:47 tot 11:58 zijn temperatuur en vocht leeg (een losse stekker, en versie 2 schrijft dan een lege cel).

CODE · 30.20

Inlezen en opschonen: logboek.py

20/29

Een analyse begint altijd met **opschonen**: wat is er binnengekomen, wat ontbreekt, wat kan niet? De functies staan in een eigen module `logboek.py`, zodat elk volgend programma ze met `from logboek import lees_week` kan gebruiken, zoals pinnen.h in C++. Het blok onderaan, na `if __name__ == "__main__":`, draait alleen als je logboek.py zelf start: dan maakt het een verslag.

Python **logboek.py** 64 regels

```
# logboek.py - de logbestanden van de klimaatmodule inlezen en opschonen
import csv
from datetime import datetime, timedelta

GRENZEN = {"T": (-10, 50), "RV": (0, 100)}      # wat in de module kan; daarbuiten is het een meetfout


def getal(tekst):
```

```
    """Tekst naar kommagetal; een lege cel (of een ontbrekende bij een halve regel) wordt None."""
    if tekst is None or tekst == "":
        return None
    return float(tekst)


def lees_dag(naam):
    """Alle regels van één dagbestand, als woordenboeken met getallen en een echte tijd."""
    rijen = []
    with open(naam, newline="", encoding="utf-8") as bestand:
        for r in csv.DictReader(bestand):
            rijen.append({"tijd": datetime.strptime(r["tijdstip"], "%Y-%m-%d %H:%M"),
                          "T": getal(r["temperatuur_C"]), "RV": getal(r["vocht_pct"]),
                          "verbruik": getal(r["verbruik_Wh"]), "zon": getal(r["zon_Wh"]),
                          "licht": getal(r["licht_lx"]), "bat": getal(r["batterij_V"])})
    return rijen


def lees_week(eerste=12, laatste=18):
    """De dagbestanden van 12 tot 18 oktober 2026 achter elkaar."""
    rijen = []
    for dag in range(eerste, laatste + 1):
        rijen += lees_dag(f"LOG/202610{dag}.CSV")
    return rijen


def buiten(waarde, grootheid):
    laag, hoog = GRENZEN[grootheid]
    return waarde is not None and not (laag <= waarde <= hoog)


def opschonen(rijen):
    """Onmogelijke waarden worden None (T en RV samen: ze komen van dezelfde sensor). Geeft hun aantal terug."""
    fout = 0
    for r in rijen:
        if buiten(r["T"], "T") or buiten(r["RV"], "RV"):
            r["T"] = r["RV"] = None
            fout += 1
    return fout


def gaten(rijen):
    """Paren van opeenvolgende regels die niet precies één minuut uit elkaar liggen."""
    return [(a["tijd"], b["tijd"]) for a, b in zip(rijen, rijen[1:]) if b["tijd"] - a["tijd"] != timedelta(minutes=1)]


if __name__ == "__main__":
    rijen = lees_week()
    print("regels ingelezen:", len(rijen), "- verwacht:", 7 * 24 * 60)
    print("lege temperatuur:", sum(1 for r in rijen if r["T"] is None))
    for r in rijen:
        if buiten(r["T"], "T") or buiten(r["RV"], "RV"):
            print("  onmogelijk:", r["tijd"], r["T"], r["RV"])
    print("opgeschoond:", opschonen(rijen), "regels")
    for a, b in gaten(rijen):
        print("gat in de tijd:", a, "->", b)
```

Verslag bij de testweek:

```
regels ingelezen: 10079 - verwacht: 10080
lege temperatuur: 12
  onmogelijk: 2026-10-12 09:12:00 130.0 100.0
  onmogelijk: 2026-10-12 09:13:00 130.0 100.0
  onmogelijk: 2026-10-13 10:05:00 130.0 100.0
```

```
opgeschoond: 3 regels
gat in de tijd: 2026-10-13 13:59:00 -> 2026-10-13 14:01:00
```

Drie soorten problemen, drie behandelingen. **Lege cellen** worden None en tellen nergens mee. **Onmogelijke waarden** (130 °C in een kas) worden None: ze zijn geen extreem weer maar een meetfout, en dat kan je bewijzen met de sensor (0xFF 0xFF). **Ontbrekende regels** vind je alleen door de tijden te vergelijken: 10 079 regels in plaats van 10 080. `datetime.strptime` maakt van de tekst een echte tijd, waarmee je kan rekenen.

UITLEG · 30.21

Een fout of een uitschieter?

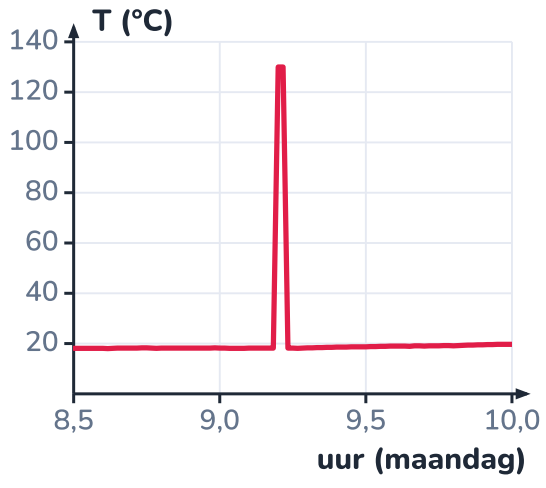
Opschonen is niet 'alles weggooien wat vreemd is'. Een waarde die sterk afwijkt, heet een **uitschieter**, en er zijn twee soorten:

- **Meetfouten**: fysisch onmogelijk (130 °C in een kas, 100,0 % en 130,0 °C samen), of met een bekende oorzaak (een losse draad). Die mag je vervangen door 'geen waarde', met een regel die je in je verslag vermeldt.
- **Echte gebeurtenissen**: een piek in de vochtigheid als iemand de plantjes besproeit, een daling van de temperatuur als het dak opengaat. Die hoor je *niet* weg te halen: juist die wil je zien.

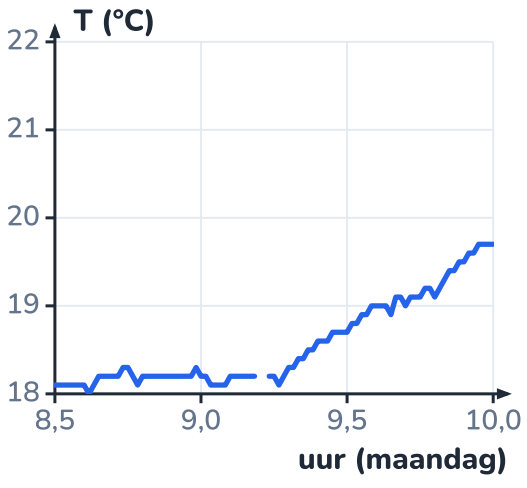
Daarom gebruikt logboek.py **vaste grenzen** die uit de fysica van de module komen (−10 tot 50 °C, 0 tot 100 %), niet 'alles wat meer dan x van het gemiddelde afwijkt'. En het past de **ruwe bestanden** nooit aan: het opschonen gebeurt in het programma, zodat iemand anders het kan nagaan en herhalen (hoofdstuk 28, het verhaal van de vijf ontbrekende landen).

"Een ontbrekende waarde vul je in met 0"

Een 0 is een meting: 0 °C trekt het daggemiddelde omlaag. Een lege waarde laat je leeg, of je vult ze in met een methode die je vermeldt (bv. het gemiddelde van de minuut ervoor en erna). Voor een daggemiddelde over 1428 in plaats van 1440 minuten is leeglaten prima.



Maandag tussen 8:30 en 10:00, zoals de logger het schreef: twee punten op 130 °C.



Hetzelfde na opschonen: een gat van twee minuten, en de echte temperatuur wordt leesbaar.

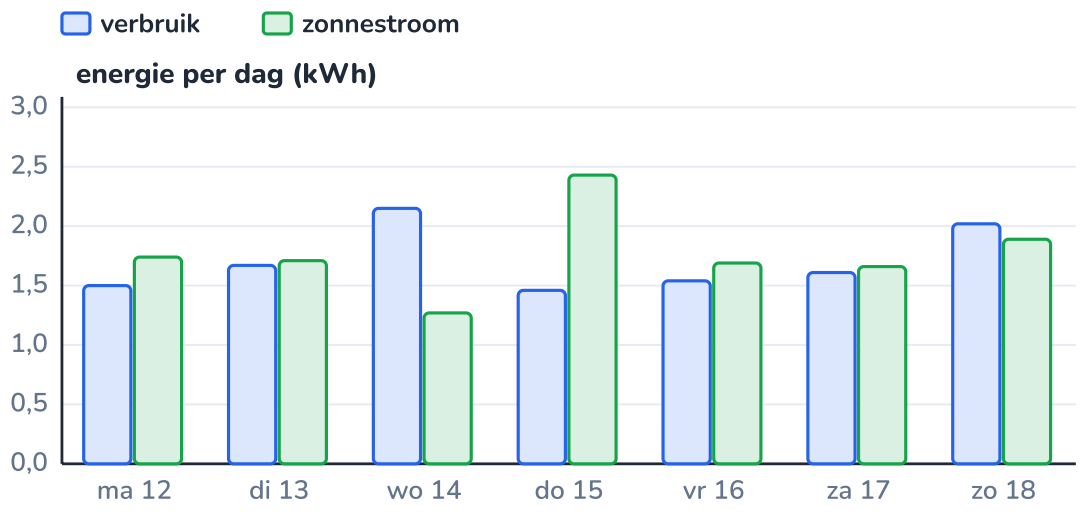
CODE · 30.22

De week per dag

Nu de gegevens schoon zijn, is een samenvatting per dag een paar regels: de rijen verzamelen per datum in een woordenboek (hoofdstuk 28), dan per dag gemiddelde, minimum, maximum en de sommen van de energie.

Python per_dag.py 18 regels

```python
# per_dag.py - de week per dag samenvatten
from logboek import lees_week, opschonen

rijen = lees_week()
opschonen(rijen)
dagen = {}                                  # datum -> de regels van die dag
for r in rijen:
    datum = r["tijd"].date()
    if datum not in dagen:
        dagen[datum] = []
    dagen[datum].append(r)

print("datum       T gem  T min  T max  geldig  verbruik   zon  saldo (kWh)")
for datum, rs in dagen.items():
    t = [r["T"] for r in rs if r["T"] is not None]
    verbruik = sum(r["verbruik"] for r in rs) / 1000
    zon = sum(r["zon"] for r in rs) / 1000
    print(f"{datum} {sum(t) / len(t):6.1f} {min(t):6.1f} {max(t):6.1f} {len(t):7d} {verbruik:9.2f} {zon:5.2f} {zon - verbruik:+6.2f}")
```



Uit dezelfde bestanden: alleen op grijze dagen is het verbruik groter dan de zonnestroom.

Uitvoer:

```
datum       T gem  T min  T max  geldig  verbruik   zon  saldo (kWh)
2026-10-12   20.0   17.7   24.7    1438      1.50  1.74  +0.23
2026-10-13   19.4   17.6   22.4    1438      1.67  1.71  +0.03
2026-10-14   19.1   17.6   21.0    1428      2.15  1.27  -0.88
2026-10-15   19.2   17.7   21.4    1440      1.46  2.43  +0.98
2026-10-16   19.7   17.7   23.7    1440      1.54  1.69  +0.14
2026-10-17   19.6   17.8   22.9    1440      1.61  1.66  +0.05
2026-10-18   19.2   17.7   21.2    1440      2.02  1.89  -0.13
```

Woensdag (grijs) is de enige dag met een groot tekort: de lampen branden langer en de zon levert weinig. Donderdag laadt de batterij dat weer bij. De warmste minuut van de week was 24,7 °C, op de heldere maandag. De kolom 'geldig' telt de minuten met een temperatuur: 1428 op woensdag door de losse stekker.

CODE · 30.23

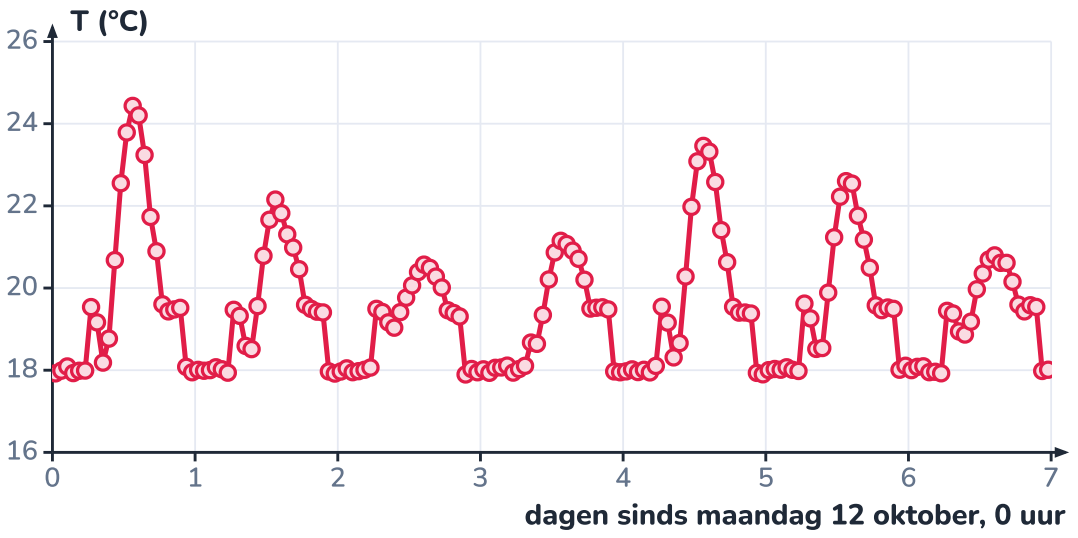
Grafieken van de week

23/29

matplotlib (hoofdstuk 28) tekent de uurgemiddelden en de energie per dag. Met `plt.subplots(2, 1)` komen twee grafieken onder elkaar in één figuur. Omdat de tijd een echte `datetime` is, zet matplotlib zelf datums op de as.

Python **grafiek_week.py** 39 regels

```python
# grafiek_week.py - de temperatuur per uur en de energie per dag, in twee grafieken
import matplotlib.pyplot as plt
from logboek import lees_week, opschonen

rijen = lees_week()
opschonen(rijen)

per_uur = {}                                # begin van het uur -> temperaturen
per_dag = {}                                # datum -> [verbruik, zon] in Wh
for r in rijen:
    if r["T"] is not None:
        uur = r["tijd"].replace(minute=0)
        if uur not in per_uur:
            per_uur[uur] = []
        per_uur[uur].append(r["T"])
    dag = r["tijd"].date()
    if dag not in per_dag:
        per_dag[dag] = [0.0, 0.0]
    per_dag[dag][0] += r["verbruik"]
    per_dag[dag][1] += r["zon"]

uren = sorted(per_uur)
gemiddeld = [sum(per_uur[u]) / len(per_uur[u]) for u in uren]
dagen = sorted(per_dag)

figuur, (boven, onder) = plt.subplots(2, 1, figsize=(9, 7))
boven.plot(uren, gemiddeld, color="tab:red")
boven.set_ylabel("temperatuur (°C)")
boven.set_title("Klimaatmodule, 12 tot 18 oktober 2026")
boven.grid(True)
x = range(len(dagen))
onder.bar([i - 0.2 for i in x], [per_dag[d][0] / 1000 for d in dagen], width=0.4, label="verbruik")
onder.bar([i + 0.2 for i in x], [per_dag[d][1] / 1000 for d in dagen], width=0.4, label="zonnestroom")
onder.set_xticks(list(x), [d.strftime("%a %d") for d in dagen])
onder.set_ylabel("energie per dag (kWh)")
onder.legend()
figuur.tight_layout()
figuur.savefig("week_oktober.png", dpi=150)
plt.show()
```



Dezelfde uurgemiddelden in de huisstijl van dit boek: elke dag een bult, het hoogst op de heldere maandag.

De code maakt de twee grafieken van dit blad en het vorige. Het uurgemiddelde van 60 minuten verbergt de korte schommelingen maar toont het dagritme: 's nachts houdt de warmte van het station de module op ongeveer 18 °C, overdag stijgt ze met de zon, en 's avonds geven de lampen nog wat warmte.

UITGEWERKT VOORBEELD · 30.24

Wat vertellen de gegevens over de module?

24/29

Bekijk het verbruik op woensdagavond en de zonnestroom op maandagvoormiddag, minuut per minuut (de energie per minuut maal 60 is het gemiddelde vermogen in W). Wat gebeurde er om 20:51 op woensdag en om 10:55 op maandag?

GEGEVEN

- de testweek (logboek.py)
- energiebeheer: onder 40 % lampen alleen op zon (hoofdstuk 20)
- batterij 2 × 120 Ah LiFePO4

GEVRAAGD

- een verklaring voor beide knikken

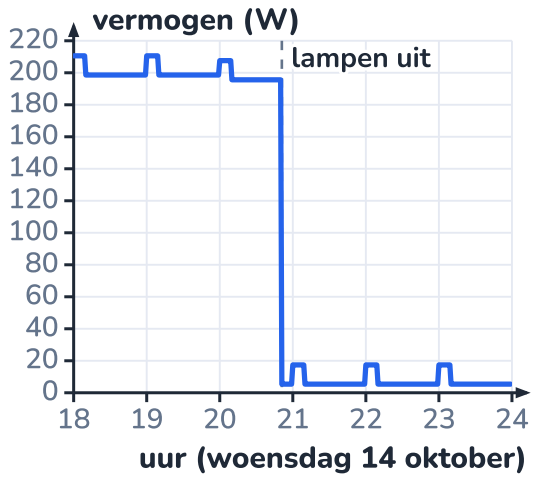
OPLOSSING

1. **Woensdag 20:51.** Het verbruik valt van ongeveer 200 W naar 5,5 W, ruim een uur vóór de lichtperiode eindigt (22 uur). Op een grijze dag daalt de laadtoestand; op dat moment zakt ze onder 40 %, en na zonsondergang betekent 'alleen op zon' gewoon 'uit'. Ook donderdagochtend blijven de lampen uit tot de zon genoeg levert.
2. **Maandag 10:55.** De zonnestroom valt van ruim 800 W naar het verbruik van de module. De zon schijnt gewoon door, maar de batterij is vol: de laadregelaar knijpt de panelen af, en de INA226 meet alleen wat er echt door de leiding gaat. De batterijspanning springt op dat moment van 14,3 naar 13,4 V.
3. **Gevolg voor de analyse.** zon_Wh is wat er geleverd werd, niet wat de panelen konden leveren. In hoofdstuk 28 was zon_Wh een model van wat de panelen gaven; een echte logger ziet de afgeknepen zon niet. Wie wil weten hoeveel zon er verloren ging, schat het uit licht_lx.

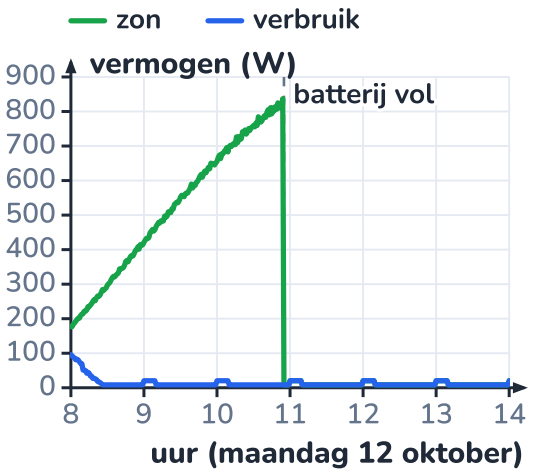
ANTWOORD

Woensdagavond grijpt het energiebeheer in; maandagvoormiddag is de batterij vol.

Klopt dit? Klopt het met de dagtotalen? Maandag leverde de zon maar 1,74 kWh, hoewel het de helderste dag was: na 10:55 werd de rest afgeknepen.



Woensdagavond: de lampen gaan uit voor de lichtperiode voorbij is.



Maandag: na het vol raken van de batterij levert de zon alleen nog het verbruik.

PROEF · 30.25

Een datalogger voor het klaslokaal

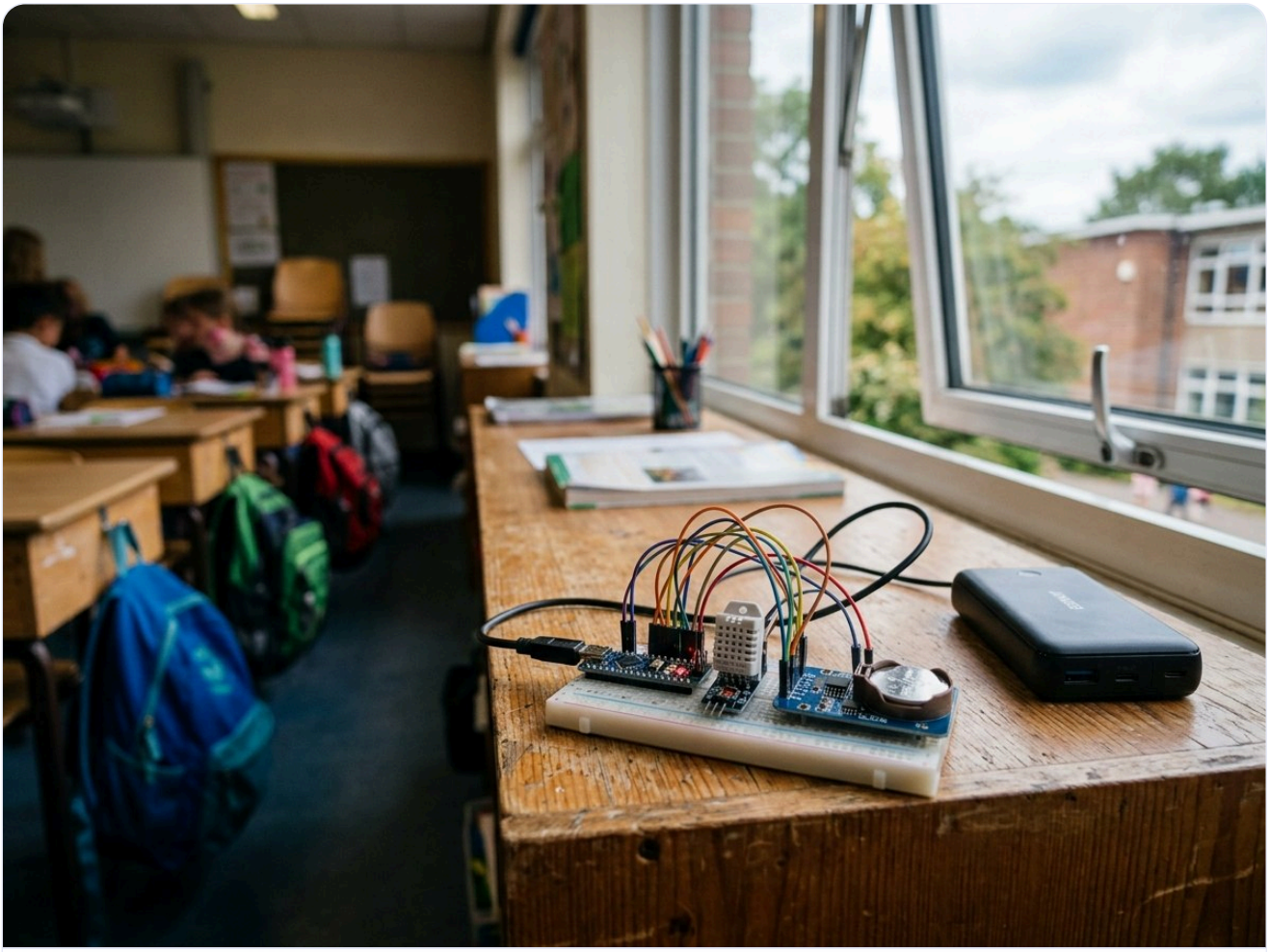
25/29

ONDERZOEKSVRAAG

Hoe veranderen de temperatuur en de luchtvochtigheid in het klaslokaal tijdens een schooldag?

MATERIAAL

- Arduino Mega of Uno, USB-kabel
- SHT31-breakout, DS3231-moduletje met knoopcel
- microSD-moduletje en een kaart (FAT32)
- breadboard en draadjes
- logger.ino, i2c_scanner.ino en klok_zetten.py (werkboek) of de seriële monitor



De logger een schooldag lang op de kast: uit de zon, weg van de verwarming.

WERKWIJZE

1. Sluit de SHT31 en de DS3231 aan op 5V, GND, SDA en SCL (Mega: 20 en 21; Uno: A4 en A5), en het SD-moduletje op 5V, GND en de SPI-pinnen (Mega: 50 tot 53; Uno: 12, 11, 13 en 10; zet dan SD_CS op 10).
2. Draai de I²C-scanner. Je verwacht 0x44, 0x57 en 0x68.
3. Laad logger.ino op. Zonder INA226 blijven de energiekolommen leeg: dat mag. Zet de klok met de opdracht T... in de seriële monitor.
4. Leg de logger op een kast op ooghoogte, uit de zon en weg van de verwarming. Laat hem een hele schooldag loggen; noteer wanneer er ramen opengaan en wanneer er leerlingen zijn.
5. Kopieer het bestand van de kaart en pas logboek.py aan (de naam van het bestand en de dagen). Maak een grafiek van temperatuur en vocht.

WAARNEMING

Zo begint het bestand (uit de simulatie van logger.ino; bij jou staan er echte waarden):

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 00:00,18.0,83.9,0.29,0.00,0,13.24
2026-10-12 00:01,18.0,84.0,0.29,0.00,0,13.24
```

Typisch: 's morgens stijgen temperatuur en vocht zodra de klas binnenkomt (mensen geven warmte en waterdamp af); bij een open raam dalen ze in enkele minuten; tijdens de middagpauze vlakken ze af. Een regel per minuut is fijn genoeg om een open raam te zien.

VERKLARING

Een klas van twintig leerlingen geeft ongeveer 2 kW warmte af (hoofdstuk 28) en elke persoon ademt waterdamp uit: daarom stijgt de relatieve vochtigheid, tenzij de temperatuur sneller stijgt (warme lucht kan meer waterdamp bevatten, dus de *relatieve* vochtigheid kan dan zelfs dalen). Een logger met een klok en een kaart laat zo'n verloop zien zonder dat iemand ernaast moet zitten, en de lege cellen en tijdstempels maken duidelijk wat er ontbreekt.

UITGEWERKT VOORBEELD · 30.26

Habitat: geheugen en eigen verbruik van de logger

26/29

De logger komt in klimaat.ino. Past alles in het werkgeheugen van de Mega? En hoeveel energie kost de logger zelf, vergeleken met de module?

GEGEVEN

- SRAM van de Mega: 8 kB
- 5 V-omvormer met een rendement van 83 % (hoofdstuk 21)
- basisverbruik van de module 230 Wh per dag (hoofdstuk 20); in de testweek 1,5 tot 2,2 kWh per dag

GEVRAAGD

- het SRAM-gebruik
- het eigen verbruik per dag

OPLOSSING

1 **Werkgeheugen.** Een schatting per onderdeel:

WAT	SRAM (ORDE VAN GROOTTE)
SD-bibliotheek (waarvan 512 bytes blokbuffer)	±0,6 kB
Wire (buffers van 32 bytes)	±0,2 kB
Serial (zend- en ontvangstbuffer van 64 bytes)	±0,2 kB
variabelen van klimaat.ino en de logger, KOP, opdracht	±0,4 kB
stack: lokale variabelen, snprintf-buffers	±0,3 kB
samen	**±1,7 kB van de 8 kB**

Samen ongeveer een vijfde van de 8 kB. De IDE toont na het compileren het exacte getal voor de globale variabelen; laat minstens een paar honderd bytes vrij voor de stack. Op een Uno (2 kB) zou het krap worden: een van de redenen voor de Mega.

2 **Stroom.** Per onderdeel:

ONDERDEEL VAN DE LOGGER	STROOM OP 5 V (GEMIDDELD)	WAAROP GEBASEERD
SD-kaartmodule	±10 mA (pieken ±100 mA bij het schrijven)	orde van grootte, hoofdstuk 21
DS3231	≤ 0,3 mA	datasheet (en een ledje op het moduletje: vaak enkele mA extra)
2 × INA226	2 × 0,33 mA	datasheet, continu meten
SHT31, om de 2 s 12,5 ms meten bij 0,6 mA	±0,004 mA	datasheet
samen	**±11 mA = ±55 mW**	de Mega zelf telt niet: die draait toch

3 **Energie per dag.** $55\,\mathrm{mW}/0{,}83 = 66\,\mathrm{mW}$ aan de 12 V-kant; maal 24 h is dat $1{,}6\,\mathrm{Wh}$ per dag. Met een ledje op het klokmoduletje (±3 mA) wordt het ±2,0 Wh.

4 **Vergelijken.** 1,6 Wh is 0,7 % van het basisverbruik van 230 Wh, en ongeveer 0,1 % van wat de module in de testweek per dag verbruikte. De SD-kaart is de grootste post; de sensoren tellen nauwelijks.

ANTWOORD

±1,7 kB van 8 kB SRAM; ±1,6 Wh per dag, minder dan 1 % van het basisverbruik.

Klopt dit? Orde van grootte: een ledje van 3 mA verbruikt al een vijfde van de hele logger. Wie zuiniger wil, knipt het ledje los en laat de kaart slapen tussen twee regels (Expert, hoofdstuk E37).

CODE · 30.27

klimaat.ino versie 2: de logger als taak

27/29

De logger wordt geen apart programma maar een deel van de sturing: de meettaak van hoofdstuk 29 meet al om de 2 s, dus telt ze nu ook op voor het logboek (`loggen()`). Drie dingen veranderen verder: de tijd van de dag komt uit de DS3231 in plaats van uit `millis()` met een startuur, de pomp draait in de eerste tien minuten van elk kloksuur (ook na een herstart), en alle meldingen beginnen met `#`, zodat een Python-programma op de computer ze van de logregels kan scheiden.

C++ (Arduino) **klimaat.ino** 421 regels

```
// klimaat.ino - de sturing van de klimaatmodule, versie 2 (Arduino Mega 2560, hoofdstuk 30)
// Nieuw tegenover versie 1: de klok komt van de DS3231, en de meettaak houdt een logboek bij: elke minuut een
// CSV-regel op de SD-kaart (map LOG, één bestand per dag) en naar de computer. Meldingen beginnen nu met #.
// I2C: INA226 zon 0x40, INA226 verbruik 0x41, SHT31 0x44, DS3231 0x68. SD op SPI (CS = 53). Klok: wintertijd.
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <RTClib.h>
#include <EEPROM.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"

// ---------------------------------------------------------------- instellingen
const unsigned long MEET_MS = 2000;
const unsigned long LAMP_MS = 200;
const unsigned long MELD_MS = 60000UL;
const unsigned long BEWAAR_MS = 3600000UL;          // de laadtoestand elk uur in het EEPROM
const unsigned long DENDER_MS = 20;
const float CAPACITEIT_AH = 240.0;                  // twee LiFePO4-batterijen van 120 Ah (hoofdstuk 20)
const int ADRES_SOC = 0;                            // waar de laadtoestand in het EEPROM staat
const float DOEL = 200.0;                           // umol/(m2 s) op het kweekbed (hoofdstuk 25)
const float LAMPEN_VOL = 200.0;                     // wat de lampen op 100 % geven
const float LUX_PER_UMOL = 54.0;
const float U_NODIG = 11.63, U_START = 2.21;        // dakmotor (hoofdstuk 26)
const unsigned long AANLOOP_MS = 1000, REMMEN_VANAF_MS = 6000, TIMEOUT_MS = 12000;
const int UITGANGEN[] = {LAMPEN, PWM_DICHT, PWM_OPEN, DAK_ENABLE, POMP, CO2_KLEP, VENTILATOREN, STATUSLED};
const int AANTAL_UIT = sizeof(UITGANGEN) / sizeof(UITGANGEN[0]);
const int SD_CS = 53;                               // logger (hoofdstuk 30)
const byte INA_ZON = 0x40, INA_VERBRUIK = 0x41;
const float R_SHUNT = 0.00075;                      // shunt 100 A / 75 mV
const char KOP[] = "tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V";

// ---------------------------------------------------------------- toestand (globaal: alle taken gebruiken ze)
Adafruit_SHT31 sht = Adafruit_SHT31();
RTC_DS3231 rtc;
DateTime klok;                                      // de tijd van de laatste meting (om de 2 s bijgewerkt)
bool kaartOk = false;
DateTime minuut;                                    // logger: de minuut die we verzamelen
```

```ino
bool bezig = false;
float somT = 0, somRV = 0, somLux = 0, somZon = 0, somVerbruik = 0, somU = 0;
int nT = 0, nMeting = 0, nVermogen = 0;
char opdracht[32];
int lengte = 0;
float luchtT = NAN, luchtRV = NAN;                    // kaslucht, van de SHT31 op I2C
float lux = 0, uBat = 0, iBat = 0, soc = 50.0;
int grens = 0;                                        // energiebeheer: de hoogste PWM die de lampen mogen
int stand = 0;                                        // de PWM die de lampen nu krijgen
bool pompAan = false, genoegWater = true;
enum Dak { STIL, SLUITEN, OPENEN };
Dak dak = STIL;
bool dakOpen = false;
unsigned long vorigeMeting = 0, vorigeLamp = 0, vorigeMelding = 0, vorigeBewaring = 0;
unsigned long startDak = 0, laatsteWissel = 0, vorigeKnipper = 0;
int vorigeLezing = LOW, stabiel = LOW;
bool ledAan = false;

// ---------------------------------------------------------------- sensoren
int gemiddeld(int pin) {                              // 16 metingen middelen tegen ruis (hoofdstuk 22)
  long som = 0;
  for (int i = 0; i < 16; i++) som += analogRead(pin);
  return som / 16;
}

float daglichtLux() {                                 // LDR onder een kapje (hoofdstuk 22 en 25)
  int n = gemiddeld(LDR);
  if (n < 1) n = 1;
  float r = 1000.0 * (1024.0 - n) / n;
  return 10.0 * pow(r / 15000.0, -1.0 / 0.7) / 0.10;
}

float batterijspanning() {                            // deler 100 kOhm / 33 kOhm (hoofdstuk 22)
  return gemiddeld(BATTERIJ) * 5.0 / 1024.0 * (133.0 / 33.0);
}

float batterijstroom() {                              // ACS712-20A: 100 metingen in 20,4 ms (hoofdstuk 16 en 25)
  long som = 0;
  unsigned long t0 = micros();
  for (int i = 0; i < 100; i++) {
    while (micros() - t0 < i * 204UL) {
    }
    som += analogRead(STROOM);
  }
  return (som / 100.0 * 5.0 / 1024.0 - 2.50) / 0.100;  // A, positief = ontladen
}

long inaLees(byte adres, byte reg) {                  // INA226: 16 bits uit een register, of -1 (hoofdstuk 30)
  Wire.beginTransmission(adres);
  Wire.write(reg);
  if (Wire.endTransmission() != 0) return -1;
  if (Wire.requestFrom(adres, (byte)2) != 2) return -1;
  byte hoog = Wire.read();
  byte laag = Wire.read();
  return ((long)hoog << 8) | laag;
}

float inaSpanning(byte adres) {
  long bus = inaLees(adres, 0x02);
  if (bus < 0) return NAN;
  return bus * 0.00125;
}

float inaVermogen(byte adres) {
  long shunt = inaLees(adres, 0x01);
  float u = inaSpanning(adres);
  if (shunt < 0 || isnan(u)) return NAN;
  return u * (int16_t)shunt * 0.0000025 / R_SHUNT;
```

```
}

// ---------------------------------------------------------------- tijd
unsigned long secondenVanDeDag() {                  // uit de klok, niet meer uit millis()
  return klok.hour() * 3600UL + klok.minute() * 60UL + klok.second();
}

void printTwee(unsigned long x) {                   // altijd twee cijfers: 7 -> 07
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void printTijd() {                                  // meldingen beginnen met #: de logger-regels niet
  Serial.print("# ");
  unsigned long s = secondenVanDeDag();
  printTwee(s / 3600);
  Serial.print(':');
  printTwee(s / 60 % 60);
  Serial.print(':');
  printTwee(s % 60);
  Serial.print("  ");
}

// ---------------------------------------------------------------- logger (hoofdstuk 30)
void wisSommen() {
  somT = somRV = somLux = somZon = somVerbruik = somU = 0;
  nT = nMeting = nVermogen = 0;
}

size_t schrijfRegel(Print &uit) {
  size_t n = 0;
  char tijd[17];
  snprintf(tijd, sizeof(tijd), "%04d-%02d-%02d %02d:%02d", minuut.year(), minuut.month(), minuut.day(),
           minuut.hour(), minuut.minute());
  n += uit.print(tijd);
  n += uit.print(',');
  if (nT > 0) n += uit.print(somT / nT, 1);
  n += uit.print(',');
  if (nT > 0) n += uit.print(somRV / nT, 1);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somVerbruik / nVermogen / 60.0, 2);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somZon / nVermogen / 60.0, 2);
  n += uit.print(',');
  if (nMeting > 0) n += uit.print(somLux / nMeting, 0);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somU / nVermogen, 2);
  n += uit.println();
  return n;
}

void opslaan() {
  size_t lang = schrijfRegel(Serial);
  if (!kaartOk) return;
  char naam[20];
  snprintf(naam, sizeof(naam), "LOG/%04d%02d%02d.CSV", minuut.year(), minuut.month(), minuut.day());
  bool nieuw = !SD.exists(naam);
  File f = SD.open(naam, FILE_WRITE);
  if (!f) {
    Serial.println("# SD: kan het bestand niet openen");
    kaartOk = false;                                // de statusled knippert snel
    return;
  }
  if (nieuw) f.println(KOP);
  if (schrijfRegel(f) < lang) {
    Serial.println("# SD: kaart vol");
    kaartOk = false;
  }
```

```
  f.close();
}

void loggen() {                                    // om de 2 s, na de metingen
  DateTime begin(klok.year(), klok.month(), klok.day(), klok.hour(), klok.minute(), 0);
  if (!bezig) {
    minuut = begin;
    bezig = true;
  } else if (begin.unixtime() != minuut.unixtime()) {
    opslaan();
    wisSommen();
    minuut = begin;
  }
  if (!isnan(luchtT)) {
    somT += luchtT;
    somRV += luchtRV;
    nT++;
  }
  somLux += lux;
  float pZon = inaVermogen(INA_ZON);
  float pVerbruik = inaVermogen(INA_VERBRUIK);
  float u = inaSpanning(INA_VERBRUIK);
  if (!isnan(pZon) && !isnan(pVerbruik) && !isnan(u)) {
    somZon += pZon;
    somVerbruik += pVerbruik;
    somU += u;
    nVermogen++;
  }
  nMeting++;
}

void leesOpdracht() {                              // T2026-10-11 20:00:00 + Enter: de klok zetten
  while (Serial.available() > 0) {
    char c = Serial.read();
    if (c == '\n' || c == '\r') {
      opdracht[lengte] = '\0';
      int j, mo, d, u, mi, s;
      if (lengte > 0 && sscanf(opdracht, "T%d-%d-%d %d:%d:%d", &j, &mo, &d, &u, &mi, &s) == 6) {
        rtc.adjust(DateTime(j, mo, d, u, mi, s));
        Serial.println("# klok gezet");
        wisSommen();
        bezig = false;
      } else if (lengte > 0) {
        Serial.println("# onbekende opdracht");
      }
      lengte = 0;
    } else if (lengte < (int)sizeof(opdracht) - 1) {
      opdracht[lengte] = c;
      lengte++;
    }
  }
}

// ------------------------------------------------------------------ taken
void taakMeten(unsigned long nu) {
  if (nu - vorigeMeting < MEET_MS) return;
  vorigeMeting += MEET_MS;
  klok = rtc.now();
  sht.readBoth(&luchtT, &luchtRV);                 // nan als de sensor niet antwoordt
  lux = daglichtLux();
  uBat = batterijspanning();
  iBat = batterijstroom();
  soc -= iBat * (MEET_MS / 3600000.0) / CAPACITEIT_AH * 100.0;    // lading tellen (hoofdstuk 16)
  if (uBat >= 14.2 && iBat < 0.0 && iBat > -2.0) soc = 100.0;     // vol: opnieuw ijken
  soc = constrain(soc, 0.0, 100.0);
  // energiebeheer (hoofdstuk 20): hoeveel licht mag er hoogstens branden?
  if (soc >= 40.0) grens = 255;
  else if (soc < 20.0) grens = 0;
```

```
  else if (iBat < 0.0) grens = min(grens + 5, 255);   // de batterij laadt: er is zon over
  else grens = max(grens - 5, 0);                     // de batterij levert: minder licht
  loggen();
}

void taakLampen(unsigned long nu) {                   // dimmen met het daglicht (hoofdstuk 25)
  if (nu - vorigeLamp < LAMP_MS) return;
  vorigeLamp += LAMP_MS;
  int wens = 0;
  unsigned long uur = secondenVanDeDag() / 3600;
  if (uur >= 6 && uur < 22) {                         // lichtperiode van 16 uur
    float f = constrain((DOEL - lux / LUX_PER_UMOL) / LAMPEN_VOL, 0.0, 1.0);
    wens = (int)(f * 255 + 0.5);
  }
  int doelStand = min(wens, grens);
  if (doelStand > stand) stand++;                     // zacht: hoogstens 1 stap per 200 ms
  else if (doelStand < stand) stand--;
  analogWrite(LAMPEN, stand);
}

void taakPomp(unsigned long nu) {                     // 10 min per uur, nooit zonder water (hoofdstuk 23)
  bool tijd = klok.minute() < 10;                     // de eerste 10 minuten van elk uur, volgens de klok
  bool water = (digitalRead(VLOTTER) == LOW);
  bool aan = tijd && water;
  if (water != genoegWater) {                         // alleen melden als er iets verandert
    printTijd();
    Serial.println(water ? "vlotter: weer genoeg water" : "ALARM vlotter: te weinig water");
    genoegWater = water;
  }
  if (aan != pompAan) {
    printTijd();
    Serial.println(aan ? "pomp aan" : "pomp uit");
    pompAan = aan;
  }
  digitalWrite(POMP, aan ? HIGH : LOW);
}

bool knopGedrukt(unsigned long nu) {                  // true bij de flank 'net ingedrukt' (hoofdstuk 24)
  int lezing = digitalRead(KNOP);
  if (lezing != vorigeLezing) {
    laatsteWissel = nu;
    vorigeLezing = lezing;
  }
  if (nu - laatsteWissel >= DENDER_MS && lezing != stabiel) {
    stabiel = lezing;
    return stabiel == HIGH;
  }
  return false;
}

void motor(int richting, int waarde) {                // +1 sluiten, -1 openen, 0 stil (hoofdstuk 26)
  analogWrite(PWM_DICHT, richting > 0 ? waarde : 0);
  analogWrite(PWM_OPEN, richting < 0 ? waarde : 0);   // nooit beide tegelijk
}

void stopDak(unsigned long nu, const char* reden) {
  motor(0, 0);
  dak = STIL;
  printTijd();
  Serial.println(reden);
}

void taakDak(unsigned long nu) {
  if (knopGedrukt(nu)) {
    if (dak != STIL) {
      stopDak(nu, "dak gestopt met de knop");         // een druk tijdens de beweging = stoppen
    } else {
      dak = dakOpen ? SLUITEN : OPENEN;
```

```
      startDak = nu;
      printTijd();
      Serial.println(dak == SLUITEN ? "dak sluiten" : "dak openen");
    }
  }
  if (dak == STIL) return;
  unsigned long t = nu - startDak;
  if (dak == SLUITEN && digitalRead(EIND_DICHT) == HIGH) {
    dakOpen = false;
    stopDak(nu, "dak dicht");
  } else if (dak == OPENEN && digitalRead(EIND_OPEN) == HIGH) {
    dakOpen = true;
    stopDak(nu, "dak open");
  } else if (t > TIMEOUT_MS) {
    stopDak(nu, "ALARM dak: eindpunt niet bereikt");
  } else {                                          // aanlopen, constant, afremmen tot 30 %
    float s = 1.0;
    if (t < AANLOOP_MS) s = t / (float)AANLOOP_MS;
    else if (t > REMMEN_VANAF_MS) s = max(0.3, 1.0 - (t - REMMEN_VANAF_MS) / (float)AANLOOP_MS);
    float u = U_START + (U_NODIG - U_START) * s;
    motor(dak == SLUITEN ? +1 : -1, constrain((int)(u / uBat * 255 + 0.5), 0, 255));
  }
}

void taakStatusled(unsigned long nu) {              // rustig knipperen = alles goed, snel = alarm
  bool alarm = !genoegWater || soc < 10.0 || !kaartOk;
  unsigned long periode = alarm ? 100 : 500;
  if (nu - vorigeKnipper < periode) return;
  vorigeKnipper = nu;                               // hier mag het: een led mag wat verlopen
  ledAan = !ledAan;
  digitalWrite(STATUSLED, ledAan ? HIGH : LOW);
}

void taakMelden(unsigned long nu) {                 // elke minuut één regel voor de seriële monitor
  if (nu - vorigeMelding < MELD_MS) return;
  vorigeMelding += MELD_MS;
  printTijd();
  Serial.print("T ");
  Serial.print(luchtT, 1);
  Serial.print(" C  RV ");
  Serial.print(luchtRV, 0);
  Serial.print(" %  licht ");
  Serial.print(lux, 0);
  Serial.print(" lx  bat ");
  Serial.print(uBat, 2);
  Serial.print(" V ");
  Serial.print(iBat, 1);
  Serial.print(" A  SoC ");
  Serial.print(soc, 1);
  Serial.print(" %  lampen ");
  Serial.print(stand);
  Serial.print("/");
  Serial.println(grens);
}

void taakBewaren(unsigned long nu) {                // de laadtoestand overleeft een stroomonderbreking
  if (nu - vorigeBewaring < BEWAAR_MS) return;
  vorigeBewaring += BEWAAR_MS;
  EEPROM.put(ADRES_SOC, soc);                       // schrijft alleen de bytes die veranderen
}

// ---------------------------------------------------------------- setup en loop
void setup() {
  for (int i = 0; i < AANTAL_UIT; i++) {
    pinMode(UITGANGEN[i], OUTPUT);
    digitalWrite(UITGANGEN[i], LOW);                // veilige begintoestand: alles uit
  }
  digitalWrite(DAK_ENABLE, HIGH);
```

```
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(KNOP, INPUT);
  pinMode(EIND_OPEN, INPUT_PULLUP);
  pinMode(EIND_DICHT, INPUT_PULLUP);
  Serial.begin(115200);
  Serial.println("# klimaatmodule v2 (hoofdstuk 30)");
  Wire.begin();
  if (!sht.begin(0x44)) Serial.println("# ALARM: geen SHT31 op adres 0x44");
  if (!rtc.begin()) Serial.println("# ALARM: geen klok (DS3231) op 0x68");
  if (rtc.lostPower()) Serial.println("# de klok heeft stilgestaan: stuur T jjjj-mm-dd uu:mm:ss");
  klok = rtc.now();
  byte ina[] = {INA_ZON, INA_VERBRUIK};
  for (int i = 0; i < 2; i++) {                      // INA226: 128 metingen middelen (282 ms)
    Wire.beginTransmission(ina[i]);
    Wire.write(0x00);
    Wire.write(0x49);
    Wire.write(0x27);
    Wire.endTransmission();
  }
  kaartOk = SD.begin(SD_CS);
  if (kaartOk && !SD.exists("LOG")) SD.mkdir("LOG");
  if (!kaartOk) Serial.println("# ALARM: geen SD-kaart");
  EEPROM.get(ADRES_SOC, soc);
  if (isnan(soc) || soc < 0.0 || soc > 100.0) soc = 50.0;   // leeg EEPROM: een ruwe schatting
  Serial.print("# laadtoestand uit het EEPROM: ");
  Serial.print(soc, 1);
  Serial.println(" %");
  uBat = batterijspanning();
  dakOpen = (digitalRead(EIND_OPEN) == HIGH);
  Serial.println(KOP);
}

void loop() {
  unsigned long nu = millis();                       // millis() blijft de klok voor de intervallen
  leesOpdracht();
  taakMeten(nu);
  taakLampen(nu);
  taakPomp(nu);
  taakDak(nu);
  taakStatusled(nu);
  taakMelden(nu);
  taakBewaren(nu);
}
```

Een test in de simulatie op maandag 12 oktober vanaf 07:59, met de laadtoestand op 35 % en een druk op de knop van het dak om 08:00:40:

```
# klimaatmodule v2 (hoofdstuk 30)
# laadtoestand uit het EEPROM: 35.0 %
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 07:59,17.9,84.0,0.85,1.11,1578,13.07
# 08:00:00  pomp aan
# 08:00:00  T 18.0 C  RV 84 %  licht 1666 lx  bat 13.05 V -0.1 A  SoC 35.0 %  lampen 86/90
# 08:00:40  dak openen
# 08:00:44  dak open
2026-10-12 08:00,18.0,83.9,1.33,1.31,1742,13.05
# 08:01:00  T 18.0 C  RV 84 %  licht 1831 lx  bat 13.05 V -0.1 A  SoC 35.0 %  lampen 86/90
2026-10-12 08:01,18.0,83.8,1.52,1.52,1908,13.05
# 08:02:00  T 18.1 C  RV 84 %  licht 1997 lx  bat 13.05 V 0.0 A  SoC 35.0 %  lampen 104/100
2026-10-12 08:02,18.1,83.7,1.73,1.74,2075,13.05
# 08:03:00  T 18.1 C  RV 84 %  licht 2161 lx  bat 13.03 V 0.2 A  SoC 35.0 %  lampen 124/120
```

De logregels staan tussen de meldingen; lees_serieel.py kan de `#`-regels overslaan. `millis()` is niet weg: ze blijft de klok voor de intervallen, want die moeten ook doorlopen als iemand de RTC verzet. De RTC geeft de datum en het uur.

NASLAG · 30.28

UART, I²C en SPI naast elkaar

28/29

stijgtijd van een I²C-lijn (30 % tot 70 %)

$$t_r = R\,C \ln\frac{0{,}7}{0{,}3} \approx 0{,}85\,R\,C$$

symbool	betekenis	eenheid
R	pull-upweerstand	Ω
C	capaciteit van de bus	F

seriële verbinding 8N1

$$\text{tekens per seconde} = \frac{\text{baudrate}}{10}$$

	UART (Serial)	I²C (Wire)	SPI (SD-kaart)
draden	TX, RX (+ GND)	SDA, SCL (+ GND)	MOSI, MISO, SCK, CS per apparaat (+ GND)
klok	geen: beide kanten kennen de baudrate	SCL, van de master	SCK, van de master
apparaten	twee	tot ±100, elk een eigen adres	meerdere, elk een eigen CS-draad
snelheid (Mega)	tot 115 200 baud en meer	100 of 400 kHz	enkele MHz
afstand	meters (tot tientallen met RS-485)	centimeters tot een paar meter	centimeters
in de module	Mega ↔ computer	SHT31, DS3231, 2 × INA226	SD-kaart

pyserial	wat het doet
`pip install pyserial`	eenmalig installeren (je importeert het als `serial`)
`python -m serial.tools.list_ports`	welke poorten er zijn (COM3, /dev/ttyACM0 …)
`poort = serial.Serial("COM3", 115200, timeout=5)`	openen; de Mega herstart, wacht 2 s
`poort.readline()`	bytes tot en met \n, of b"" na de timeout
`.decode("utf-8").strip()`	bytes naar tekst, zonder \r\n
`poort.write(b"T2026-10-11 20:00:00\n")`	tekst naar de Mega sturen (als bytes)
`poort.close()`	sluiten: pas dan kan de seriële monitor de poort weer openen

HET GEHEEL · 30.29

Waar dit hoofdstuk in het geheel past

29/29

Datalogging is waar de leerlijnen elkaar ontmoeten: elektronica (pull-ups, shunts), programmeren in twee talen, onderzoeken (tijdstempels, lege cellen, eerlijk rapporteren) en data (opschonen, samenvatten, grafieken). De logger is ook het begin van de digitale tweeling van het station in Expert: een model dat je alleen kan bouwen en controleren met goede metingen.

I28 Functies, lijsten en data in Python

Het CSV-formaat, DictReader, woordenboeken per dag en matplotlib: hier op echte logbestanden.

I29 Microcontrollers programmeren in C++

Taken met millis(), Serial, bibliotheken en klimaat.ino, waar de logger een taak in wordt.

I24 Condensatoren en de RC-tijd

De RC-tijd achter de stijgtijd van een I²C-lijn, en de supercondensator die het bestand laat sluiten.

I25 Van analoog naar digitaal

Bits, bytes, hexadecimaal en aliasing met de PWM van de lampen.

I21 De wetten van Kirchhoff

De knooppuntwet: batterijstroom = zonnestroom – verbruik.

I16 Redox en batterijen

De laadtoestand: met twee INA226's nauwkeuriger dan met de ACS712 alleen.

I20 Zonne-energie dimensioneren

Het energiebeheer dat je in de gegevens van woensdagavond terugziet.

I31 Regelsystemen

Een regeling beoordelen met een meting: daar is het logboek voor.

I39 Data visualiseren en eerste stappen in AI

Grafieken kiezen en een beslisboom trainen op de logbestanden.

E6 Statistiek en meetonzekerheid

Meetonzekerheid en standaardfout op minuutgemiddelden.

E37 Embedded systemen en IoT

SPI en I²C in detail, interrupts, slaapstanden en de gegevens draadloos naar de centrale computer (MQTT).

E41 Van data naar model: regressie

Het verbruik van morgen voorspellen uit een logboek van maanden.

Habitat

De klimaatmodule houdt nu een logboek bij: logger.ino (of klimaat.ino versie 2) meet om de 2 s en schrijft elke minuut een regel met tijdstip, temperatuur, vocht, verbruik, zonnestroom, licht en batterijspanning, op een SD-kaart in LOG/JJJJMMDD.CSV en naar de computer. De sensoren hangen aan één I²C-bus: SHT31 (0x44, in de sensorkast op 3 m, met pull-ups van 2,2 kΩ extra), DS3231 (0x68, wintertijd) en twee INA226 (0x40 zon, 0x41 verbruik). Een dag is ±68,5 kB (1 GB is ±40 jaar); de logger zelf kost ±1,6 Wh per dag. Opdracht 30.13 is het ontwerp voor je portfolio; in hoofdstuk 39 en 40 analyseer je maanden aan gegevens.

SAMENVATTING

In het kort

- Een seriële verbinding (UART) stuurt elk teken als startbit, acht databits (b0 eerst) en stopbit; beide kanten gebruiken dezelfde baudrate. Tekens per seconde = baudrate / 10.
- Een eenvoudig protocol: één CSV-regel per meting, een kop, # voor meldingen, een lege cel als er geen geldige meting is.
- Python leest de regels met pyserial: readline() geeft bytes, decode() en strip() maken er tekst van; controleer het aantal velden.
- I²C: twee draden (SDA, SCL), open drain met pull-ups, elke chip een eigen adres van 7 bits; START, adres + R/W, ACK, data, STOP.
- Een lange kabel verhoogt de capaciteit: de stijgtijd 0,85 · R · C moet onder 1 µs blijven (100 kHz), dus sterkere pull-ups.
- De bytes van een sensor reken je om met de datasheet; een CRC toont of er onderweg een bit omviel.
- millis() is geen klok: geen datum, weg na een reset, onnauwkeurig. Een DS3231 wel: BCD in registers, ±2 ppm, knoopcel. Log het hele jaar in wintertijd.
- De SD-bibliotheek kent alleen 8.3-namen; pas flush() of close() legt de gegevens echt vast. Een dag logboek is ±68 kB: 1 GB is ±40 jaar.
- Opschonen: lege cellen worden None, onmogelijke waarden ook (met een vaste regel), ontbrekende regels vind je met de tijdstempels. Pas de ruwe bestanden nooit aan.
- Een logger kost weinig: ±1,7 kB SRAM en ±1,6 Wh per dag in de klimaatmodule.

BEGRIPPEN

Wat je nu kent

datalogger
Toestel of programma dat metingen automatisch met een tijdstempel bewaart.

UART
Seriële verbinding zonder klokdraad: elk teken gaat als startbit, databits en stopbit over één draad (TX naar RX).

baudrate
Aantal bits per seconde op een seriële verbinding; beide kanten moeten dezelfde gebruiken.

protocol
Afspraak over hoe twee toestellen gegevens uitwisselen: vorm, volgorde en betekenis.

I²C
Bus met twee draden (SDA voor data, SCL voor de klok) waarop veel chips hangen, elk met een eigen adres.

pull-upweerstand
Weerstand naar de voedingsspanning die een lijn hoog houdt als niemand ze laag trekt.

CRC
Controlegetal dat uit de data berekend wordt: klopt het niet, dan is er onderweg iets misgegaan.

RTC (real-time clock)
Klokchip die datum en tijd bijhoudt, ook zonder voeding van de microcontroller (knoopcel).

BCD
Binair gecodeerd decimaal: elk decimaal cijfer apart in 4 bits, zoals 0x34 voor 34.

SPI
Snelle bus met een klokdraad, twee datadraden en een selectiedraad per apparaat; hier voor de SD-kaart.

tijdstempel
Datum en tijd die bij een meting horen.

opschonen (data)
Fouten in data opsporen en behandelen: lege cellen, onmogelijke waarden, dubbele of ontbrekende regels.

shunt
Weerstand van enkele milliohm in een stroomleiding; de spanning erover is evenredig met de stroom.

I INTERMEDIATE · DEEL I8 · REGELEN EN ROBOTICA

HANDBOEK · HOOFDSTUK 31

Regelsystemen

Meten, vergelijken, bijsturen: de verwarming en de ventilatie van de klimaatmodule met hysterese

Een thermostaat, een cruisecontrol en je eigen lichaam doen hetzelfde: ze meten, vergelijken met wat ze willen en sturen bij, en dan meten ze opnieuw. In dit hoofdstuk leer je de taal van het blokschema, bouw je een aan/uit-regeling met hysterese voor de klimaatmodule, simuleer je ze met een warmtemodel, en beoordeel je ze met getallen: hoe ver schommelt de temperatuur, hoe vaak schakelt de verwarming, en wat kost het?

10× uitleg · 3× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 3× code · ± 8 lestijden

R Regelen en robotica · B Leven en systemen

Buiten vriest het, binnen houdt een regeling de sla tussen 16 en 17 °C.

Na dit hoofdstuk

- Je kan het verschil tussen een open en een gesloten lus uitleggen en systemen uit techniek en natuur indelen.
- Je kan een regelkring tekenen als blokschema met setpoint, fout, regelaar, actuator, proces, sensor, meting en storing.
- Je kan uitleggen hoe een aan/uit-regeling met hysterese werkt en ze programmeren in C++.
- Je kan met een eenvoudig warmtemodel de aan- en uittijd, het aantal schakelingen per uur en de inschakelduur berekenen.
- Je kan verklaren hoe de breedte van de hysterese en de vertraging van de sensor de schommeling en het schakelen beïnvloeden.
- Je kan een regeling beoordelen met kengetallen uit een meting: gemiddelde afwijking, schommeling, schakelingen per uur, energie.
- Je kan grenzen kiezen en verantwoorden voor de verwarming en de ventilatie van de klimaatmodule.

VERHAAL · 31.1

Een blinde uitvinder en een schokkerige chauffeur

1/21

De Amerikaanse ingenieur **Ralph Teetor** werd blind toen hij vijf jaar was. Hij werd toch ingenieur, en een van de productiefste uitvinders van de auto-industrie: hij kreeg meer dan veertig octrooien. Zijn bekendste uitvinding begon met ergernis. Teetor reed vaak mee met zijn advocaat, en die had een

gewoonte: als hij luisterde, gaf hij gas, als hij praatte, liet hij het gaspedaal los. De auto versnelde en vertraagde voortdurend. Teetor, die de beweging voelde in plaats van zag, werd er misselijk van.

Hij bedacht een toestel dat de snelheid zelf vasthoudt. Het meet hoe snel de auto rijdt, vergelijkt dat met de snelheid die de bestuurder gekozen heeft, en geeft meer of minder gas. In 1950 kreeg hij er een octrooi op. In 1958 zat het als 'Auto-Pilot' in de Chrysler Imperial; een jaar later bood Cadillac het aan onder de naam die bleef: **cruise control**.

Teetors toestel doet wat elke regeling doet: **meten, vergelijken, bijsturen**, en dan opnieuw meten. De thermostaat van je verwarming doet het, de koelkast, het vlotterventiel in een toilet, en je eigen lichaam, dat zijn temperatuur op 37 °C houdt. In dit hoofdstuk leer je de taal waarin ingenieurs zulke regelingen tekenen en beoordelen, en je regelt de verwarming en de ventilatie van de klimaatmodule.

Waarom dit hoofdstuk ertoe doet

De meetkas gaf water onder 30 % en opende haar dakraam boven 28 °C (Fundamental, hoofdstuk 25). In de klimaatmodule werken mensen en groeit sla die niet mag bevriezen en niet mag koken. Waar leg je de grenzen? Hoe vaak mag een klep schakelen? Wat als de sensor traag is of uitvalt? En hoe bewijs je met metingen dat je regeling goed werkt?

De eerste cruisecontrols hadden een draaiknop: de bestuurder stelde de gewenste snelheid in.

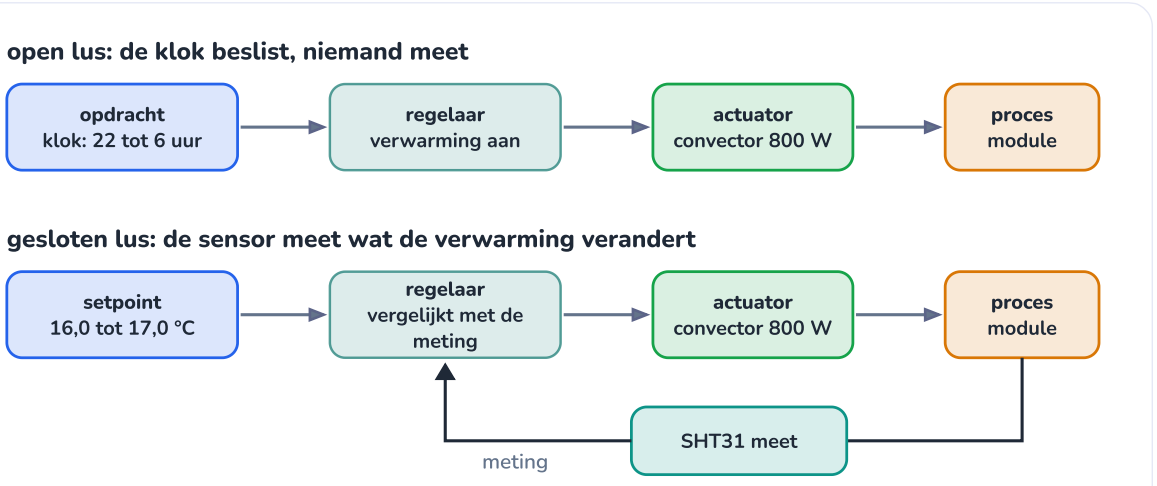
In Fundamental (hoofdstuk 25) leerde je het patroon **meten, beslissen, doen**, en het verschil tussen een tuinsproeier met een timer en een pomp die pas water geeft als de grond droog is. Nu maak je dat precies.

Een **open lus** stuurt volgens een plan of een opdracht, zonder te meten wat er gebeurt: 'verwarming aan van 22 tot 6 uur'. Is de nacht zachter dan verwacht, dan wordt het te warm; is ze kouder, te koud. Een **gesloten lus** meet de **geregelde grootheid** (de luchttemperatuur) en past de actie aan zolang er een verschil is met wat je wil. Die pijl terug van de sensor naar de regelaar heet **terugkoppeling**.

systeem	lus	wat wordt er gemeten?
broodrooster met draaiknop	open	niets: de tijd beslist
pomp van de module, 10 min per uur (hoofdstuk 29)	open	niets: de klok beslist
stappenmotor zonder encoder (hoofdstuk 26)	open	niets: de motor telt zijn stappen
groeilampen dimmen met het daglicht (hoofdstuk 25)	open, met voorsturing	het daglicht, niet het licht op de planten
servomotor (hoofdstuk 26)	gesloten	de hoek, met een potmeter in de servo
wasmachine die water inlaat	gesloten	het waterniveau in de trommel
verwarming van de module (dit hoofdstuk)	gesloten	de luchttemperatuur

Een open lus is eenvoudig, goedkoop en snel, en kan nooit gaan slingeren. Maar ze corrigeert niets: een storing (een koude nacht, een open deur) of een fout in het plan werkt volledig door. Een gesloten lus corrigeert storingen die ze niet kent, maar ze heeft een sensor nodig die het juiste meet, en ze kan schommelen of doorschieten. Daarover gaat dit hoofdstuk.

> **Voorsturing: de storing meten**
>
> De groeilampen meten het daglicht en dimmen vooraf, nog voor de planten te veel licht krijgen. Dat heet **voorsturing**: je meet de storing, niet het resultaat. Snel, maar blind voor alles wat je niet meet: als een lamp veroudert en minder licht geeft, merkt de regeling niets. Echte installaties combineren daarom vaak voorsturing met terugkoppeling.



Dezelfde verwarming, twee manieren van sturen. Alleen de onderste ziet wat de verwarming echt doet.

UITLEG · 31.3

De taal van het blokschema

Ingenieurs tekenen elke regelkring op dezelfde manier. De blokken zijn apparaten of processen, de pijlen zijn signalen: grootheden die van het ene blok naar het andere gaan.

- Het **proces** is wat je wil regelen: de lucht in de module, een auto, een lichaam.
- De **actuator** grijpt in het proces in: de convector die warmte geeft, de gasklep, de zweetklieren.
- De **sensor** meet de **geregelde grootheid** y en geeft de **meting** y_m.
- De **regelaar** vergelijkt de meting met de **gewenste waarde** of **setpoint** r en berekent de **stuurgrootheid** u: wat de actuator moet doen.
- Een **storing** d werkt van buitenaf in: kou, wind, een helling, een open deur.

Wat de actuator aan het proces geeft (een warmtestroom, een kracht), heet in de figuur q.

Het rondje met Σ is een **vergelijkingspunt**: het telt de signalen op met hun teken. Zo ontstaat de **regelafwijking** of **fout**:

REGELAFWIJKING (FOUT)

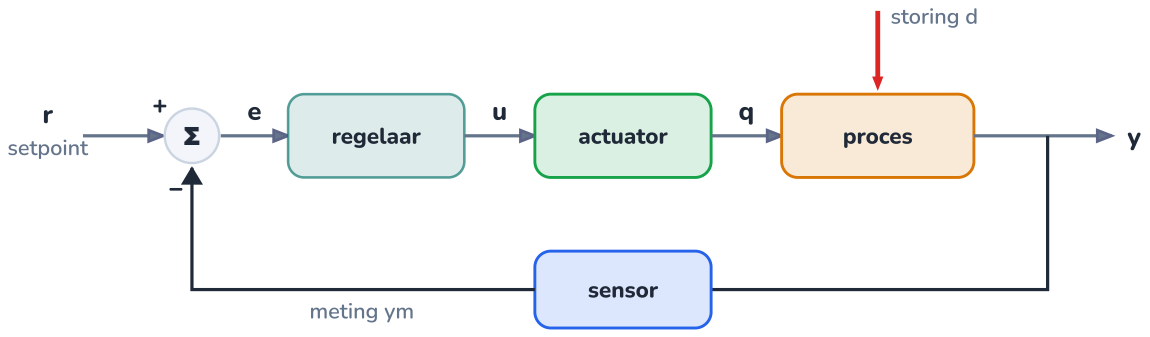
$$e = r - y_m$$

Is het 16,2 °C en wil je 16,5 °C, dan is $e = +0{,}3\,\mathrm{K}$: te koud. Een negatieve fout betekent te warm.

Het minteken maakt er een **negatieve terugkoppeling** van: een te hoge meting maakt de fout kleiner of negatief, en de regelaar doet minder. Zo dempt de lus afwijkingen uit. Bij **positieve terugkoppeling** versterkt een afwijking zichzelf: een microfoon die zijn eigen luidspreker opvangt, gaat fluiten (rondzingen).

De regelaar kent alleen de meting

Hangt de sensor in de zon of vlak boven de convector, dan regelt de lus perfect, maar naar de verkeerde temperatuur. Een regelkring is nooit beter dan zijn meting.



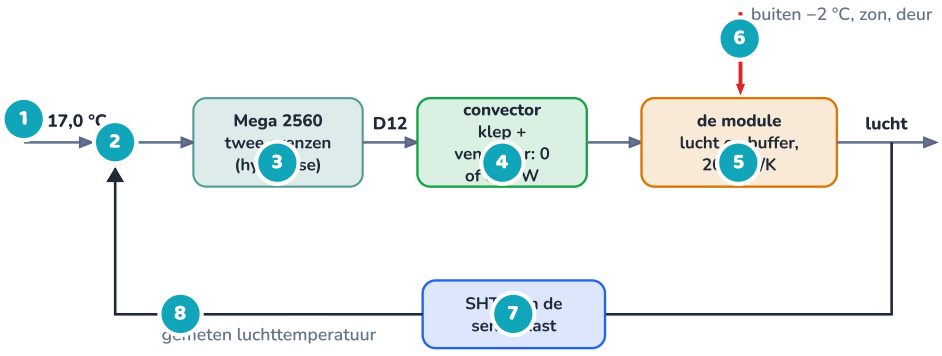
Elke regelkring in dezelfde taal: de fout e gaat naar de regelaar, de meting komt terug via de sensor.

ONTLEED · 31.4

Het blokschema van de verwarming van de module

De klimaatmodule heeft geen eigen verwarming op haar batterij: de warmte komt van het station. Een warmwaterleiding van de stationsverwarming loopt naar een **ventiloconvector** in de module: een klein radiatortje met een ventilator erachter. Als de Mega

verwarmt, zet pin D12 (nieuw in `pinnen.h`) via een MOSFET een magneetklep open en het ventilatortje aan; dan komt er ongeveer **800 W** warmte in de module.



De verwarming van de klimaatmodule als gesloten regelkring (nachtstand).

1 Setpoint
Geen enkel getal maar een band: 's nachts aan onder 16,0 °C en uit boven 17,0 °C, overdag 18,0 en 19,0 °C.

2 Vergelijken
In de code: `luchtT < laag` en `luchtT > hoog`. De Mega doet het om de 2 s.

3 Regelaar
Een aan/uit-regelaar met hysterese: hij onthoudt of de verwarming aan staat, en verandert alleen voorbij een grens.

4 Actuator
De convector geeft 0 of 800 W: meer standen heeft hij niet. Bij −8 °C buiten verliest de module 495 W bij 16,5 °C: er blijft reserve om op te warmen.

5 Proces
De lucht (en alles wat snel meewarmt) en de buffer: het vat van 200 L en het reservoir. Verlies naar buiten: 20,2 W per graad verschil (hoofdstuk 19).

6 Storingen
De buitentemperatuur, de zon door het kasdak, de lampen (190 W warmte), een deur die opengaat. De regelaar kent ze niet: hij ziet alleen hun gevolg.

7 Sensor
De SHT31 hangt in de sensorkast (hoofdstuk 30). Die kast volgt de lucht met een vertraging van ongeveer een minuut.

8 Terugkoppeling
De meting gaat terug naar het vergelijkingspunt: daardoor is dit een gesloten lus.

UITLEG · 31.5

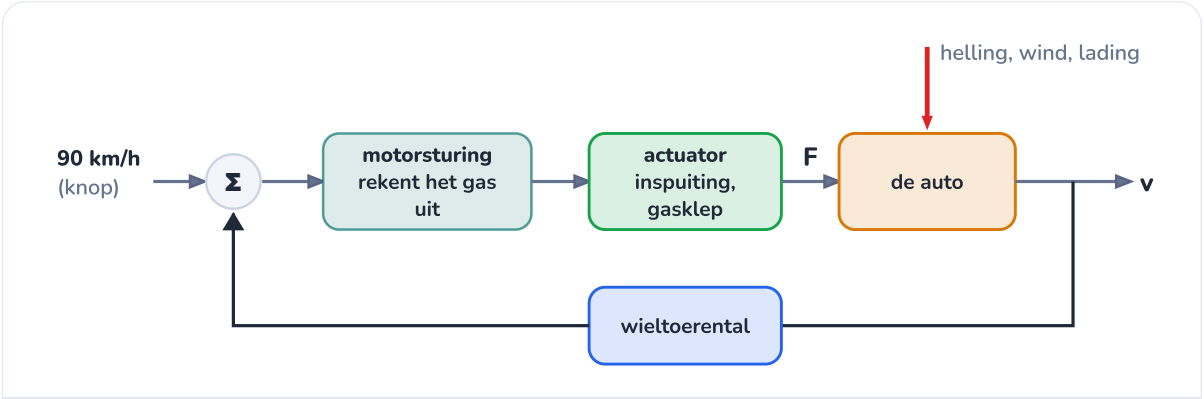
Cruisecontrol: dezelfde lus op de weg

5/21

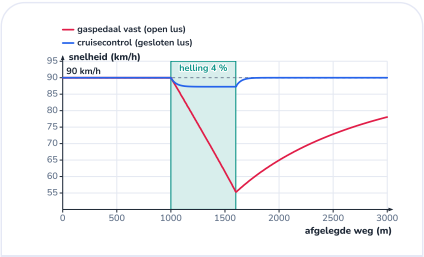
Bij een cruisecontrol is het **proces** de auto en de **geregelde grootheid** de snelheid. De **sensor** telt de omwentelingen van een wiel, de **regelaar** is de motorsturing, de **actuator** de inspuiting of de gasklep. De **storingen** zijn een helling, tegenwind, een zware lading.

De grafiek toont een simulatie: een auto van 1200 kg rijdt aan 90 km/h en komt aan een helling van 4 % over 600 m. Met het gaspedaal vast (een open lus) zakt de snelheid tot 55 km/h: de helling vraagt 471 N extra, en niemand geeft meer gas. Met cruisecontrol zakt ze maar tot 87,3 km/h. Dat is geen aan/uit-regeling: de motorsturing geeft **evenredig** met de fout meer kracht, hier 600 N extra per m/s te traag. Aan/uit zou betekenen: gas vol of gas dicht, en een schokkende auto, precies wat Teetor wilde vermijden.

Merk op dat de auto op de helling niet helemaal 90 km/h haalt. Een regelaar die evenredig met de fout werkt, geeft alleen extra gas *als er een fout is*: er blijft een kleine **blijvende afwijking**. Echte cruisecontrols tellen daarom ook op hoe lang de fout al duurt (een I-actie); dat leer je in Expert (hoofdstuk 39). Een adaptieve cruisecontrol heeft nog een tweede lus: een radar meet de afstand tot de auto ervoor.



De snelheid is de geregelde grootheid; de helling is een storing die de regelaar niet kent.



Simulatie: op de helling verliest de auto met het gaspedaal vast 35 km/h, met cruisecontrol ongeveer 3 km/h.

UITGEWERKT VOORBEELD · 31.6

Hoeveel trager op de helling?

6/21

De auto uit de simulatie (1200 kg) rijdt 90 km/h ($25\,\mathrm{m/s}$) op een vlakke weg. Hij komt aan een helling van 4 %. (a) Hoeveel extra kracht vraagt de helling? (b) Hoe snel vertraagt de auto in het begin als het gaspedaal niet verandert? (c) De cruisecontrol geeft 600 N extra per m/s die de auto te traag rijdt. Welke snelheid houdt hij op de helling?

GEGEVEN

- $m = 1200\,\mathrm{kg}$, $g = 9{,}81\,\mathrm{N/kg}$, helling 4 % (4 m hoger per 100 m)
- regelaar: $F_{\text{extra}} = 600\,\mathrm{N/(m/s)} \cdot (v_{\text{gewenst}} - v)$

GEVRAAGD

- extra kracht
- vertraging
- snelheid met cruisecontrol

OPLOSSING

1. (a) Een helling van 4 % maakt een hoek $\alpha = \arctan 0{,}04 = 2{,}29°$ (hoofdstuk 6). De component van de zwaartekracht langs de weg: $F = m\,g\sin\alpha = 1200 \cdot 9{,}81 \cdot 0{,}0400 = 471\,\mathrm{N}$.

2. (b) Met het gaspedaal vast verandert de motorkracht niet: die 471 N is een netto kracht naar achter. $a = F/m = 471/1200 = 0{,}39\,\mathrm{m/s^2}$ (hoofdstuk 8). Na 10 s is dat al $3{,}9\,\mathrm{m/s}$ of 14 km/h trager.

3. (c) De cruisecontrol vindt een evenwicht als zijn extra kracht de helling dekt: $600 \cdot \Delta v = 471$, dus $\Delta v = 0{,}78\,\mathrm{m/s} = 2{,}8\,\mathrm{km/h}$. De auto rijdt $90 - 2{,}8 = 87{,}2\,\mathrm{km/h}$.

ANTWOORD

(a) 471 N extra. (b) In het begin 0,39 m/s² vertraging. (c) Ongeveer 87,2 km/h: een blijvende afwijking van bijna 3 km/h.

Klopt dit? De simulatie geeft 87,3 km/h: iets beter, omdat de luchtweerstand bij een lagere snelheid ook iets kleiner wordt. Wil je een kleinere afwijking, dan kies je een sterkere regelaar (meer N per m/s), maar dan reageert de auto schokkeriger op elke kleine fout.

UITLEG · 31.7

Je lichaam: een regeling met vele actuatoren

7/21

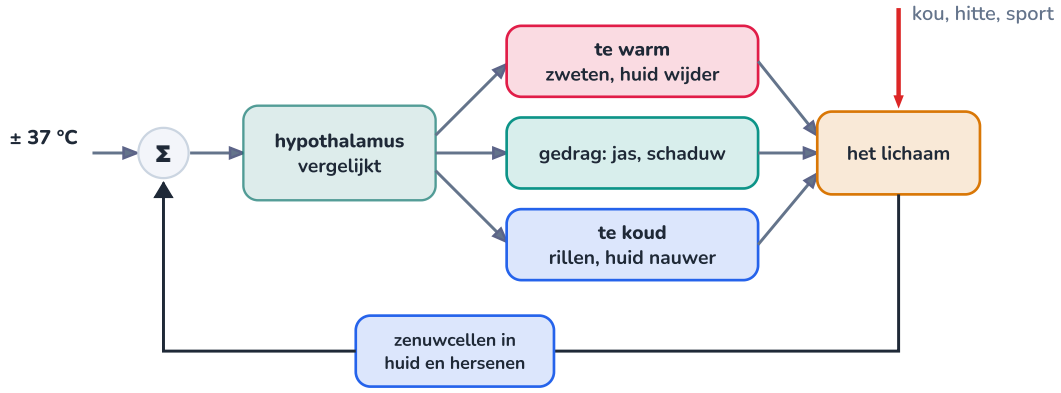
Je lichaam houdt zijn kerntemperatuur tussen ongeveer 36,5 en 37,5 °C (Fundamental, hoofdstuk 29). Zenuwcellen in de huid en in de hersenen zijn de **sensoren**, een deel van de hersenen, de **hypothalamus**, is de **regelaar**. Dat constant houden heet **homeostase**.

Het lichaam gebruikt zijn actuatoren **in trappen**. Wijkt de temperatuur een beetje af, dan past het alleen de doorbloeding van de huid aan: wijdere bloedvaten geven meer warmte af, nauwere houden ze binnen. Dat kost bijna niets. Pas bij een grotere afwijking volgen **zweten** of **rillen**. Zweten koelt sterk: een liter zweet die verdampt, neemt 2,4 MJ mee (hoofdstuk 18); een halve liter per uur is al $1{,}2\,\mathrm{MJ}/3600\,\mathrm{s} = 333\,\mathrm{W}$ koeling. Rillen laat spieren warmte maken. En je **gedrag** is ook een actuator: een jas aantrekken, de schaduw zoeken.

Twee dingen zijn anders dan bij een eenvoudige thermostaat. Het lichaam regelt **geleidelijk**: een beetje te warm geeft een beetje zweet. En de huid meet de omgeving al voor de kern afkoelt: het lichaam reageert op kou voor het echt te koud is, een vorm van voorsturing. De klimaatmodule doet iets gelijkaardigs in trappen: eerst de ventilatoren, pas bij meer warmte het dak.

Koorts: een verschoven setpoint

Bij een infectie zet de hypothalamus de gewenste waarde hoger, bijvoorbeeld op 39 °C. Je lichaam is dan 'te koud' tegenover zijn nieuwe setpoint: je rilt en hebt het koud, terwijl je temperatuur stijgt. De regelkring werkt perfect, alleen met een ander setpoint.



Eén regelaar, meerdere actuatoren: de hypothalamus kiest de actie die past bij de grootte van de fout.

UITLEG · 31.8

Drie soorten regelaars

8/21

Wat doet de regelaar met de fout? Drie mogelijkheden, van eenvoudig naar verfijnd:

- **Aan/uit met één drempel.** Kouder dan 16,5 °C: aan; warmer: uit. Eenvoudig, maar rond de drempel schakelt de actuator voortdurend, want elke kleine schommeling van de meting zit er afwisselend boven en onder.
- **Aan/uit met hysterese.** Twee grenzen: aan onder 16,0 °C, uit boven 17,0 °C. Daartussen verandert er niets: de regelaar **onthoudt** of de verwarming aan staat. Warmt de lucht op, dan blijft de verwarming aan tot 17,0 °C; koelt ze af, dan blijft ze uit tot 16,0 °C. De grafiek is een lus die je maar in één richting doorloopt.

- **Evenredig (P).** Hoe groter de fout, hoe meer vermogen: $u = K_p \cdot e$, begrensd tussen 0 en het maximum. Daarvoor moet de actuator tussenstanden hebben (een dimbare lamp, een klep die half open kan, een motor met PWM). De cruisecontrol werkt zo.

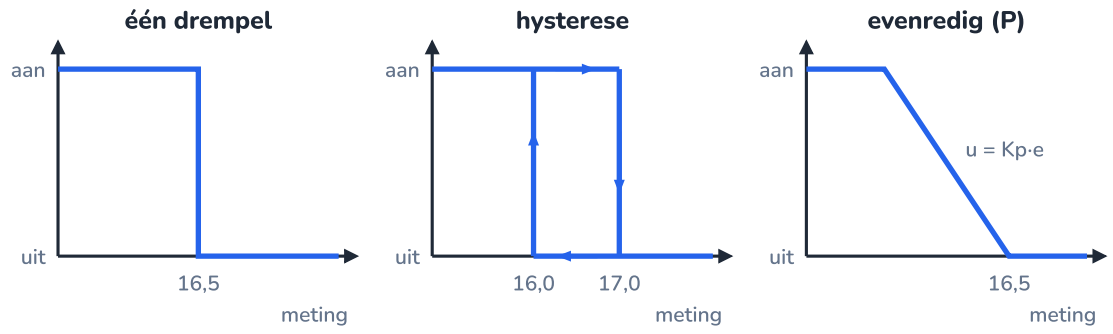
In code is hysterese twee regels. Belangrijk is het woord `else` en de variabele die de toestand onthoudt:

```
if (!verwarmen && luchtT < laag) verwarmen = true;        // koud genoeg: aan
else if (verwarmen && luchtT > hoog) verwarmen = false;  // warm genoeg: uit
// daartussen: niets veranderen
```

Hysterese ken je al: de Schmitt-trigger 74HC14 schakelt bij 2,6 V en terug bij 1,6 V (hoofdstuk 24), de groeilampen gaan aan onder 4000 lx en uit boven 6000 lx (hoofdstuk 22), en het dakraam van de meetkas opent boven 28 °C en sluit onder 25 °C.

De thermostaat hoger zetten warmt sneller op

Een aan/uit-verwarming geeft altijd haar volle vermogen, hier 800 W. Een hoger setpoint maakt de module niet sneller warm; de verwarming blijft alleen langer aan, en de temperatuur gaat hoger. Wie de thermostaat op 25 °C zet om sneller 19 °C te hebben, krijgt na een tijd 25 °C.



Links schakelt de regelaar op één punt, midden onthoudt hij zijn toestand tussen twee grenzen, rechts doseert hij.

PROEF · 31.9

Een thermostaat in een doos

9/21

ONDERZOEKSVRAAG

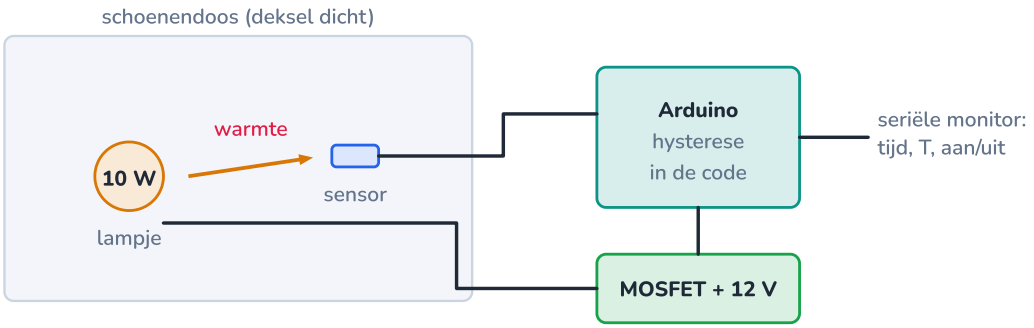
Hoe vaak schakelt een verwarming met hysterese, en hoe ver schiet de temperatuur voorbij de grenzen? Hangt dat af van de breedte van de band en van waar de sensor zit?

MATERIAAL

- schoenendoos met deksel
- lampje 12 V / 10 W in een metalen fitting (of vermogensweerstand 15 Ω / 10 W)
- logic-level MOSFET (IRLZ44N) met 220 Ω naar de gate en 10 kΩ naar GND, 12 V-voeding
- Arduino met temperatuursensor (DS18B20, of TMP36 met het gemiddelde van 16 metingen)
- computer met de seriële monitor
- stopwatch of de klok van het programma

Veiligheid

Het lampje wordt heet: laat minstens 3 cm ruimte tot het karton en laat de opstelling nooit alleen. Alleen 12 V. Stop de proef als het in de doos warmer wordt dan 40 °C.



De opstelling: een lampje als verwarming, een sensor op 10 cm, een Arduino met een MOSFET.

WERKWIJZE

1. Bouw de schakeling van hoofdstuk 23: de MOSFET schakelt het lampje, de Arduino meet de temperatuur. Leg de sensor op 10 cm van het lampje.
2. Programmeer een regeling met hysterese rond 27 °C: aan onder LAAG, uit boven HOOG. Laat elke seconde de tijd, de temperatuur en de stand (0 of 1) naar de seriële monitor sturen.
3. Begin met een band van 1,0 K (26,5 en 27,5 °C). Wacht tot de doos opgewarmd is (ongeveer 20 minuten) en meet dan 20 minuten.
4. Tel hoe vaak het lampje aangaat en noteer de hoogste en laagste temperatuur in die 20 minuten.
5. Herhaal met 0,5 K en 2,0 K. Leg daarna de sensor vlak tegen het lampje en herhaal met 1,0 K.

WAARNEMING

band	keer aan in 20 min	laagste (°C)	hoogste (°C)
0,5 K	11	26,4	27,5
1,0 K	7	26,1	27,7
2,0 K	5	25,6	28,2

Voorbeeldmeting (sensor op 10 cm). Met de sensor tegen het lampje schakelt het lampje veel sneller, en is de lucht in de rest van de doos kouder dan de sensor aangeeft.

VERKLARING

Hoe smaller de band, hoe vaker het lampje schakelt: de temperatuur heeft minder tijd nodig om van de ene grens naar de andere te gaan. Maar de temperatuur blijft niet tussen de grenzen: bij 1,0 K schommelt ze tussen 26,1 en 27,7 °C. Twee oorzaken. De warmte moet eerst van het lampje naar de sensor, en de sensor zelf moet opwarmen: de meting loopt achter. En een uitgeschakeld lampje is nog heet en geeft nog even warmte. Beide laten de temperatuur **doorschieten** voorbij de grens. Een sensor tegen het lampje regelt de temperatuur van het lampje, niet die van de doos: de meting moet de juiste grootheid meten.

Een TMP36 geeft per stap van de ADC ongeveer 0,49 °C (5 V / 1024 / 10 mV per °C): voor een band van 0,5 K is dat te grof. Neem het gemiddelde van 16 metingen, of een digitale sensor zoals de DS18B20 (0,0625 °C per stap).

Het warmtemodel van de module

Om een regeling te ontwerpen voor je ze bouwt, heb je een **model** nodig: rekenregels die voorspellen hoe de temperatuur verandert. Voor de klimaatmodule volstaan twee temperaturen:

- de **lucht** T_L, met alles wat in enkele minuten mee opwarmt (planten, goten, lampen, de binnenkant van de wanden): samen $C_L = 80\,\mathrm{kJ/K}$. De lucht alleen is maar 14 kJ/K;
- de **buffer** T_W: het vat van 200 L en het reservoir van 60 L, samen $C_W = 1{,}10\,\mathrm{MJ/K}$ (hoofdstuk 18).

De lucht verliest $H = 20{,}2\,\mathrm{W}$ per graad verschil met buiten (hoofdstuk 19) en wisselt $G = 30\,\mathrm{W}$ per graad verschil uit met de buffer. Elke 2 s rekent het model de warmte die binnenkomt en buitengaat, en zet de temperatuur een stapje verder, zoals in hoofdstuk 18 ($Q = C \cdot \Delta T$):

EEN STAP VAN HET MODEL

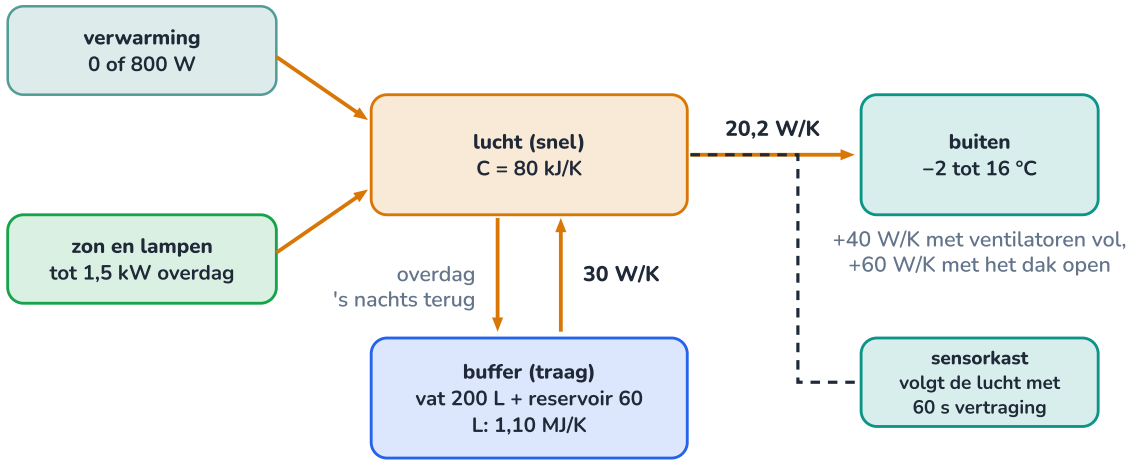
$$T_L \leftarrow T_L + \frac{P_{\mathrm{verw}} + P_{\mathrm{zon}} - H\,(T_L - T_b) - G\,(T_L - T_W)}{C_L}\,\Delta t$$

T_b: buitentemperatuur, $\Delta t = 2\,\mathrm{s}$. De buffer krijgt op dezelfde manier $G\,(T_L - T_W)\,\Delta t / C_W$ bij. De sensor volgt de lucht met een tijdconstante $\tau_s = 60\,\mathrm{s}$: $T_m \leftarrow T_m + (T_L - T_m)\,\Delta t / \tau_s$, zoals een condensator die oplaadt (hoofdstuk 24).

Waarom twee temperaturen? In hoofdstuk 18 koelde alles samen af, als één blok van 1,1 MJ/K. Voor een hele nacht klopt dat ongeveer. Voor een regeling die om de paar minuten schakelt niet: de lucht warmt in enkele minuten een graad op, het water in de vaten merkt daar nauwelijks iets van. Een model moet passen bij de vraag die je stelt.

Waarom 800 W?

Bij de ontwerptemperatuur van −8 °C (hoofdstuk 19) en 16,5 °C binnen verliest de module $20{,}2 \cdot 24{,}5 = 495\,\mathrm{W}$. Een verwarming die net 495 W geeft, houdt de temperatuur, maar warmt nooit op na een open deur. Met 800 W blijft er 305 W reserve. De warmte komt uit de verwarmingskring van het station; de 12 V-batterij van de module (hoofdstuk 20) moet er niets voor leveren.

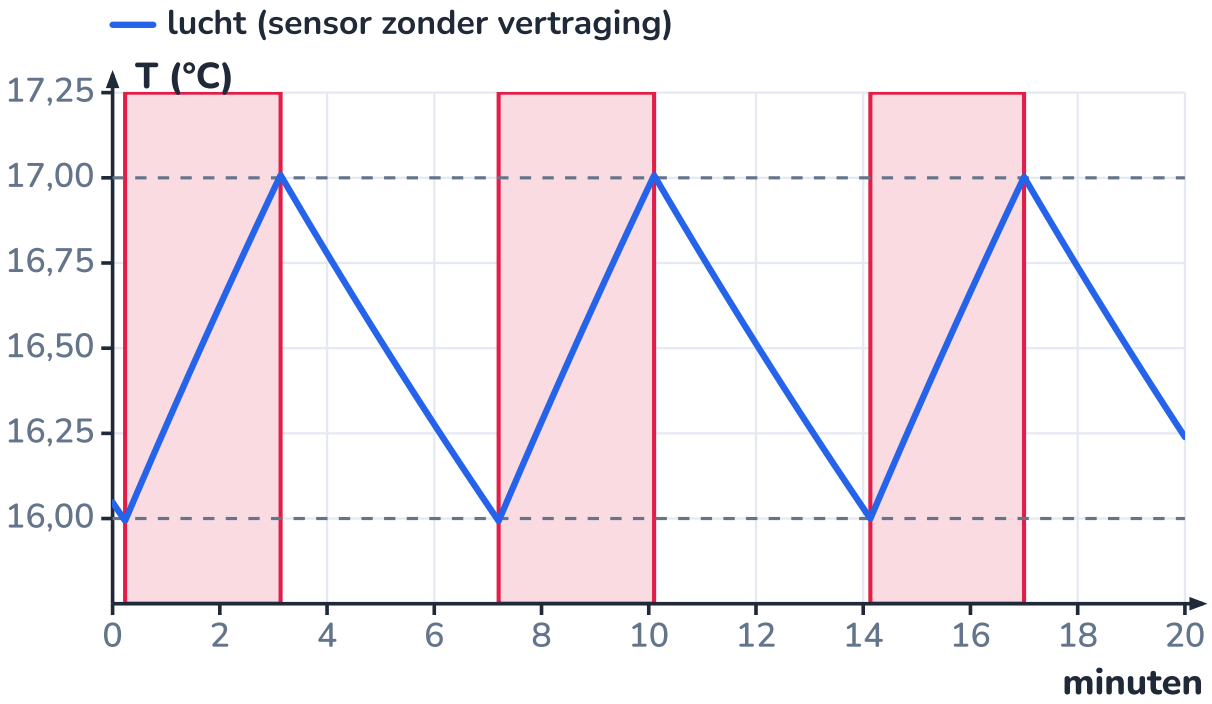


Twee temperaturen: de lucht reageert in minuten, de buffer in uren.

UITGEWERKT VOORBEELD · 31.11

Hoe lang staat de verwarming aan?

Het is een windstille januarinacht, buiten 0 °C. De verwarming (800 W) gaat aan onder 16,0 °C en uit boven 17,0 °C. De buffer heeft ongeveer de gemiddelde luchttemperatuur, 16,5 °C, zodat hij gemiddeld geen warmte geeft of neemt. Neem aan dat de sensor zonder vertraging meet. Bereken hoe lang de verwarming telkens aan en uit staat, hoe vaak ze per uur aangaat, welk deel van de tijd ze aan staat, en hoeveel warmte een nacht van 8 uur kost.



De zaagtand uit het voorbeeld: stijgen tegen 0,35 °C per minuut, dalen tegen 0,25 °C per minuut. Rood: verwarming aan.

GEGEVEN

- $C_L = 80\,000\,\mathrm{J/K}$, $H = 20{,}2\,\mathrm{W/K}$, $P = 800\,\mathrm{W}$
- band van 16,0 tot 17,0 °C: $\Delta T = 1{,}0\,\mathrm{K}$; buiten $0\,^\circ\mathrm{C}$

GEVRAAGD

- aantijd en uittijd
- keer aan per uur
- inschakelduur
- warmte per nacht

OPLOSSING

1. Verlies bij gemiddeld 16,5 °C: $V = H \cdot (T_L - T_b) = 20{,}2 \cdot 16{,}5 = 333\,\mathrm{W}$.
2. Opwarmen: er komt netto $P - V = 800 - 333 = 467\,\mathrm{W}$ binnen. De lucht moet $C_L \cdot \Delta T = 80\,000\,\mathrm{J}$ opnemen: $t_{\text{aan}} = \dfrac{80\,000}{467} = 171\,\mathrm{s}$.
3. Afkoelen: nu gaat er $333\,\mathrm{W}$ uit en komt er niets binnen: $t_{\text{uit}} = \dfrac{80\,000}{333} = 240\,\mathrm{s}$.
4. Eén volledige zaagtand duurt $171 + 240 = 411\,\mathrm{s}$ (6,9 min): $\dfrac{3600}{411} = 8{,}75$ keer aan per uur.
5. Inschakelduur: $\dfrac{171}{411} = 0{,}417$, of 41,7 %. Dat is precies $V/P = 333/800$: gemiddeld moet de verwarming juist het verlies dekken.
6. Warmte per nacht: gemiddeld levert de verwarming precies het verlies, dus $333{,}3\,\mathrm{W} \cdot 8\,\mathrm{h} = 2666\,\mathrm{Wh} \approx 2{,}7\,\mathrm{kWh}$.

ANTWOORD

Telkens 171 s aan en 240 s uit; 8,7 keer aan per uur; 42 % van de tijd aan; ongeveer 2,7 kWh warmte per nacht.

Klopt dit? De simulatie met hetzelfde model (zonder vertraging en zonder ruis) geeft 8,57 keer aan per uur en 41,5 % aan. Het kleine verschil komt van de buffer, die toch wat warmte uitwisselt, en van het meten om de 2 s. Merk op: de inschakelduur hangt niet af van de band, het aantal schakelingen wel.

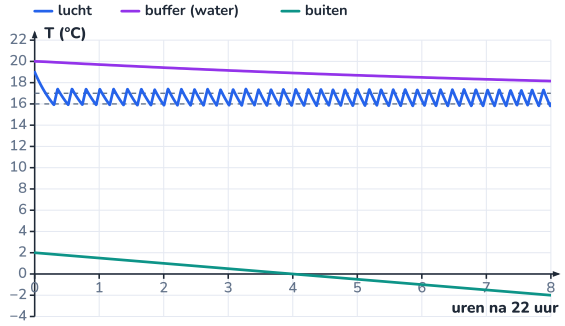
UITLEG · 31.12

De zaagtand van een echte nacht

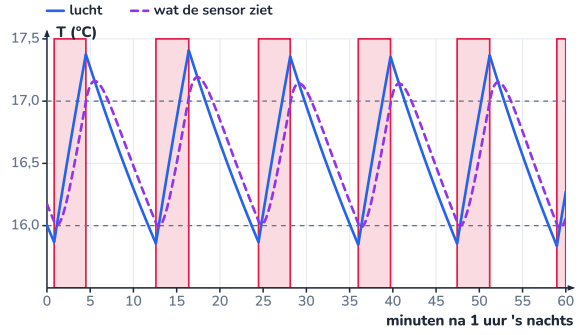
12/21

Nu een realistische januarinacht: om 22 uur gaan de lampen uit, de lucht is 19 °C, de buffer 20 °C, en buiten daalt de temperatuur van 2 naar −2 °C. De sensor hangt in de sensorkast en loopt een minuut achter, en hij ruist een beetje (±0,03 °C). De regelaar gebruikt de nachtband: aan onder 16,0, uit boven 17,0 °C.

- Na 18 minuten is de lucht onder 16 °C gezakt en gaat de verwarming voor het eerst aan.
- Daarna tekent de lucht een **zaagtand**: 5,3 keer aan per uur, tussen 15,8 en 17,4 °C. Dat is breder dan de band van 1,0 K: de trage sensor laat de lucht voorbij beide grenzen gaan (verderop).
- Hoe kouder het buiten wordt, hoe langer de verwarming aan staat: van 24 % van de tijd tussen 23 en 24 uur tot 38 % tussen 5 en 6 uur.
- De buffer koelt de hele nacht langzaam af, van 20 naar 18,2 °C. Zolang hij warmer is dan de lucht, geeft hij warmte: de verwarming hoeft minder.
- Van 23 tot 6 uur geeft de verwarming 1,88 kWh warmte (gemiddeld 269 W).



De hele nacht: de lucht zaagt tussen de twee grenzen, de buffer daalt traag, buiten wordt het kouder.



Eén uur ingezoomd, met de grenzen 16,0 en 17,0 °C. Rood: verwarming aan. De sensor (stippellijn) loopt achter op de lucht.

CODE · 31.13

De thermostaat simuleren

13/21

Het model past in een Python-functie van twintig regels. Ze simuleert een nacht van 8 uur bij 0 °C en telt hoe vaak de verwarming aangaat. Zo probeer je in een paar seconden grenzen uit die je op de echte module weken zou moeten testen.

Python **thermostaat.py** 33 regels

```python
# thermostaat.py - de verwarming van de klimaatmodule met hysterese
C_L, C_W = 80e3, 1.10e6      # J/K: de lucht (met wat snel meewarmt) en de buffer (vat + reservoir)
H, G = 20.2, 30.0            # W/K: lucht -> buiten, lucht <-> buffer
P = 800.0                    # W: verwarming aan
DT = 2.0                     # s: om de 2 s meten en beslissen
TAU_S = 60.0                 # s: tijdconstante van de sensor in de sensorkast
```

```python


def nacht(laag, hoog, uren=8.0, buiten=0.0):
    lucht = buffer = meting = 16.5
    aan = False
    keer_aan, tijd_aan = 0, 0.0
    t_min, t_max = 99.0, -99.0
    for stap in range(int(uren * 3600 / DT)):
        if not aan and meting < laag:          # de regelaar: twee grenzen
            aan = True
            keer_aan += 1
        elif aan and meting > hoog:
            aan = False
        warmte = (P if aan else 0.0) - H * (lucht - buiten) - G * (lucht - buffer)   # W
        lucht_nieuw = lucht + warmte * DT / C_L
        buffer += G * (lucht - buffer) * DT / C_W
        meting += (lucht - meting) * DT / TAU_S
        lucht = lucht_nieuw
        if aan:
            tijd_aan += DT
        t_min, t_max = min(t_min, lucht), max(t_max, lucht)
    return keer_aan / uren, 100 * tijd_aan / (uren * 3600), t_min, t_max


for breedte in (0.2, 0.5, 1.0, 2.0):
    per_uur, pct, lo, hi = nacht(16.5 - breedte / 2, 16.5 + breedte / 2)
    print(f"hysterese {breedte:.1f} K: {per_uur:4.1f} keer aan per uur, {pct:.0f} % aan, lucht {lo:.2f} tot {hi:.2f}
°C")
```

Uitvoer:

```
hysterese 0.2 K: 12.8 keer aan per uur, 42 % aan, lucht 16.18 tot 16.88 °C
hysterese 0.5 K:  8.4 keer aan per uur, 42 % aan, lucht 16.01 tot 17.07 °C
hysterese 1.0 K:  5.6 keer aan per uur, 42 % aan, lucht 15.76 tot 17.34 °C
hysterese 2.0 K:  3.5 keer aan per uur, 42 % aan, lucht 15.28 tot 17.82 °C
```

Drie dingen vallen op. De inschakelduur is altijd dezelfde: de verwarming moet het verlies dekken, welke band je ook kiest. Het aantal schakelingen daalt sterk met de breedte. En de lucht schommelt telkens meer dan de band breed is. De volgorde van de regels in de lus is belangrijk: eerst beslissen met de *meting*, dan de lucht, de buffer en de sensor één stap verder zetten, allemaal met de *oude* waarden.

UITLEG · 31.14

Hoe breed moet de hysterese zijn?

14/21

Met één drempel en een sensor die een beetje ruist, is het een ramp. De meting springt afwisselend boven en onder 16,5 °C, en de verwarming gaat 382 keer per uur aan: om de paar seconden. Een relais of een magneetklep is dan na enkele weken versleten. Een band van 0,2 K brengt dat al terug tot 14 keer per uur.

Een bredere band geeft minder schakelingen, maar een grotere schommeling. De simulatie (buiten 0 °C, sensor met 60 s vertraging, gemeten van het eerste tot het achtste uur) geeft:

hysterese	keer aan per uur	lucht (°C)	gem. afwijking (K)	warmte in 7 h (kWh)
één drempel	381,7	16,36 tot 16,67	0,03	2,34
0,2 K	14,3	16,17 tot 16,88	0,16	2,35
0,5 K	8,7	16,00 tot 17,10	0,25	2,34
1,0 K	5,7	15,75 tot 17,36	0,38	2,33

hysterese	keer aan per uur	lucht (°C)	gem. afwijking (K)	warmte in 7 h (kWh)
1,5 K	4,3	15,51 tot 17,59	0,50	2,37
2,0 K	3,4	15,26 tot 17,84	0,63	2,36
3,0 K	2,4	14,79 tot 18,31	0,87	2,35

Zonder vertraging volgt het aantal keer per uur uit het rekenvoorbeeld: het is omgekeerd evenredig met de breedte van de band. Met een trage sensor schiet de lucht aan elke kant verder door; dat werkt alsof de band breder is. Een vuistregel voor een korte vertraging:

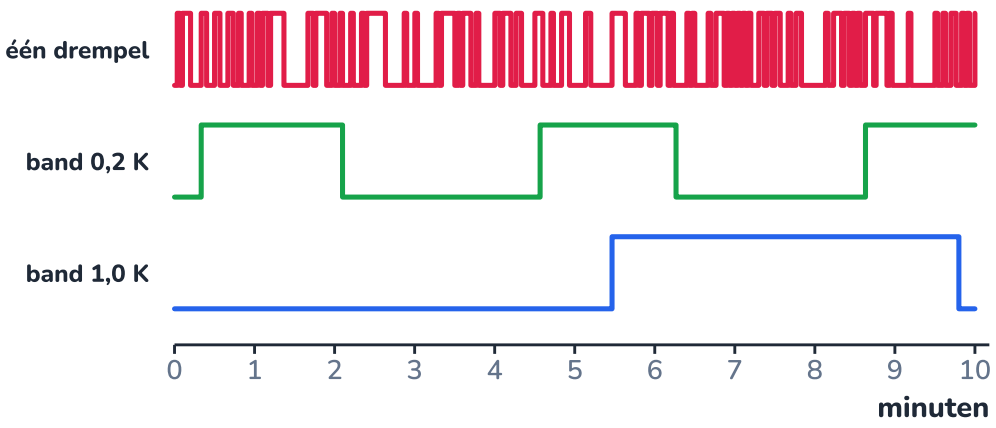
EFFECTIEVE BREEDTE VAN DE BAND

$$\Delta T_{\text{eff}} \approx \Delta T + \frac{P \cdot \tau_s}{C_L}$$

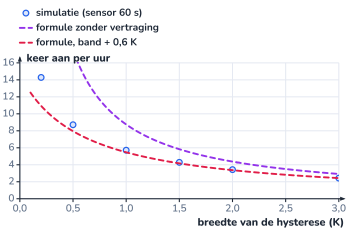
Voor de module: $800 \cdot 60/80\,000 = 0{,}6\,\text{K}$ extra. Een band van 1,0 K gedraagt zich als 1,6 K: formule 5,5 keer per uur, simulatie 5,7.

Een smallere band kost meer energie

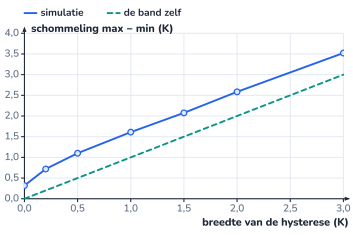
Nee: de warmte die de verwarming levert, is altijd het verlies van de module. In de tabel verschilt ze tussen 0,2 en 3,0 K nauwelijks. Een smallere band laat de verwarming alleen vaker en korter aangaan. Wat wel energie spaart: een lagere gemiddelde temperatuur, of minder verlies (een nachtscherm).



Tien minuten uit dezelfde nacht. Met één drempel flikkert de verwarming; met een band van 1,0 K schakelt ze rustig.



Aantal keer aan per uur: de simulatie volgt de formule met de effectieve band (band + 0,6 K).



Hoe breder de band, hoe groter de schommeling; door de vertraging altijd meer dan de band zelf.

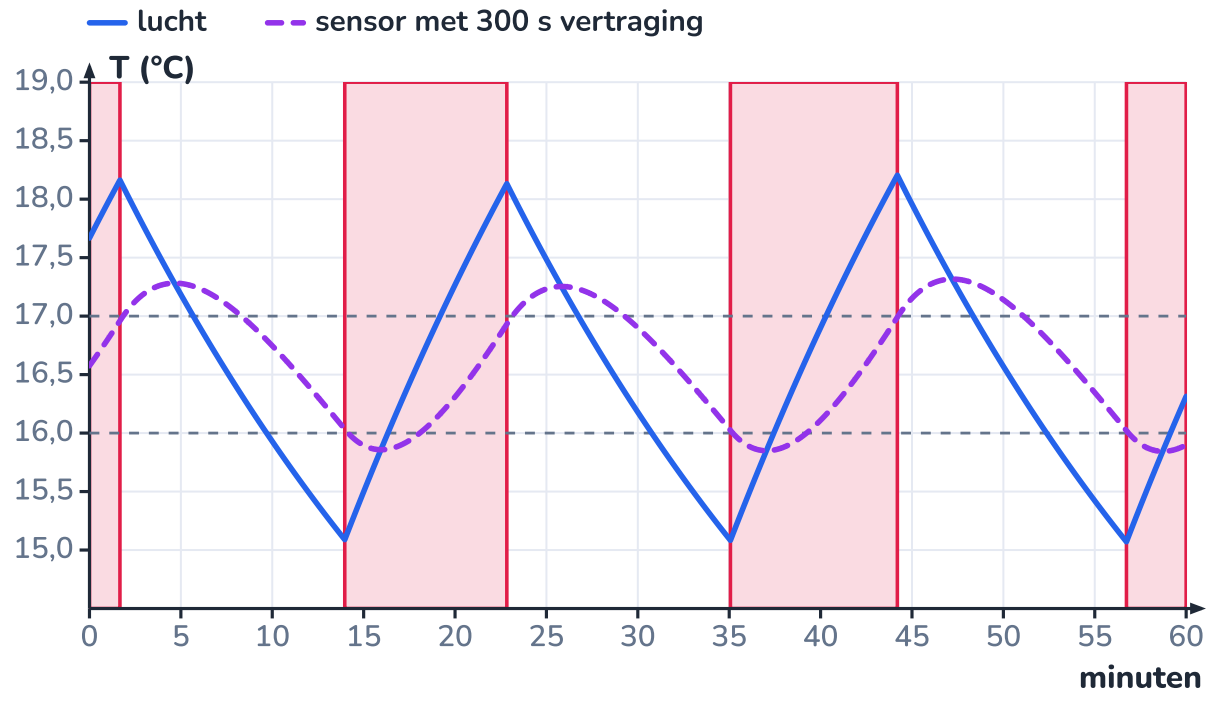
Meetvertraging: de regeling schiet door

Een sensor ziet een verandering nooit meteen. De SHT31 zelf reageert in enkele seconden, maar hij hangt in een sensorkast, en de lucht in die kast moet eerst meewarmen. Zet de verwarming uit boven 17,0 °C, dan is de lucht rond de convector al warmer, en hij warmt nog verder op terwijl de sensor achterloopt. De temperatuur **schiet door** voorbij de grens, boven én onder.

tijdconstante sensor	keer aan per uur	boven 17,0 (K)	onder 16,0 (K)	gem. afwijking (K)
0 s	9,1	0,04	0,02	0,24
60 s	5,7	0,36	0,25	0,38
180 s	3,6	0,87	0,64	0,61
300 s	2,9	1,25	0,96	0,78

Een trage sensor laat de verwarming dus minder vaak schakelen, maar de lucht schommelt meer. Het is geen gratis bredere band: de regeling weet niet meer precies waar ze is. Met 300 s vertraging zakt de lucht tot 15,0 °C, onder de grens van 16,0 °C die je voor de sla gekozen had.

De doorschot is ongeveer de snelheid waarmee de lucht verandert maal de vertraging. Opwarmen gaat hier sneller (0,35 K per minuut) dan afkoelen (0,25 K per minuut): daarom schiet de lucht boven verder door dan onder. Waar hang je de sensor dus? Niet in de luchtstroom van de convector (dan schakelt hij te snel en regelt hij de convector in plaats van de module), niet in een dichte doos (te traag), maar midden in de ruimte, op de hoogte van de planten, in een kastje met openingen of een klein ventilatortje dat lucht aanzuigt.



Met een sensor die 300 s achterloopt, gaat de lucht meer dan een graad voorbij de grenzen (16,0 en 17,0 °C, stippellijnen).

UITGEWERKT VOORBEELD · 31.16

Grenzen kiezen met eisen

16/21

De magneetklep van de convector gaat volgens de fabrikant 200 000 schakelingen mee (rekenwaarde). Het team stelt drie eisen: (1) de lucht zakt nooit onder 15,5 °C, (2) de klep houdt minstens tien stookseizoenen, (3) de gemiddelde afwijking van 16,5 °C is zo klein mogelijk. Een stookseizoen telt 200 dagen, met gemiddeld 12 uur per dag in het ritme van een koude nacht. Welke band kies je: 0,5, 1,0 of 2,0 K?

GEGEVEN

- de tabel van 'Hoe breed moet de hysterese zijn?'
- klep: 200 000 schakelingen; 200 dagen × 12 h per seizoen

GEVRAAGD

- de band die aan alle eisen voldoet

OPLOSSING

1. Eis 1: de laagste temperatuur is 16,00 °C (0,5 K), 15,75 °C (1,0 K) en 15,26 °C (2,0 K). De band van 2,0 K valt af.
2. Eis 2, schakelingen per seizoen: $8{,}7 \cdot 12 \cdot 200 = 20\,913$ (0,5 K) en $5{,}7 \cdot 12 \cdot 200 = 13\,713$ (1,0 K).
3. De klep houdt $200\,000 / 20\,913 = 9{,}6$ seizoenen (0,5 K) of $200\,000 / 13\,713 = 14{,}6$ seizoenen (1,0 K). De band van 0,5 K valt net af.
4. Eis 3 kiest dan tussen wat overblijft: alleen 1,0 K, met een gemiddelde afwijking van 0,38 K.

ANTWOORD

Een band van 1,0 K: 16,0 tot 17,0 °C 's nachts. Dezelfde breedte krijgt de dagband: 18,0 tot 19,0 °C.

Klopt dit? Een ontwerpkeuze is altijd een afweging. Een betere klep (een miljoen schakelingen) laat een smallere band toe. Een snellere sensor (een kastje met een ventilatortje) geeft met dezelfde band minder doorschot, maar meer schakelingen: de effectieve band wordt smaller. Reken elke combinatie na met de tabel en de vuistregel voor de effectieve band.

UITLEG · 31.17

Koelen in twee trappen: ventilatoren en dak

17/21

Overdag is het probleem omgekeerd. Op een heldere aprildag komt er door het kasdak tot 1,5 kW zonnewarmte binnen. Zonder koelen zou de lucht in het model tot 53 °C stijgen: sla verbrandt. De module koelt in twee **trappen**, elk met hysterese:

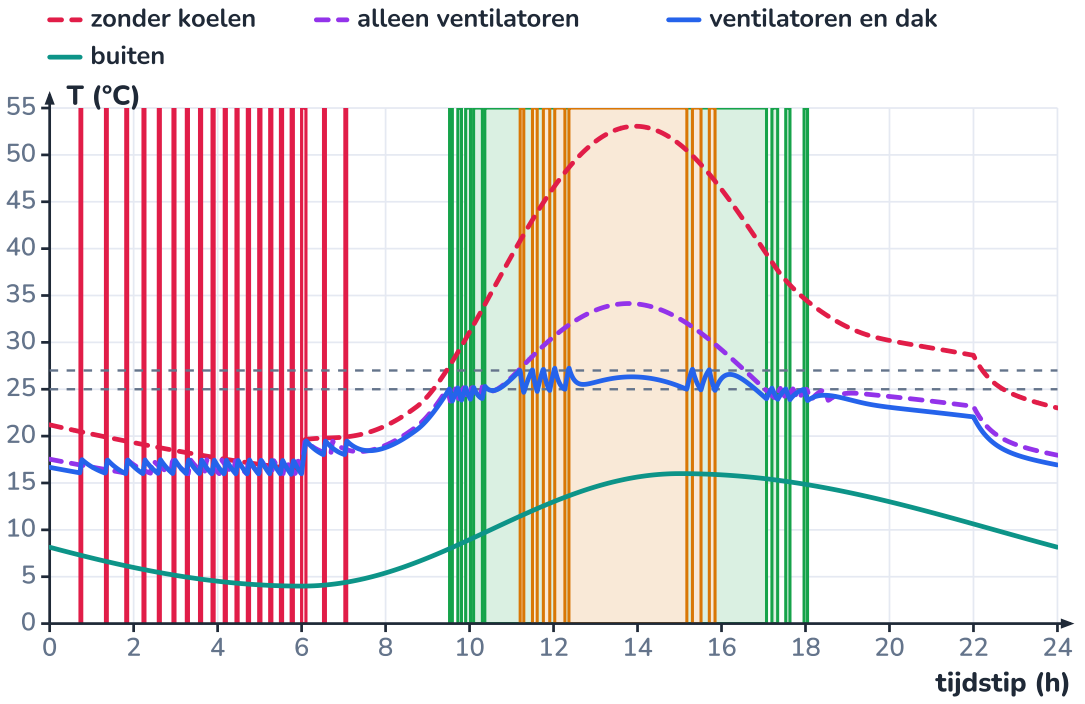
- **Trap 1, de ventilatoren:** ze draaien altijd zacht voor de basisventilatie (hoofdstuk 14). Boven 25 °C draaien ze vol (ongeveer 120 m³ lucht per uur), onder 24 °C weer zacht.
- **Trap 2, het dak:** boven 27 °C gaat het kasdak open, onder 25 °C weer dicht. Een open dak ververst ongeveer 180 m³ per uur.

Met alleen de ventilatoren haalt de lucht nog 34,1 °C; met ventilatoren en dak blijft ze onder 27,3 °C. Die dag draaien de ventilatoren 8 keer vol, samen 7,5 uur; het dak gaat 7 keer open, samen 3,7 uur. De verwarming gaat 19 keer aan, vooral 's nachts en vroeg in de ochtend (dagband vanaf 6 uur: 18,0 tot 19,0 °C).

Tussen 19 °C (verwarming uit) en 24 °C (ventilatoren vol) ligt een **dode zone** waarin geen enkele actuator iets doet. Die zone is bewust: verwarmen en koelen mogen nooit tegelijk, want dan betaal je voor warmte die je meteen weer wegblaast. In de code staat daarom een extra regel: als de verwarming aan is, draaien de ventilatoren niet vol.

Het dak is geen lampje

Een dak openen duurt 7 s en een motor start met een grote stroom (hoofdstuk 26). De hysterese moet daarom breder zijn dan bij de ventilatoren: 2 K in plaats van 1 K. Het openen en sluiten zelf, met eindeloopschakelaars en time-out, is een toestandsmachine: dat programmeer je in hoofdstuk 32.



Rood: verwarming aan; groen: ventilatoren vol; oranje: dak open. De stippellijnen tonen wat er zonder koeling gebeurt.

CODE · 31.18

De regeling in C++

18/21

In `pinnen.h` komt één regel bij: `const int VERWARMING = 12;`. De nieuwe taak meet om de 2 s de SHT31, kiest de dag- of nachtband volgens de klok (DS3231, hoofdstuk 30) en schakelt de verwarming en de ventilatoren met hysterese. Ze komt in `klimaat.ino` (versie 3); hier staat ze als losse schets, zoals ze getest is. Twee dingen zijn nieuw: een **uuroverzicht** waarmee de regeling zichzelf beoordeelt, en een **noodprogramma** als de sensor niet antwoordt.

C++ (Arduino) **klimaatregeling.ino** 125 regels

```cpp
// klimaatregeling.ino - verwarming en ventilatoren van de klimaatmodule met hysterese (hoofdstuk 31)
// Deze taak komt in klimaat.ino (versie 3) naast taakMeten(); hier als losse schets om ze te testen.
#include <Wire.h>
#include <RTClib.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"

// ------------------------------------------------------------- grenzen (graden C): aan onder LAAG, uit boven HOOG
const float DAG_LAAG = 18.0, DAG_HOOG = 19.0;        // verwarming van 6 tot 22 uur
const float NACHT_LAAG = 16.0, NACHT_HOOG = 17.0;    // verwarming 's nachts
const float KOEL_UIT = 24.0, KOEL_AAN = 25.0;        // ventilatoren vol boven 25, terug laag onder 24
const int VENT_LAAG = 90, VENT_VOL = 255;            // PWM: basisventilatie (hoofdstuk 14) of koelen
const unsigned long MEET_MS = 2000;                  // om de 2 s meten en beslissen
const unsigned long VENSTER_MS = 600000UL;           // noodprogramma: vensters van 10 minuten

// ------------------------------------------------------------- toestand
Adafruit_SHT31 sht;
RTC_DS3231 rtc;
```

```
float luchtT = NAN;
bool verwarmen = false, koelen = false, noodprogramma = false;
float gemAan = 0.5;                                  // gemiddelde inschakelduur (0 tot 1) van het laatste uur
unsigned long vorigeMeting = 0, vensterStart = 0;
int starts = 0, vorigUur = -1;                       // voor het uuroverzicht
unsigned long aanMs = 0;
float tMin = 99.0, tMax = -99.0;

// ------------------------------------------------------------------ meldingen
void printTwee(int x) {
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void melding(const char* tekst) {                    // "# 22:13:24 tekst"
  DateTime k = rtc.now();
  Serial.print("# ");
  printTwee(k.hour());
  Serial.print(':');
  printTwee(k.minute());
  Serial.print(':');
  printTwee(k.second());
  Serial.print(' ');
  Serial.println(tekst);
}

void uurOverzicht(int uur) {                         // elk vol uur: hoe deed de regeling het?
  if (vorigUur >= 0) {
    melding("uuroverzicht");
    Serial.print("  keer aan: ");
    Serial.print(starts);
    Serial.print("   aan: ");
    Serial.print(aanMs / 36000.0, 0);                // ms per uur -> procent
    Serial.print(" %   lucht: ");
    Serial.print(tMin, 1);
    Serial.print(" tot ");
    Serial.print(tMax, 1);
    Serial.println(" C");
  }
  vorigUur = uur;
  starts = 0;
  aanMs = 0;
  tMin = 99.0;
  tMax = -99.0;
}

// ------------------------------------------------------------------ de regeling
void regelen(unsigned long nu, int uur) {
  bool dag = (uur >= 6 && uur < 22);
  float laag = dag ? DAG_LAAG : NACHT_LAAG;
  float hoog = dag ? DAG_HOOG : NACHT_HOOG;
  bool was = verwarmen, wasKoelen = koelen;
  if (isnan(luchtT)) {                               // geen meting: open lus met het gemiddelde van het laatste uur
    if (!noodprogramma) {
      noodprogramma = true;
      vensterStart = nu;
      melding("ALARM: geen temperatuur, noodprogramma");
    }
    if (nu - vensterStart >= VENSTER_MS) vensterStart += VENSTER_MS;
    verwarmen = (nu - vensterStart) < (unsigned long)(gemAan * VENSTER_MS);
    koelen = false;
  } else {
    if (noodprogramma) {
      noodprogramma = false;
      melding("temperatuur terug: gewone regeling");
    }
    if (!verwarmen && luchtT < laag) verwarmen = true;          // hysterese: twee grenzen
    else if (verwarmen && luchtT > hoog) verwarmen = false;
```

```
    if (!koelen && luchtT > KOEL_AAN) koelen = true;
    else if (koelen && luchtT < KOEL_UIT) koelen = false;
    if (verwarmen) koelen = false;                          // nooit tegelijk verwarmen en koelen
    gemAan += ((verwarmen ? 1.0 : 0.0) - gemAan) * (MEET_MS / 3600000.0);
    tMin = min(tMin, luchtT);
    tMax = max(tMax, luchtT);
  }
  if (verwarmen && !was) starts++;
  if (verwarmen) aanMs += MEET_MS;
  if (koelen != wasKoelen) melding(koelen ? "ventilatoren vol: koelen" : "ventilatoren terug laag");
  digitalWrite(VERWARMING, verwarmen ? HIGH : LOW);
  analogWrite(VENTILATOREN, koelen ? VENT_VOL : VENT_LAAG);
}

void taakKlimaat(unsigned long nu) {
  if (nu - vorigeMeting < MEET_MS) return;
  vorigeMeting += MEET_MS;
  float t, rv;
  if (sht.readBoth(&t, &rv)) luchtT = t;
  else luchtT = NAN;                                  // de sensor antwoordt niet (draad los, sensor stuk)
  int uur = rtc.now().hour();
  if (uur != vorigUur) uurOverzicht(uur);
  regelen(nu, uur);
}

void setup() {
  pinMode(VERWARMING, OUTPUT);
  digitalWrite(VERWARMING, LOW);                      // veilig starten: verwarming uit
  pinMode(VENTILATOREN, OUTPUT);
  Serial.begin(115200);
  Wire.begin();
  rtc.begin();
  if (!sht.begin(0x44)) Serial.println("# ALARM: geen SHT31 op adres 0x44");
  melding("klimaatregeling gestart");
}

void loop() {
  taakKlimaat(millis());
}
```

Getest in een gesloten lus: een C++-versie van het warmtemodel speelt de module en geeft de temperatuur door aan een gesimuleerde SHT31. Om 2:10 gaat het stekkertje van de sensor een halfuur los. De seriële monitor:

```
# 22:00:00 klimaatregeling gestart
# 23:00:00 uuroverzicht
  keer aan: 4   aan: 21 %   lucht: 15.9 tot 19.1 C
# 00:00:00 uuroverzicht
  keer aan: 4   aan: 26 %   lucht: 15.9 tot 17.3 C
# 01:00:00 uuroverzicht
  keer aan: 5   aan: 30 %   lucht: 16.0 tot 17.3 C
# 02:00:00 uuroverzicht
  keer aan: 5   aan: 31 %   lucht: 15.9 tot 17.2 C
# 02:10:02 ALARM: geen temperatuur, noodprogramma
# 02:40:02 temperatuur terug: gewone regeling
# 03:00:00 uuroverzicht
  keer aan: 6   aan: 36 %   lucht: 15.8 tot 17.2 C
# 04:00:00 uuroverzicht
  keer aan: 6   aan: 37 %   lucht: 15.9 tot 17.2 C
# 05:00:00 uuroverzicht
  keer aan: 5   aan: 37 %   lucht: 15.9 tot 17.2 C
```

```
# 06:00:00 uuroverzicht
  keer aan: 6   aan: 42 %   lucht: 15.9 tot 17.2 C
```

Het uuroverzicht bevestigt de simulatie in Python: vijf à zes keer aan per uur, de meting tussen 15,8 en 17,3 °C, en een inschakelduur die stijgt naarmate het kouder wordt. Tijdens het noodprogramma regelt de Mega **blind**: hij zet de verwarming elke 10 minuten even lang aan als ze het afgelopen uur gemiddeld aan stond. Dat is een open lus, maar een verstandige: de lucht bleef in de test tussen 15,7 en 17,4 °C. Zonder de test `isnan(luchtT)` was elke vergelijking met `nan` onwaar geweest, en was de verwarming blijven staan zoals ze stond, aan of uit.

Een tweede test op een zonnige aprildag (10 tot 18 uur) toont de grens van trap 1: de ventilatoren draaien vanaf 10:11 vol, maar de lucht stijgt toch tot 34,0 °C om 13:47. Zonder dak lukt het niet.

CODE · 31.19

Een regeling beoordelen met een meting

19/21

Werkt de regeling op de echte module zoals in de simulatie? Dat bewijs je met een meting. Voor een testnacht laat je de Mega elke 10 s een regel naar de computer sturen (hoofdstuk 30): tijdstip, gemeten temperatuur, stand van de verwarming. De eerste regels van het bestand `nacht.csv`:

```
tijdstip,temperatuur_C,verwarming
23:00:00,16.78,1
23:00:10,16.78,1
23:00:20,16.81,1
23:00:30,16.92,1
23:00:40,17.00,0
23:00:50,17.11,0
```

Vier **kengetallen** vatten een aan/uit-regeling samen: de gemiddelde afwijking van het midden van de band (nauwkeurigheid), de laagste en hoogste waarde (schommeling), het aantal keer aan per uur (slijtage) en de inschakelduur (energie).

Python **beoordeel.py** 23 regels

```python
# beoordeel.py - hoe goed regelde de verwarming vannacht? (testmeting: elke 10 s een regel)
import csv

MIDDEN = 16.5            # midden van de band 16,0 tot 17,0 °C
STAP = 10                # s tussen twee regels

temps, aan = [], []
with open("nacht.csv", newline="") as f:
    for rij in csv.DictReader(f):
        temps.append(float(rij["temperatuur_C"]))
        aan.append(rij["verwarming"] == "1")

uren = len(temps) * STAP / 3600
keer_aan = sum(1 for vorige, nu in zip(aan, aan[1:]) if nu and not vorige)
gem = sum(temps) / len(temps)
afwijking = sum(abs(t - MIDDEN) for t in temps) / len(temps)
inschakelduur = sum(aan) / len(aan)

print(f"meetduur {uren:.1f} h, {len(temps)} metingen")
print(f"temperatuur gemiddeld {gem:.2f} °C, van {min(temps):.2f} tot {max(temps):.2f} °C")
print(f"gemiddelde afwijking van {MIDDEN} °C: {afwijking:.2f} °C")
print(f"verwarming {keer_aan} keer aan: {keer_aan / uren:.1f} keer per uur, {100 * inschakelduur:.0f} % van de tijd")
print(f"warmte van het station: {0.8 * inschakelduur * uren:.2f} kWh")
```

Uitvoer:

```
meetduur 7.0 h, 2520 metingen
temperatuur gemiddeld 16.59 °C, van 15.93 tot 17.24 °C
gemiddelde afwijking van 16.5 °C: 0.34 °C
verwarming 37 keer aan: 5.3 keer per uur, 34 % van de tijd
warmte van het station: 1.88 kWh
```

De gemeten waarden liggen dichter bij elkaar dan de echte luchttemperatuur (15,80 tot 17,42 °C): de sensor loopt achter en vlakt de toppen af. Een meting beoordeelt dus altijd de lucht zoals de sensor ze ziet. Wie de echte schommeling wil kennen, hangt voor de test een tweede, snelle sensor in de ruimte.

NASLAG · 31.20

Formules, kengetallen en grenzen

regelafwijking

$$e = r - y_m$$

symbool	betekenis	eenheid
r	setpoint, gewenste waarde	°C
y_m	meting	°C

aan- en uittijd (zonder vertraging)

$$t_{\text{aan}} = \frac{C\,\Delta T}{P - V} \qquad t_{\text{uit}} = \frac{C\,\Delta T}{V}$$

symbool	betekenis	eenheid
C	warmtecapaciteit die meewarmt	J/K
ΔT	breedte van de band	K
P	vermogen van de verwarming	W
V	verlies bij het midden van de band	W

keer aan per uur en inschakelduur

$$f = \frac{3600}{t_{\text{aan}} + t_{\text{uit}}} \qquad \frac{t_{\text{aan}}}{t_{\text{aan}} + t_{\text{uit}}} = \frac{V}{P}$$

symbool	betekenis	eenheid
f	keer aan per uur	1/h

effectieve band met een trage sensor

$$\Delta T_{\text{eff}} \approx \Delta T + \frac{P\,\tau_s}{C}$$

symbool	betekenis	eenheid
τ_s	tijdconstante van de sensor	s

één stap van een warmtemodel

$$T \leftarrow T + \frac{P_{\text{in}} - P_{\text{uit}}}{C}\,\Delta t$$

symbool	betekenis	eenheid
Δt	tijdstap	s

kengetal	wat het zegt	module, januarinacht
gemiddelde afwijking van het midden	nauwkeurigheid	0,39 K
laagste en hoogste waarde	schommeling	15,8 tot 17,4 °C
keer aan per uur	slijtage van klep of relais	5,3
inschakelduur	energie	34 %, 1,88 kWh in 7 h

klimaatmodule	aan / vol / open	uit / laag / dicht	pin
verwarming overdag (6-22 h)	onder 18,0 °C	boven 19,0 °C	D12
verwarming 's nachts	onder 16,0 °C	boven 17,0 °C	D12
ventilatoren vol (anders zacht)	boven 25 °C	onder 24 °C	D11 (PWM 255 / 90)
kasdak (hoofdstuk 32)	boven 27 °C	onder 25 °C	D6-D8
sensor weg	verwarming blind, even lang als gemiddeld het laatste uur		

HET GEHEEL · 31.21

Waar regelen vandaan komt en waar het heen gaat

21/21

Regelen is de leerlijn waar de andere samenkomen: het proces is energie en warmte (hoofdstukken 18 en 19), de meting is elektronica en sensoren (22 en 30), de uitvoering is programmeren (29), en het ontwerp is een afweging met getallen. Ook de natuur regelt zo: homeostase is een regelkring. In Expert leer je regelaars die niet schommelen (PID) en modellen met afgeleiden in plaats van stapjes.

F25 Sensoren en automatisch regelen
Meten, beslissen, doen: de eerste drempel en het eerste dakraam met twee grenzen.

F29 Het lichaam als systeem
Homeostase en de lichaamstemperatuur als regelkring.

I18 Warmte berekenen
Q = C · ΔT en het watervat als nachtbuffer: de bouwstenen van het warmtemodel.

I19 Warmtetransport en isolatie
De 20,2 W/K van de module: het verlies dat de verwarming moet dekken.

I24 Condensatoren en de RC-tijd
De vertraging van de sensor gedraagt zich als een RC-kring; de Schmitt-trigger als hysterese in hardware.

I29 Microcontrollers programmeren in C++
millis(), taken en pinnen.h: de basis van de nieuwe taak.

I30 Datalogging en communicatie
Het logboek waarmee je een regeling beoordeelt.

I32 Toestandsmachines
Het dak, de modi van de module en de veilige toestand bij een fout.

E38 Dynamische systemen modelleren
Tijdconstanten en modellen met differentiaalvergelijkingen in plaats van stapjes.

E39 PID-regeling
Regelaars die doseren in plaats van schakelen: P, I en D, zonder zaagtand.

E33 Het lichaam als regelsysteem
Bloedsuiker en temperatuur: regelkringen in het lichaam, kwantitatief.

Habitat

De klimaatmodule krijgt haar klimaatregeling: verwarming met de warmte van het station (convector op D12) in een dag- en een nachtband, ventilatoren en dak als twee koeltrappen, een dode zone ertussen, en een noodprogramma als de sensor uitvalt. In hoofdstuk 32 worden dag, nacht, alarm en onderhoud modi van een toestandsmachine; in het integratieproject (hoofdstuk 40) test je de regeling met een logboek van een week.

SAMENVATTING

In het kort

- Een open lus stuurt zonder te meten; een gesloten lus meet de geregelde grootheid en corrigeert zo ook storingen die ze niet kent.
- Een blokschema toont regelaar, actuator, proces en sensor; de regelaar werkt met de fout $e = r - y_m$.
- Cruisecontrol en lichaamstemperatuur zijn gesloten lussen; het lichaam gebruikt zijn actuatoren in trappen en regelt geleidelijk.
- Een aan/uit-regeling met hysterese heeft twee grenzen en onthoudt haar toestand; één drempel laat de actuator flikkeren.
- Zonder vertraging: $t_{\text{aan}} = C\,\Delta T/(P - V)$ en $t_{\text{uit}} = C\,\Delta T/V$; de inschakelduur is V/P, onafhankelijk van de band.
- Een bredere band en een tragere sensor geven minder schakelingen maar meer schommeling; een trage sensor laat de lucht voorbij de grenzen schieten.
- Een regeling beoordeel je met kengetallen uit een meting: gemiddelde afwijking, schommeling, keer aan per uur en energie.

- De module verwarmt 's nachts tussen 16 en 17 °C en overdag tussen 18 en 19 °C, en koelt in twee trappen: ventilatoren boven 25 °C, dak boven 27 °C.

BEGRIPPEN

Wat je nu kent

gesloten lus
Regeling die de geregelde grootheid meet en haar actie aanpast aan het verschil met de gewenste waarde (terugkoppeling).

open lus
Sturing volgens een plan of opdracht, zonder te meten wat het resultaat is.

blokschema
Tekening van een regelkring met blokken (regelaar, actuator, proces, sensor) en pijlen voor de signalen.

regelafwijking (fout)
Het verschil $e = r - y_m$ tussen de gewenste waarde en de meting.

storing
Een invloed van buiten die de geregelde grootheid verandert, zoals de buitentemperatuur of een open deur.

hysterese
Een regelaar met twee grenzen: aan onder de ene, uit boven de andere; daartussen blijft hij in zijn toestand.

inschakelduur
Het deel van de tijd dat een aan/uit-actuator aan staat; gemiddeld dekt ze precies het verlies.

meetvertraging
De tijd die een sensor nodig heeft om een verandering te volgen; ze laat de regeling voorbij haar grenzen schieten.

negatieve terugkoppeling
Terugkoppeling die een afwijking verkleint: te warm geeft minder verwarming.

voorsturing
Een storing meten en er vooraf op reageren, zonder het resultaat te meten (open lus).

I INTERMEDIATE · DEEL I8 · REGELEN EN ROBOTICA

HANDBOEK · HOOFDSTUK 32

Toestandsmachines

Een programma dat onthoudt waar het is: van verkeerslicht tot de modi van de klimaatmodule

Een druk op dezelfde knop opent het kasdak, sluit het, of stopt het, naargelang wat het dak op dat ogenblik doet. Wie dat met losse als-dan-regels programmeert, verdwaalt snel. Een toestandsmachine ordent het: in elke toestand staat wat elke gebeurtenis doet. Je leert toestandsdiagrammen lezen en tekenen, programmeert ze met switch-case, en ontwerpt wat er gebeurt als er iets misloopt: de veilige toestand, time-outs en de waakhond. De klimaatmodule krijgt haar modi: dag, nacht, alarm en onderhoud.

6× uitleg · 2× uitgewerkt voorbeeld · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 5× code · ± 8 lestijden

R Regelen en robotica · P Programmeren en computationeel denken

Een knop, een paar lampen en een klok: het klassieke voorbeeld van een toestandsmachine.

Na dit hoofdstuk

- Je kan in een systeem toestanden, gebeurtenissen, overgangen en acties herkennen en benoemen.
- Je kan een toestandsdiagram lezen en tekenen met de notatie gebeurtenis [voorwaarde] / actie.
- Je kan een toestandsdiagram omzetten in een toestandstabel en ermee controleren of elke combinatie bedacht is.
- Je kan een toestandsmachine programmeren met enum en switch-case, zonder delay(), met de tijd in een toestand.
- Je kan fouten opsporen met time-outs en onmogelijke combinaties, en per fout een veilige toestand kiezen.
- Je kan uitleggen wat fail-safe betekent en wat een waakhond doet.
- Je kan de modi van de klimaatmodule ontwerpen, programmeren en met een testscenario controleren.

UITLEG · 32.1

Een programma dat onthoudt waar het is

1/17

Wat je al weet: in Fundamental reageerde een blokprogramma op gebeurtenissen ('wanneer op de knop gedrukt', hoofdstuk 23) en liet je een verkeerslicht branden met `delay()` (hoofdstuk 24). In hoofdstuk

26 en 29 kreeg het kasdak een variabele met drie waarden: STIL, SLUITEN en OPENEN. Dat was al een kleine **toestandsmachine**.

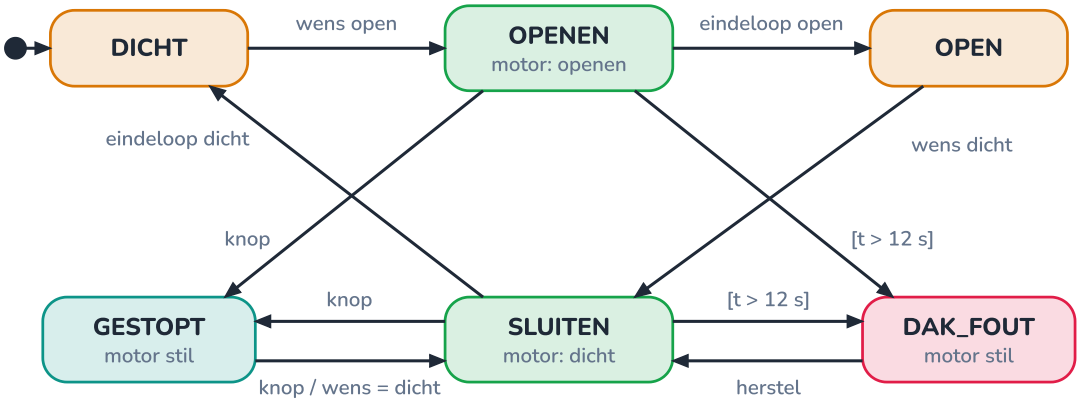
Een toestandsmachine bestaat uit vier soorten dingen:

- **Toestanden**: situaties waarin het systeem kan blijven, met een vast gedrag. Het dak is DICHT, OPENEN, OPEN, SLUITEN, GESTOPT of DAK_FOUT: altijd precies één daarvan.
- **Gebeurtenissen**: iets dat gebeurt. Een knopdruk, een eindeloopschakelaar die schakelt, een tijd die om is, een temperatuur die een grens passeert.
- **Overgangen**: van de ene toestand naar een andere, als een bepaalde gebeurtenis komt.
- **Acties**: wat het systeem doet bij een overgang of bij het binnenkomen van een toestand: de motor aanzetten, een lamp doven, een melding sturen.

Waarom zo? Omdat dezelfde gebeurtenis iets anders betekent in een andere toestand. Een druk op S1 opent het dak als het dicht is, sluit het als het open is, en stopt het als het beweegt. Met losse als-dan-regels verlies je dat snel uit het oog; met een toestandsmachine staat per toestand wat elke gebeurtenis doet, en wat ze niet doet.

Zes toestanden in plaats van drie

In hoofdstuk 29 onthield het programma de toestand in twee variabelen: `dak` (STIL, SLUITEN, OPENEN) en `dakOpen` (ja of nee). Twee variabelen maken samen zes combinaties, waarvan sommige geen zin hebben. Hier zit alles in één variabele: DICHT, OPEN en GESTOPT zijn drie verschillende soorten 'stil', en DAK_FOUT is nieuw.



Het kasdak: 'wens open' en 'wens dicht' komen van de knop of van de temperatuurregeling (hoofdstuk 31).

UITLEG · 32.2

Het toestandsdiagram

2/17

Een **toestandsdiagram** tekent een toestandsmachine. De afspraken:

- Een **toestand** is een afgeronde rechthoek met een naam die een situatie beschrijft: DICHT, WACHTEN, OPENEN. Een actie die gebeurt telkens je de toestand binnenkomt, schrijf je erin: 'bij binnenkomen: motor aan'.
- Het **beginpunt**, een zwart bolletje, wijst naar de toestand waarin de machine start.
- Een **overgang** is een pijl met het label **gebeurtenis [voorwaarde] / actie**. Elk deel mag ontbreken. 'eindeloop open' is alleen een gebeurtenis; '[t > 12 s]' is alleen een voorwaarde op de tijd in de

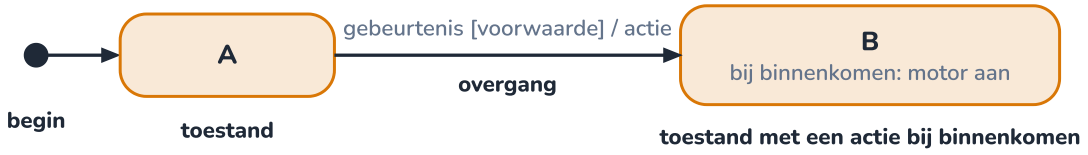
toestand; 'knop / wens = dicht' is een gebeurtenis met een actie.

- Een **zelflus** is een overgang naar dezelfde toestand: er gebeurt iets, maar de toestand blijft.
- Een gebeurtenis die in een toestand geen pijl heeft, wordt daar **genegeerd**. Dat moet een bewuste keuze zijn, geen vergetelheid.

Voor dat laatste is de **toestandstabel** handig: de toestanden als rijen, de gebeurtenissen als kolommen. Elk vakje zegt wat er gebeurt: een nieuwe toestand, een actie, of niets (–). Een diagram toont het verhaal, een tabel dwingt je om over elke combinatie na te denken. Die van het dak:

toestand	knop	wens open	wens dicht	eindeloop open	eindeloop dicht	t > 12 s
DICHT	– (wens = open)	OPENEN	–	–	–	–
OPENEN	GESTOPT	–	–	OPEN	–	DAK_FOUT
OPEN	– (wens = dicht)	–	SLUITEN	–	–	–
SLUITEN	GESTOPT	–	–	–	DICHT	DAK_FOUT
GESTOPT	SLUITEN (wens = dicht)	–	–	–	–	–
DAK_FOUT	–	–	–	–	–	–

DAK_FOUT verlaat je alleen met `dakHerstel()`, na een bevestigd alarm.




```
Voorbeeld: knop [dak dicht] / wens = open
```

als de knop ingedrukt wordt, en alleen als het dak dicht is: onthoud dat het dak open moet.

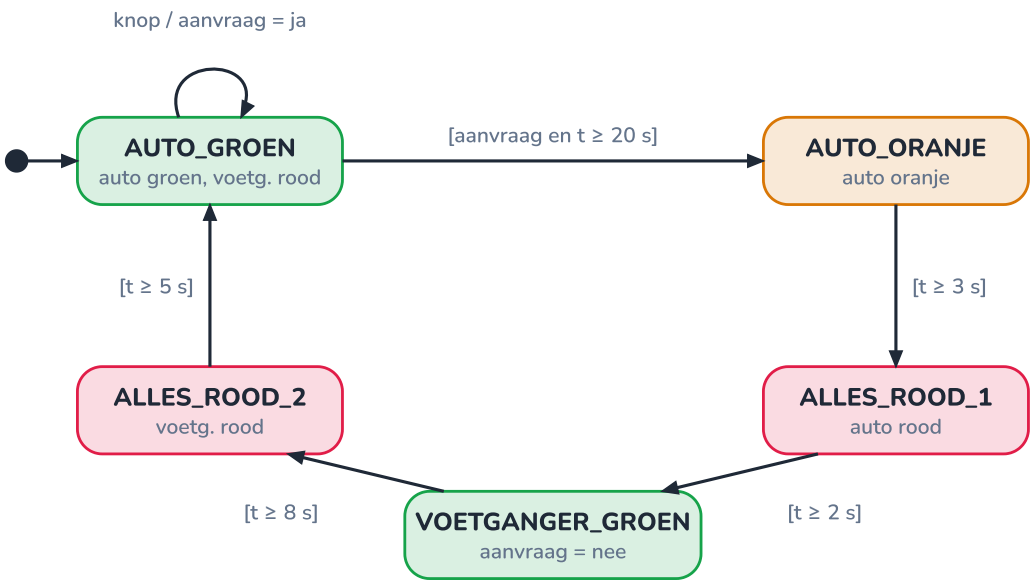
Alles wat je nodig hebt om een toestandsdiagram te lezen.

UITGEWERKT VOORBEELD · 32.3

Een voetgangersoversteek ontwerpen

3/17

Ontwerp de toestandsmachine van een voetgangersoversteek met drukknop. De auto's hebben groen tot er een voetganger drukt, maar altijd minstens 20 s. Na een druk: 3 s oranje, 2 s alles rood (het kruispunt leeg laten rijden), 8 s groen voor de voetgangers, 5 s alles rood (de laatste voetgangers halen de overkant), en dan weer groen voor de auto's. Een lampje 'wacht' toont dat de aanvraag genoteerd is. Een druk terwijl de voetgangers al groen hebben, telt niet.



De knop kan in elke toestand behalve VOETGANGER_GROEN een aanvraag noteren; de zelflus staat hier bij AUTO_GROEN.

GEGEVEN

- tijden: groen minstens 20 s, oranje 3 s, alles rood 2 s, voetgangers 8 s, alles rood 5 s
- een drukknop en een lampje 'wacht'

GEVRAAGD

- toestanden, overgangen en acties
- een toestandsdiagram

OPLOSSING

1. Toestanden zijn situaties met een ander gedrag. Hier: de lampen. AUTO_GROEN, AUTO_ORANJE, ALLES_ROOD_1, VOETGANGER_GROEN en ALLES_ROOD_2. Alles rood komt twee keer voor met dezelfde lampen, maar wat erna komt, verschilt: dus twee toestanden.
2. Gebeurtenissen: de knop, en de tijd in een toestand. De tijd schrijf je als voorwaarde: [t ≥ 3 s].
3. Een druk moet onthouden worden, ook als de auto's nog 15 s groen hebben. Daarvoor dient een extra variabele *aanvraag* (ja of nee): een zelflus 'knop / aanvraag = ja'. In VOETGANGER_GROEN heeft de knop geen pijl: genegeerd.
4. De overgang uit AUTO_GROEN heeft een samengestelde voorwaarde: [aanvraag en t ≥ 20 s]. Alle andere overgangen hangen alleen van de tijd af.
5. Acties bij binnenkomen: de lampen van elke toestand. In VOETGANGER_GROEN ook: aanvraag = nee, lampje 'wacht' uit.

ANTWOORD

Vijf toestanden in een kring, met één zelflus voor de knop (zie het diagram).

Klopt dit? Loop elke toestand na: wat als de knop komt? In AUTO_GROEN, AUTO_ORANJE en de twee rode toestanden wordt hij onthouden, in VOETGANGER_GROEN genegeerd. De langste wachttijd voor een voetganger: net na het begin van AUTO_GROEN drukken geeft 20 + 3 + 2 = 25 s tot groen.

CODE · 32.4

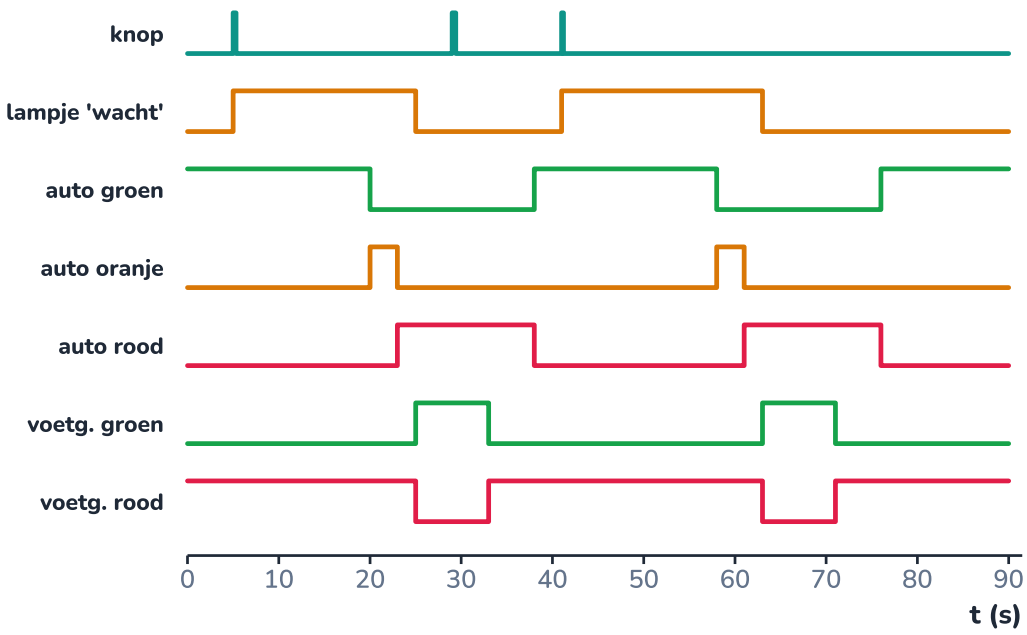
Het voetgangerslicht in C++

Het diagram wordt code in vier stappen: een `enum` met de toestanden, één variabele `toestand`, een functie `naar()` die de overgang doet en de acties bij binnenkomen uitvoert, en een `switch` in `loop()` die per toestand de gebeurtenissen bekijkt. Nergens `delay()`: de tijd in een toestand is `nu - sinds` (hoofdstuk 29).

C++ (Arduino) **verkeerslicht.ino** 81 regels

```cpp
// verkeerslicht.ino - een voetgangersoversteek met drukknop als toestandsmachine (hoofdstuk 32)
// Auto's: rood D4, oranje D5, groen D6. Voetgangers: rood D7, groen D8. Lampje 'wacht' D9. Knop D2 naar GND.

const int KNOP = 2;
const int A_ROOD = 4, A_ORANJE = 5, A_GROEN = 6, V_ROOD = 7, V_GROEN = 8, WACHT = 9;

const unsigned long GROEN_MIN_MS = 20000;   // auto's hebben minstens 20 s groen
const unsigned long ORANJE_MS = 3000;
const unsigned long ONTRUIM_MS = 2000;      // alles rood: het kruispunt leeg laten rijden
const unsigned long OVERSTEEK_MS = 8000;    // voetgangers groen
const unsigned long NALOOP_MS = 5000;       // alles rood: de laatste voetgangers halen de overkant

enum Toestand { AUTO_GROEN, AUTO_ORANJE, ALLES_ROOD_1, VOETGANGER_GROEN, ALLES_ROOD_2 };
const char* NAAM[] = {"AUTO_GROEN", "AUTO_ORANJE", "ALLES_ROOD_1", "VOETGANGER_GROEN", "ALLES_ROOD_2"};

Toestand toestand = AUTO_GROEN;
unsigned long sinds = 0;                    // wanneer de huidige toestand begon
bool aanvraag = false;                      // heeft een voetganger op de knop gedrukt?

// bij binnenkomen van een toestand: de lampen die erbij horen
void lampen(bool ar, bool ao, bool ag, bool vr, bool vg) {
  digitalWrite(A_ROOD, ar);
  digitalWrite(A_ORANJE, ao);
  digitalWrite(A_GROEN, ag);
  digitalWrite(V_ROOD, vr);
  digitalWrite(V_GROEN, vg);
}

void naar(Toestand nieuw, unsigned long nu) {
  toestand = nieuw;
  sinds = nu;
  switch (nieuw) {
    case AUTO_GROEN:       lampen(0, 0, 1, 1, 0); break;
    case AUTO_ORANJE:      lampen(0, 1, 0, 1, 0); break;
    case ALLES_ROOD_1:     lampen(1, 0, 0, 1, 0); break;
    case VOETGANGER_GROEN: lampen(1, 0, 0, 0, 1);
                           aanvraag = false;          // de aanvraag is beantwoord
                           digitalWrite(WACHT, LOW);
                           break;
    case ALLES_ROOD_2:     lampen(1, 0, 0, 1, 0); break;
  }
  Serial.print(nu / 1000.0, 1);
  Serial.print(" s  ");
  Serial.println(NAAM[nieuw]);
}

void setup() {
  pinMode(KNOP, INPUT_PULLUP);
  for (int p = A_ROOD; p <= WACHT; p++) pinMode(p, OUTPUT);
  Serial.begin(115200);
  naar(AUTO_GROEN, millis());
}

void loop() {
  unsigned long nu = millis();
  // gebeurtenis: knop ingedrukt (onthouden, behalve als de voetgangers al groen hebben)
  if (digitalRead(KNOP) == LOW && !aanvraag && toestand != VOETGANGER_GROEN) {
```

```
    aanvraag = true;
    digitalWrite(WACHT, HIGH);
    Serial.print(nu / 1000.0, 1);
    Serial.println(" s  knop: aanvraag");
  }
  unsigned long t = nu - sinds;              // hoe lang zijn we al in deze toestand?
  switch (toestand) {
    case AUTO_GROEN:
      if (aanvraag && t >= GROEN_MIN_MS) naar(AUTO_ORANJE, nu);
      break;
    case AUTO_ORANJE:
      if (t >= ORANJE_MS) naar(ALLES_ROOD_1, nu);
      break;
    case ALLES_ROOD_1:
      if (t >= ONTRUIM_MS) naar(VOETGANGER_GROEN, nu);
      break;
    case VOETGANGER_GROEN:
      if (t >= OVERSTEEK_MS) naar(ALLES_ROOD_2, nu);
      break;
    case ALLES_ROOD_2:
      if (t >= NALOOP_MS) naar(AUTO_GROEN, nu);
      break;
  }
}
```



De pinnen die het programma in de test schakelde: nooit twee groene lampen tegelijk.

Getest met drie knopdrukken: op 5 s, op 29 s en op 41 s. De seriële monitor:

```
0.0 s  AUTO_GROEN
5.0 s  knop: aanvraag
20.0 s  AUTO_ORANJE
23.0 s  ALLES_ROOD_1
25.0 s  VOETGANGER_GROEN
33.0 s  ALLES_ROOD_2
38.0 s  AUTO_GROEN
41.0 s  knop: aanvraag
58.0 s  AUTO_ORANJE
```

```
61.0 s  ALLES_ROOD_1
63.0 s  VOETGANGER_GROEN
71.0 s  ALLES_ROOD_2
76.0 s  AUTO_GROEN
```

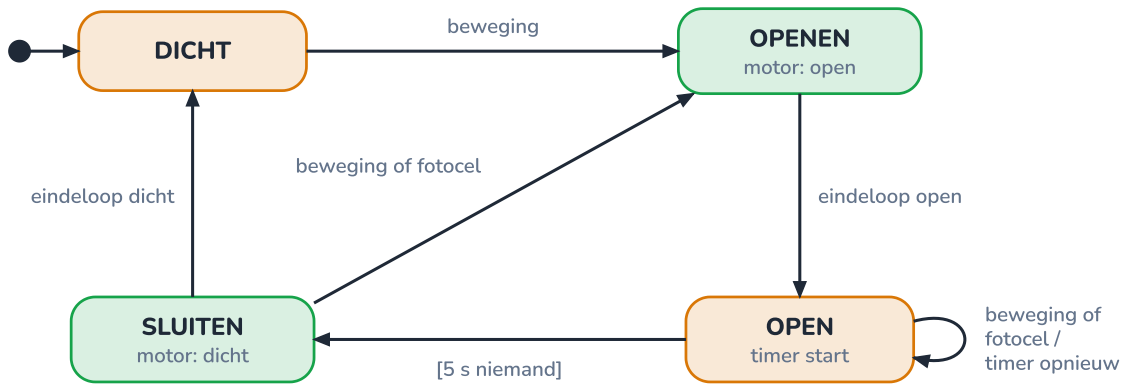
De druk op 29 s valt in VOETGANGER_GROEN en wordt genegeerd. De druk op 41 s komt 3 s nadat de auto's groen kregen: de aanvraag wacht tot de 20 s om zijn (58 s). De knop hoeft hier niet ontdenderd te worden: een tweede of derde 'druk' van dender zet `aanvraag` opnieuw op ja, en dat verandert niets.

UITGEWERKT VOORBEELD · 32.5

De automatische deur: van diagram naar tabel

5/17

De ingang van het station heeft een automatische schuifdeur. Een radar ziet beweging, twee eindeloopschakelaars melden 'open' en 'dicht', een fotocel tussen de deurstijlen ziet of er iemand in de opening staat. De deur blijft open zolang er beweging is, en sluit 5 s nadat niemand meer beweegt. Stel de toestandstabel op en zoek de overgangen die over veiligheid gaan.



De pijl van SLUITEN naar OPENEN is de veiligheidsovergang.

GEGEVEN

- gebeurtenissen: beweging, eindeloop open, eindeloop dicht, 5 s niemand, fotocel onderbroken

GEVRAAGD

- toestanden
- toestandstabel
- veiligheid

OPLOSSING

1. Toestanden waarin de deur anders doet: DICHT, OPENEN, OPEN (timer loopt), SLUITEN.
2. Vul voor elke toestand elke kolom in. In DICHT doet alleen beweging iets; in OPENEN alleen 'eindeloop open'.
3. In OPEN start 'beweging' de timer opnieuw: een zelflus. Ook de fotocel: wie stilstaat in de opening, beweegt misschien niet voor de radar.
4. Veiligheid: in SLUITEN zijn 'beweging' en 'fotocel' de belangrijkste vakjes. De deur keert om naar OPENEN: niemand raakt gekneld.
5. Lege vakjes nakijken. DICHT + fotocel: negeren kan. Maar OPENEN + 'eindeloop dicht' kan niet gebeuren als alles werkt: dat is een **onmogelijke combinatie**, en dus een sensorfout. Een goed ontwerp gaat dan naar een toestand FOUT.

ANTWOORD

TOESTAND	BEWEGING	EINDELOOP OPEN	EINDELOOP DICHT	5 S NIEMAND	FOTOCEL ONDERBROKEN
DICHT	OPENEN / motor open	–	–	–	–
OPENEN	–	OPEN / motor stil, timer	–	–	–
OPEN	OPEN / timer opnieuw	–	–	SLUITEN / motor dicht	OPEN / timer opnieuw
SLUITEN	OPENEN / motor open	–	DICHT / motor stil	–	OPENEN / motor open

Klopt dit? De tabel heeft 4 × 5 = 20 vakjes, waarvan 8 ingevuld; het diagram toont ze met 5 pijlen (twee met twee gebeurtenissen) en een zelflus. Wie alleen tekent, vergeet makkelijk de lege vakjes; de tabel laat ze zien. Opdracht 32.12 maakt van de tabel een programma.

UITLEG · 32.6

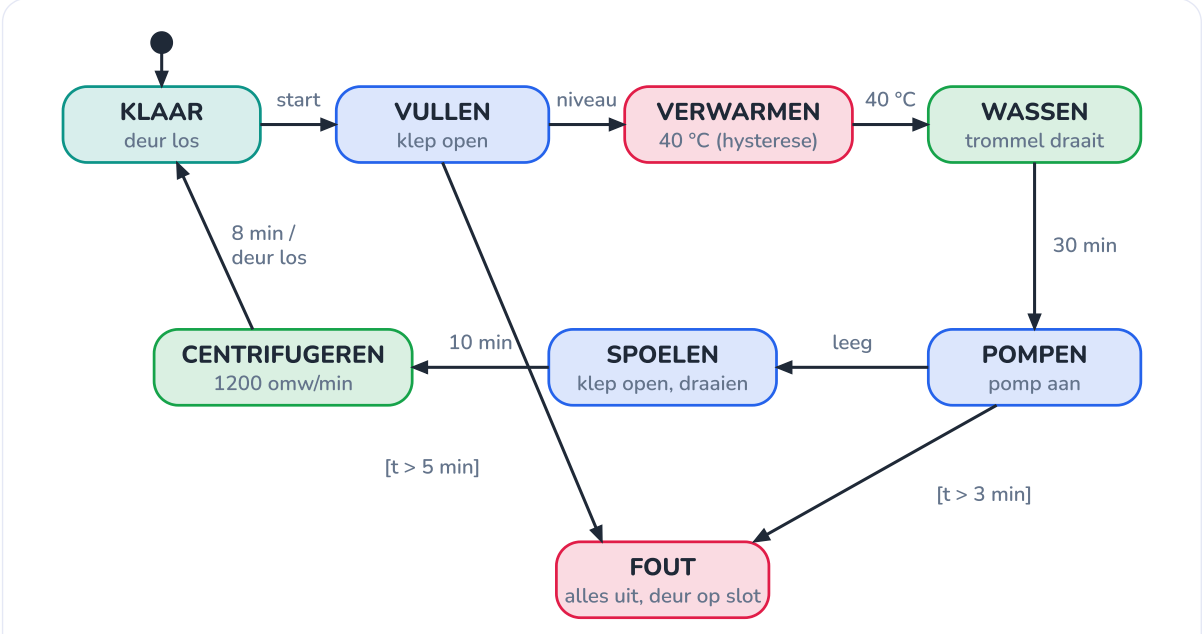
De wasmachine: een programma in stappen

6/17

Een wasmachine doorloopt een vaste reeks toestanden. Sommige overgangen komen van een sensor (het niveau van het water, de temperatuur), andere van de tijd. Drie dingen maken dit voorbeeld leerrijk:

- **Een regeling binnen een toestand.** In VERWARMEN warmt het water op tot 40 °C, en tijdens het wassen houdt een aan/uit-regeling met hysterese het daar (hoofdstuk 31); in VULLEN staat de klep open tot de niveauschakelaar schakelt. Toestandsmachines en regelingen werken samen: de machine kiest *wat* er geregeld wordt, de regeling doet het.
- **Time-outs vangen fouten op.** Staat de kraan dicht, dan komt 'niveau bereikt' nooit. Zonder time-out wacht de machine eeuwig met een open klep. Met [t > 5 min] gaat ze naar FOUT. Elke toestand die op een sensor wacht, hoort een time-out te hebben.
- **De veilige toestand is niet 'alles uit en alles open'.** In FOUT staan klep, verwarming en trommel stil, maar de deur blijft op slot zolang er water in de trommel staat: anders loopt de badkamer onder.

Echte wasmachines spoelen twee of drie keer. Dat teken je niet met drie keer dezelfde toestanden, maar met een teller en een voorwaarde: uit POMPEN naar SPOELEN [spoelbeurt < 2], anders naar CENTRIFUGEREN.



Rechte lijn door het programma, en twee time-outs die naar FOUT leiden.

CODE · 32.7

Een toestandsmachine als tabel

7/17

Een toestandsmachine hoeft geen `switch` te zijn. In Python kan je de toestandstabel letterlijk als woordenboek schrijven (hoofdstuk 28): de sleutel is het paar (toestand, gebeurtenis), de waarde het paar (nieuwe toestand, actie). Het programma zoekt alleen nog op.

Python **wasmachine.py** 29 regels

```python
# wasmachine.py - een toestandsmachine als tabel: (toestand, gebeurtenis) -> (nieuwe toestand, actie)
OVERGANGEN = {
    ("KLAAR", "start"): ("VULLEN", "deur vergrendelen, waterklep open"),
    ("VULLEN", "niveau bereikt"): ("VERWARMEN", "waterklep dicht, verwarming aan"),
    ("VULLEN", "5 min voorbij"): ("FOUT", "waterklep dicht: komt er geen water?"),
    ("VERWARMEN", "40 graden"): ("WASSEN", "verwarming uit, trommel draait"),
    ("WASSEN", "30 min voorbij"): ("POMPEN", "trommel stil, pomp aan"),
    ("POMPEN", "leeg"): ("SPOELEN", "pomp uit, waterklep open, trommel draait"),
    ("POMPEN", "3 min voorbij"): ("FOUT", "pomp uit: filter verstopt?"),
    ("SPOELEN", "10 min voorbij"): ("CENTRIFUGEREN", "waterklep dicht, pomp aan, 1200 omw/min"),
    ("CENTRIFUGEREN", "8 min voorbij"): ("KLAAR", "trommel stil, pomp uit, deur ontgrendelen"),
}


def verwerk(toestand, gebeurtenis):
    """Zoekt de overgang op. Staat de combinatie niet in de tabel, dan gebeurt er niets."""
    if (toestand, gebeurtenis) in OVERGANGEN:
        nieuw, actie = OVERGANGEN[(toestand, gebeurtenis)]
        print(f"{toestand:13} --{gebeurtenis}--> {nieuw:13} {actie}")
        return nieuw
    print(f"{toestand:13} --{gebeurtenis}--> (genegeerd)")
    return toestand


toestand = "KLAAR"
for g in ["deur open", "start", "niveau bereikt", "start", "40 graden", "deur open", "30 min voorbij",
          "3 min voorbij"]:
    toestand = verwerk(toestand, g)
print("eindtoestand:", toestand)
```

Uitvoer:

```
KLAAR         --deur open--> (genegeerd)
KLAAR         --start--> VULLEN        deur vergrendelen, waterklep open
VULLEN        --niveau bereikt--> VERWARMEN     waterklep dicht, verwarming aan
VERWARMEN     --start--> (genegeerd)
VERWARMEN     --40 graden--> WASSEN        verwarming uit, trommel draait
WASSEN        --deur open--> (genegeerd)
WASSEN        --30 min voorbij--> POMPEN        trommel stil, pomp aan
POMPEN        --3 min voorbij--> FOUT          pomp uit: filter verstopt?
eindtoestand: FOUT
```

Combinaties die niet in de tabel staan, worden genegeerd: 'deur open' tijdens het wassen, een tweede druk op start. Het voordeel van een tabel: de machine is *data*, die je kan nakijken, afdrukken of uit een bestand laden. Het nadeel: voorwaarden zoals [spoelbeurt < 2] en acties bij binnenkomen passen er minder vlot in. Op een microcontroller kiest men daarom meestal de `switch`.

UITLEG · 32.8

Van diagram naar code: het patroon

8/17

Elke toestandsmachine in dit hoofdstuk volgt hetzelfde patroon:

```
enum Toestand { A, B, C };           // 1. de toestanden krijgen een naam
Toestand toestand = A;               // 2. één variabele onthoudt waar we zijn
unsigned long sinds = 0;             //    en sinds wanneer

void naar(Toestand nieuw, unsigned long nu) {
  toestand = nieuw;                  // 3. een overgang: toestand en tijd bijwerken,
  sinds = nu;
  // ... acties bij binnenkomen (motor aan, lamp uit, melding)
}

void taak(unsigned long nu) {        // 4. elke ronde van loop(): alleen kijken,
nooit wachten
  unsigned long t = nu - sinds;      //    hoe lang al in deze toestand?
  switch (toestand) {
    case A:
      if (gebeurtenis1) naar(B, nu);
      break;                         // 5. break: anders loopt de code door in de
volgende case
    case B:
      if (t > TIMEOUT_MS) naar(C, nu);
      break;
    case C:
      break;
  }
}
```

in het diagram	in de code
toestand	een waarde van de `enum`
beginpunt	de beginwaarde van `toestand`, of `naar()` in `setup()`
overgang	`naar(...)` in de `case` van de vertrektoestand
gebeurtenis	een test: een pin, een vlag, `t > ...`
[voorwaarde]	een extra test met `&&`
actie bij binnenkomen	in `naar()`
zelflus met een actie	de actie, zonder `naar()`

Vier fouten komen vaak voor:

- **break vergeten.** Dan loopt de code door in de volgende `case`, en test ze gebeurtenissen van een toestand waarin je niet bent.
- **De toestand elke ronde opnieuw zetten.** Wie in de `case` schrijft 'motor aan' in plaats van dat bij de overgang te doen, voert de actie duizenden keren per seconde uit, en `sinds` schuift mee: een time-out loopt dan nooit af.
- **Wachten in een toestand.** Een `delay()` of een `while` die op een sensor wacht, legt alle andere taken stil. Een toestand wacht door *niets* te doen tot de gebeurtenis er is.
- **De toestand in meerdere variabelen.** `dak` en `dakOpen` samen geven combinaties die niet kunnen bestaan. Eén variabele, één waarheid.

CODE · 32.9

Het kasdak als toestandsmachine

Het dak krijgt zijn eigen bestand, zoals `sensoren.cpp` in hoofdstuk 29. Het headerbestand zegt wat de rest van het programma mag vragen:

```
// dak.h - het kasdak als toestandsmachine (hoofdstuk 32)
#ifndef DAK_H
#define DAK_H

enum DakToestand { DICHT, OPENEN, OPEN, SLUITEN, GESTOPT, DAK_FOUT };

extern DakToestand dak;
void dakStart(unsigned long nu);                  // in setup(): waar staat het dak?
void dakVraag(bool open);                         // de klimaatregeling vraagt: open of dicht
void dakKnop(unsigned long nu);                   // korte druk op S1 (met de hand)
void dakHerstel(unsigned long nu);                // na een alarm: opnieuw proberen te sluiten
void dakBlokkeer(bool aan, unsigned long nu);     // onderhoud: de motor mag niet bewegen
void taakDak(unsigned long nu);
const char* dakNaam(DakToestand t);

#endif
```

De temperatuurregeling van hoofdstuk 31 stuurt de motor niet meer zelf aan: ze vraagt alleen `dakVraag(true)` als de lucht boven 27 °C komt en `dakVraag(false)` onder 25 °C. De hysterese maakt van een temperatuur een **gebeurtenis**; de toestandsmachine beslist of en hoe het dak beweegt. Zo zit elke verantwoordelijkheid op één plaats.

C++ (Arduino) **dak.cpp** 87 regels

```
// dak.cpp - het kasdak als toestandsmachine (hoofdstuk 32; motor en eindeloop: hoofdstuk 26)
#include <Arduino.h>
#include "pinnen.h"
#include "dak.h"

void melding(const char* tekst);                  // staat in de schets zelf (met de tijd van de klok)

const unsigned long TIMEOUT_MS = 12000;           // langer dan 12 s bewegen: er klopt iets niet
const int PWM_DAK = 232;                          // 0,10 m/s bij 12,8 V (hoofdstuk 26)
const char* DAK_NAAM[] = {"DICHT", "OPENEN", "OPEN", "SLUITEN", "GESTOPT", "DAK_FOUT"};

DakToestand dak = DICHT;
unsigned long dakSinds = 0;
bool wensOpen = false, geblokkeerd = false;

const char* dakNaam(DakToestand t) { return DAK_NAAM[t]; }

void motor(int richting) {                        // +1 sluiten, -1 openen, 0 stil: nooit beide tegelijk
  analogWrite(PWM_DICHT, richting > 0 ? PWM_DAK : 0);
  analogWrite(PWM_OPEN, richting < 0 ? PWM_DAK : 0);
}

void dakNaar(DakToestand nieuw, unsigned long nu) {
  dak = nieuw;
  dakSinds = nu;
  if (nieuw == OPENEN) motor(-1);                 // bij binnenkomen: de motor
  else if (nieuw == SLUITEN) motor(+1);
  else motor(0);
  char tekst[24] = "dak: ";
  strcat(tekst, DAK_NAAM[nieuw]);
  melding(tekst);
}
```

```
33
34  void dakStart(unsigned long nu) {
35    if (digitalRead(EIND_DICHT) == HIGH) dakNaar(DICHT, nu);
36    else if (digitalRead(EIND_OPEN) == HIGH) { wensOpen = true; dakNaar(OPEN, nu); }
37    else dakNaar(SLUITEN, nu);                      // onbekende stand: veilig dichtdoen
38  }
39
40  void dakVraag(bool open) { wensOpen = open; }
41
42  void dakKnop(unsigned long nu) {
43    if (geblokkeerd) return;
44    switch (dak) {
45      case DICHT:   wensOpen = true; break;
46      case OPEN:    wensOpen = false; break;
47      case OPENEN:
48      case SLUITEN: dakNaar(GESTOPT, nu); break;    // een druk tijdens de beweging = stoppen
49      case GESTOPT: wensOpen = false; dakNaar(SLUITEN, nu); break;
50      case DAK_FOUT: break;                         // alleen dakHerstel() haalt het dak hieruit
51    }
52  }
53
54  void dakHerstel(unsigned long nu) {
55    if (dak == DAK_FOUT) {
56      wensOpen = false;
57      dakNaar(SLUITEN, nu);
58    }
59  }
60
61  void dakBlokkeer(bool aan, unsigned long nu) {
62    geblokkeerd = aan;
63    if (aan && (dak == OPENEN || dak == SLUITEN)) dakNaar(GESTOPT, nu);
64  }
65
66  void taakDak(unsigned long nu) {
67    unsigned long t = nu - dakSinds;
68    switch (dak) {
69      case DICHT:
70        if (wensOpen && !geblokkeerd) dakNaar(OPENEN, nu);
71        break;
72      case OPEN:
73        if (!wensOpen && !geblokkeerd) dakNaar(SLUITEN, nu);
74        break;
75      case OPENEN:
76        if (digitalRead(EIND_OPEN) == HIGH) dakNaar(OPEN, nu);
77        else if (t > TIMEOUT_MS) dakNaar(DAK_FOUT, nu);
78        break;
79      case SLUITEN:
80        if (digitalRead(EIND_DICHT) == HIGH) dakNaar(DICHT, nu);
81        else if (t > TIMEOUT_MS) dakNaar(DAK_FOUT, nu);
82        break;
83      case GESTOPT:                                 // wacht op een knopdruk
84      case DAK_FOUT:                                // wacht op dakHerstel()
85        break;
86    }
87  }
```

Let op `dakStart()`: na een stroomuitval weet het programma niet waar het dak staat. Het leest de eindeloopschakelaars; staat het dak ergens tussenin, dan sluit het. Hetzelfde principe had de meetkas (eis G5: na een stroomuitval herstarten met het raam dicht). In DAK_FOUT probeert het programma niet uit zichzelf opnieuw: een motor die tegen een blokkade blijft duwen, kan de riem of het paneel beschadigen. Een mens moet eerst kijken.

VERHAAL · 32.10

Een lift die niet valt

Rond 1850 waren liften in fabrieken en pakhuizen gevaarlijk: platformen aan een touw of een ketting, en als die brak, stortte alles naar beneden. Niemand wilde er zelf in. In 1852 bouwde de Amerikaanse monteur **Elisha Otis** in Yonkers een lift met een **veiligheidsrem**. Aan het platform zat een veer die door de spanning in het touw platgedrukt bleef. Brak het touw, dan viel die spanning weg, en duwde de veer grijpers in tanden langs de geleiders: het platform bleef hangen.

In mei 1854 toonde Otis zijn lift op de wereldtentoonstelling in het Crystal Palace van New York. Hij liet zich met het platform hoog optrekken en liet het touw doorsnijden. Het platform zakte een klein stukje en bleef hangen. Het publiek was overtuigd, en de veilige lift maakte later wolkenkrabbers mogelijk.

Het geniale zit niet in de rem, maar in *wanneer* ze werkt: de rem heeft niets nodig om in te grijpen. Geen stroom, geen bediening, geen werkend onderdeel. Het defect zelf (een gebroken touw) brengt de lift in de veilige toestand. Dat heet **fail-safe**.

Fail-safe in de klimaatmodule

De eindeloopschakelaars van het dak zijn verbreekcontacten met een pull-up: een losse draad leest als 'eindpunt bereikt', en de motor stopt. De vlotter op D2 leest bij een losse draad 'droog', en de pomp stopt. De MOSFET van de verwarming heeft een pull-down van 10 kΩ: als de Mega uitvalt, gaat de gate laag en de verwarming uit. Bij elke keuze vraag je: wat gebeurt er als deze draad breekt?

Een reconstructie: de lift bleef hangen toen het touw doorgesneden werd.

Foutsituaties en de veilige toestand

Wat kan er misgaan, en hoe ziet een programma dat?

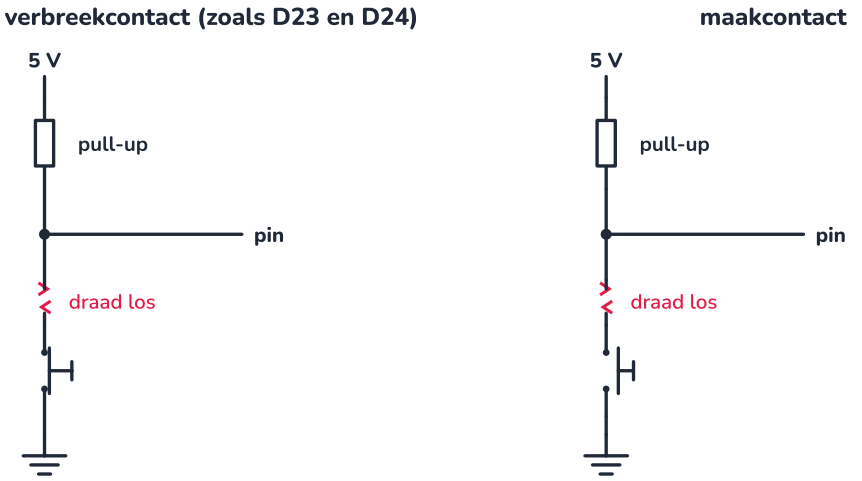
- Een sensor antwoordt niet: de SHT31 geeft `nan`. Na 15 mislukte metingen (30 s) is het geen toeval meer.
- Een sensor geeft een **onmogelijke waarde** (−40 °C in een verwarmde module) of een **onmogelijke combinatie** (beide eindeloopschakelaars tegelijk ingedrukt).
- Een actuator doet niet wat hij moet: het dak haalt zijn eindpunt niet binnen 12 s. Een **time-out** ziet het.
- Het programma zelf hangt: de **waakhond** ziet het (volgend blad).
- De stroom valt weg: bij het herstarten begint alles in OPSTART, met alle uitgangen uit.

Dan de vraag: wat is **veilig**? Niet altijd 'alles uit'. Een verwarming uitzetten is veilig in juli, maar laat de sla bevriezen in januari. Het dak dichtdoen is veilig bij storm, maar laat de module koken op een zonnige middag. De veilige reactie hangt af van de fout en van de omstandigheden, en ze doet alleen iets aan wat de fout raakt: een droge vlotter stopt de pomp, niet de verwarming.

fout	hoe de Mega ze ziet	veilige reactie	weg als
temperatuursensor weg	15 metingen na elkaar `nan` (30 s)	verwarming blind op het gemiddelde; dak blijft staan	er weer metingen zijn
vlotter droog	D2 hoog, 10 s lang (ook bij een losse draad)	pomp uit	de vlotter weer drijft
reservoir onder 10 cm	MPX5010 op A5	pomp uit	het niveau boven 12 cm is
dak klemt	eindeloop niet gehaald binnen 12 s	motor stil, geen nieuwe poging	iemand bevestigt: één nieuwe poging

Veilig betekent alles uitzetten

Een lift die bij een fout de rem loslaat om 'alles uit' te zetten, valt. Een schuifdeur op een vluchtweg moet bij een stroomuitval juist open. Kies per fout de toestand met de minste schade, en laat de rest van het systeem zo goed mogelijk verder werken.



Normaal: contact dicht, pin LOW.
Eindpunt: contact open, pin HIGH.
Draad los: ook HIGH, de motor stopt.

Normaal: contact open, pin HIGH.
Eindpunt: contact dicht, pin LOW.
Draad los: nooit LOW, de motor loopt door.

Dezelfde losse draad, twee soorten contact: alleen het verbreekcontact is fail-safe.

CODE · 32.12

De waakhond

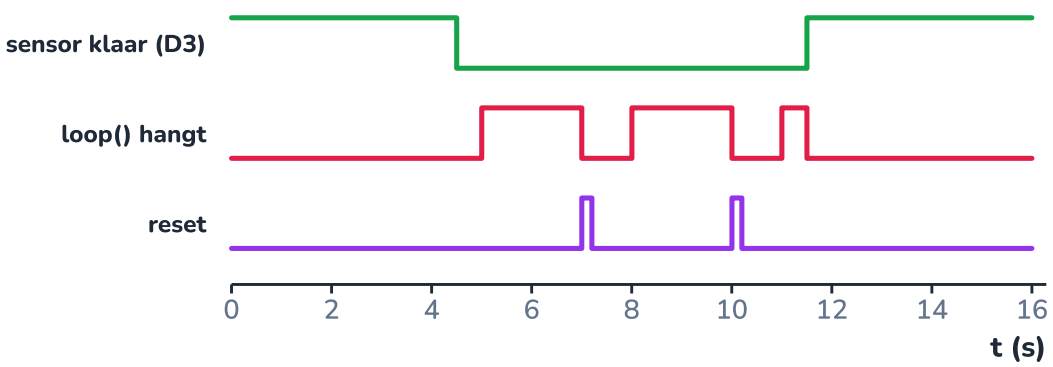
12/17

Een programma kan blijven hangen: een `while` die wacht op een sensor die nooit antwoordt, een oneindige lus door een fout. Dan draait niets meer, ook de veiligheid niet. De ATmega heeft daarvoor een **waakhond** (watchdog): een aparte timer met een eigen oscillator. Het programma moet hem regelmatig geruststellen met `wdt_reset()`. Blijft dat uit, bijvoorbeeld 2 s lang, dan herstart de waakhond de microcontroller: `setup()` begint opnieuw, alle uitgangen gaan in hun veilige beginstand.

C++ (Arduino) **waakhond.ino** 31 regels

```cpp
// waakhond.ino - de waakhond (watchdog) herstart de Mega als loop() blijft hangen (hoofdstuk 32)
#include <avr/wdt.h>

const int KLAAR = 3;                 // 'meting klaar'-lijn van een sensor: HIGH = er is een meting
unsigned long vorige = 0;
int metingen = 0;

void setup() {
  wdt_disable();                     // eerst uit: misschien zijn we net door de waakhond herstart
  Serial.begin(115200);
  pinMode(KLAAR, INPUT);
  vorige = 0;
  metingen = 0;
  Serial.println("opgestart");
  wdt_enable(WDTO_2S);               // komt loop() 2 s niet meer langs wdt_reset(): herstart
}

void loop() {
  wdt_reset();                       // 'ik leef nog', elke ronde
  unsigned long nu = millis();
  if (nu - vorige >= 1000) {
    vorige += 1000;
    while (digitalRead(KLAAR) == LOW) {
      // wachten op de sensor, zonder uitweg: hangt de sensor, dan hangt alles
    }
    metingen++;
    Serial.print(nu / 1000.0, 1);
    Serial.print(" s  meting ");
    Serial.println(metingen);
  }
```

```
31  }
```



Hangt loop() langer dan 2 s, dan volgt een herstart.

In de test hangt de sensor van 4,5 tot 11,5 s. De seriële monitor (de regels met --- komen van de simulatie):

```
opgestart
1.0 s  meting 1
2.0 s  meting 2
3.0 s  meting 3
4.0 s  meting 4
--- 7.0 s: de waakhond herstart de Mega ---
opgestart
--- 10.0 s: de waakhond herstart de Mega ---
opgestart
1.0 s  meting 1
2.0 s  meting 2
3.0 s  meting 3
4.0 s  meting 4
5.0 s  meting 5
6.0 s  meting 6
```

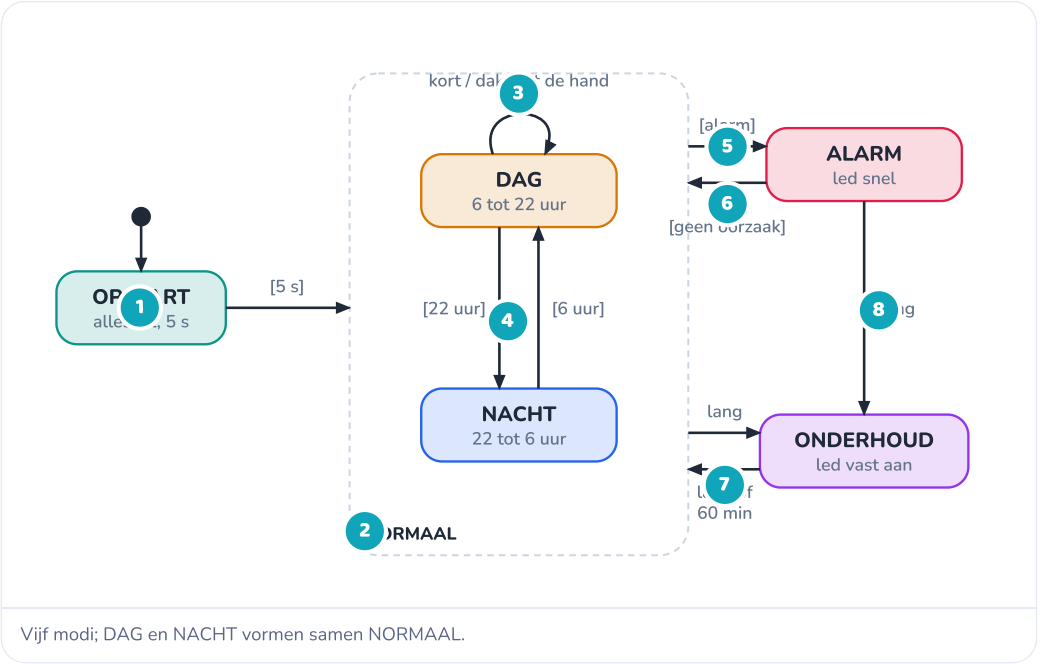
Na een herstart begint `millis()` opnieuw bij 0. De waakhond lost de oorzaak niet op: zolang de sensor hangt, herstart de Mega om de 2 à 3 s. Hij beperkt wel de schade: het programma komt telkens in een veilige begintoestand. De echte oplossing is een wachtlus met een uitweg (een time-out). Zet `wdt_reset()` nooit in zo'n wachtlus: dan kan de waakhond niets meer zien. Test de waakhond op je eigen bord voor je erop vertrouwt.

ONTLEED · 32.13

De modi van de klimaatmodule

13/17

De klimaatmodule doet niet altijd hetzelfde. Overdag branden de lampen en volgt het dak de temperatuur, 's nachts niet. Bij een fout moet ze voorzichtig zijn, en als iemand binnen werkt, mag er niets onverwacht bewegen. Die **modi** vormen een toestandsmachine. DAG en NACHT reageren op de meeste gebeurtenissen hetzelfde: ze zitten samen in een **samengestelde toestand** NORMAAL. Een pijl vanaf de rand van NORMAAL geldt voor DAG en NACHT allebei.



Vijf modi; DAG en NACHT vormen samen NORMAAL.

1 OPSTART
Na het aanzetten, na een stroomuitval of na de waakhond: alle uitgangen uit, 5 s om de sensoren en de klok te lezen, de stand van het dak zoeken.

2 Samengestelde toestand
NORMAAL bevat DAG en NACHT. Na OPSTART kiest de klok welke van de twee.

3 Een korte druk
In DAG en NACHT opent of sluit een korte druk op S1 het dak met de hand, zoals in hoofdstuk 29.

4 De klok
Om 22 uur naar NACHT (lampen uit, dak dicht), om 6 uur naar DAG. De klok komt van de DS3231 (hoofdstuk 30).

5 Naar ALARM
Zodra er een oorzaak is: sensor weg, vlotter droog, reservoir te laag, dak klemt. Vanuit DAG en NACHT.

6 Bevestigen
Een korte druk, alleen als de oorzaak weg is. Een alarm gaat niet vanzelf uit: iemand moet het gezien hebben.

7 Onderhoud
Een lange druk (3 s) zet ONDERHOUD aan en weer uit. Vergeten? Na 60 minuten keert de module vanzelf terug.

8 Van ALARM naar ONDERHOUD
Wie een fout komt herstellen, zet onderhoud aan: dan beweegt het dak niet terwijl hij binnen werkt.

UITLEG · 32.14

Wat elke modus doet

14/17

Een toestandsmachine zegt wanneer de module van modus verandert; de **actietabel** zegt wat elke modus met de uitgangen doet. Die tabel is het echte ontwerp:

modus	groeilampen	pomp	verwarming	ventilatoren	kasdak	CO_2-klep	statusled
OPSTART	uit	uit	uit	zacht	eindeloop lezen	dicht	uit
DAG	aan	min. 0-9	18/19 °C	vol boven 25 °C	volgt 27/25 °C	30 s per halfuur	500 ms
NACHT	uit	min. 0-9	16/17 °C	zacht	dicht	dicht	500 ms
ALARM	zoals dag of nacht	uit bij een wateralarm	blind bij een sensorfout	zoals dag of nacht	stil bij een dakfout	dicht	100 ms

modus	groeilampen	pomp	verwarming	ventilatoren	kasdak	CO_2-klep	statusled
ONDERHOUD	werklicht	uit	zoals dag of nacht	vol	geblokkeerd	dicht	vast aan

Drie keuzes vallen op. In ONDERHOUD is het dak **geblokkeerd**, gaat de CO_2-klep dicht (wie binnen werkt, mag geen extra CO_2 inademen, hoofdstuk 14) en staat de pomp stil (handen in het reservoir). In ALARM werkt de module **zo gewoon mogelijk verder**: alleen wat de fout raakt, stopt. En de alarmen zijn **bits**: er kunnen er meerdere tegelijk zijn, en de module weet welke.

> **Eén variabele per onafhankelijk aspect**
>
> Dag of nacht hangt van de klok af; alarm of niet van de sensoren. Wie ook nog 'batterij bijna leeg' als modus wil toevoegen, krijgt DAG_ZUINIG, NACHT_ZUINIG, … : het aantal modi verdubbelt. Dan is een tweede, kleine toestandsmachine voor het energiebeheer beter (opdracht 32.13).

CODE · 32.15

klimaat.ino versie 4: de modi in C++

15/17

De modi komen bovenop de taken van hoofdstuk 29 tot 31. Om het overzichtelijk te houden, staan hier alleen de modi, de alarmen, de verwarming, de ventilatoren, de pomp, de CO_2-klep en het dak (uit `dak.cpp`); de logger van hoofdstuk 30 en het dimmen van de lampen van hoofdstuk 25 blijven zoals ze waren. De knop onderscheidt een korte en een lange druk.

C++ (Arduino) **modi.ino** 212 regels

```
// modi.ino - de modi van de klimaatmodule als toestandsmachine (hoofdstuk 32; klimaat.ino versie 4)
// OPSTART -> DAG of NACHT volgens de klok; ALARM bij een fout; ONDERHOUD met een lange druk op S1 (3 s).
#include <Wire.h>
#include <RTClib.h>
#include <Adafruit_SHT31.h>
#include "pinnen.h"
#include "dak.h"

enum Modus { OPSTART, DAG, NACHT, ALARM, ONDERHOUD };
const char* NAAM[] = {"OPSTART", "DAG", "NACHT", "ALARM", "ONDERHOUD"};
const byte A_SENSOR = 1, A_WATER = 2, A_NIVEAU = 4, A_DAK = 8;     // oorzaken van een alarm, als bits

const unsigned long MEET_MS = 2000;
const unsigned long LANG_MS = 3000;                 // zo lang indrukken = onderhoud aan of uit
const unsigned long ONDERHOUD_MAX_MS = 3600000UL;   // vergeten uit te zetten: na een uur terug
const int WERKLICHT = 60, LAMPEN_DAG = 200;         // PWM van de groeilampen
const int VENT_LAAG = 90, VENT_VOL = 255;

Adafruit_SHT31 sht;
RTC_DS3231 rtc;
DateTime klok;
Modus modus = OPSTART;
unsigned long modusSinds = 0, vorigeMeting = 0, knopSinds = 0, droogSinds = 0, ledSinds = 0;
byte alarmen = 0;                                   // welke alarmen er nu zijn
float luchtT = NAN, gemAan = 0.5;
int geenMeting = 0;
bool verwarmen = false, koelen = false, warm = false, knopWas = false, langGemeld = false, niveauLaag = false;
bool ledAan = false;

// ------------------------------------------------------------------ meldingen
void printTwee(int x) {
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void melding(const char* tekst) {
  DateTime k = rtc.now();
```

```
  Serial.print("# ");
  printTwee(k.hour());
  Serial.print(':');
  printTwee(k.minute());
  Serial.print(':');
  printTwee(k.second());
  Serial.print(' ');
  Serial.println(tekst);
}

// ---------------------------------------------------------------- sensoren en alarmen
float niveauCm() {                              // MPX5010 onderaan het reservoir (hoofdstuk 11 en 29)
  float u = analogRead(DRUK) * 5.0 / 1024.0;
  float p = (u / 5.0 - 0.04) / 0.09 * 1000.0;
  return p / (1000.0 * 9.81) * 100.0;
}

void meten(unsigned long nu) {
  float t, rv;
  if (sht.readBoth(&t, &rv)) { luchtT = t; geenMeting = 0; }
  else { luchtT = NAN; geenMeting++; }
  bool droog = digitalRead(VLOTTER) == HIGH;    // verbreekt ook als de draad los is
  if (!droog) droogSinds = nu;
  float h = niveauCm();
  if (!niveauLaag && h < 10.0) niveauLaag = true;   // hysterese: 10 en 12 cm
  else if (niveauLaag && h > 12.0) niveauLaag = false;
  byte nieuw = 0;
  if (geenMeting >= 15) nieuw |= A_SENSOR;      // 30 s geen temperatuur
  if (nu - droogSinds >= 10000) nieuw |= A_WATER;   // 10 s droog: geen dender van een golfje
  if (niveauLaag) nieuw |= A_NIVEAU;
  if (dak == DAK_FOUT) nieuw |= A_DAK;
  if (nieuw & ~alarmen & A_SENSOR) melding("oorzaak: geen temperatuur");
  if (nieuw & ~alarmen & A_WATER) melding("oorzaak: vlotter droog");
  if (nieuw & ~alarmen & A_NIVEAU) melding("oorzaak: reservoir onder 10 cm");
  if (nieuw & ~alarmen & A_DAK) melding("oorzaak: dak haalt zijn eindpunt niet");
  alarmen = nieuw;
}

// ---------------------------------------------------------------- de knop: kort of lang
void leesKnop(unsigned long nu, bool& kort, bool& lang) {
  bool in = digitalRead(KNOP) == HIGH;          // na de 74HC14: hoog = ingedrukt (hoofdstuk 24)
  kort = lang = false;
  if (in && !knopWas) { knopSinds = nu; langGemeld = false; }
  if (in && !langGemeld && nu - knopSinds >= LANG_MS) { lang = true; langGemeld = true; }
  if (!in && knopWas && !langGemeld) kort = true;
  knopWas = in;
}

// ---------------------------------------------------------------- de modi
bool isDag() { return klok.hour() >= 6 && klok.hour() < 22; }
Modus gewoon() { return isDag() ? DAG : NACHT; }

void naar(Modus nieuw, unsigned long nu) {
  Modus oud = modus;
  modus = nieuw;
  modusSinds = nu;
  char tekst[32] = "modus ";
  strcat(tekst, NAAM[nieuw]);
  melding(tekst);
  // bij binnenkomen en bij verlaten
  if (nieuw == ONDERHOUD) dakBlokkeer(true, nu);    // niemand onder een bewegend dak
  if (oud == ONDERHOUD) dakBlokkeer(false, nu);
  if (nieuw == NACHT) dakVraag(false);
  if (nieuw == DAG) dakVraag(warm);                 // overdag volgt het dak de temperatuur
}

void taakModus(unsigned long nu, bool kort, bool lang) {
  switch (modus) {
```

```
    case OPSTART:                                  // 5 s: sensoren en klok lezen
      if (nu - modusSinds >= 5000) naar(gewoon(), nu);   // een alarm volgt dan meteen vanuit DAG of NACHT
      break;
    case DAG:
    case NACHT:
      if (alarmen) naar(ALARM, nu);
      else if (lang) naar(ONDERHOUD, nu);
      else if (gewoon() != modus) naar(gewoon(), nu);
      else if (kort) dakKnop(nu);                  // korte druk: het dak met de hand
      break;
    case ALARM:
      if (kort) {                                  // bevestigen
        if (alarmen & A_DAK) {                     // het dak nog eens laten sluiten
          dakHerstel(nu);
          alarmen &= ~A_DAK;                       // het dak is niet meer in DAK_FOUT
        }
        if (alarmen == 0) naar(gewoon(), nu);
        else melding("alarm blijft: oorzaak nog aanwezig");
      } else if (lang) naar(ONDERHOUD, nu);
      break;
    case ONDERHOUD:
      if (lang || nu - modusSinds >= ONDERHOUD_MAX_MS) naar(gewoon(), nu);
      break;
  }
}

// ------------------------------------------------------------------ wat elke modus doet
void acties(unsigned long nu) {
  bool dag = isDag();
  // verwarming en ventilatoren met hysterese (hoofdstuk 31); zonder meting blind
  float laag = dag ? 18.0 : 16.0, hoog = laag + 1.0;
  if (isnan(luchtT)) {
    verwarmen = (nu / 1000UL) % 600UL < (unsigned long)(gemAan * 600);
    koelen = false;
  } else {
    if (!verwarmen && luchtT < laag) verwarmen = true;
    else if (verwarmen && luchtT > hoog) verwarmen = false;
    if (!koelen && luchtT > 25.0) koelen = true;
    else if (koelen && luchtT < 24.0) koelen = false;
    if (verwarmen) koelen = false;
    gemAan += ((verwarmen ? 1.0 : 0.0) - gemAan) * (MEET_MS / 3600000.0);
    // het dak: hysterese 25/27 geeft gebeurtenissen; overdag in DAG en ALARM (niet bij een dakfout)
    bool automatisch = dag && (modus == DAG || (modus == ALARM && !(alarmen & A_DAK)));
    if (!warm && luchtT > 27.0) { warm = true; if (automatisch) dakVraag(true); }
    else if (warm && luchtT < 25.0) { warm = false; if (automatisch) dakVraag(false); }
  }
  if (modus == ONDERHOUD) koelen = true;           // verse lucht voor wie binnen werkt
  digitalWrite(VERWARMING, (verwarmen && modus != OPSTART) ? HIGH : LOW);
  analogWrite(VENTILATOREN, koelen ? VENT_VOL : VENT_LAAG);
  // groeilampen: overdag aan (het dimmen van hoofdstuk 25 laten we hier weg), werklicht bij onderhoud
  int lamp = 0;
  if (modus == ONDERHOUD) lamp = WERKLICHT;
  else if (dag && modus != OPSTART) lamp = LAMPEN_DAG;
  analogWrite(LAMPEN, lamp);
  // pomp: minuut 0 tot 9 van elk uur, nooit zonder water, niet bij onderhoud
  bool pomp = (modus == DAG || modus == NACHT || modus == ALARM) && klok.minute() < 10
              && !(alarmen & (A_WATER | A_NIVEAU));
  digitalWrite(POMP, pomp ? HIGH : LOW);
  // CO2: alleen in DAG, met het dak dicht en zonder koelen: om het halfuur 30 s
  bool co2 = modus == DAG && dak == DICHT && !koelen && klok.minute() % 30 == 0 && klok.second() < 30;
  digitalWrite(CO2_KLEP, co2 ? HIGH : LOW);
}

void taakStatusled(unsigned long nu) {             // DAG en NACHT: rustig, ALARM: snel, ONDERHOUD: vast aan
  if (modus == ONDERHOUD) { digitalWrite(STATUSLED, HIGH); return; }
  unsigned long periode = (modus == ALARM) ? 100 : 500;
  if (nu - ledSinds < periode) return;
  ledSinds = nu;
```

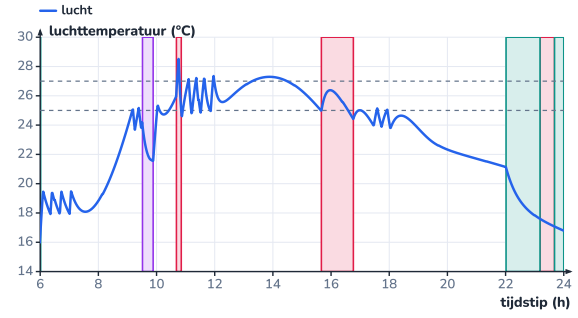
```
  ledAan = !ledAan;
  digitalWrite(STATUSLED, (ledAan && modus != OPSTART) ? HIGH : LOW);
}

// ------------------------------------------------------------- setup en loop
void setup() {
  const int UIT[] = {LAMPEN, PWM_DICHT, PWM_OPEN, DAK_ENABLE, POMP, CO2_KLEP, VENTILATOREN, VERWARMING, STATUSLED};
  for (int p : UIT) { pinMode(p, OUTPUT); digitalWrite(p, LOW); }   // veilig starten: alles uit
  digitalWrite(DAK_ENABLE, HIGH);
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(KNOP, INPUT);
  pinMode(EIND_OPEN, INPUT_PULLUP);
  pinMode(EIND_DICHT, INPUT_PULLUP);
  Serial.begin(115200);
  Wire.begin();
  rtc.begin();
  sht.begin(0x44);
  klok = rtc.now();
  unsigned long nu = millis();
  droogSinds = nu;
  melding("klimaatmodule v4 (hoofdstuk 32)");
  naar(OPSTART, nu);
  dakStart(nu);
}

void loop() {
  unsigned long nu = millis();
  bool kort, lang;
  leesKnop(nu, kort, lang);
  if (nu - vorigeMeting >= MEET_MS) {
    vorigeMeting += MEET_MS;
    klok = rtc.now();
    meten(nu);
    acties(nu);
  }
  taakModus(nu, kort, lang);
  taakDak(nu);
  taakStatusled(nu);
}
```

Getest op een zonnige aprildag, van 5:50 tot middernacht, met het warmtemodel van hoofdstuk 31 en een gesimuleerd dak. Het scenario: onderhoud van 9:30 tot 9:52, het stekkertje van de SHT31 los van 10:40 tot 10:45, een dak dat klemt van 15:30 tot 16:40, en een droge vlotter van 23:10 tot 23:35. De seriële monitor:

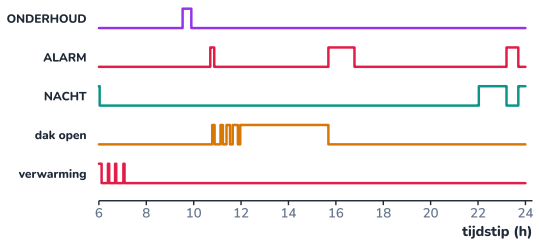
```
# 05:50:00 klimaatmodule v4 (hoofdstuk 32)
# 05:50:00 modus OPSTART
# 05:50:00 dak: DICHT
# 05:50:05 modus NACHT
# 06:00:00 modus DAG
# 09:30:03 modus ONDERHOUD
# 09:52:03 modus DAG
# 10:40:28 oorzaak: geen temperatuur
# 10:40:28 modus ALARM
# 10:45:00 dak: OPENEN
# 10:45:07 dak: OPEN
# 10:50:00 modus DAG
# 10:51:38 dak: SLUITEN
# 10:51:45 dak: DICHT
# 11:06:10 dak: OPENEN
# 11:06:17 dak: OPEN
# 11:12:24 dak: SLUITEN
```

```
# 11:12:31 dak: DICHT
# 11:21:52 dak: OPENEN
# 11:21:59 dak: OPEN
# 11:30:14 dak: SLUITEN
# 11:30:21 dak: DICHT
# 11:37:30 dak: OPENEN
# 11:37:37 dak: OPEN
# 11:50:54 dak: SLUITEN
# 11:51:01 dak: DICHT
# 11:56:44 dak: OPENEN
# 11:56:51 dak: OPEN
# 15:39:34 dak: SLUITEN
# 15:39:46 dak: DAK_FOUT
# 15:39:48 oorzaak: dak haalt zijn eindpunt niet
# 15:39:48 modus ALARM
# 16:45:00 dak: SLUITEN
# 16:45:00 modus DAG
# 16:45:03 dak: DICHT
# 22:00:00 modus NACHT
# 23:10:08 oorzaak: vlotter droog
# 23:10:08 modus ALARM
# 23:40:00 modus NACHT
```

De test vond een fout die in een eerdere versie zat. Tijdens het sensoralarm steeg de lucht boven 27 °C. Die gebeurtenis ging verloren, en na het bevestigen bleef het dak dicht tot 35 °C. De oplossing staat in `naar()`: bij het binnenkomen van DAG vraagt het programma het dak opnieuw wat de temperatuur vraagt (`dakVraag(warm)`), en in ALARM volgt het dak de temperatuur gewoon verder. Een gebeurtenis die komt terwijl niemand luistert, is weg: zet wat telt ook in de actie bij binnenkomen.



De luchttemperatuur met de modi: rood ALARM, paars ONDERHOUD, blauwgroen NACHT.



Rond 11 uur pendelt het dak: vier keer open in een uur. Opdracht 32.14 zoekt een oplossing.

NASLAG · 32.16

Notatie, patroon en controlelijst

16/17

tijd in de huidige toestand (ms)

$$t = \text{nu} - \text{sinds}$$

symbool	betekenis	eenheid
nu	millis() in deze ronde	ms
$sinds$	millis() bij het binnenkomen	ms

teken	betekenis
zwart bolletje met pijl	begin
afgeronde rechthoek	toestand, eventueel met 'bij binnenkomen: actie'
pijl met gebeurtenis [voorwaarde] / actie	overgang
pijl naar dezelfde toestand	zelflus: actie zonder van toestand te veranderen
rechthoek rond toestanden	samengestelde toestand: een pijl vanaf de rand geldt voor alle binnenste

controlelijst voor een toestandsmachine
Is er precies één begintoestand, en is ze veilig (alles uit, stand van actuatoren lezen)?
Is elk vakje van de toestandstabel bewust ingevuld, ook de lege?
Heeft elke toestand die op een sensor wacht, een time-out?
Zijn onmogelijke combinaties een fout in plaats van 'niets'?
Is voor elke fout de veilige toestand gekozen, per seizoen of omstandigheid?
Staat de toestand in één variabele, en gebeuren acties bij de overgang, niet elke ronde?
Wacht er nergens een `delay()` of `while` zonder uitweg?
Is elke overgang minstens één keer getest met een scenario?

HET GEHEEL · 32.17

Waar toestandsmachines vandaan komen en waar ze heen gaan

17/17

Toestandsmachines zitten overal waar iets in stappen gebeurt of op gebeurtenissen reageert: verkeerslichten, liften, wasmachines, drankautomaten, protocollen tussen computers, spelfiguren in een game. In de informatica heten ze **eindige automaten**; in Expert bouw je ze ook met poorten en flipflops (hoofdstuk 23), en worden ze bewezen veilig ontworpen met een FMEA (hoofdstuk 46).

F23 Programmeren met blokken
Gebeurtenissen en 'wanneer'-blokken: de eerste stap naar toestanden.

F24 De microcontroller
Het verkeerslicht met delay(): nu zonder wachten en met een knop.

I26 Motoren en actuatoren
De dakmotor, de H-brug, de eindeloopschakelaars en de time-out.

I29 Microcontrollers programmeren in C++
millis(), taken, enum en eigen bestanden (.h en .cpp).

I31 Regelsystemen
Hysterese maakt van een temperatuur een gebeurtenis voor de toestandsmachine.

I20 Zonne-energie dimensioneren
Het energiebeheer met drempels: een toestandsmachine voor de batterij.

I33 Een mobiele robot
Lijn volgen, meten, obstakels vermijden: de robot als toestandsmachine.

E23 Digitale logica
Toestandsmachines in hardware: flipflops en de noodstop-logica.

E37 Embedded systemen en IoT
Interrupts, timers en real-time eisen.

E46 Betrouwbaarheid en risico
Faalwijzen systematisch zoeken (FMEA) en fail-safe ontwerpen.

Habitat
De klimaatmodule heeft nu vijf modi (OPSTART, DAG, NACHT, ALARM, ONDERHOUD), een actietabel en een alarmtabel, en het kasdak is een toestandsmachine met time-out en veilige start. In hoofdstuk 33 wordt het inspectierobotje een

toestandsmachine; in het integratieproject (hoofdstuk 40) test je alle modi met een scenario op de echte module.

SAMENVATTING

In het kort

- Een toestandsmachine is altijd in precies één toestand; gebeurtenissen veroorzaken overgangen, met acties.
- In een diagram staat op elke pijl gebeurtenis [voorwaarde] / actie; acties bij binnenkomen staan in de toestand.
- Een toestandstabel (toestanden × gebeurtenissen) dwingt je om elke combinatie te bedenken, ook de lege.
- In C++: een enum, één variabele, naar() voor overgangen en acties, een switch met break, de tijd als nu – sinds.
- Time-outs en onmogelijke combinaties verraden fouten; de veilige toestand hangt af van de fout en de omstandigheden.
- Fail-safe: een defect brengt het systeem vanzelf in de veilige toestand (verbreekcontact, pull-up, rem van Otis).
- De waakhond herstart een programma dat blijft hangen; hij vervangt geen goede time-outs.
- De klimaatmodule heeft vijf modi; DAG en NACHT vormen samen NORMAAL, ALARM laat zoveel mogelijk verder werken.

BEGRIPPEN

Wat je nu kent

toestandsmachine
Een systeem dat op elk ogenblik in precies één van een beperkt aantal toestanden is, en bij gebeurtenissen van toestand verandert.

toestand
Een situatie waarin een systeem kan blijven, met een vast gedrag (DICHT, OPENEN, NACHT).

gebeurtenis
Iets dat gebeurt en een overgang kan veroorzaken: een knopdruk, een eindeloopschakelaar, een tijd die om is.

overgang
De stap van een toestand naar een andere (of dezelfde), bij een gebeurtenis, eventueel alleen als een voorwaarde klopt.

toestandstabel
Tabel met de toestanden als rijen en de gebeurtenissen als kolommen; elk vakje zegt wat er gebeurt.

time-out
Een grens op de tijd in een toestand; wordt ze overschreden, dan is er iets mis.

veilige toestand
De toestand waarin een systeem bij een fout de minste schade doet; ze hangt af van de fout en van de omstandigheden.

fail-safe
Zo ontworpen dat een defect (een losse draad, een stroomuitval) vanzelf tot de veilige toestand leidt.

waakhond (watchdog)
Een aparte timer in de microcontroller die hem herstart als het programma niet regelmatig laat weten dat het nog loopt.

I INTERMEDIATE · DEEL I8 · REGELEN EN ROBOTICA

HANDBOEK · HOOFDSTUK 33

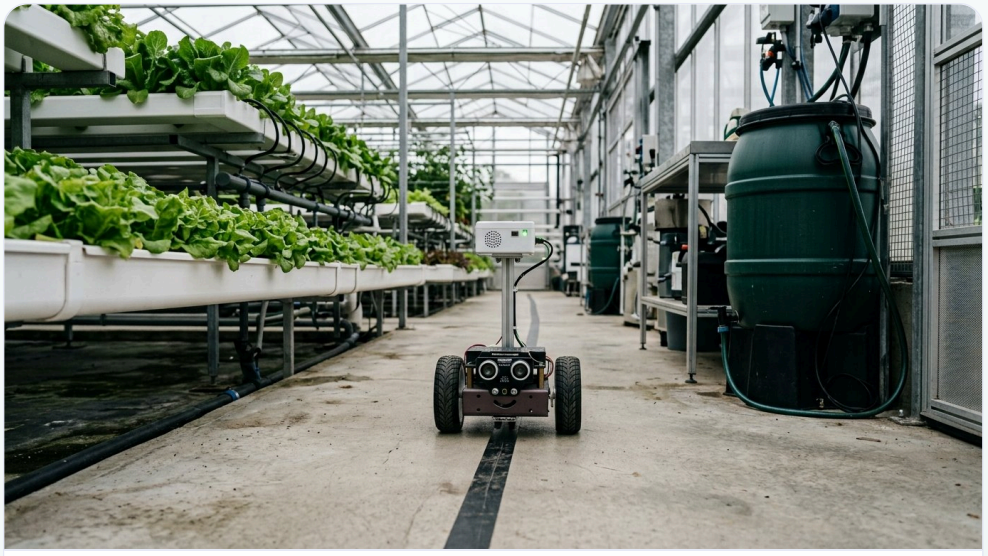
Een mobiele robot

Twee wielen, drie lichtsensoren en een oor: een inspectierobotje voor de gang van de klimaatmodule

Een robot die zelf door een ruimte rijdt, moet drie dingen kunnen: bewegen en sturen, weten waar hij is, en stoppen voor wat in de weg staat. In dit hoofdstuk bouw je de natuurkunde en de code van een klein robotje met twee aangedreven wielen: hoe het draait, hoe snel het rijdt door de tandwielkast, hoe het een lijn volgt, hoe het afstand meet met geluid, en hoe een toestandsmachine alles samenhoudt. Het robotje rijdt in de gang van de klimaatmodule en meet langs het kweekbed temperatuur, vocht en lekken.

8× uitleg · 3× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 9 lestijden

R Regelen en robotica · K Krachten, beweging en constructies · P Programmeren en computationeel denken · L Elektriciteit en elektronica



Het inspectierobotje meet waar de vaste sensorkast niet komt: laag, langs het hele kweekbed.

Na dit hoofdstuk

- Je kan met v = (vL + vR)/2 en ω = (vR − vL)/b de snelheid, de hoeksnelheid en de draaistraal van een robot met twee wielen berekenen.
- Je kan uit de motordata, de overbrengingsverhouding en de wieldiameter de snelheid van een robot berekenen, en de PWM-waarde kiezen.
- Je kan uitleggen hoe een reflectiesensor werkt, hem ijken, en een lijnvolger maken met aan/uit en met een P-regelaar.
- Je kan met s = v · t / 2 een afstand berekenen uit een echo, en de invloed van de temperatuur op de geluidssnelheid inschatten.
- Je kan het gedrag van een robot ontwerpen als toestandsmachine, met obstakels, meetpunten, time-outs en een veilige toestand.
- Je kan een simulatie van een robotpad lezen en gebruiken om een versterking of een snelheid te kiezen.

VERHAAL · 33.1

Een trekker die een draad volgt

1/18

In 1953 bouwde het Amerikaanse bedrijf Barrett Electronics een trekker zonder bestuurder. Hij trok karren door een magazijn en vond zijn weg door een draad te volgen. Het was de eerste **AGV**

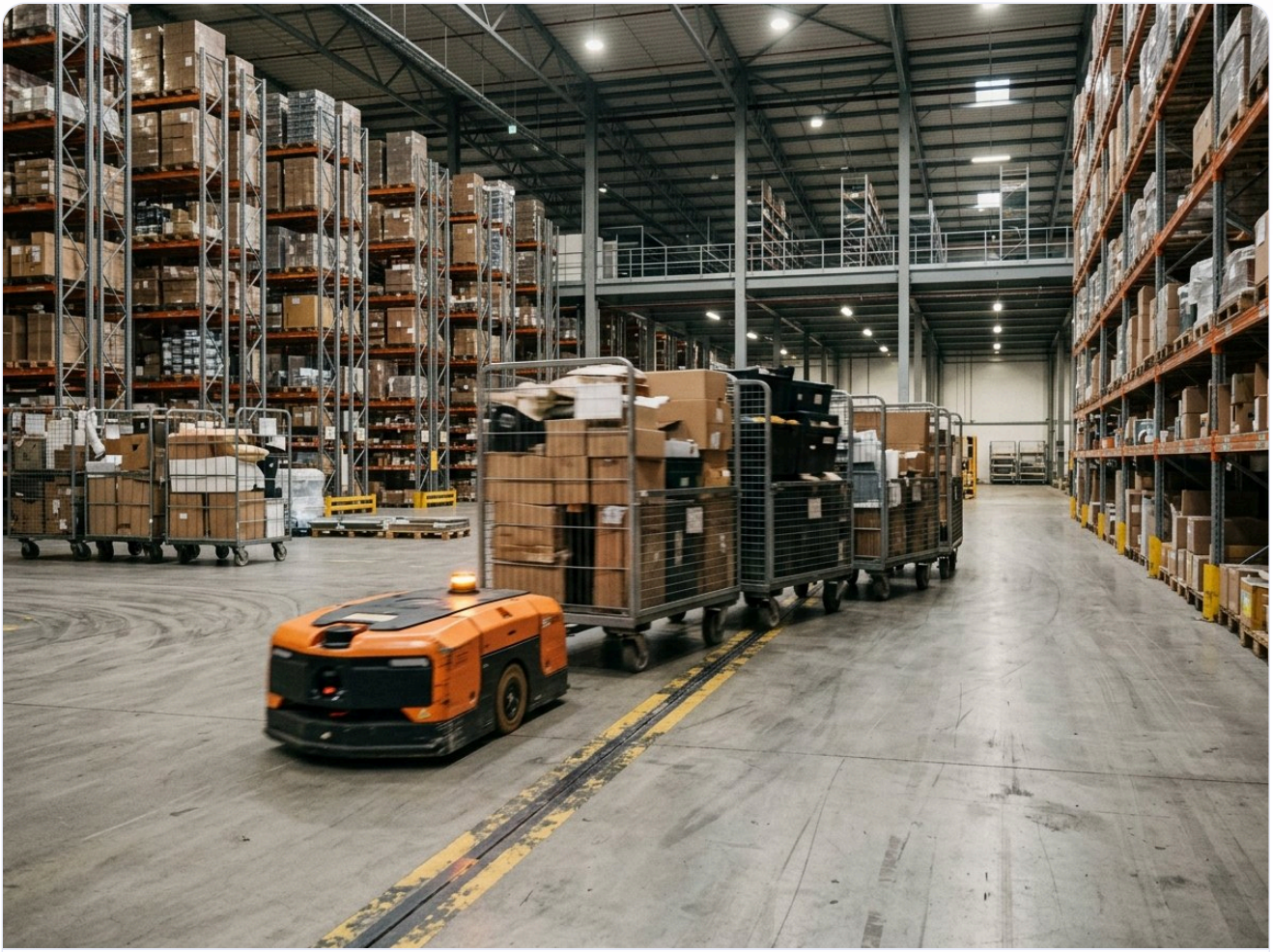
(automated guided vehicle), een 'automatisch geleid voertuig'. Vandaag rijden er duizenden door fabrieken, magazijnen en ziekenhuizen: ze volgen een lijn of magneetband op de vloer, of rekenen hun plaats uit met lasers en camera's.

De meeste kleine robots, ook stofzuigrobots, hebben geen stuur. Ze rijden op **twee wielen met elk een eigen motor**, en een zwenkwieltje dat meeloopt. Draaien ze beide wielen even snel, dan rijden ze rechtdoor; draait het ene sneller, dan maken ze een bocht; draaien ze tegengesteld, dan draaien ze op de plaats. Dat maakt ze wendbaar in smalle ruimtes, precies wat nodig is in de gang van de klimaatmodule.

Het inspectierobotje van Habitat is zo'n kleine AGV. De vaste sensorkast meet op één plaats, maar sla groeit langs het hele kweekbed, en een lek onder een goot zie je niet vanuit de sensorkast. Het robotje rijdt een ronde van bijna 3 m, stopt op drie meetpunten, meet er temperatuur, luchtvochtigheid en of de vloer nat is, en rijdt terug naar zijn laadstation.

Wat je al weet

Overbrenging en tandwielen (Fundamental, hoofdstuk 12), snelheid en beweging (hoofdstukken 7 tot 9), de spanningsdeler en sensoren (22), PWM (25), motoren en de H-brug (26), millis() en taken (29), regelen (31) en toestandsmachines (32). Dit hoofdstuk zet ze samen in één machine.

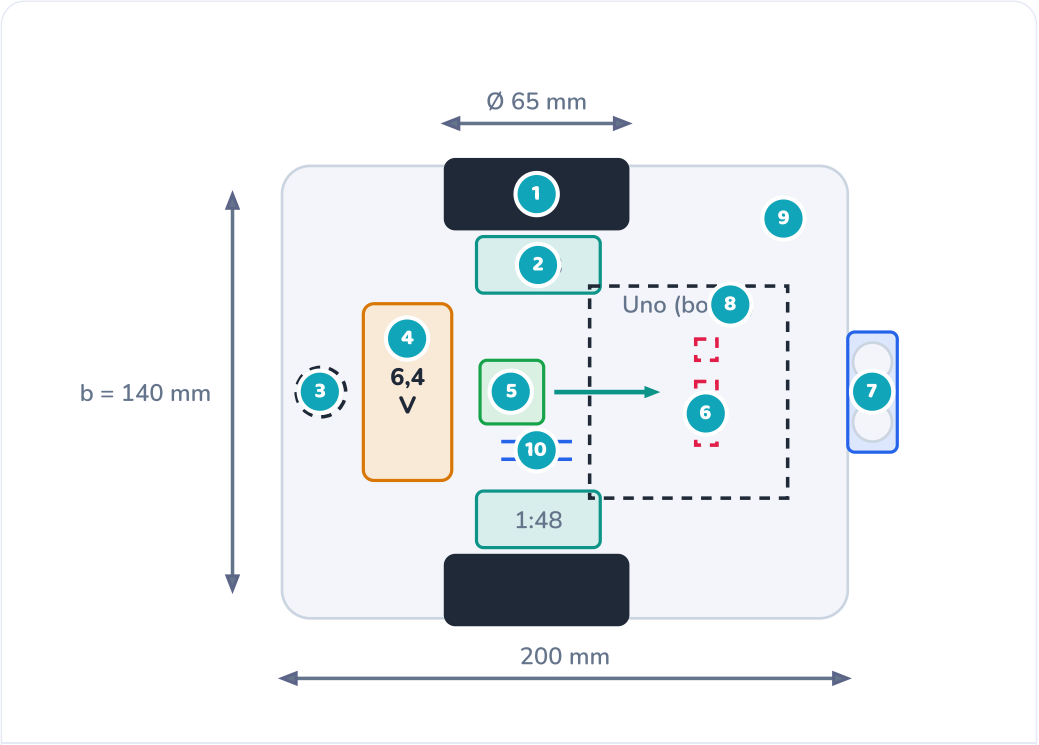


Een AGV in een magazijn volgt een lijn op de vloer, net als het robotje in de module.

ONTLEED · 33.2

Het inspectierobotje

Het robotje is 20 cm lang en 16 cm breed: klein genoeg voor de gang van 80 cm tussen de buitenwand en het kweekbed, en groot genoeg voor een batterij die een dag meegaat. Onder staan de wielen, de motoren en de sensoren; op een tweede verdieping de Arduino Uno.



Bovenaanzicht; de neus wijst naar rechts. Stippellijnen: onder de bodemplaat of op de bovenste verdieping.

1 Wielen
Twee wielen van 65 mm, 140 mm uit elkaar (de spoorbreedte b). Elk heeft een eigen motor: zo stuurt het robotje.

2 Motor met tandwielkast
Een motortje van 6 V met een tandwielkast 1:48: de wielen draaien 48 keer trager, met bijna 48 keer meer koppel.

3 Zwenkwiel
Een balletje of wieltje onder de achterkant: het draagt, maar stuurt niet.

4 Batterij
Twee LiFePO4-cellen in serie: 6,4 V, ongeveer 1,5 Ah (±10 Wh). LiFePO4, zoals de batterij van de module (hoofdstuk 16).

5 Motordriver
Een TB6612FNG: twee H-bruggen (hoofdstuk 26) op één printje, tot 1,2 A per motor. Per motor een PWM-pin en twee richtingspinnen.

6 Reflectiesensoren
Drie sensoren onder de neus, 60 mm voor de as en 15 mm uit elkaar, 5 mm boven de vloer: ze zien de zwarte tape.

7 Ultrasone sensor
De HC-SR04 met zender en ontvanger: meet de afstand tot wat voor het robotje staat, van 2 cm tot 4 m.

8 Arduino Uno
Op de bovenste verdieping. Het robotje heeft zijn eigen microcontroller; de Mega van de module blijft voor de module.

9 Mast met SHT31
Een dun mastje met een SHT31 op 0,50 m, de hoogte van de onderste goot: daar meet de vaste sensorkast niet.

10 Lekdetectie
Twee koperstrips 3 mm boven de vloer, als spanningsdeler op A3: water tussen de strips laat de spanning zakken.

UITLEG · 33.3

Twee wielen, één draaipunt

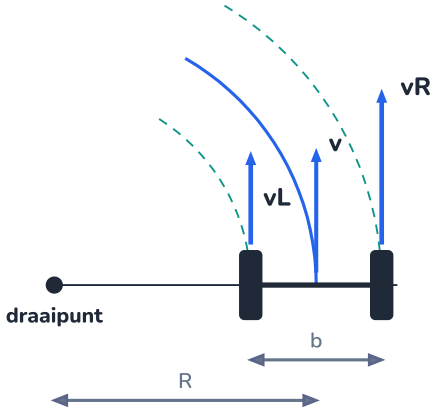
Rijden beide wielen even snel, dan gaat het robotje rechtdoor. Draait het rechterwiel sneller dan het linker, dan legt het rechterwiel in dezelfde tijd een langere weg af: het robotje draait naar links. Beide wielen draaien dan rond hetzelfde punt op de verlengde as, het **draaipunt**. Het rechterwiel ligt op afstand $R + b/2$ ervan, het linker op $R - b/2$, met R de straal van het midden van de as en b de spoorbreedte. Omdat het hele robotje met dezelfde hoeksnelheid ω (in rad/s) draait, geldt $v_R = \omega\,(R + b/2)$ en $v_L = \omega\,(R - b/2)$. Tel op en trek af:

DIFFERENTIËLE AANDRIJVING

$$v = \frac{v_L + v_R}{2} \qquad \omega = \frac{v_R - v_L}{b} \qquad R = \frac{v}{\omega}$$

v: snelheid van het midden van de as (m/s); ω: hoeksnelheid (rad/s), positief = naar links (tegen de klok in); b: spoorbreedte (m); R: draaistraal (m).

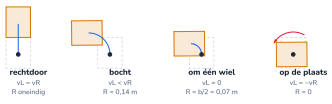
Vier gevallen vallen op. Rechtdoor: $v_L = v_R$, dus $\omega = 0$ en R oneindig. Bocht: v_L en v_R verschillend maar met hetzelfde teken. Om één wiel: $v_L = 0$ geeft $R = b/2$. Op de plaats: $v_L = -v_R$ geeft $v = 0$ en $R = 0$. Dat laatste kan een auto niet: die heeft een stuur en een kleinste draaicirkel.




```
v = (vL + vR) / 2
ω = (vR − vL) / b
R = v / ω
```

voorbeeld:
vL = 0,15 m/s
vR = 0,25 m/s
b = 0,14 m
R = 0,28 m

De snelheid van elk wiel is evenredig met zijn afstand tot het draaipunt.



Uitgerekend met de formules: vier keer ongeveer een kwartslag, telkens met een andere draaistraal.

UITGEWERKT VOORBEELD · 33.4

Bochten en draaien in de gang

4/18

Het robotje ($b = 0{,}14\,\mathrm{m}$) krijgt drie opdrachten. (a) $v_L = 0{,}15\,\mathrm{m/s}$ en $v_R = 0{,}25\,\mathrm{m/s}$: bereken v, ω en R. (b) De lus in de gang heeft bochten met $R = 0{,}15\,\mathrm{m}$. Welke wielsnelheden horen bij $v = 0{,}20\,\mathrm{m/s}$ in zo'n bocht naar rechts? (c) Hoe lang duurt een halve draai op de plaats als elk wiel $0{,}10\,\mathrm{m/s}$ rijdt?

GEGEVEN

- $b = 0{,}14\,\text{m}$
- (a) $v_L = 0{,}15$, $v_R = 0{,}25\,\text{m/s}$
- (b) $R = 0{,}15\,\text{m}$, $v = 0{,}20\,\text{m/s}$
- (c) $v_L = -0{,}10$, $v_R = +0{,}10\,\text{m/s}$

GEVRAAGD

- v, ω, R
- v_L, v_R
- tijd voor 180°

OPLOSSING

1. (a) $v = (0{,}15 + 0{,}25)/2 = 0{,}20\,\text{m/s}$; $\omega = (0{,}25 - 0{,}15)/0{,}14 = 0{,}714\,\text{rad/s}$; $R = 0{,}20/0{,}714 = 0{,}28\,\text{m}$.
2. (b) $\omega = v/R = 0{,}20/0{,}15 = 1{,}33\,\text{rad/s}$, naar rechts dus negatief. Het buitenste (linker) wiel: $v_L = v + |\omega|\,b/2 = 0{,}20 + 1{,}33 \cdot 0{,}07 = 0{,}293\,\text{m/s}$; het binnenste: $v_R = 0{,}20 - 1{,}33 \cdot 0{,}07 = 0{,}107\,\text{m/s}$.
3. (c) Op de plaats: $v = 0$ en $\omega = (0{,}10 - (-0{,}10))/0{,}14 = 1{,}43\,\text{rad/s}$. Een halve draai is π rad: $t = \pi/1{,}43 = 2{,}20\,\text{s}$.
4. Controle voor (c): elk wiel rijdt een halve cirkel met straal $b/2$: $\pi \cdot 0{,}07 = 0{,}220\,\text{m}$, en dat duurt aan $0{,}10\,\text{m/s}$ ook 2,20 s.

ANTWOORD

(a) 0,20 m/s, 0,71 rad/s, R = 0,28 m. (b) Buitenwiel 0,29 m/s, binnenwiel 0,11 m/s. (c) 2,2 s.

Klopt dit? Een bocht met een straal kleiner dan b/2 = 0,07 m laat het binnenste wiel achteruit draaien: probeer R = 0,05 m. In de gang van 0,80 m kan het robotje overal omkeren: bij een halve draai op de plaats beschrijft zijn verste hoek een cirkel van 0,14 m straal.

UITLEG · 33.5

Van motor tot wiel: de overbrenging

5/18

Een klein gelijkstroommotortje draait snel maar zwak: onbelast ongeveer 9000 omwentelingen per minuut bij 6 V, met een piepklein koppel. Een wiel van 65 mm aan die as zou $\pi \cdot 0{,}065 \cdot 9000/60 = 31\,\text{m/s}$ willen rijden, maar kreeg het robotje niet eens in beweging. Een **tandwielkast** ruilt toerental voor koppel, zoals de versnellingen van een fiets (Fundamental, hoofdstuk 12).

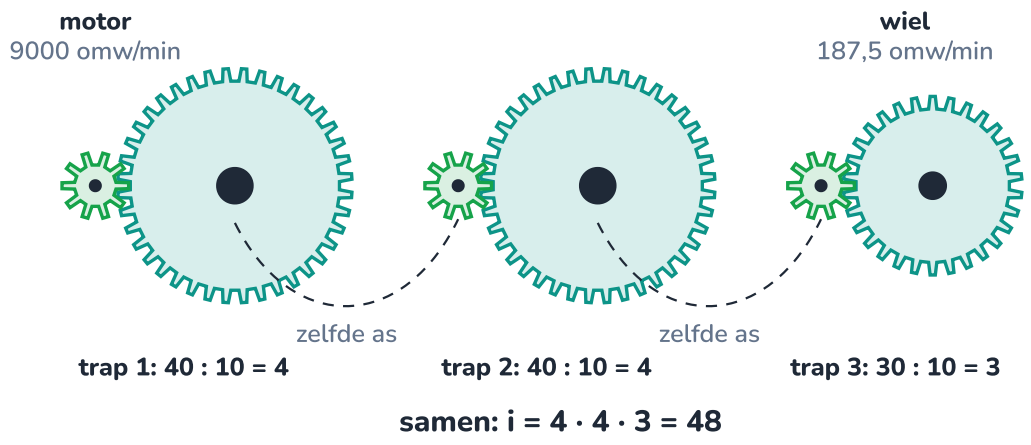
OVERBRENGING EN WIELSNELHEID

$$i = \frac{z_2}{z_1} = \frac{n_1}{n_2} \qquad v = \pi \cdot d \cdot \frac{n_{\text{wiel}}}{60}$$

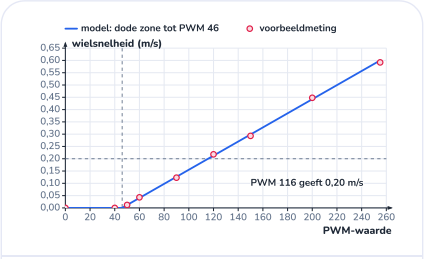
Bij meerdere trappen vermenigvuldig je de verhoudingen. n in omw/min, d in m, v in m/s.

De kast van het robotje heeft drie trappen: 10 tanden op 40, nog eens 10 op 40, en 10 op 30. Samen $i = 4 \cdot 4 \cdot 3 = 48$. Het wiel draait dan onbelast $9000/48 = 187{,}5$ omw/min, en het robotje rijdt $\pi \cdot 0{,}065 \cdot 187{,}5/60 = 0{,}64\,\text{m/s}$. Het koppel aan het wiel is bijna 48 keer groter dan dat van de motor; 'bijna', omdat elk tandwielpaar wat wrijving verliest (hoofdstuk 9).

Onder belasting en met wat wrijving rekenen we met hoogstens 0,60 m/s bij PWM 255. En er is een **dode zone**: bij een kleine PWM-waarde draait het wiel niet, omdat de motor de wrijving in de kast niet overwint. Bij dit motortje ligt die grens rond 18 % (PWM 46). Daarboven loopt de snelheid ongeveer recht evenredig op. De code overbrugt de dode zone: wie 0,20 m/s wil, stuurt PWM 116, niet PWM 85.



Drie keer een klein tandwiel op een groot: 4 · 4 · 3 = 48 keer trager en sterker.



Onder PWM 46 staat het wiel stil; daarboven stijgt de snelheid recht evenredig.

UITGEWERKT VOORBEELD · 33.6

Van motordata tot rijsnelheid

6/18

Het robotje moet 0,20 m/s rijden. Hoe snel draaien het wiel en de motoras dan? Welke PWM-waarde stuurt de code? Hoeveel kracht en vermogen vraagt het rijden, als de rolweerstand 2,5 % van het gewicht is (hoofdstuk 8) en het robotje 0,80 kg weegt?

GEGEVEN

- $d = 0{,}065\,\text{m}$, $i = 48$
- model: $v = 0{,}60\,\text{m/s} \cdot \dfrac{D - 0{,}18}{0{,}82}$, met D = PWM/255
- $m = 0{,}80\,\text{kg}$, rolweerstand 2,5 %

GEVRAAGD

- n_{wiel}, n_{motor}
- PWM-waarde
- kracht en vermogen

OPLOSSING

1. $n_{\text{wiel}} = \dfrac{60\,v}{\pi d} = \dfrac{60 \cdot 0{,}20}{\pi \cdot 0{,}065} = 58{,}8\,\text{omw/min}.$
2. $n_{\text{motor}} = i \cdot n_{\text{wiel}} = 48 \cdot 58{,}8 = 2821\,\text{omw/min}.$
3. PWM: $D = 0{,}18 + 0{,}82 \cdot \dfrac{0{,}20}{0{,}60} = 0{,}453$, dus PWM $= 255 \cdot 0{,}453 \approx 116$.
4. Rolweerstand: $F = 0{,}025 \cdot 0{,}80 \cdot 9{,}81 = 0{,}196\,\text{N}$ voor het hele robotje, 0,098 N per wiel. Koppel per wiel: $M = F \cdot r = 0{,}098 \cdot 0{,}0325 = 0{,}0032\,\text{N} \cdot \text{m}$ (hoofdstuk 10).
5. Vermogen om te rijden: $P = F \cdot v = 0{,}196 \cdot 0{,}20 = 0{,}039\,\text{W}$ (hoofdstuk 9).

ANTWOORD

Wiel 58,8 omw/min, motoras 2821 omw/min, PWM 116. Rijden kost maar 0,04 W mechanisch.

Klopt dit? De motoren nemen samen toch ongeveer 2 W op: bijna alles gaat verloren in de weerstand van de wikkelingen, de wrijving in de kast en de driver. Bij zo'n klein vermogen is het rendement laag (hoofdstuk 26). Een ronde van ongeveer 50 s kost zo, met de elektronica erbij, ongeveer 0,024 Wh: op één lading haalt het robotje honderden rondes.

PROEF · 33.7

De dode zone van een wielmotor

7/18

ONDERZOEKSVRAAG

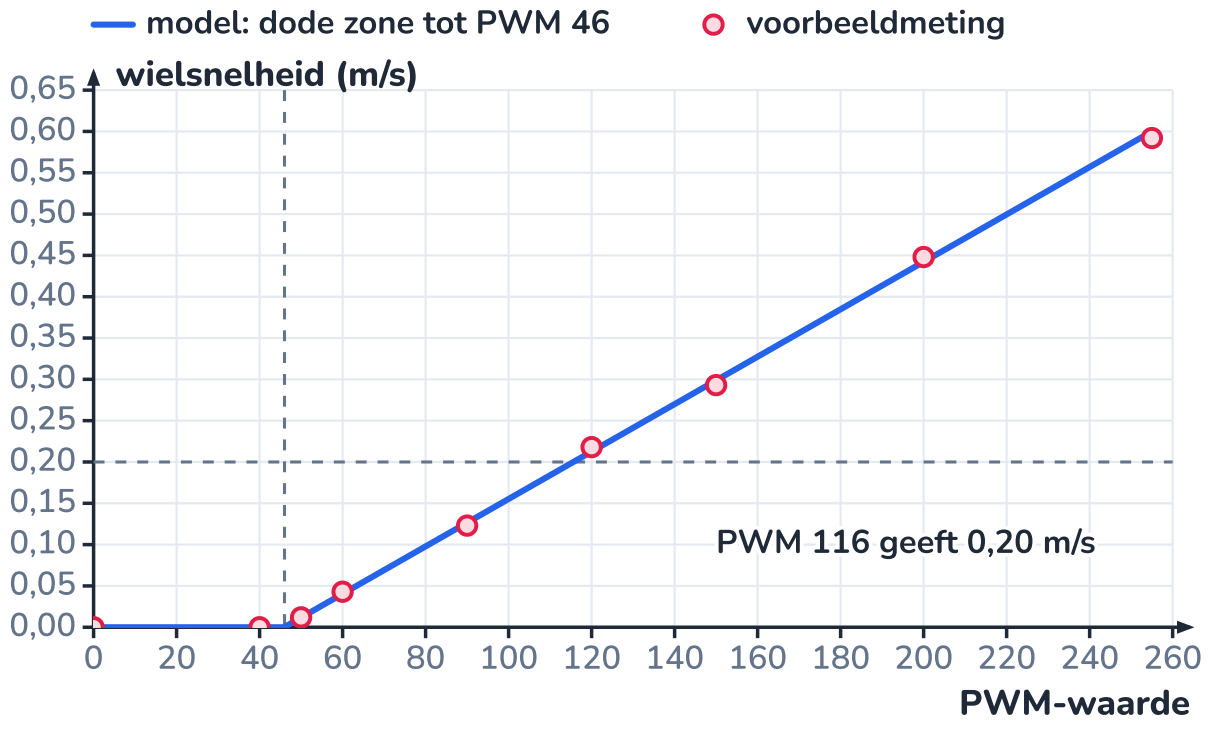
Vanaf welke PWM-waarde draait een wiel, en hoe hangt de snelheid van de PWM-waarde af?

MATERIAAL

- het robotje (of één motor met wiel en driver)
- Arduino met een programma dat een vaste PWM-waarde stuurt
- meetlint en stopwatch (of een smartphone die filmt)
- tape: een rechte lijn van 1,0 m

Veiligheid

Laat het robotje niet van een tafel rijden: meet op de vloer, of zet het op een blokje met de wielen vrij.



Voorbeeldmeting en model: de rechte snijdt de as bij de grens van de dode zone.

WERKWIJZE

1. Laat beide motoren draaien met dezelfde PWM-waarde, te beginnen bij 0, en verhoog telkens een beetje.
2. Noteer de kleinste waarde waarbij het robotje zelf vertrekt.

3 Meet voor een reeks PWM-waarden hoe lang het robotje over 1,0 m doet (na een aanloop van 20 cm). Herhaal elke meting drie keer.

4 Zet de snelheid uit tegen de PWM-waarde en trek een rechte door de punten boven de dode zone.

WAARNEMING

PWM	0	40	50	60	90	120	150
v (m/s)	0,000	0,000	0,012	0,043	0,123	0,218	0,293

Voorbeeldmeting. Tot PWM 40 blijft het robotje staan; bij 50 kruipt het nauwelijks vooruit; daarboven stijgt de snelheid netjes recht evenredig.

VERKLARING

Een motor levert een koppel dat evenredig is met de stroom (hoofdstuk 26). Bij een kleine PWM-waarde is de gemiddelde spanning te laag om de wrijving in de tandwielkast en de rolweerstand te overwinnen: het wiel staat stil, de motor wordt alleen wat warm. Boven die drempel is de snelheid ongeveer evenredig met wat er boven de drempel uitkomt. Het snijpunt van de rechte met de horizontale as is de dode zone, hier rond PWM 46. Een regelaar die niets van de dode zone weet, stuurt bij kleine correcties niets: het binnenste wiel in een bocht zou stilvallen.

UITLEG · 33.8

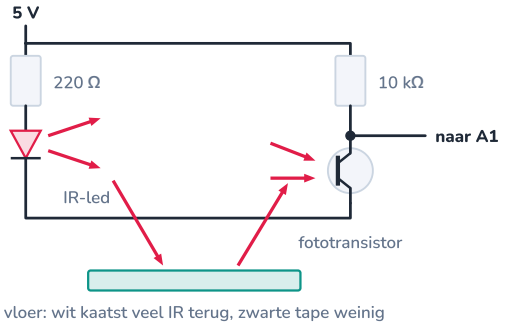
Een lijn zien: de reflectiesensor

8/18

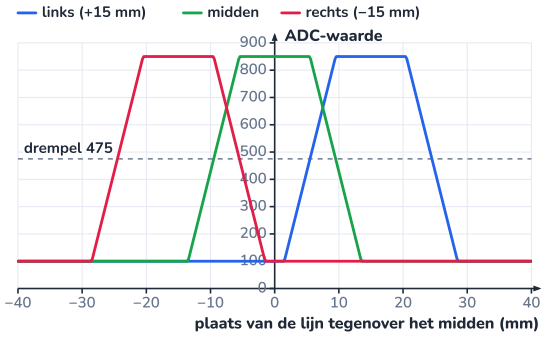
Een reflectiesensor (bijvoorbeeld de TCRT5000) heeft een infrarood-led en een fototransistor naast elkaar, met een filter tegen daglicht. De led schijnt op de vloer; hoe meer licht er terugkaatst, hoe beter de fototransistor geleidt. Met een weerstand van 10 kΩ naar 5 V vormt hij een **spanningsdeler** (hoofdstuk 22): op een witte vloer geleidt hij goed en zakt de spanning op de pin; boven zwarte tape komt er weinig licht terug en blijft de spanning hoog. De sensor werkt het best vlak boven de vloer (de datasheet noemt 2,5 mm als beste afstand); het robotje heeft hem op ongeveer 5 mm, veilig boven drempels en voegen.

IJken is nodig, want elke sensor en elke vloer is anders. Houd elke sensor boven de vloer en boven de tape, en noteer de ADC-waarden. Hier: ongeveer 100 op wit en 850 op zwart. Het getal `(ruw - wit) / (zwart - wit)` geeft dan 0 voor wit en 1 voor zwart; de drempel in het midden is 475.

Het robotje heeft drie sensoren, 15 mm uit elkaar. De tape is 19 mm breed: ligt ze in het midden, dan ziet alleen de middelste sensor zwart. Schuift de lijn opzij, dan wordt een buitenste sensor geleidelijk zwarter. Die overgang is niet scherp, omdat elke sensor een vlekje van enkele millimeter ziet.



Veel teruggekaatst IR-licht: de transistor geleidt, de spanning op A1 zakt.



Model van de drie sensoren: elke sensor ziet de tape over ongeveer 25 mm.

Lijnvolgen: aan/uit of evenredig

De eenvoudigste lijnvolger werkt **aan/uit** (hoofdstuk 31): elke sensor is zwart of wit, en een tabel zegt wat het robotje doet.

links	midden	rechts	betekenis	aan/uit doet
wit	zwart	wit	lijn in het midden	rechtdoor
zwart	zwart	wit	lijn iets links	bijsturen naar links
zwart	wit	wit	lijn ver links	bijsturen naar links
wit	zwart	zwart	lijn iets rechts	bijsturen naar rechts
wit	wit	zwart	lijn ver rechts	bijsturen naar rechts
zwart	zwart	zwart	dwarsstreep	meetpunt
wit	wit	wit	lijn kwijt	zoeken aan de laatste kant

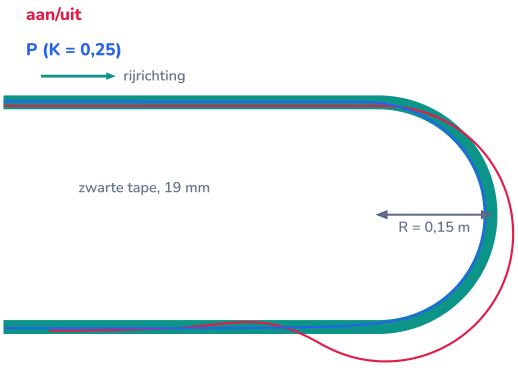
Dat werkt, maar het robotje zigzagt: het stuurt altijd even hard bij, of het nu 2 of 12 mm naast de lijn rijdt. Beter is een **evenredige** of P-regelaar: hoe verder de lijn opzij ligt, hoe harder het robotje draait. De plaats van de lijn volgt uit de drie sensoren als **gewogen gemiddelde**:

P-REGELAAR VOOR EEN LIJNVOLGER

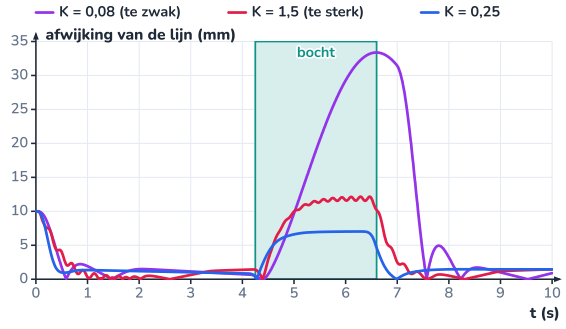
$$p = \frac{15\,w_L + 0\,w_M - 15\,w_R}{w_L + w_M + w_R} \qquad \omega = K \cdot p$$

w: hoe zwart elke sensor ziet (0 tot 1); p in mm, positief = de lijn ligt links; K in rad/s per mm. De wielen krijgen $v_L = v_0 - \omega b/2$ en $v_R = v_0 + \omega b/2$.

In de simulatie (hierna) start het robotje 1 cm naast de lijn en rijdt het de bocht van de lus. Met aan/uit wijkt het midden van de as tot 48 mm af van de lijn: zijn vaste correctie (het ene wiel $0{,}5\,v_0$, het andere $1{,}2\,v_0$) geeft een draaistraal van 0,17 m, te ruim voor de bocht van 0,15 m, en het zwaait naar buiten. Met de P-regelaar ($K = 0{,}25$) is dat maar 7 mm. Kies je K te klein, dan draait het robotje in de bocht niet genoeg en verliest het bijna de lijn (33 mm). Te groot, en het slingert na elke correctie (12 mm): de wielen hebben tijd nodig om een nieuwe snelheid te halen, en de regelaar overcorrigeert. Hetzelfde verschijnsel ken je van de trage sensor in hoofdstuk 31.



Exact getekend uit de simulatie: aan/uit zwaait in de bocht naar buiten, P blijft bij de tape.



Afwijking van de as: te zwak mist de bocht bijna, te sterk slingert, K = 0,25 is een goede keuze.

CODE · 33.10

De lijnvolger simuleren

10/18

De simulatie rekent elke 2 ms uit waar het robotje staat: uit de wielsnelheden volgen v en ω, en daaruit de nieuwe plaats en richting (hoofdstuk 7: $x \leftarrow x + v\cos\theta\,\Delta t$). Elke 20 ms kijkt de regelaar naar de sensoren, zoals de Arduino. Het model kent ook de dode zone en de traagheid van de wielen.

Python **lijnvolger.py** 77 regels

```python
# lijnvolger.py - een robot met twee wielen volgt de lus in de gang (P-regelaar of aan/uit)
import math

B = 0.14                  # m tussen de wielen
A = 0.06                  # m: de reflectiesensoren zitten voor de as
V_MAX, D0 = 0.60, 0.18    # m/s bij PWM 255; dode zone van de motoren
TAU = 0.08                # s: zo snel volgt een wiel een nieuwe snelheid
DT, T_REGEL = 0.002, 0.02
XA, XB, YA, YB, YC, R = 0.45, 1.40, 0.55, 0.25, 0.40, 0.15    # de lus (m)


def tot_lijn(x, y):
    d = 9.0
    if XA <= x <= XB:
        d = min(abs(y - YA), abs(y - YB))
    if x >= XB:
        d = min(d, abs(math.hypot(x - XB, y - YC) - R))
    if x <= XA:
        d = min(d, abs(math.hypot(x - XA, y - YC) - R))
    return d


def zwart(dy, x, y, th):
    """Hoe zwart ziet de sensor op dy (m, + = links) het? 0 = wit, 1 = zwart (tape 19 mm, vlekje 8 mm)."""
    sx = x + A * math.cos(th) - dy * math.sin(th)
    sy = y + A * math.sin(th) + dy * math.cos(th)
    return min(1.0, max(0.0, (0.0135 - tot_lijn(sx, sy)) / 0.008))


def wiel(v):
    """Wat het wiel echt doet: de code overbrugt de dode zone, PWM in hele stappen."""
    if v == 0:
        return 0.0
    pwm = round(255 * (D0 + (1 - D0) * min(abs(v) / V_MAX, 1.0)))
    return math.copysign(V_MAX * max(0.0, pwm / 255 - D0) / (1 - D0), v)


def rij(regelaar, K=0.25, v0=0.20, x=0.55, y=0.56, th=0.0, T=10.0):
    vl = vr = doel_l = doel_r = 0.0
    kant, pad = 1, []
    for k in range(int(T / DT)):
        if k % int(T_REGEL / DT) == 0:                      # 50 keer per seconde beslissen
            w = [zwart(dy, x, y, th) for dy in (0.015, 0.0, -0.015)]
            if regelaar == "P":
                if sum(w) > 0.2:
                    pos = (w[0] * 15 - w[2] * 15) / sum(w)   # mm, + = lijn links
                    kant = 1 if pos > 0 else -1
                else:
                    pos = 15 * kant                         # kwijt: zoek aan de laatste kant
                om = K * pos
                doel_l, doel_r = v0 - om * B / 2, v0 + om * B / 2
            else:                                           # aan/uit met drempel 0,5
                links, midden, rechts = (s > 0.5 for s in w)
                if links and not rechts:
                    kant = 1
                elif rechts and not links:
                    kant = -1
```

```
                if midden and not (links or rechts):
                    doel_l = doel_r = v0
                else:                                   # bijsturen naar de kant van de lijn
                    doel_l, doel_r = (0.5 * v0, 1.2 * v0) if kant > 0 else (1.2 * v0, 0.5 * v0)
        vl += (wiel(doel_l) - vl) * DT / TAU
        vr += (wiel(doel_r) - vr) * DT / TAU
        v, om = (vl + vr) / 2, (vr - vl) / B
        x += v * math.cos(th) * DT
        y += v * math.sin(th) * DT
        th += om * DT
        pad.append((k * DT, x, y, tot_lijn(x, y)))
    return pad


if __name__ == "__main__":
    for naam, K in (("aan/uit", 0), ("P, K = 0,08", 0.08), ("P, K = 0,25", 0.25), ("P, K = 1,5", 1.5)):
        pad = rij("aanuit" if K == 0 else "P", K)
        afw = [p[3] * 1000 for p in pad if p[0] > 0.5]
        weg = sum(math.hypot(b[1] - a[1], b[2] - a[2]) for a, b in zip(pad, pad[1:]))
        print(f"{naam:12} grootste afwijking {max(afw):5.1f} mm, gemiddeld {sum(afw) / len(afw):4.1f} mm, afgelegd
{weg:.2f} m")
```

Uitvoer:

```
aan/uit      grootste afwijking  47.8 mm, gemiddeld 14.1 mm, afgelegd 1.85 m
P, K = 0,08  grootste afwijking  33.4 mm, gemiddeld  7.6 mm, afgelegd 1.99 m
P, K = 0,25  grootste afwijking   7.0 mm, gemiddeld  2.4 mm, afgelegd 1.99 m
P, K = 1,5   grootste afwijking  12.2 mm, gemiddeld  3.2 mm, afgelegd 1.96 m
```

'Afgelegd' is de weg van het midden van de as in 10 s. Aan/uit komt minder ver: het zigzagt en rijdt bij elke correctie met één wiel trager. Alle paden in de figuren hiervoor komen uit deze functie.

UITLEG · 33.11

Afstand meten met een echo

11/18

De HC-SR04 meet afstand zoals een vleermuis: met geluid dat we niet horen. Op een puls van 10 µs op TRIG zendt hij 8 trillingen van 40 kHz uit. Daarna maakt hij ECHO hoog, tot de echo terugkomt. De breedte van de ECHO-puls is de tijd t die het geluid nodig had om heen en terug te gaan. Omdat het geluid de afstand twee keer aflegt:

AFSTAND UIT EEN ECHO

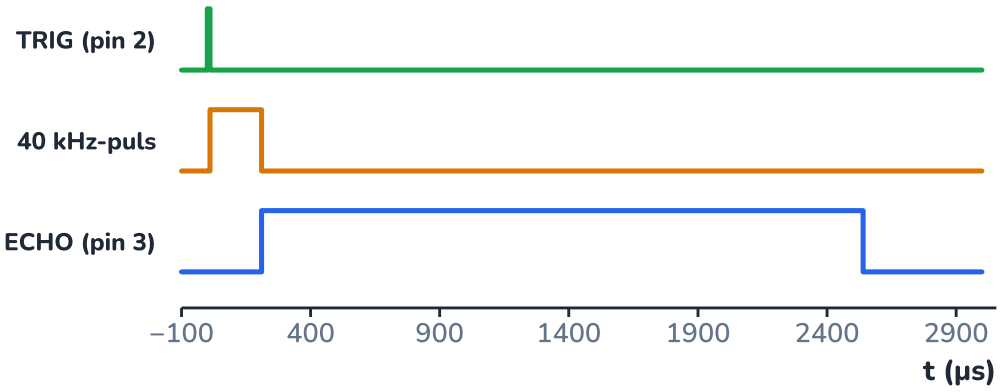
$$s = \frac{v \cdot t}{2} \qquad v \approx 331{,}3 + 0{,}606 \cdot T$$

v: geluidssnelheid in m/s, T: luchttemperatuur in °C. Bij 20 °C is $v = 343{,}4\,\mathrm{m/s}$.

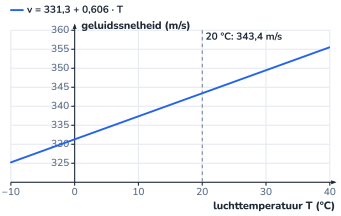
Op 40 cm duurt de echo $2 \cdot 0{,}40 / 343{,}4 = 2{,}33\,\mathrm{ms}$. De datasheet geeft de vuistregel 'microseconden gedeeld door 58 = centimeter': dat hoort bij ongeveer 22 °C. Geluid gaat sneller in warme lucht, dus meet het robotje de temperatuur met zijn SHT31 en rekent het met de echte geluidssnelheid.

In C++ doet `pulseIn()` het meetwerk: de functie wacht tot ECHO hoog wordt en meet hoe lang het hoog blijft, in microseconden. Met een time-out van 25 ms (4 m heen en terug) blijft het programma nooit hangen als er geen echo komt:

```
unsigned long us = pulseIn(ECHO, HIGH, 25000UL);   // 0 als er geen echo kwam
float v = 331.3 + 0.606 * luchtT;                  // m/s
float cm = us * 1e-6 * v / 2.0 * 100.0;
```



De 40 kHz-puls duurt 200 µs; de breedte van ECHO (2,33 ms) is de tijd heen en terug.



Tussen 5 en 35 °C verandert de geluidssnelheid met ongeveer 5 %.

UITGEWERKT VOORBEELD · 33.12

Een echo omrekenen

12/18

De HC-SR04 meet een echo van 1,75 ms. (a) Hoe ver staat het voorwerp bij 20 °C? (b) Het is 35 °C in de module, maar de code rekent met 343,4 m/s. Hoe groot is de fout? (c) Het robotje rijdt 0,20 m/s en meet om de 60 ms. Hoeveel rijdt het tussen twee metingen?

GEGEVEN

- $t = 1{,}75\,\text{ms}$
- $v = 331{,}3 + 0{,}606\,T$
- meten om de 60 ms, 0,20 m/s

GEVRAAGD

- afstand
- fout bij 35 °C
- weg tussen twee metingen

OPLOSSING

1. (a) $s = \dfrac{343{,}4 \cdot 0{,}00175}{2} = 0{,}300\,\text{m}$.
2. (b) Bij 35 °C: $v = 331{,}3 + 0{,}606 \cdot 35 = 352{,}5\,\text{m/s}$; de echte afstand is $\dfrac{352{,}5 \cdot 0{,}00175}{2} = 0{,}308\,\text{m}$. De code geeft 0,300 m: 8 mm te weinig, of 2,6 %.
3. (c) $0{,}20\,\text{m/s} \cdot 0{,}060\,\text{s} = 0{,}012\,\text{m}$: 12 mm per meting.

ANTWOORD

(a) 0,300 m. (b) ongeveer 8 mm (2,6 %) te kort. (c) 12 mm.

Klopt dit? Een fout van 2,6 % op 30 cm is 8 mm: klein tegenover de marge die het robotje neemt (het stopt op 25 cm). Voor het stoppen is de temperatuurcorrectie dus niet nodig, voor een meting van de gang wel nuttig.

UITLEG · 33.13

Wat een ultrasone sensor niet ziet

13/18

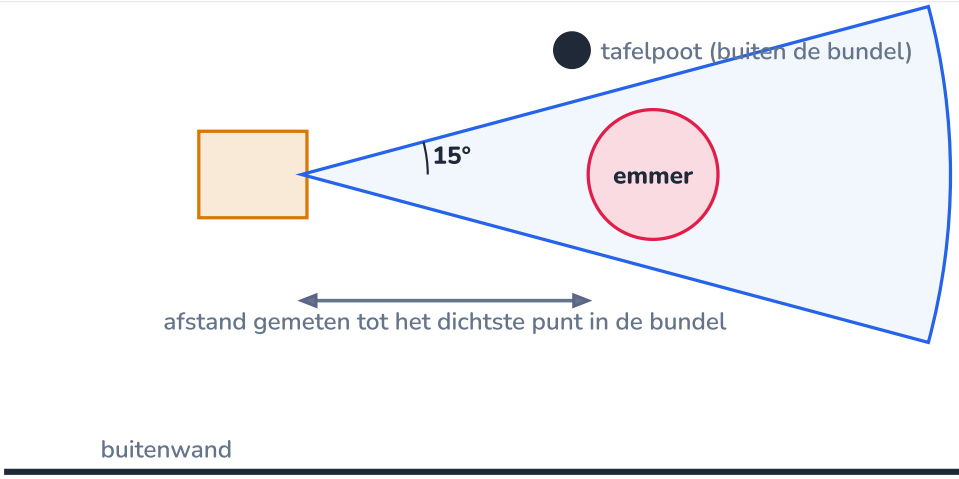
De HC-SR04 meet van 2 cm tot ongeveer 4 m, maar niet alles:

- **Een bundel, geen straal.** Het geluid waaiert uit (de datasheet noemt een meethoek van 15°; in de simulatie rekenen we met 15° naar elke kant). Alles in die kegel kan een echo geven, en de sensor meldt de dichtste. Een smalle tafelpoot net naast de kegel ziet hij niet.
- **Schuine en zachte oppervlakken.** Een schuine wand kaatst het geluid weg, weg van de sensor; zachte bladeren en stof slorpen het op. Het robotje ziet een krop sla slecht, een emmer goed.
- **Te snel na elkaar meten.** Een late echo van de vorige meting kan als nieuwe echo tellen. Daarom meet het robotje hoogstens om de 60 ms.
- **Andere sensoren.** Twee robots met een HC-SR04 in dezelfde ruimte kunnen elkaars puls horen.

Het robotje stopt daarom niet op 'net niet', maar met marge: bij minder dan 25 cm staat het stil, en het rijdt pas verder boven 30 cm. Dat verschil is **hysterese** (hoofdstuk 31): zonder dat zou het robotje rond 25 cm schokkend stoppen en vertrekken, want de meting schommelt een paar millimeter.

Een robot tussen mensen

Een robotje van 0,8 kg aan 0,20 m/s doet niemand pijn, maar een voet of een knie die het niet ziet, kan het wel laten vallen of vastlopen. Traag rijden, stoppen met marge, en nooit rijden in de modus ONDERHOUD van de module (hoofdstuk 32) zijn eenvoudige regels.



Alles in de kegel geeft een echo; de sensor meldt de kortste afstand.

UITLEG · 33.14

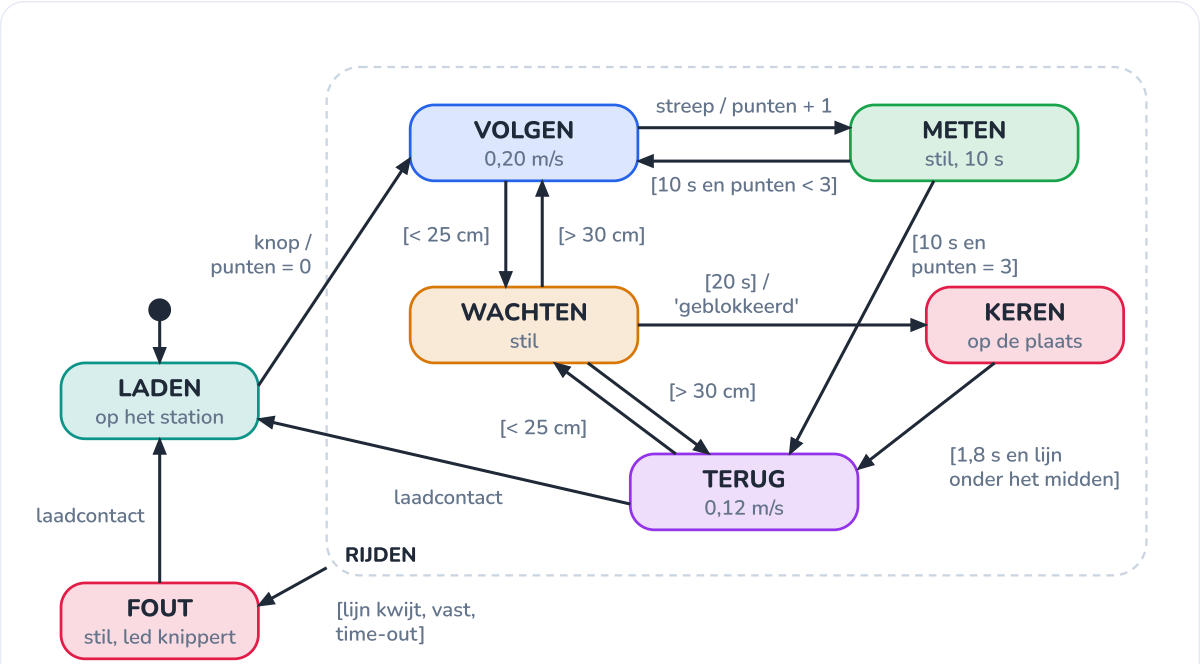
Het gedrag als toestandsmachine

14/18

Een lijn volgen en een afstand meten is niet genoeg: het robotje moet ook weten wat het wil. Dat is een toestandsmachine (hoofdstuk 32) met zeven toestanden. RIJDEN is een samengestelde toestand: uit elk van de vijf binnenste toestanden leidt een fout (lijn 1,5 s kwijt, vast, een time-out) naar FOUT.

- **LADEN**: op het laadstation. Een druk op de knop (of later een opdracht van de module) start een ronde.
- **VOLGEN**: lijn volgen aan 0,20 m/s met de P-regelaar. Een dwarsstreep (alle drie de sensoren zwart) is een meetpunt.
- **METEN**: 10 s stilstaan, meten en melden.
- **WACHTEN**: er staat iets op minder dan 25 cm. Wordt het vrij, dan verder. Blijft het 20 s, dan is de gang geblokkeerd.
- **KEREN**: op de plaats draaien tot de lijn weer onder de middelste sensor ligt, maar niet voor 1,8 s voorbij is: anders ziet het robotje meteen de lijn waar het al op stond.
- **TERUG**: trager naar het laadstation; dwarsstrepen tellen niet meer.
- **FOUT**: stil, de led knippert, tot iemand het robotje op het laadstation zet.

Merk op hoe dezelfde meting in verschillende toestanden iets anders betekent. Een dwarsstreep in VOLGEN is een meetpunt, in TERUG niets. Een afstand van 20 cm in VOLGEN betekent stoppen; in KEREN wordt ze niet eens bekeken.



Het robotje heeft zeven toestanden; alle fouten tijdens het rijden leiden naar FOUT.

CODE · 33.15

robot.ino: het robotje in C++

15/18

Het programma volgt het patroon van hoofdstuk 32: een `enum`, één toestandsvariabele, `naar()` met acties bij binnenkomen, en een `switch` die 50 keer per seconde de gebeurtenissen bekijkt. De afstand wordt om de 60 ms gemeten, los van het regelen. De rechtermotor is gespiegeld gemonteerd; in de bedrading zijn daarom zijn twee motordraden verwisseld, zodat 'vooruit' voor beide motoren hetzelfde betekent.

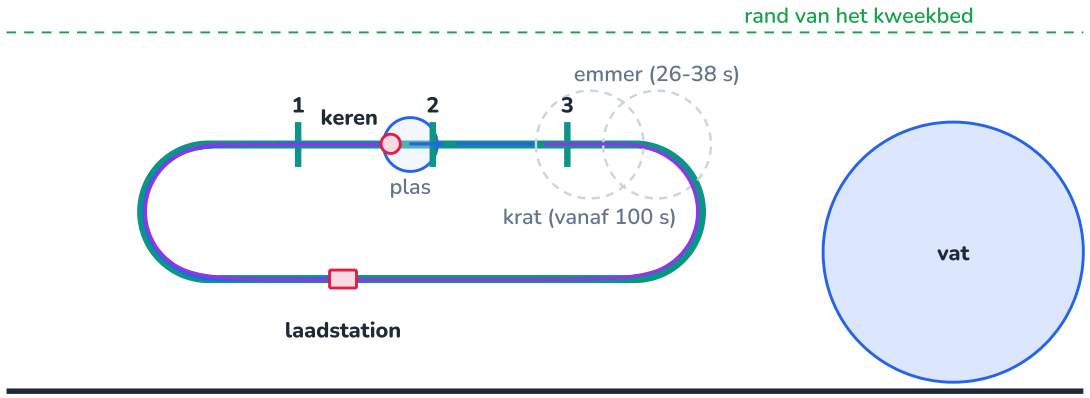
C++ (Arduino) **robot.ino** 183 regels

```cpp
// robot.ino - het inspectierobotje van de klimaatmodule (hoofdstuk 33)
// Arduino Uno, motordriver TB6612FNG, drie reflectiesensoren (A0-A2), lekdetectie (A3), HC-SR04, SHT31 op een mastje.
#include <Wire.h>
#include <Adafruit_SHT31.h>

// ---------------------------------------------------------------- pinnen
const int TRIG = 2, ECHO = 3, STBY = 4;
const int PWM_L = 5, IN1_L = 7, IN2_L = 8;          // linkermotor (kanaal A)
const int PWM_R = 6, IN1_R = 9, IN2_R = 10;         // rechtermotor (kanaal B)
const int KNOP = 11, LAAD = 12, LED = 13;           // startknop, laadcontacten, statusled
const int REFL_L = A0, REFL_M = A1, REFL_R = A2, LEK = A3;

// ---------------------------------------------------------------- instellingen
const float B_SPOOR = 0.14;                         // m tussen de twee wielen
const float V_MAX = 0.60;                           // m/s bij PWM 255
const float D0 = 0.18;                              // dode zone: onder 18 % PWM draait het wiel niet
const float V_RIJ = 0.20, V_TERUG = 0.12, V_DRAAI = 0.10;  // m/s
const float K_P = 0.25;                             // rad/s per mm dat de lijn opzij ligt
const int WIT = 100, ZWART = 850;                   // ADC-waarden na het ijken
const float STOP_CM = 25.0, VRIJ_CM = 30.0;         // hysterese op de afstand
const unsigned long REGEL_MS = 20, SONAR_MS = 60;
const unsigned long METEN_MS = 10000, WACHT_MAX_MS = 20000, KWIJT_MS = 1500;
const unsigned long DRAAI_MIN_MS = 1800, DRAAI_MAX_MS = 5000;

enum Toestand { LADEN, VOLGEN, METEN, WACHTEN, KEREN, TERUG, FOUT };
const char* NAAM[] = {"LADEN", "VOLGEN", "METEN", "WACHTEN", "KEREN", "TERUG", "FOUT"};

Adafruit_SHT31 sht;
```

```
Toestand toestand = LADEN, naWachten = VOLGEN;
unsigned long sinds = 0, vorigeRegel = 0, vorigeSonar = 0, kwijtSinds = 0;
int punten = 0, kant = 1;                                  // meetpunten deze ronde; waar de lijn laatst lag
bool opStreep = false, kwijt = false;
float afstand = 999.0, luchtT = 20.0;

// ---------------------------------------------------------------- motoren (TB6612FNG)
void zetMotor(int pwm, int in1, int in2, float v) {        // v in m/s, negatief = achteruit
  float f = min(fabs(v) / V_MAX, 1.0);
  int d = (v == 0.0) ? 0 : (int)(255 * (D0 + (1.0 - D0) * f) + 0.5);   // de dode zone overbruggen
  digitalWrite(in1, v > 0 ? HIGH : LOW);
  digitalWrite(in2, v < 0 ? HIGH : LOW);
  analogWrite(pwm, d);
}

void rijden(float vL, float vR) {
  zetMotor(PWM_L, IN1_L, IN2_L, vL);
  zetMotor(PWM_R, IN1_R, IN2_R, vR);
}

// ---------------------------------------------------------------- sensoren
float wit01(int pin) {                                     // 0 = wit, 1 = zwart
  return constrain((analogRead(pin) - WIT) / (float)(ZWART - WIT), 0.0, 1.0);
}

float lijnPositie(bool& streep) {                          // mm, + = de lijn ligt links; NAN = kwijt
  float wl = wit01(REFL_L), wm = wit01(REFL_M), wr = wit01(REFL_R);
  streep = wl > 0.6 && wm > 0.6 && wr > 0.6;               // dwarsstreep: alle drie zwart
  float g = wl + wm + wr;
  if (g < 0.2) return NAN;
  return (wl * 15.0 - wr * 15.0) / g;                      // gewogen gemiddelde van +15, 0 en -15 mm
}

float meetAfstand() {                                      // HC-SR04: s = v * t / 2
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  unsigned long us = pulseIn(ECHO, HIGH, 25000UL);         // na 25 ms (4 m heen en terug) opgeven
  if (us == 0) return 999.0;                               // geen echo: niets binnen bereik
  float v = 331.3 + 0.606 * luchtT;                        // geluidssnelheid in m/s
  return us * 1e-6 * v / 2.0 * 100.0;                      // cm
}

// ---------------------------------------------------------------- toestanden
void melding(const char* tekst) {
  Serial.print(millis() / 1000.0, 1);
  Serial.print(" s  ");
  Serial.println(tekst);
}

void meetpunt() {                                          // temperatuur, vocht en lekdetectie
  float t, rv;
  sht.readBoth(&t, &rv);
  if (!isnan(t)) luchtT = t;
  Serial.print(millis() / 1000.0, 1);
  Serial.print(" s  meetpunt ");
  Serial.print(punten);
  Serial.print(": ");
  Serial.print(t, 1);
  Serial.print(" C, ");
  Serial.print(rv, 0);
  Serial.println(analogRead(LEK) < 500 ? " %, NAT: lek onder de goot!" : " %, droog");
}

void naar(Toestand nieuw, unsigned long nu) {
  toestand = nieuw;
```

```
  sinds = nu;
  melding(NAAM[nieuw]);
  switch (nieuw) {                                  // acties bij binnenkomen
    case KEREN:  rijden(-V_DRAAI, V_DRAAI); break;  // op de plaats draaien (tegen de klok in)
    case METEN:  rijden(0, 0); meetpunt(); break;
    case VOLGEN:
    case TERUG:  kwijt = false; break;
    default:     rijden(0, 0); break;               // LADEN, WACHTEN, FOUT: stilstaan
  }
}

void volg(float pos, float v0) {                    // P-regelaar: draai naar de lijn toe
  if (isnan(pos)) pos = 15.0 * kant;                // kwijt: zoek aan de kant waar ze laatst lag
  else kant = pos > 0 ? 1 : -1;
  float omega = K_P * pos;                          // rad/s
  rijden(v0 - omega * B_SPOOR / 2, v0 + omega * B_SPOOR / 2);
}

bool lijnKwijt(float pos, unsigned long nu) {
  if (!isnan(pos)) { kwijt = false; return false; }
  if (!kwijt) { kwijt = true; kwijtSinds = nu; }
  return nu - kwijtSinds > KWIJT_MS;
}

void setup() {
  int uit[] = {TRIG, STBY, PWM_L, IN1_L, IN2_L, PWM_R, IN1_R, IN2_R, LED};
  for (int p : uit) pinMode(p, OUTPUT);
  pinMode(ECHO, INPUT);
  pinMode(KNOP, INPUT_PULLUP);
  pinMode(LAAD, INPUT);
  digitalWrite(STBY, HIGH);                         // motordriver aan
  Serial.begin(115200);
  Wire.begin();
  sht.begin(0x44);
  naar(digitalRead(LAAD) == HIGH ? LADEN : FOUT, millis());   // niet op het laadstation: eerst terugzetten
}

void loop() {
  unsigned long nu = millis();
  if (nu - vorigeSonar >= SONAR_MS) {               // hoogstens om de 60 ms meten (datasheet)
    vorigeSonar += SONAR_MS;
    afstand = meetAfstand();
  }
  if (nu - vorigeRegel < REGEL_MS) return;          // 50 keer per seconde beslissen
  vorigeRegel += REGEL_MS;
  bool streep;
  float pos = lijnPositie(streep);
  bool nieuweStreep = streep && !opStreep;
  opStreep = streep;
  unsigned long t = nu - sinds;
  digitalWrite(LED, toestand == FOUT ? (nu / 200) % 2 : toestand != LADEN);

  switch (toestand) {
    case LADEN:
      if (digitalRead(KNOP) == LOW) { punten = 0; naar(VOLGEN, nu); }
      break;
    case VOLGEN:
      if (afstand < STOP_CM) { naWachten = VOLGEN; naar(WACHTEN, nu); }
      else if (nieuweStreep) { punten++; naar(METEN, nu); }
      else if (lijnKwijt(pos, nu)) naar(FOUT, nu);
      else volg(pos, V_RIJ);
      break;
    case METEN:
      if (t >= METEN_MS) naar(punten >= 3 ? TERUG : VOLGEN, nu);
      break;
    case WACHTEN:
      if (afstand > VRIJ_CM && t >= 1000) naar(naWachten, nu);
      else if (t >= WACHT_MAX_MS) {
```

```
        if (naWachten == VOLGEN) { melding("gang geblokkeerd: omkeren"); naar(KEREN, nu); }
        else { melding("vast: de weg terug is ook geblokkeerd"); naar(FOUT, nu); }
      }
      break;
    case KEREN:
      if (t >= DRAAI_MIN_MS && wit01(REFL_M) > 0.6) naar(TERUG, nu);   // lijn weer onder het midden
      else if (t >= DRAAI_MAX_MS) naar(FOUT, nu);
      break;
    case TERUG:                                          // trager, dwarsstrepen tellen niet meer
      if (digitalRead(LAAD) == HIGH) naar(LADEN, nu);
      else if (afstand < STOP_CM) { naWachten = TERUG; naar(WACHTEN, nu); }
      else if (lijnKwijt(pos, nu)) naar(FOUT, nu);
      else volg(pos, V_TERUG);
      break;
    case FOUT:                                           // stil en knipperen tot iemand hem op het laadstation zet
      if (digitalRead(LAAD) == HIGH) naar(LADEN, nu);
      break;
  }
}
```



Het gelogde pad uit de test. Blauw: VOLGEN, paars: TERUG, rood: KEREN. De cijfers zijn de meetpunten.

Getest in een gesimuleerde gang: de sensoren krijgen hun waarden uit de plaats van het robotje, de echo uit de afstand tot de wanden, het vat en wat er in de weg staat. In ronde 1 staat er van 26 tot 38 s een emmer op de lijn, en ligt er een plas bij het tweede meetpunt. In ronde 2 blijft er een krat staan. De seriële monitor:

```
0.0 s  LADEN
2.0 s  VOLGEN
6.5 s  METEN
6.5 s  meetpunt 1: 20.0 C, 68 %, droog
16.5 s  VOLGEN
18.0 s  METEN
18.0 s  meetpunt 2: 20.0 C, 68 %, NAT: lek onder de goot!
28.0 s  VOLGEN
28.6 s  WACHTEN
38.0 s  VOLGEN
39.0 s  METEN
39.1 s  meetpunt 3: 20.0 C, 68 %, droog
49.0 s  TERUG
59.8 s  LADEN
110.0 s  VOLGEN
114.5 s  METEN
114.5 s  meetpunt 1: 20.0 C, 68 %, droog
124.5 s  VOLGEN
```

```
125.8 s  WACHTEN
145.8 s  gang geblokkeerd: omkeren
145.8 s  KEREN
148.0 s  TERUG
157.6 s  LADEN
```

Ronde 1 duurt 57,8 s, inclusief 30 s meten en 9 s wachten. In ronde 2 wacht het robotje 20 s voor de krat, keert om op de plaats (2,2 s) en rijdt tegen de draairichting in terug naar het laadstation, via de andere bocht. Het midden van de as bleef tijdens het volgen overal binnen 1 cm van de lijn.

UITLEG · 33.16

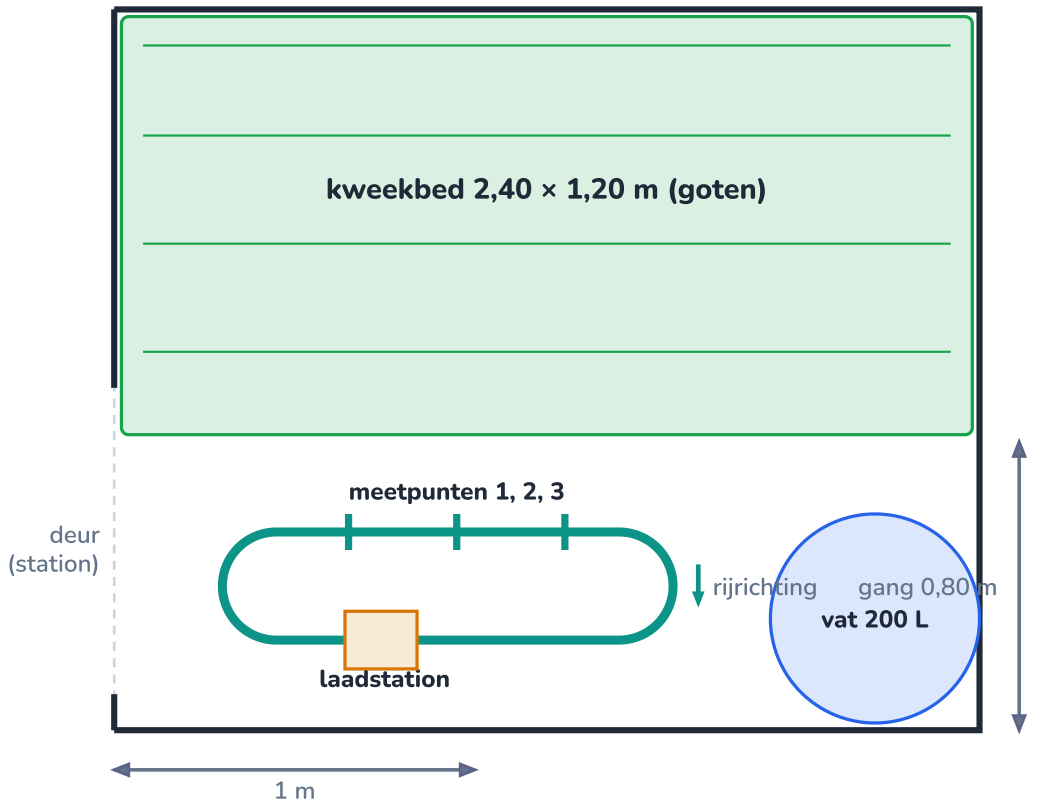
Habitat: de route in de module

16/18

In de klimaatmodule ligt tussen de buitenwand en het kweekbed een gang van 0,80 m breed. Daarin loopt een lus van zwarte tape (19 mm) met twee rechte stroken (strook A langs het kweekbed, strook B langs de wand) en twee bochten met een straal van 0,15 m. Op strook A liggen drie dwarsstrepen: de meetpunten. Op strook B staat het laadstation, met contacten in de vloer. Het vat van 200 L staat in de hoek: de lus blijft er bijna 30 cm van weg, zodat de ultrasone sensor het in de bocht niet als obstakel ziet.

De lus is $2 \cdot 0{,}95 + 2\pi \cdot 0{,}15 = 2{,}84\,\text{m}$ lang. Een ronde (heen aan 0,20 m/s, terug aan 0,12 m/s, drie keer 10 s meten) duurt ongeveer 50 s en kost ongeveer 0,024 Wh. Overdag elk uur een ronde (16 per dag) is 0,4 Wh. Tussen de rondes staat het robotje op zijn laadstation; daar gebruikt vooral de Uno stroom (±0,3 W), en die levert het laadstation. Wat het robotje meet, stuurt het later draadloos naar de module (Expert, hoofdstuk 37); voorlopig schrijft het elke meting naar zijn seriële poort.

Waarom een lijn en geen vrije route? Een robotje dat zijn plaats alleen uit de omwentelingen van zijn wielen berekent (**odometrie**), dwaalt af: een wiel dat 1 % groter is dan het andere, laat het na 10 m rijden 41° scheef staan. De lijn verbetert die fout bij elke meting. Zonder lijn heb je andere sensoren nodig en een slimme combinatie ervan (Expert, hoofdstuk 40).



De lus in de gang, de drie meetpunten, het laadstation en het vat.

NASLAG · 33.17

Formules en gegevens van het robotje

17/18

differentiële aandrijving

$$v = \frac{v_L + v_R}{2} \qquad \omega = \frac{v_R - v_L}{b} \qquad R = \frac{v}{\omega}$$

symbool	betekenis	eenheid
v_L, v_R	snelheid van linker- en rechterwiel	m/s
b	spoorbreedte	m
ω	hoeksnelheid	rad/s

overbrenging en wielsnelheid

$$i = \frac{z_2}{z_1} = \frac{n_1}{n_2} \qquad v = \frac{\pi\, d\, n}{60}$$

symbool	betekenis	eenheid
n	toerental	omw/min
d	wieldiameter	m

P-lijnvolger

$$p = \frac{15\, w_L - 15\, w_R}{w_L + w_M + w_R} \qquad \omega = K \cdot p$$

symbool	betekenis	eenheid
w	zwartheid van een sensor (0 tot 1)	–
p	plaats van de lijn	mm
K	versterking	rad/(s·mm)

afstand uit een echo

$$s = \frac{v\, t}{2} \qquad v = 331{,}3 + 0{,}606\, T$$

symbool	betekenis	eenheid
t	tijd heen en terug	s
T	temperatuur	°C

plaats bijhouden (simulatie, odometrie)

$$x \leftarrow x + v \cos\theta\, \Delta t \qquad y \leftarrow y + v \sin\theta\, \Delta t \qquad \theta \leftarrow \theta + \omega\, \Delta t$$

symbool	betekenis	eenheid
θ	rijrichting	rad

onderdeel	waarde
maten	200 × 160 mm, b = 140 mm, wielen Ø 65 mm, ±0,8 kg
motoren	6 V, tandwielkast 1:48; wiel 187,5 omw/min onbelast; dode zone PWM 46

onderdeel	waarde
snelheden	volgen 0,20 m/s (PWM 116), terug 0,12 m/s, draaien 0,10 m/s per wiel
pinnen (Uno)	TRIG D2, ECHO D3, STBY D4, motor links D5/D7/D8, rechts D6/D9/D10, knop D11, laadcontact D12, led D13, sensoren A0-A2, lek A3, SHT31 op A4/A5 (I²C)
regelaar	K = 0,25 rad/s per mm, 50 keer per seconde
afstand	stop onder 25 cm, verder boven 30 cm, meten om de 60 ms
batterij	2 × LiFePO4, 6,4 V, ±1,5 Ah

HET GEHEEL · 33.18

Waar robotica vandaan komt en waar ze heen gaat

18/18

Een mobiele robot is een klein systeem waarin bijna alle leerlijnen samenkomen: krachten en beweging (de wielen, de overbrenging), elektronica (sensoren, H-brug, PWM), programmeren (taken, toestandsmachines), regelen (de lijnvolger) en ontwerpen (maten, veiligheid). In Expert leert het robotje zijn eigen weg zoeken en zijn plaats schatten met meerdere sensoren.

F12 Hefbomen en mechanismen
Tandwielen en de overbrengingsverhouding: de tandwielkast van de motoren.

I7 Beweging beschrijven
Snelheid, plaats en tijd: de plaats van de robot stap voor stap bijhouden.

I9 Arbeid, energie en vermogen
Vermogen = kracht × snelheid; rendement van een kleine motor.

I22 De spanningsdeler en sensoren
De reflectiesensor is een spanningsdeler met een fototransistor.

I25 Van analoog naar digitaal
PWM stuurt de snelheid van de motoren.

I26 Motoren en actuatoren
Gelijkstroommotor, koppel en de H-brug.

I31 Regelsystemen
Aan/uit tegenover evenredig, en hysterese op de afstand.

I32 Toestandsmachines
Het gedrag van het robotje als toestandsmachine met veilige toestand.

E34 Algoritmen en complexiteit
De kortste route door het station voor de inspectierobot.

E40 Sensorfusie en schatten
De plaats schatten uit wielen, lijn en afstandssensor samen.

E39 PID-regeling
Een PID-lijnvolger: sneller en zonder blijvende afwijking in de bocht.

Habitat
De klimaatmodule krijgt een inspectierobotje: een lus in de gang met drie meetpunten langs het kweekbed, een laadstation, en een gedrag dat stopt voor obstakels en omkeert als de gang geblokkeerd is. In het integratieproject (hoofdstuk 40) koppel je zijn metingen aan het logboek van de module; in Expert zoekt het zijn weg door het hele station (hoofdstuk 34).

SAMENVATTING

In het kort

- Met twee aangedreven wielen: $v = (v_L + v_R)/2$, $\omega = (v_R - v_L)/b$ en $R = v/\omega$; tegengesteld draaien geeft $R = 0$.
- De tandwielkast maakt het toerental i keer kleiner en het koppel bijna i keer groter; $v = \pi d n/60$.
- Een motor heeft een dode zone; de code moet ze overbruggen om kleine snelheden te kunnen rijden.

- Een reflectiesensor is een spanningsdeler met een fototransistor; ijk op wit en zwart en werk met 0 tot 1.
- Een P-lijnvolger draait evenredig met hoe ver de lijn opzij ligt; te kleine K mist bochten, te grote K slingert.
- Een ultrasone sensor meet $s = vt/2$ met $v = 331{,}3 + 0{,}606\,T$; hij ziet een kegel, geen punt.
- Het gedrag van een robot is een toestandsmachine; dezelfde meting betekent iets anders in een andere toestand.
- Het inspectierobotje rijdt een lus van ±3 m in de gang, meet op drie punten en keert terug naar zijn laadstation.

BEGRIPPEN

Wat je nu kent

differentiële aandrijving
Twee wielen met elk een eigen motor; de robot stuurt door de wielen verschillend snel te laten draaien.

hoeksnelheid
Hoe snel een voorwerp draait, in radialen per seconde (ω).

draaistraal
De straal R van de cirkel die het midden van de as beschrijft; R = v/ω.

dode zone
Het gebied van kleine stuursignalen waarin een motor nog niet draait, omdat de wrijving te groot is.

reflectiesensor
Een IR-led en een fototransistor naast elkaar: hoe meer licht de vloer terugkaatst, hoe meer stroom de transistor geleidt.

lijnvolger
Een robot die met sensoren onder zijn neus een lijn op de vloer volgt.

versterking
Bij een P-regelaar: hoeveel harder de robot draait per millimeter dat de lijn opzij ligt.

echo
Geluid dat terugkaatst op een voorwerp; de tijd heen en terug geeft de afstand.

odometrie
De plaats van een robot bijhouden uit hoeveel elk wiel gedraaid heeft.

I INTERMEDIATE · DEEL I9 · LEVEN EN BIOTECHNOLOGIE

HANDBOEK · HOOFDSTUK 34

Enzymen en stofwisseling

Waarom een cel niet ontploft en niet stilvalt: katalysatoren met een optimum, en de weg van licht naar ATP

In elke cel lopen duizenden reacties tegelijk, bij lichaamstemperatuur en in water. Zonder hulp zouden de meeste ervan jaren duren. Die hulp komt van **enzymen**: eiwitten die een reactie miljoenen keren versnellen zonder zelf op te raken. In dit hoofdstuk zie je hoe ze werken, waarom ze een optimale temperatuur en pH hebben, hoe de energie van het licht via suiker in **ATP** terechtkomt, en voor Habitat: waarom de sla het best groeit rond 20 °C en 's nachts beter wat koeler staat.

10× uitleg | 5× uitgewerkt voorbeeld | 1× proef | 1× ontleed | 1× naslag | 2× verhaal | 1× het geheel | 1× code | ± 9 lestijden

B Leven en systemen | M Materie en chemie

Een schijfje aardappel in waterstofperoxide: het enzym katalase maakt in seconden wat anders maanden duurt.

Na dit hoofdstuk

- Je kan uitleggen wat een katalysator doet met de activeringsenergie, en waarom een enzym niet opgebruikt wordt.
- Je kan de werking van een enzym beschrijven met het actieve centrum, het enzym-substraatcomplex en de specificiteit.
- Je kan het temperatuuroptimum van een enzym verklaren met twee effecten, en rekenen met de vuistregel Q10 ≈ 2.
- Je kan uitleggen waarom een enzym een pH-optimum heeft, en pH-grafieken van enzymen lezen.
- Je kan de betekenis van Vmax en Km uitleggen en ze uit een grafiek van de snelheid tegenover de substraatconcentratie halen.
- Je kan fotosynthese, celademhaling en gisting als kloppende reactievergelijking schrijven en vergelijken.
- Je kan de energieketen van licht tot ATP beschrijven en uitleggen waarom ATP een energiedrager is.
- Je kan een enzymproef opzetten met één veranderlijke, en de resultaten verklaren.
- Je kan met een model uitleggen waarom de sla in de klimaatmodule een temperatuuroptimum heeft en 's nachts koeler mag.

UITLEG · 34.1

Wat je al weet, en wat nu komt

In F27 leerde je dat een plant met licht suiker maakt uit koolstofdioxide en water, en dat elke cel die suiker weer 'verbrandt' om energie te krijgen. In hoofdstuk 14 schreef je die reacties als kloppende vergelijkingen, en zag je dat een reactie eerst over een drempel moet: de activeringsenergie. Glucose verbrandt in een vlam bij honderden graden. Jouw spiercellen doen hetzelfde bij 37 °C, in water, zonder vlam. Hoe kan dat?

wat je al weet	waar	wat er nu bij komt
cellen met bladgroenkorrels en mitochondriën	F26	wat er in die korrels precies gebeurt, stap voor stap
fotosynthese en celademhaling als woordvergelijking	F27	als kloppende reactievergelijking, met energie
temperatuur heeft een optimum voor de groei	F27	waarom: enzymen, met een model
activeringsenergie, exotherm en endotherm; gist ontbindt waterstofperoxide	hoofdstuk 14	het enzym katalase en hoe een katalysator werkt
pH, zuren en basen	hoofdstuk 15	waarom een enzym bij de verkeerde pH stilvalt
energie uit voeding, 10 MJ per dag	F29	hoe die energie in je cellen als ATP terechtkomt
de temperatuurregeling van de module: 18/19 °C overdag, 16/17 °C 's nachts	hoofdstuk 31	waarom die grenzen goed zijn voor de sla

In dit hoofdstuk

- leer je hoe **enzymen** reacties versnellen, en waarom elk enzym maar één soort reactie doet;
- verklaar je met een **model** waarom een enzym een optimale temperatuur en pH heeft, en meet je dat zelf met katalase;
- zie je wat er gebeurt als er meer substraat is: de snelheid stijgt, maar niet eindeloos;
- volg je de energie van het licht via glucose tot in **ATP**, de 'batterij' van elke cel;
- reken je uit hoeveel ATP een bewoner van het station per dag maakt, en wat hij eigenlijk verbrandt;
- verklaar je voor Habitat de temperaturen waarop de klimaatmodule de sla houdt.

Een spiercel en een bladcel draaien op dezelfde soort chemie: enzymen, suiker en ATP.

VERHAAL · 34.2

Gist zonder gist, en een kromme uit Berlijn

2/22

In 1833 haalden de Franse chemici **Anselme Payen** en **Jean-François Persoz** uit kiemende gerst een stof die zetmeel omzet in suiker. Ze noemden ze *diastase*; vandaag heet ze amylase. Het was de eerste keer dat iemand zo'n werkzame stof uit een levend wezen apart hield. Maar bij de gisting van suiker tot alcohol lukte dat niet. **Louis Pasteur** toonde rond 1860 dat gisting het werk is van levende gistcellen, en velen besloten dat daar een 'levenskracht' voor nodig was die buiten een levende cel niet bestaat. De Duitse fysioloog **Wilhelm Kühne** gaf de werkzame stoffen in 1877 een naam: *enzym*, Grieks voor 'in de gist'.

In 1897 maalde **Eduard Buchner** in München gistcellen fijn met kwartszand en werd er met een pers een sap uit geperst, zonder één levende cel. Toen hij er veel suiker aan toevoegde (om het sap te bewaren, zoals je fruit in suiker bewaart), begon het te borrelen: koolstofdioxide en alcohol. Gisting zonder gist. Het waren de enzymen in het sap die het werk deden, niet het leven zelf. Buchner kreeg er in 1907 de Nobelprijs voor scheikunde voor. In 1926 kristalliseerde de Amerikaan **James Sumner** het enzym urease en toonde hij dat enzymen **eiwitten** zijn.

Intussen wilden onderzoekers weten *hoe snel* een enzym werkt. In 1913 publiceerden **Leonor Michaelis** en de Canadese arts **Maud Menten** in Berlijn hun metingen met invertase, het gistenzym dat sacharose splitst. Ze toonden dat de snelheid eerst evenredig stijgt met de hoeveelheid suiker en dan afvlakt: het enzym raakt verzadigd. Die kromme, en de twee getallen die ze beschrijven, gebruik je in dit hoofdstuk nog steeds.

Waarom dit verhaal ertoe doet

Dat een levend proces uit gewone chemie bestaat die je in een reageerbuis kan nabootsen, was in 1897 een schok. Het is de basis van de biotechnologie: bakkers, brouwers, wasmiddelen, geneesmiddelen (hoofdstuk 35), en de PCR-test die met een enzym uit een hete bron werkt.

Gist, zand en een pers: met zo'n opstelling maakte Buchner een gistsap zonder levende cellen dat toch gistte.

UITLEG · 34.3

Enzymen: katalysatoren van het leven

Een **katalysator** is een stof die een reactie versnelt en er zelf onveranderd uitkomt. Ze doet dat door een andere weg aan te bieden, met een **lagere activeringsenergie** (hoofdstuk 14). Meer botsingen zijn dan krachtig genoeg om de reactie te laten doorgaan. Wat de katalysator *niet* verandert: de beginstoffen, de producten en de energie die vrijkomt of opgenomen wordt. De drempel wordt lager, het hoogteverschil blijft.

Enzymen zijn de katalysatoren van levende wezens. Het zijn eiwitten: lange ketens van aminozuren die zich opvouwen tot een precieze ruimtelijke vorm. Enzymen zijn verbazend snel. Katalase, dat het giftige waterstofperoxide in je cellen afbreekt tot water en zuurstof, verwerkt tot tientallen miljoenen moleculen per seconde. Zonder katalase ontbindt waterstofperoxide zo traag dat een fles uit de apotheek maandenlang meegaat.

DE REACTIE VAN KATALASE

$$2\,H_2O_2 \longrightarrow 2\,H_2O + O_2$$

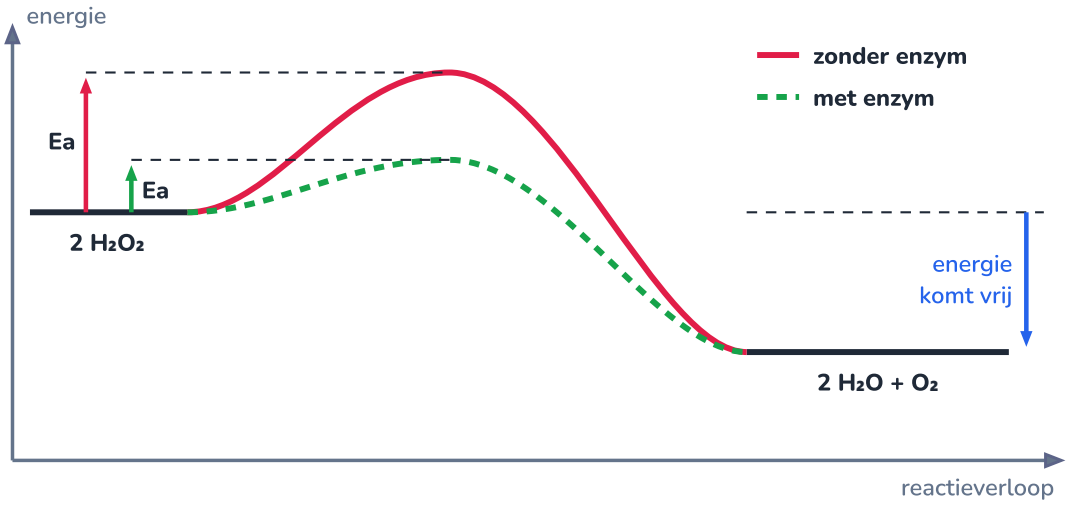
Exotherm. Zonder katalysator verloopt ze bij kamertemperatuur heel traag.

De namen van de meeste enzymen eindigen op **-ase**, en zeggen vaak wat ze doen:

enzym	waar	substraat → producten
katalase	bijna alle cellen (lever, aardappel, gist)	waterstofperoxide → water + zuurstof
amylase	speeksel, alvleesklier, kiemend graan	zetmeel → maltose (een suiker)
pepsine	maag	eiwitten → stukken eiwit
lactase	dunne darm	lactose (melksuiker) → glucose + galactose
invertase	gist	sacharose → glucose + fructose
Rubisco	bladgroenkorrels	legt CO_2 vast in de Calvincyclus
ATP-synthase	mitochondriën, bladgroenkorrels	ADP + fosfaat → ATP

Een enzym wordt verbruikt

Een enzymmolecule doet dezelfde reactie keer op keer. Het wordt niet opgebruikt zoals een beginstof. Wel gaat het na verloop van tijd kapot (door warmte, zuur, beschadiging) en maakt de cel nieuwe. Een cel heeft dus maar weinig enzym nodig voor veel substraat.



Schets: met katalase is de drempel veel lager. Begin, einde en de energie die vrijkomt blijven gelijk.

UITLEG · 34.4

Slot en sleutel: het actieve centrum

4/22

Waarom doet katalase alleen iets met waterstofperoxide, en amylase alleen met zetmeel? In het opgevouwen eiwit zit een kuiltje of spleet: het **actieve centrum**. Alleen een molecule met de juiste vorm en de juiste ladingen op de juiste plaats past erin. Dat molecule heet het **substraat**. De Duitse chemicus Emil Fischer vergeleek het in 1894 met een **slot en een sleutel**.

1. Het substraat botst op het enzym en valt in het actieve centrum.
2. Het **enzym-substraatcomplex** ontstaat. Het enzym trekt aan de bindingen die moeten breken, of houdt twee moleculen precies zo tegen elkaar dat ze kunnen binden. Daardoor is minder energie

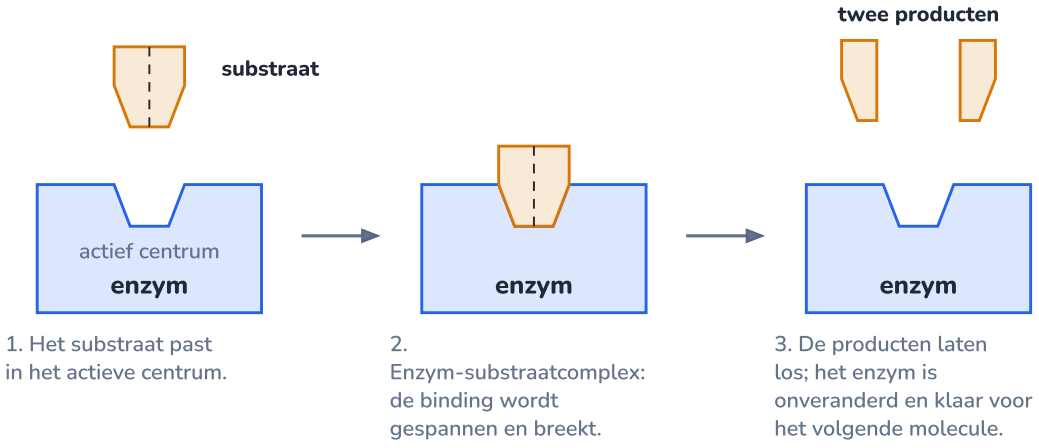
nodig.

3. De **producten** hebben een andere vorm, passen niet meer en laten los. Het enzym is klaar voor het volgende molecule.

Het slot is niet helemaal stijf: vaak sluit het enzym zich een beetje rond het substraat als dat binnenkomt (*induced fit*, Daniel Koshland, 1958). Het principe blijft: vorm bepaalt functie. Die **specificiteit** is essentieel. In één cel werken duizenden enzymen door elkaar, en toch doet elk precies zijn eigen reactie.

Vorm is alles

Alles wat de vorm van het enzym verandert, verandert zijn werking: te veel warmte, een te zure of te basische oplossing, zware metalen, of een molecule dat op het substraat lijkt en het actieve centrum bezet houdt (een **remmer**). Veel geneesmiddelen zijn zulke remmers: aspirine blokkeert bijvoorbeeld een enzym dat stoffen maakt die pijn en koorts veroorzaken.



Het substraat past in het actieve centrum, de binding breekt, de producten laten los. Het enzym blijft hetzelfde.

PROEF · 34.5

Katalase: hoeveel schuim bij welke temperatuur?

5/22

ONDERZOEKSVRAAG

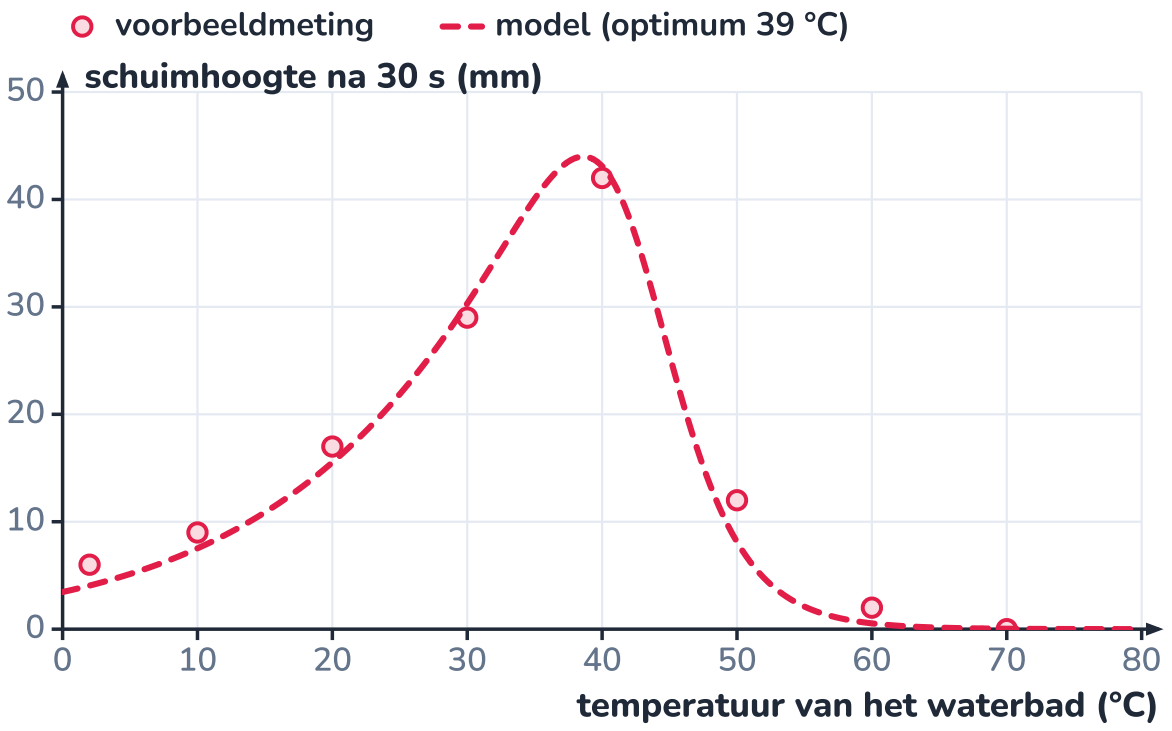
Rauwe aardappel bevat katalase. Hoe hangt de snelheid waarmee het enzym waterstofperoxide ontbindt af van de temperatuur?

MATERIAAL

- rauwe aardappel, dunschiller of mesje, kurkboor (of een lever in een afgesloten bakje, met handschoenen)
- waterstofperoxide 3 % (apotheek), afwasmiddel
- acht maatcilinders van 50 mL, liniaal, stopwatch
- waterbaden van ongeveer 2 (ijswater), 10, 20, 30, 40, 50, 60 en 70 °C, thermometer
- veiligheidsbril

Veiligheid

Bril op: het schuim spat. Waterstofperoxide van 3 % draagt volgens de Europese CLP-regels geen gevarenpictogram (pas vanaf 5 % 'oogirriterend', GHS07, en vanaf 8 % 'ernstig oogletsel', GHS05), maar het hoort niet in je ogen of mond. Spoel spatten meteen met veel water. Werk je met lever: handschoenen, een apart snijplankje, en achteraf handen wassen (rauw vlees kan ziekteverwekkers dragen, hoofdstuk 35).



De metingen stijgen tot ongeveer 40 °C en vallen dan snel terug; het model volgt dat verloop.

WERKWIJZE

1. Snijd met de kurkboor acht gelijke cilindertjes aardappel en snijd elk in vijf schijfjes van ongeveer 2 mm.
2. Doe in elke maatcilinder 5 mL waterstofperoxide en één druppel afwasmiddel (dat houdt het zuurstofgas vast als schuim).
3. Zet elke maatcilinder en een bakje met vijf schijfjes 5 minuten in een waterbad: telkens een andere temperatuur.
4. Laat de schijfjes in de cilinder vallen, start de stopwatch en lees na 30 s de hoogte van het schuim af boven de vloeistof.
5. Herhaal de hele reeks nog twee keer en neem per temperatuur het gemiddelde. Zet de metingen in een grafiek.

WAARNEMING

temperatuur (°C)	2	10	20	30	40	50	60	70
schuimhoogte na 30 s (mm)	6	9	17	29	42	12	2	0

Voorbeeldmeting, gemiddelde van drie reeksen. Tot 40 °C stijgt de schuimhoogte, daarboven daalt ze snel. Bij 70 °C schuimt het niet meer; gekookte aardappel schuimt ook na afkoelen niet meer.

VERKLARING

Onder het optimum werkt het enzym sneller naarmate het warmer is: de moleculen bewegen sneller, er zijn meer botsingen, en meer botsingen zijn krachtig genoeg. Tussen 10 en 30 °C verdrievoudigt de schuimhoogte ongeveer: per 10 °C wordt het ongeveer 1,8 keer zo veel. Boven ongeveer 40 °C verliest het enzym zijn vorm: het **denatureert**. Het actieve centrum past niet meer, en dat komt niet terug als je afkoelt. Het model (stippellijn) combineert beide effecten; hoe, zie je op het volgende blad. Let op: het gemeten optimum hangt af van hoe lang de schijfjes in het warme bad lagen (opdracht 34.17).

Acht temperaturen naast elkaar: de middelste cilinders schuimen het hoogst.

UITLEG · 34.6

Temperatuur: twee effecten, één optimum

6/22

Waarom heeft een enzym een optimale temperatuur? Omdat twee effecten tegen elkaar in werken.

1. Meer botsingen met genoeg energie. Hoe warmer, hoe sneller de moleculen bewegen, en hoe groter het deel van de botsingen dat over de drempel geraakt. De Zweedse chemicus Svante Arrhenius beschreef dat in 1889 met een exponentiële formule (hoofdstuk E25). Voor veel enzymreacties komt die neer op een eenvoudige vuistregel: per 10 °C warmer gaat de reactie ongeveer **twee keer zo snel**. Die factor heet Q_{10}.

VUISTREGEL VOOR DE TEMPERATUUR (ONDER HET OPTIMUM)

$$v_2 = v_1 \cdot Q_{10}^{(T_2 - T_1)/10} \qquad Q_{10} \approx 2$$

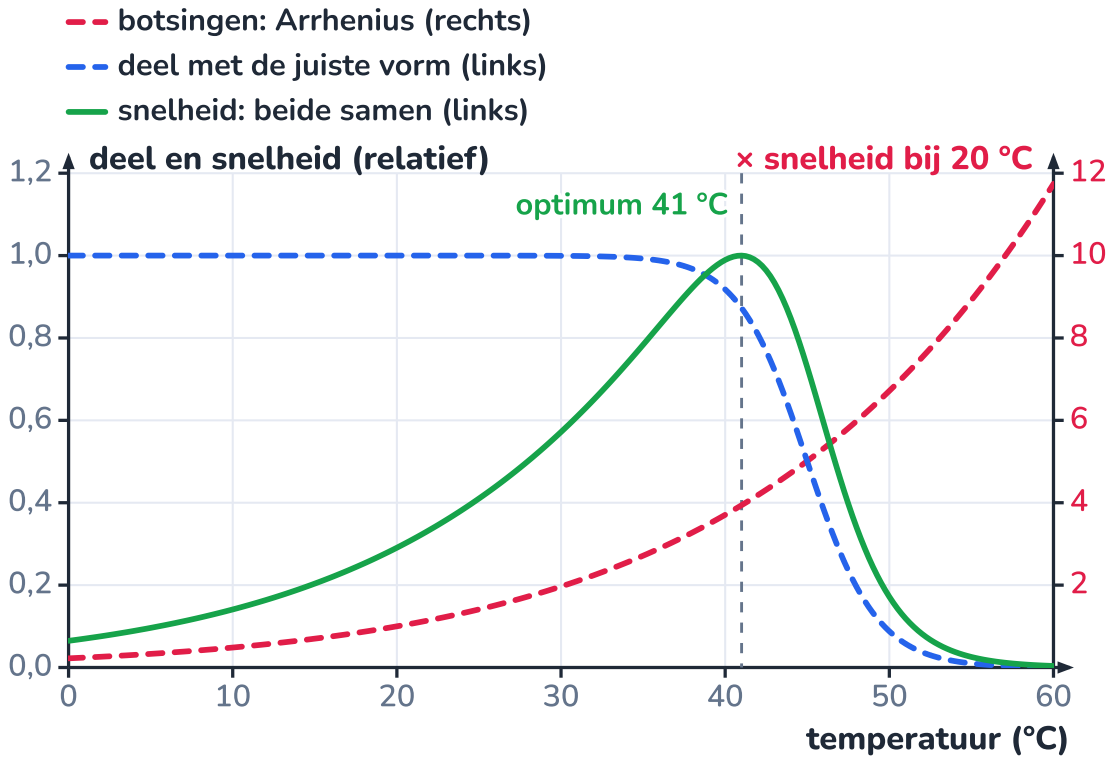
v de reactiesnelheid, T in °C. Een exponentieel model zoals in hoofdstuk 5: de groeifactor per 10 °C is Q_{10}.

2. Het enzym verliest zijn vorm. Een eiwit houdt zijn vorm door vele zwakke bindingen, zoals waterstofbruggen (hoofdstuk 13). Bij hogere temperatuur trillen de ketens zo hevig dat die bindingen breken: het eiwit ontvouwt, het **denatureert**. Boven een bepaalde temperatuur gebeurt dat plots bij bijna alle moleculen, zoals het eiwit van een ei in de pan wit en vast wordt.

In een **model** vermenigvuldig je beide: de snelheid is het Arrhenius-effect maal het deel van de enzymen dat zijn vorm nog heeft. De grafiek toont het voor een enzym dat bij 45 °C voor de helft ontvouwen is. De rode lijn stijgt steeds sneller, de blauwe valt boven 40 °C plots weg. Het product, de groene lijn, heeft

een top: het **optimum**, hier bij 41 °C. Voorbij het optimum valt de snelheid veel steiler dan ze ervoor steeg.

> ⚠ **Een model, geen wet**
>
> Het model gaat ervan uit dat het ontvouwen meteen gebeurt en dat de temperatuur de enige oorzaak is. In werkelijkheid is denaturatie ook een kwestie van **tijd**: een enzym houdt het een paar seconden uit bij een temperatuur die het na een uur kapotmaakt. Daarom hangt een gemeten optimum af van de werkwijze. Het model verklaart de vorm van de kromme, niet elk getal.



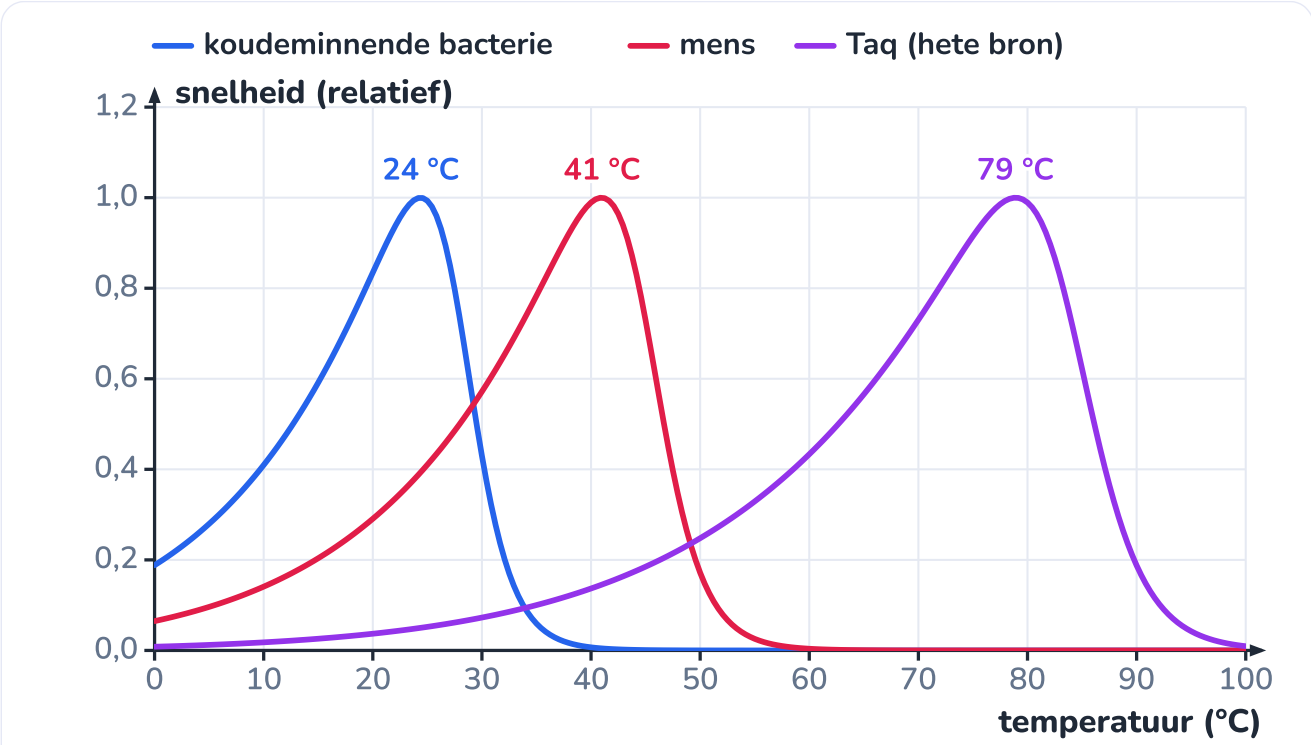
Rood: meer botsingen (rechter as). Blauw: deel met de juiste vorm. Groen: beide samen, met een optimum.

UITGEWERKT VOORBEELD · 34.7

Rekenen met Q10, en waar de regel ophoudt

7/22

Een enzym breekt bij 20 °C 5,0 µmol substraat per minuut af. Neem $Q_{10} = 2{,}0$. (a) Hoe snel gaat het bij 30 °C en bij 37 °C? (b) Hoe snel in de koelkast, bij 4 °C? (c) Voor het menselijke enzym uit het model is de snelheid bij 50 °C maar 17 % van die bij 40 °C. Wat zou Q_{10} voorspellen, en waarom klopt dat niet?



Elk organisme heeft enzymen die passen bij zijn leefomgeving: koud zeewater, 37 °C of een hete bron.

GEGEVEN

- $v_{20} = 5{,}0\,\mu\text{mol/min}$, $Q_{10} = 2{,}0$
- model: $v(50\,^\circ\text{C})/v(40\,^\circ\text{C}) = 0{,}17$

GEVRAAGD

- v_{30}, v_{37}, v_4
- wat Q_{10} voorspelt tussen 40 en 50 °C

OPLOSSING

1. (a) Van 20 naar 30 °C is één stap van 10 °C: $v_{30} = 5{,}0 \cdot 2{,}0^{1} = 10\,\mu\text{mol/min}$. Naar 37 °C: $v_{37} = 5{,}0 \cdot 2{,}0^{17/10} = 5{,}0 \cdot 3{,}25 = 16\,\mu\text{mol/min}$.

2. (b) Naar 4 °C is een stap van -16 °C: $v_4 = 5{,}0 \cdot 2{,}0^{-1{,}6} = 5{,}0/3{,}03 = 1{,}65\,\mu\text{mol/min}$. In de koelkast gaat het dus ongeveer drie keer trager.

3. (c) Q_{10} voorspelt $2{,}0$ keer sneller bij 50 °C. Het model geeft 0,17 keer zo snel: het enzym denatureert. De vuistregel beschrijft alleen het eerste effect (botsingen) en geldt dus alleen onder het optimum.

ANTWOORD

10 µmol/min bij 30 °C, 16 µmol/min bij 37 °C, 1,65 µmol/min bij 4 °C. Tussen 40 en 50 °C valt de snelheid in het model tot 17 %, waar Q_{10} een verdubbeling voorspelt: boven het optimum geldt de regel niet.

Klopt dit? Dit is de reden waarom je eten in de koelkast bewaart: ook de enzymen van het voedsel zelf (en van bacteriën, hoofdstuk 35) werken er ongeveer drie keer trager. Blancheren (groenten kort in kokend water dompelen voor je ze invriest) werkt met het tweede effect: het denatureert de enzymen die anders in de diepvries kleur en smaak afbreken.

VERHAAL · 34.8

Enzymen van koud tot kokend

8/22

Het optimum van een enzym past bij waar zijn eigenaar leeft. Menselijke enzymen werken het best rond lichaamstemperatuur, met een marge tot ongeveer 40 °C. Daarom wordt koorts boven ongeveer 41 °C gevaarlijk: eiwitten beginnen hun vorm te verliezen en cellen raken beschadigd.

Bacteriën uit het ijskoude water rond Antarctica hebben enzymen die bij 0 °C nog vlot werken: hun vouwing is losser, met minder bindingen. Ze denatureren daardoor ook al bij een temperatuur waarbij jij het lekker vindt.

Aan het andere uiterste vond de microbioloog Thomas Brock in 1969 in de hete bronnen van Yellowstone de bacterie *Thermus aquaticus*. Haar DNA-polymerase, het enzym dat DNA kopieert, heet **Taq-polymerase**. Het overleeft zelfs een verhitting tot 95 °C. Daarom werkt de PCR-test met Taq: bij elke cyclus wordt het DNA bij ongeveer 95 °C in twee strengen gesplitst, en het enzym overleeft dat telkens (hoofdstuk E30).

Enzymen in je wasmiddel

Moderne wasmiddelen bevatten enzymen: proteasen tegen eiwitvlekken (bloed, gras, ei), amylasen tegen zetmeel, lipasen tegen vet. Ze zijn gekozen of aangepast om te werken bij 20 tot 40 °C. Wie op 30 °C wast in plaats van op 60 °C, spaart meer dan de helft van de energie van een wasbeurt, omdat de enzymen het werk doen dat vroeger de hitte deed.

De gekleurde randen van een hete bron zijn bacteriematten: elke kleur leeft bij haar eigen temperatuur.

UITLEG · 34.9

pH: de vorm hangt aan ladingen

9/22

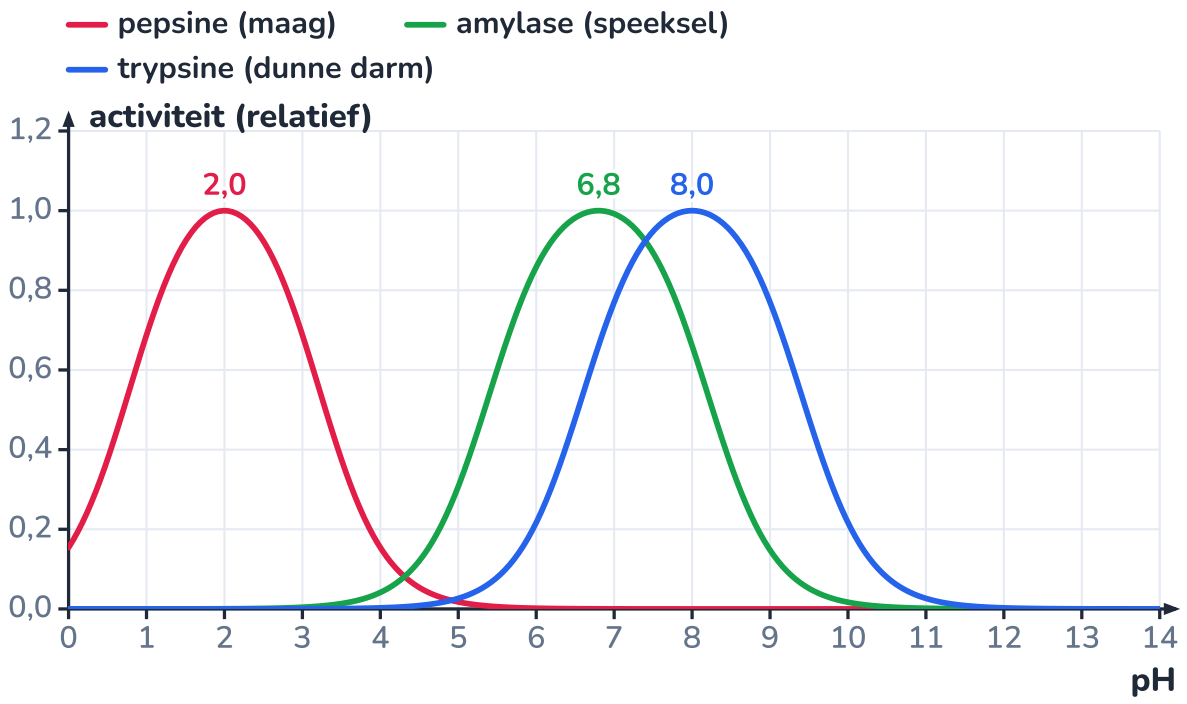
Ook de pH (hoofdstuk 15) verandert de vorm van een enzym. Aminozuren hebben zijgroepen die, afhankelijk van de pH, een proton opnemen of afgeven: -COOH wordt $-COO^-$, $-NH_2$ wordt $-NH_3^+$. Daarmee veranderen de ladingen in het eiwit en in het actieve centrum. Het substraat wordt dan niet meer vastgehouden, of het eiwit valt uit zijn plooi.

Elk enzym heeft zo een **pH-optimum**, en ook dat past bij de plek waar het werkt. Pepsine knipt eiwitten in je maag, waar maagzuur de pH tot ongeveer 2 laat zakken. Amylase uit je speeksel werkt het best rond pH 7 en valt stil in de maag. Trypsine werkt verder in de dunne darm, waar de alvleesklier het zure voedsel neutraliseert tot een licht basische pH.

In de grafiek is elk enzym beschreven met een eenvoudig **model**: twee zuurgroepen in het actieve centrum moeten allebei in de juiste vorm staan. Daardoor krijg je een klokvorm, met het optimum in het midden. Echte enzymen hebben soms een scheve of bredere top.

Een zuur 'eet' het enzym op

Een enzym dat in zuur stilvalt, is niet opgelost of verbrand. Het heeft een andere lading en vaak een andere vorm. Een kleine pH-verandering is omkeerbaar: breng je de pH terug, dan werkt het weer. Een extreme pH denatureert het eiwit wel blijvend, zoals melk die schift als je er citroensap in doet.



Drie spijsverteringsenzymen, drie optima: 2,0 in de maag, 6,8 in de mond, 8,0 in de dunne darm.

UITLEG · 34.10

Meer substraat: sneller, tot het enzym verzadigd is

10/22

Wat gebeurt er met de snelheid als je meer substraat toevoegt, bij een vaste hoeveelheid enzym? Bij weinig substraat staan de meeste actieve centra leeg: dubbel zoveel substraat geeft bijna dubbel zoveel botsingen en dubbel zoveel product per minuut. Maar een enzym heeft maar een beperkt aantal actieve centra, en elk heeft een vaste tijd nodig per molecule. Bij veel substraat zijn ze allemaal bezet: het enzym is **verzadigd**. Meer substraat helpt dan niet meer. Michaelis en Menten beschreven die kromme met

DE VERGELIJKING VAN MICHAELIS EN MENTEN

$$v = \frac{V_{\max} \cdot [S]}{K_m + [S]}$$

v de snelheid (bv. in µmol/min), $[S]$ de concentratie van het substraat (mmol/L). Twee getallen typeren het enzym: $V_{\max}$ en K_m. Hoe je deze formule afleidt, zie je in hoofdstuk E31.

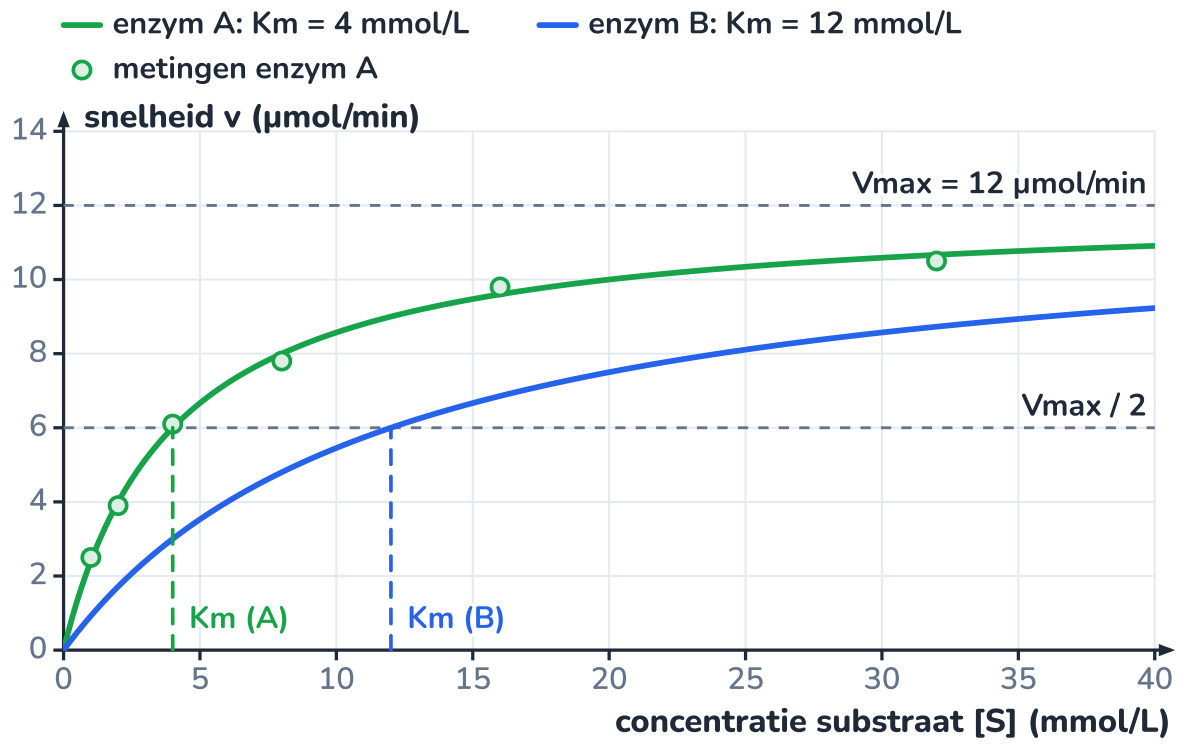
- $V_{\max}$ is de **grootste snelheid**: alle actieve centra bezet. Dubbel zoveel enzym geeft dubbel zo'n grote $V_{\max}$.
- K_m is de substraatconcentratie waarbij het enzym op de **helft** van $V_{\max}$ werkt. Vul $[S] = K_m$ in: $v = V_{\max} \cdot K_m/(2K_m) = V_{\max}/2$. Een **kleine** K_m betekent dat het enzym zijn substraat al bij een lage

concentratie goed vangt.

In de grafiek hebben enzym A en B dezelfde $V_{\max}$, maar A heeft een drie keer kleinere K_m: bij weinig substraat is A veel sneller. Bij heel veel substraat werken ze even snel.

Twee gebieden onthouden

Bij $[S] \ll K_m$ is $v \approx (V_{\max}/K_m) \cdot [S]$: recht evenredig (de rechte door de oorsprong). Bij $[S] \gg K_m$ is $v \approx V_{\max}$: vlak. Dezelfde vorm zie je terug bij de groei van bacteriën met meer of minder voeding (hoofdstuk 35 en E31), en bij de fotosynthese met meer of minder licht (F27).



Beide enzymen halen dezelfde Vmax. A haalt de helft ervan al bij 4 mmol/L, B pas bij 12 mmol/L.

UITGEWERKT VOORBEELD · 34.11

Vmax en Km aflezen en gebruiken

11/22

De punten in de grafiek zijn metingen met enzym A. (a) Lees $V_{\max}$ en K_m af. (b) Bereken de snelheid bij 2,0 mmol/L. (c) Bij welke concentratie werkt het enzym op 90 % van $V_{\max}$? (d) Een leerling verdubbelt de concentratie van 20 naar 40 mmol/L en verwacht dubbel zoveel product per minuut. Wat meet ze?

GEGEVEN

- de metingen en de kromme van enzym A in de grafiek
- $v = V_{\max}[S]/(K_m + [S])$

GEVRAAGD

- $V_{\max}$, K_m
- $v(2{,}0)$
- $[S]$ bij $0{,}9\,V_{\max}$
- $v(20)$ en $v(40)$

OPLOSSING

1 (a) De kromme vlakt af naar 12 µmol/min: $V_{\max} = 12\,\mu\text{mol/min}$. De helft, 6 µmol/min, wordt gehaald bij 4 mmol/L: $K_m = 4\,\text{mmol/L}$.

2 (b) $v = 12 \cdot 2{,}0/(4 + 2{,}0) = 24/6{,}0 = 4{,}0\ \mu\text{mol/min}$ (de meting gaf 3,9).

3 (c) $0{,}9 = [S]/(4 + [S])$ geeft $0{,}9 \cdot 4 + 0{,}9\,[S] = [S]$, dus $0{,}1\,[S] = 3{,}6$ en $[S] = 36\ \text{mmol/L}$: negen keer K_m.

4 (d) $v(20) = 12 \cdot 20/24 = 10{,}0$ en $v(40) = 12 \cdot 40/44 = 10{,}9\ \mu\text{mol/min}$: maar 9 % meer.

ANTWOORD

$V_{\max} = 12\ \mu\text{mol/min}$, $K_m = 4\ \text{mmol/L}$; bij 2,0 mmol/L gaat het 4,0 µmol/min; 90 % van $V_{\max}$ pas bij 36 mmol/L; van 20 naar 40 mmol/L stijgt de snelheid maar van 10,0 naar 10,9 µmol/min.

Klopt dit? Wie in de verzadiging meer product wil, heeft niets aan meer substraat: hij heeft meer **enzym** nodig. Een cel regelt de snelheid van een reactie daarom vaak door meer of minder enzym te maken, of door het enzym aan of uit te zetten (volgend blad).

UITLEG · 34.12

Stofwisseling: ketens van enzymen, met een rem

12/22

Alle reacties in een organisme samen heten de **stofwisseling** of het **metabolisme**. Er zijn twee soorten:

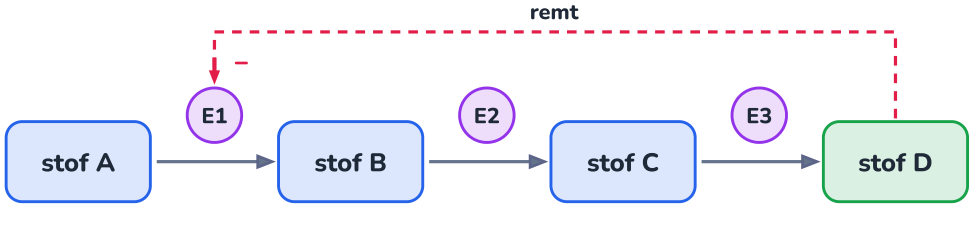
- **afbreken** (katabolisme): grote moleculen worden kleine, en er komt energie vrij. Voorbeelden: de vertering van zetmeel tot glucose, de celademhaling;
- **opbouwen** (anabolisme): uit kleine moleculen maakt de cel grote, en dat kost energie. Voorbeelden: eiwitten uit aminozuren, zetmeel en cellulose uit glucose, de fotosynthese.

Bijna nooit gaat een omzetting in één stap. De afbraak van glucose in de glycolyse telt tien stappen, elk met een eigen enzym: een **enzymketen**. Het product van het ene enzym is het substraat van het volgende.

Zo'n keten heeft een rem nodig. Vaak **remt het eindproduct het eerste enzym** van zijn eigen keten. Is er veel eindproduct, dan zit het vaak op dat enzym en loopt de keten trager; wordt het verbruikt, dan komt het enzym vrij en versnelt de keten. Een voorbeeld: als een spiercel genoeg ATP heeft, remt ATP het enzym fosfofructokinase in de glycolyse, en wordt er minder glucose afgebroken.

Een regelkring in één cel

Dit is **negatieve terugkoppeling**, precies zoals de thermostaat van de klimaatmodule (hoofdstuk 31): de uitgang (het eindproduct) stuurt de ingang (het eerste enzym) bij, in de richting die een afwijking tegenwerkt. Je lichaam regelt zo ook zijn bloedsuiker en zijn temperatuur (E33). Leven is regelen.



eindproduct D remt het eerste enzym E1: negatieve terugkoppeling

Drie enzymen zetten A om in D. Veel D remt E1: de keten regelt zichzelf.

UITLEG · 34.13

Fotosynthese en celademhaling als reactievergelijking

In F27 schreef je beide processen als woordvergelijking, in hoofdstuk 14 als kloppende vergelijking. Nu komt de energie erbij, en de plek waar ze gebeuren.

FOTOSYNTHESE (IN DE BLADGROENKORRELS, MET LICHT)

$$6\,CO_2 + 6\,H_2O \longrightarrow C_6H_{12}O_6 + 6\,O_2 \qquad +2803\,\text{kJ per mol glucose}$$

Endotherm: de energie komt van het licht. Bladgroen vangt het licht, tientallen enzymen doen de rest.

CELADEMHALING (IN CYTOPLASMA EN MITOCHONDRIËN)

$$C_6H_{12}O_6 + 6\,O_2 \longrightarrow 6\,CO_2 + 6\,H_2O \qquad -2803\,\text{kJ per mol glucose}$$

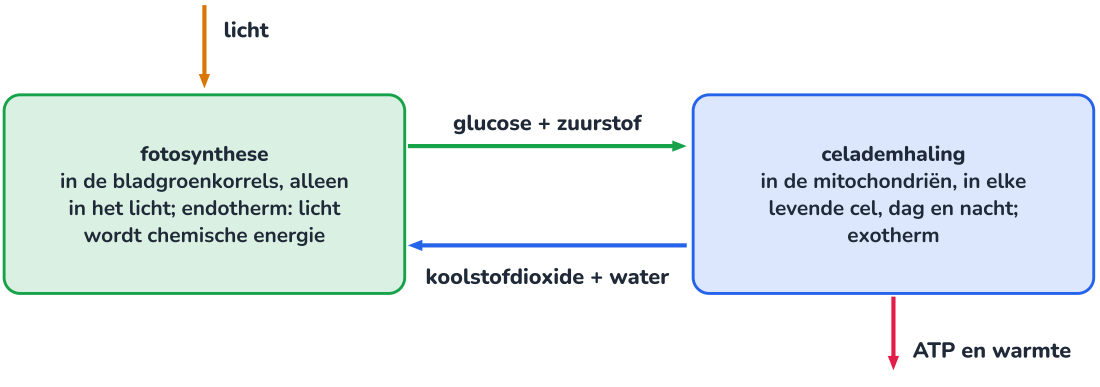
Exotherm: dezelfde energie komt vrij. Ongeveer een derde wordt ATP, de rest warmte.

Is er geen zuurstof, dan kan een cel toch nog een beetje energie uit glucose halen, met **gisting**. Gist maakt er alcohol en koolstofdioxide van; je spieren bij een sprint, en melkzuurbacteriën in yoghurt, maken er melkzuur van (hoofdstuk 35):

$$C_6H_{12}O_6 \longrightarrow 2\,C_2H_5OH + 2\,CO_2 \qquad C_6H_{12}O_6 \longrightarrow 2\,C_3H_6O_3$$

	fotosynthese	celademhaling	gisting (zonder zuurstof)
waar	bladgroenkorrels	cytoplasma en mitochondriën	cytoplasma
wanneer	alleen in het licht	altijd	als er geen zuurstof is
energie	neemt 2803 kJ per mol glucose op	geeft 2803 kJ per mol glucose vrij; ± 30 ATP	2 ATP per glucose
gassen	CO_2 in, O_2 uit	O_2 in, CO_2 uit	CO_2 uit (bij alcoholgisting)

Gisting levert maar **2 ATP** per glucose, tegenover ongeveer 30 bij de celademhaling: 15 keer minder. Het meeste van de energie zit na de gisting nog in de alcohol of het melkzuur. Daarom kan je alcohol als brandstof gebruiken, en daarom verbruikt gist die gist enorm veel suiker.



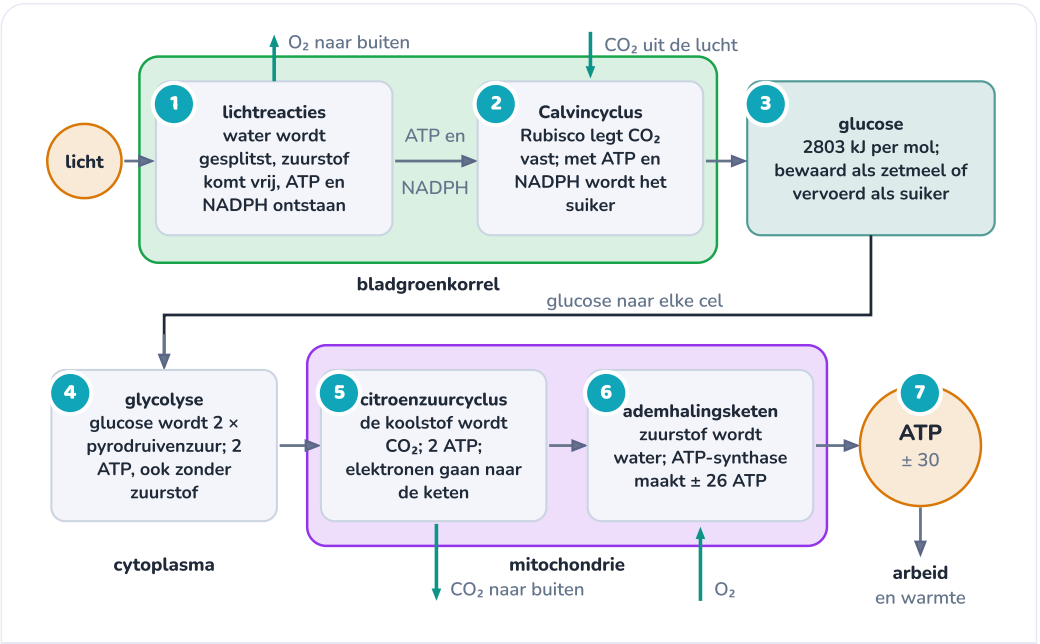
Wat de bladgroenkorrel maakt, gebruikt de mitochondrie. De koolstof gaat in een kringloop, de energie niet: die eindigt als warmte.

ONTLEED · 34.14

De energieketen: van licht tot ATP

Volg de energie van één lichtdeeltje tot in een spiercel. In de bladgroenkorrel wordt licht chemische energie, in elke cel wordt die chemische energie ATP. Beide helften gebruiken hetzelfde soort motortje: **ATP-synthase**. In de sla van Habitat belandt ongeveer

3,5 % van het lamplicht dat op het kweekbed valt, als chemische energie in de droge stof (per dag 256 kJ van 7,2 MJ licht).



Boven de bladgroenkorrel (fotosynthese), onder de celademhaling in cytoplasma en mitochondrie.

1 Lichtreacties
Bladgroen in de thylakoïden (plaatjes in de bladgroenkorrel) vangt het licht. De energie splitst water: de zuurstof die vrijkomt, is de zuurstof die jij inademt. De elektronen maken ATP en NADPH, een tweede energiedrager.

2 Calvincyclus
In het stroma legt het enzym Rubisco koolstofdioxide vast. Met de energie van ATP en NADPH wordt het suiker. Rubisco is traag (enkele moleculen per seconde), dus maakt de plant er veel van: tot de helft van het oplosbare eiwit in een blad. Het is het meest voorkomende eiwit op aarde.

3 Glucose
De chemische energie zit in de bindingen van glucose: 2803 kJ per mol. De plant bewaart ze als zetmeel of vervoert ze als sacharose. Wie de plant eet, krijgt ze erbij.

4 Glycolyse
Tien enzymen breken glucose in het cytoplasma af tot twee moleculen pyrodruivenzuur. Dat levert 2 ATP op, ook zonder zuurstof. Zonder zuurstof stopt het hier: gisting.

5 Citroenzuurcyclus
In de mitochondrie wordt de koolstof volledig afgebroken tot koolstofdioxide: de CO_2 die je uitademt. De energie gaat vooral naar elektronendragers.

6 Ademhalingsketen
De elektronen worden via een keten van eiwitten aan zuurstof gegeven, dat water wordt. Die keten pompt protonen over een membraan; terugstromend door ATP-synthase, een draaiend moleculair motortje, maken ze de meeste ATP.

7 ATP
Ongeveer 30 ATP per glucose. Met ATP doet de cel arbeid: spieren trekken samen, ionen worden gepompt, eiwitten gebouwd. Alles wat niet in ATP terechtkomt, en uiteindelijk ook de energie van het ATP zelf, wordt warmte.

UITLEG · 34.15

ATP: de oplaadbare batterij van de cel

15/22

Glucose is een goede brandstof, maar te groot om mee te betalen: één molecule levert 2803 kJ per mol, terwijl een spiereiwit per stapje maar een fractie nodig heeft. Daarom zet de cel de energie eerst om in

kleine, handige pakjes: **ATP**, adenosinetrifosfaat. Het molecule heeft drie fosfaatgroepen op een rij. Als de laatste afsplitst, komt er energie vrij:

ATP GEEFT ENERGIE AF

$$\mathrm{ATP} + \mathrm{H_2O} \longrightarrow \mathrm{ADP} + \text{fosfaat} \qquad \Delta G^{\circ\prime} = -30{,}5\,\mathrm{kJ/mol}$$

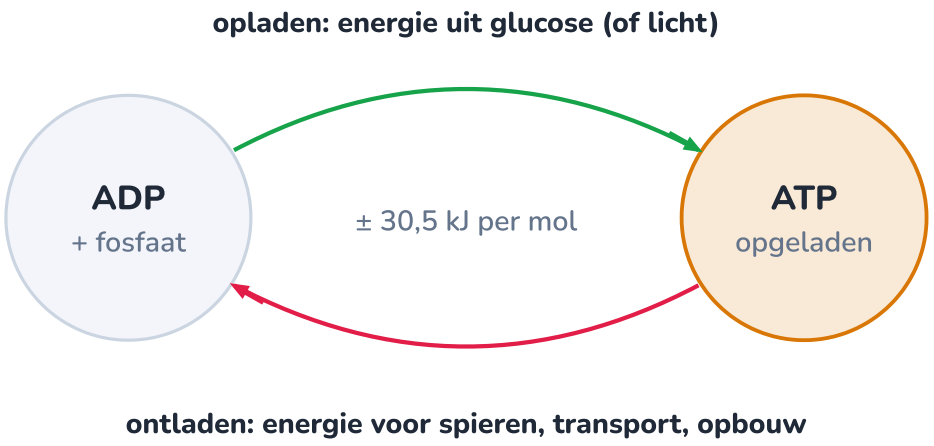
ADP: adenosinedifosfaat. De waarde $-30{,}5\,\mathrm{kJ/mol}$ geldt bij standaardomstandigheden; in een levende cel is het ongeveer $-50\,\mathrm{kJ/mol}$, omdat er veel meer ATP dan ADP is.

Denk aan een **oplaadbare batterij**: ADP is de lege batterij, ATP de volle. De celademhaling (en in een plant ook het licht) laadt op, spierwerk, transport en opbouw ontladen. Een cel heeft maar een klein voorraadje ATP, en laadt het elke minuut opnieuw op. Een volwassene zet zo per dag ongeveer zijn eigen lichaamsgewicht aan ATP om (voorbeeld hieronder).

Per glucosemolecule maakt de celademhaling ongeveer **30 ATP** (oudere boeken zeggen 36 tot 38; nauwkeurigere metingen gaven minder). Met de standaardwaarde komt $30 \cdot 30{,}5 = 915\,\mathrm{kJ}$ van de $2803\,\mathrm{kJ}$ in ATP terecht: ongeveer **een derde**. De rest wordt meteen warmte. Die warmte is niet verloren: ze houdt jou op 37 °C.

Energie zit 'in de binding' die breekt

Het breken van een binding kost altijd energie. Dat ATP energie levert, komt doordat de producten (ADP, fosfaat en water, met hun nieuwe bindingen en omringende watermoleculen) samen lager in energie liggen dan ATP en water. Het is de hele reactie die energie oplevert, zoals bij de verbranding in hoofdstuk 14.



Opladen met energie uit glucose of licht, ontladen voor arbeid: elke minuut opnieuw.

UITGEWERKT VOORBEELD · 34.16

Hoeveel ATP maakt een bewoner per dag?

16/22

Een bewoner van het station eet per dag voedsel met 10 MJ energie (F29). Neem aan dat een derde daarvan in ATP terechtkomt, met 30,5 kJ per mol. Hoeveel mol en hoeveel kilogram ATP maakt hij per dag? De molaire massa van ATP ($\mathrm{C_{10}H_{16}N_5O_{13}P_3}$) is 507,18 g/mol.

GEGEVEN

- $E = 10\,\mathrm{MJ}$ per dag, een derde naar ATP
- 30,5 kJ per mol ATP
- $M = 507{,}18\,\mathrm{g/mol}$

GEVRAAGD

- n(ATP) en m(ATP) per dag

OPLOSSING

1. Energie naar ATP: $10\,000\,\text{kJ}/3 = 3333\,\text{kJ}$.
2. Aantal mol: $n = 3333/30{,}5 = 109\,\text{mol}$ ATP per dag.
3. Massa: $m = 109 \cdot 507{,}18 = 55\,430\,\text{g} \approx 55\,\text{kg}$.

ANTWOORD

Ongeveer 109 mol ATP, of 55 kg per dag: ongeveer het lichaamsgewicht van een volwassene.

Klopt dit? Rekent men met de 50 kJ per mol die in een cel geldt, dan is het ongeveer 34 kg. Beide antwoorden vallen binnen wat biochemici opgeven: 100 à 150 mol per dag. Omdat je maar een klein voorraadje ATP hebt, wordt elk ATP-molecule honderden keren per dag opnieuw opgeladen.

UITGEWERKT VOORBEELD · 34.17

Wat verbrandt de bewoner? Het ademhalingsquotiënt

17/22

Een bewoner ademt per dag ongeveer 0,84 kg zuurstof in en 1,0 kg koolstofdioxide uit (hoofdstuk 14). Bereken het **ademhalingsquotiënt** $RQ = n(\text{CO}_2)/n(\text{O}_2)$. Bij de verbranding van glucose is $RQ = 1$ (zie de vergelijking), bij vet ongeveer 0,70. Wat verbrandt de bewoner vooral?

GEGEVEN

- $m(\text{O}_2) = 840\,\text{g}$, $m(\text{CO}_2) = 1000\,\text{g}$
- $M(\text{O}_2) = 32{,}00\,\text{g/mol}$, $M(\text{CO}_2) = 44{,}01\,\text{g/mol}$
- vet (tripalmitine): $2\,\text{C}_{51}\text{H}_{98}\text{O}_6 + 145\,\text{O}_2 \longrightarrow 102\,\text{CO}_2 + 98\,\text{H}_2\text{O}$

GEVRAAGD

- RQ en een schatting van de verhouding koolhydraten en vet

OPLOSSING

1. $n(\text{O}_2) = 840/32{,}00 = 26{,}3\,\text{mol}$; $n(\text{CO}_2) = 1000/44{,}01 = 22{,}7\,\text{mol}$.
2. $RQ = 22{,}7/26{,}3 = 0{,}87$.
3. Glucose: 6 mol CO_2 per 6 mol O_2, $RQ = 1$. Vet: $102/145 = 0{,}70$.
4. $0{,}87$ ligt tussen beide in. Als een deel x van de zuurstof naar koolhydraten gaat: $x \cdot 1 + (1 - x) \cdot 0{,}70 = 0{,}87$, dus $x = 0{,}55$.

ANTWOORD

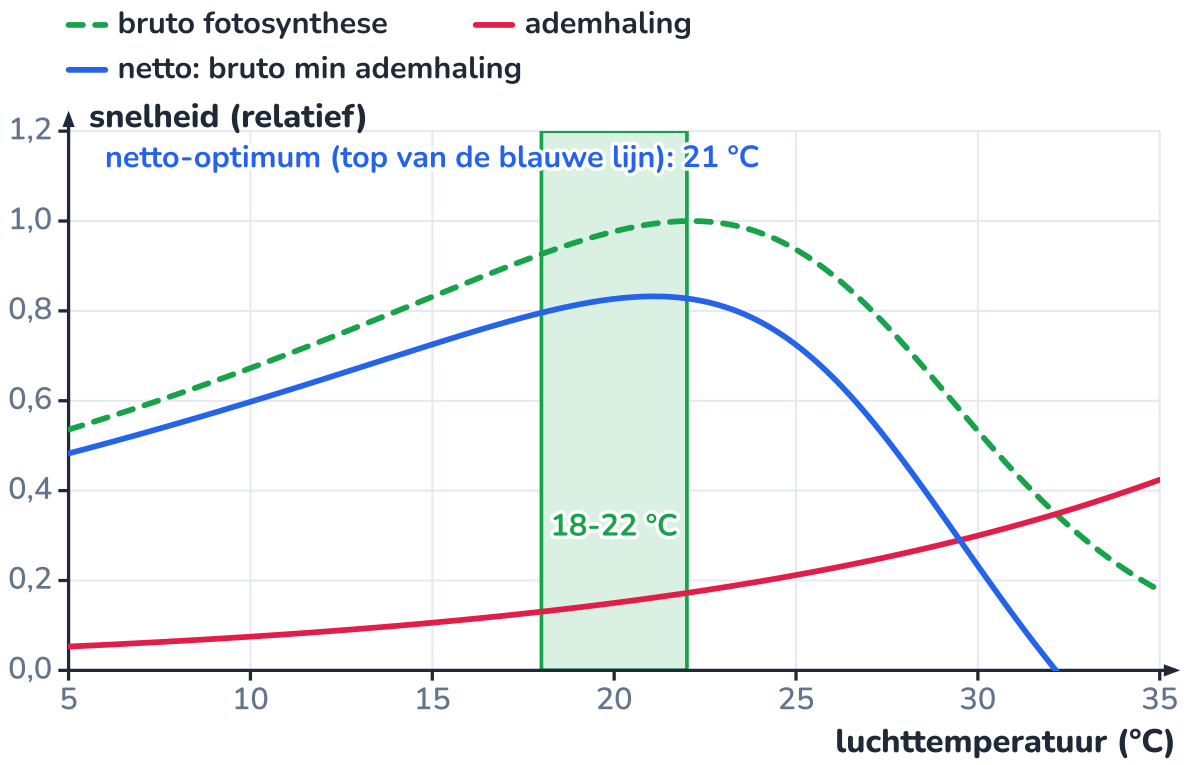
$RQ \approx 0{,}87$: ruwweg de helft van de zuurstof verbrandt koolhydraten, de rest vooral vet (eiwit, met $RQ \approx 0{,}8$, laten we hier buiten beschouwing).

Klopt dit? Dit is een gewone waarde voor een gemengd dieet. Voor het station betekent het: per mol zuurstof die de bewoner verbruikt, komt er maar 0,87 mol koolstofdioxide bij. Planten doen het omgekeerde met RQ ≈ 1: wie de lucht met planten wil verversen, moet dat verschil in rekening brengen (E31).

UITGEWERKT VOORBEELD · 34.18

Habitat: de optimale temperatuur voor de sla, verklaard

De klimaatmodule houdt de kweekruimte overdag tussen 18 en 19 °C en 's nachts tussen 16 en 17 °C (hoofdstuk 31). Waarom niet warmer, als enzymen tot 40 °C sneller werken? En waarom 's nachts koeler? Gebruik een model: de bruto fotosynthese heeft een optimum zoals een enzym, de ademhaling stijgt met $Q_{10} = 2$ en loopt dag en nacht door.



Model: de ademhaling (rood) blijft stijgen, dus ligt het netto-optimum (blauw) lager dan het bruto optimum (groen).

GEGEVEN

- model bruto fotosynthese bij 200 µmol/(m²·s): optimum 22 °C
- ademhaling: 0,15 van het bruto maximum bij 20 °C, $Q_{10} = 2$
- 16 uur licht, 8 uur donker; warmteverlies van de module 20,2 W/K (hoofdstuk 19)

GEVRAAGD

- het optimum van de netto fotosynthese en van de dagbalans
- het effect van een koelere nacht

OPLOSSING

1. Netto fotosynthese = bruto min ademhaling. Omdat de ademhaling met de temperatuur blijft stijgen, ligt het netto-optimum lager dan het bruto optimum: 21 °C in plaats van 22 °C (grafiek).

2. De dagbalans (16 uur netto min 8 uur nachtademhaling) is het grootst bij 21 °C overdag. Bij 18 °C haalt de sla nog 95 % daarvan, bij 28 °C maar 52 %: te warm kost veel meer dan iets te koel.

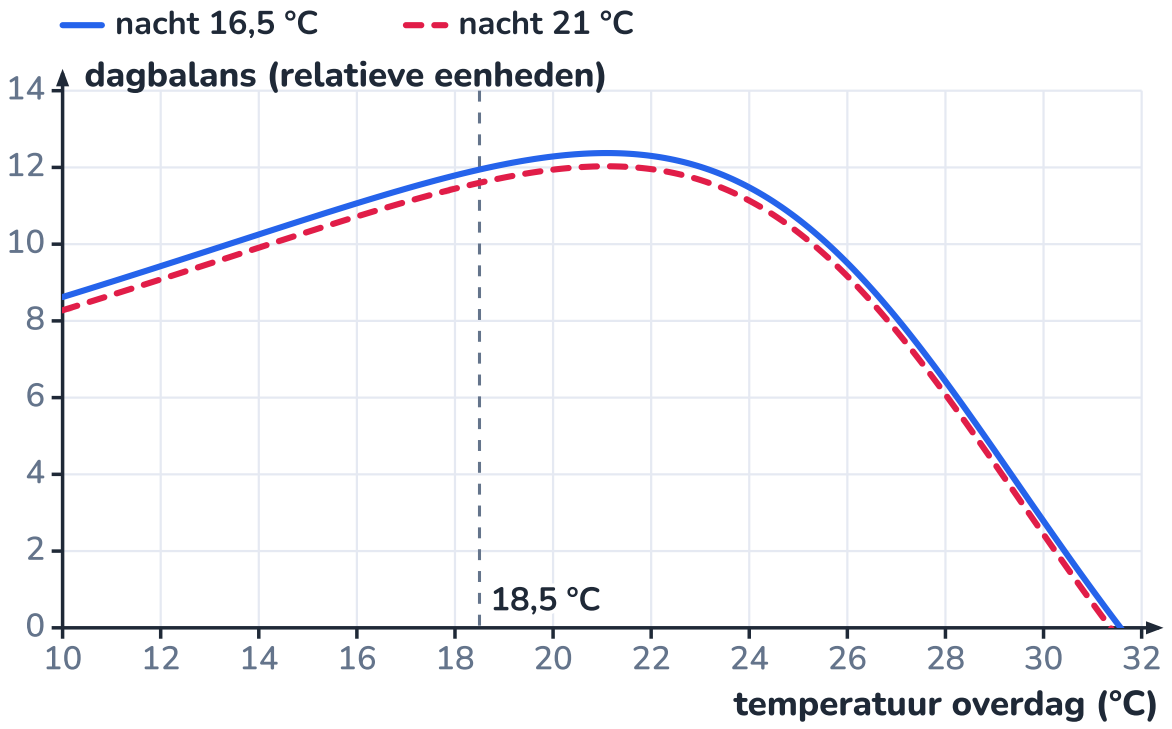
3. 's Nachts is er geen fotosynthese, alleen ademhaling. Bij 16,5 in plaats van 21 °C is de ademhaling $2^{-0{,}45} = 0{,}73$ keer zo groot: 27 % minder suiker verbrand. De dagbalans stijgt daardoor met 3,0 %.

4. De grootste winst is energie: 2 °C kouder gedurende 8 uur bespaart $20{,}2 \cdot 2 \cdot 8 = 323\ \text{Wh}$ warmte per nacht, 0,32 kWh.

ANTWOORD

In het model groeit de sla het best rond 21 °C overdag; 18 tot 22 °C kost hoogstens enkele procenten. Boven 25 °C daalt de groei snel, omdat de ademhaling doorstijgt en de fotosynthese na haar optimum valt. 's Nachts is koeler beter: minder ademhaling, en een lagere stookkost.

Klopt dit? Het model laat twee dingen weg die telers in de praktijk nog strenger maken. Boven ongeveer 24 °C gaat sla 'schieten' (ze maakt een bloemstengel en wordt bitter), en groeit ze zo snel dat calcium de jonge bladranden niet bijhoudt: ze verbranden (randjes, tipburn). Beide duwen het praktische optimum naar 18 à 22 °C. Een model met enzymen verklaart de vorm; de getallen komen uit metingen.



Een koelere nacht verschuift de hele dagbalans iets omhoog; het optimum overdag blijft bij 21 °C.

UITLEG · 34.19

Habitat: ook de wortels hebben een temperatuur

19/22

De wortels hangen in de voedingsoplossing van het reservoir (60 L). Ook zij ademen: dag en nacht, met zuurstof die opgelost zit in het water. Hoe warmer het water, hoe **sneller** de wortels ademen ($Q_{10} \approx 2$) en hoe **minder** zuurstof erin kan oplossen:

temperatuur van het water	10 °C	15 °C	20 °C	25 °C	30 °C
opgeloste zuurstof bij verzadiging (mg/L)	11,29	10,08	9,09	8,26	7,56

Van 20 naar 30 °C verdubbelt de vraag ongeveer, terwijl het aanbod met een zesde daalt. Warm water is bovendien ideaal voor *Pythium*, een waterschimmel die wortelrot veroorzaakt: bruine, slijmerige wortels, en slaplanten die op een warme dag verwelken (hoofdstuk 35).

Voor Habitat leggen we vast: de voedingsoplossing blijft tussen **18 en 22 °C**; boven **24 °C** geeft de NTC in het reservoir (hoofdstuk 22) een waarschuwing. Het reservoir staat op de vloer, in de schaduw van het rek; de stroming in de goten en het terugvallen in het reservoir brengen zuurstof in het water.

Waarom het rondpompen ook om zuurstof gaat

In een NFT-systeem stroomt de oplossing als een dun filmpje over de wortels: veel contact met de lucht, dus veel zuurstof. Staat de pomp stil, dan houdt de vochtige wortelmat het een tijd vol. Bij warm water mag die pauze korter zijn: nog een reden om het reservoir koel te houden (hoofdstuk 36).

Gezonde wortels zijn wit en stevig. Bruine, slijmerige wortels wijzen op te warm water en wortelrot.

CODE · 34.20

Python: het enzymmodel doorrekenen

20/22

Het model van dit hoofdstuk in een klein programma: twee functies voor de twee effecten, hun product als snelheid, en een zoektocht naar het optimum door alle temperaturen van 0,0 tot 100,0 °C af te lopen (`max` met een `key`, hoofdstuk 28). Tm is de temperatuur waarbij de helft van de enzymen ontvouwen is. De grafiek met de drie enzymen komt uit dezelfde formules.

Python **enzymmodel.py** 35 regels

```python
# enzymmodel.py - hoe snel werkt een enzym bij elke temperatuur? (een model)
import math

R = 8.314                          # gasconstante in J/(mol K)


def botsingen(t, Ea=50000):
    """Arrhenius: hoeveel keer sneller dan bij 20 graden, alleen door de temperatuur."""
    T = t + 273.15
    return math.exp(-Ea / R * (1 / T - 1 / 293.15))


```

```python
def juiste_vorm(t, Tm, dH=400000):
    """Deel van de enzymen dat zijn vorm nog heeft: 1 = alle, 0 = geen."""
    T = t + 273.15
    return 1 / (1 + math.exp(dH / R * (1 / (Tm + 273.15) - 1 / T)))


def snelheid(t, Tm):
    return botsingen(t) * juiste_vorm(t, Tm)


enzymen = {"koudeminnend": 28, "mens": 45, "Taq": 84}   # Tm: dan is de helft ontvouwen
temps = [x / 10 for x in range(0, 1001)]               # 0,0 tot 100,0 graden
beste = {}
for naam, Tm in enzymen.items():
    t_opt = max(temps, key=lambda t: snelheid(t, Tm))
    beste[naam] = snelheid(t_opt, Tm)
    print(f"{naam:13s} optimum bij {t_opt:5.1f} graden")

print()
print("  t   koudeminnend   mens    Taq    (tegenover het eigen maximum)")
for t in range(0, 101, 10):
    rij = [snelheid(t, Tm) / beste[naam] for naam, Tm in enzymen.items()]
    print(f"{t:3d}       {rij[0]:5.2f}     {rij[1]:5.2f}   {rij[2]:5.2f}")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
koudeminnend  optimum bij  24.4 graden
mens          optimum bij  41.0 graden
Taq           optimum bij  78.9 graden

  t   koudeminnend   mens    Taq    (tegenover het eigen maximum)
  0        0.19      0.06    0.01
 10        0.41      0.14    0.02
 20        0.83      0.29    0.04
 30        0.43      0.57    0.07
 40        0.01      0.99    0.14
 50        0.00      0.17    0.25
 60        0.00      0.00    0.43
 70        0.00      0.00    0.73
 80        0.00      0.00    0.99
 90        0.00      0.00    0.19
100        0.00      0.00    0.01
```

Het optimum ligt telkens een paar graden onder Tm: net voor de vorm massaal verloren gaat. In opdracht 34.12 zoek je uit wat er gebeurt met een grotere of kleinere activeringsenergie.

NASLAG · 34.21

Overzicht: enzymen en stofwisseling

vuistregel temperatuur (onder het optimum)

$$v_2 = v_1 \cdot Q_{10}^{(T_2 - T_1)/10}$$

symbool	betekenis	eenheid
v	reactiesnelheid	bv. µmol/min
T	temperatuur	°C
Q_{10}	factor per 10 °C, ≈ 2	–

Michaelis en Menten

$$v = \frac{V_{\max} \cdot [S]}{K_m + [S]}$$

symbool	betekenis	eenheid
$V_{\max}$	grootste snelheid (verzadiging)	µmol/min
K_m	[S] bij de halve Vmax	mmol/L
$[S]$	concentratie substraat	mmol/L

fotosynthese: +2803 kJ per mol glucose

$$6\,CO_2 + 6\,H_2O \longrightarrow C_6H_{12}O_6 + 6\,O_2$$

celademhaling: −2803 kJ per mol glucose, ± 30 ATP

$$C_6H_{12}O_6 + 6\,O_2 \longrightarrow 6\,CO_2 + 6\,H_2O$$

alcoholgisting: 2 ATP

$$C_6H_{12}O_6 \longrightarrow 2\,C_2H_5OH + 2\,CO_2$$

−30,5 kJ/mol (standaard), ± −50 kJ/mol in de cel

$$ATP + H_2O \longrightarrow ADP + \text{fosfaat}$$

ademhalingsquotiënt: 1 voor koolhydraten, ± 0,7 voor vet

$$RQ = \frac{n(CO_2)}{n(O_2)}$$

enzym	optimum temperatuur	optimum pH	waar
pepsine	± 37 °C	± 2	maag
speekselamylase	± 37 °C	± 6,8	mond
trypsine	± 37 °C	± 8	dunne darm
katalase (aardappel, deze proef)	± 40 °C	± 7	bijna alle cellen
Taq-polymerase	± 75-80 °C	± 8	bacterie uit een hete bron

Afgeronde, typische waarden; het gemeten optimum hangt af van de werkwijze.

Habitat: rekenwaarden uit dit hoofdstuk	
temperatuuroptimum sla (model, 200 µmol/(m²·s))	netto ± 21 °C; dagbalans bij 18 °C 95 %, bij 28 °C 52 %
nacht 16,5 in plaats van 21 °C	27 % minder nachtademhaling; 0,32 kWh minder warmte per nacht per 2 °C
voedingsoplossing	18-22 °C; waarschuwing boven 24 °C (zuurstof, wortelrot)
bewoner	± 109 mol ATP per dag; ademhalingsquotiënt 0,87
sla, hele kweekbed	± 3,5 % van het lamplicht wordt chemische energie in de droge stof

HET GEHEEL · 34.22

Waar enzymen en stofwisseling terugkomen

Enzymen verbinden de chemie met het leven. Wat je in hoofdstuk 14 leerde over reacties en activeringsenergie, en in hoofdstuk 15 over pH, gebeurt in elke cel. Het temperatuuroptimum verklaart waarom een regeling (hoofdstuk 31) de kweekruimte op 18 °C houdt, en de energieketen verklaart waar de energie van de lampen (hoofdstuk 20) en van het voedsel van de bewoners (F29) naartoe gaat.

F26 Cellen en organismen

Bladgroenkorrels en mitochondriën: hier zie je wat ze precies doen.

F27 Planten: fotosynthese en groei

Fotosynthese, ademhaling en het temperatuuroptimum, eerst als woorden en schetsen.

F29 Het lichaam als systeem

Energie uit voeding: hier volg je die energie tot in ATP.

I5 Groei en verval: exponentiële modellen

Q10 is een groeifactor; denaturatie in de tijd is exponentieel verval.

I14 Reacties en stoichiometrie

Reactievergelijkingen, activeringsenergie, exotherm en endotherm, katalase in gist.

I15 Zuren, basen en pH

pH verandert de ladingen in een enzym, en dus zijn vorm.

I31 Regelsystemen

De temperatuurgrenzen van de module, en terugkoppeling in een enzymketen.

I35 Micro-organismen en biotechnologie

Gisting en melkzuurgisting in brood, yoghurt en compost; groei van bacteriën met de temperatuur.

I36 Hydroponie en voedselproductie

Licht, CO2 en temperatuur samen: hoeveel sla per kWh?

E25 Thermochemie en reactiesnelheid

De vergelijking van Arrhenius en reactiesnelheid, kwantitatief.

E29 DNA, genen en eiwitten

Hoe een eiwit gebouwd en opgevouwen wordt, en wat een mutatie met een enzym doet.

E30 Biotechnologie en genetische technieken

PCR met Taq-polymerase, knipenzymen en CRISPR: enzymen als gereedschap.

E31 Bioprocestechniek

Verzadiging bij microben: het Monod-model is Michaelis-Menten voor een hele cel.

E33 Het lichaam als regelsysteem

Bloedsuiker en temperatuur: regelkringen met enzymen en hormonen.

Habitat

Voor Habitat leggen we vast: het temperatuuroptimum van de sla ligt in het model rond 21 °C overdag, en de grenzen van hoofdstuk 31 (18/19 °C overdag, 16/17 °C 's nachts) liggen aan de veilige, koele kant: weinig verlies aan groei, minder schieten en randjes, minder stookkost. De voedingsoplossing blijft tussen 18 en 22 °C, met een waarschuwing boven 24 °C. In opdracht 34.13 stel je zelf de temperatuurstrategie van de kweekruimte op.

SAMENVATTING

In het kort

- Een enzym is een eiwit dat een reactie versnelt door de activeringsenergie te verlagen; het wordt niet verbruikt.
- Het substraat past in het actieve centrum (slot en sleutel): elk enzym doet één soort reactie.
- Warmer is sneller (Q10 ≈ 2, meer botsingen) tot het enzym zijn vorm verliest (denaturatie): daartussen ligt het temperatuuroptimum.
- Ook de pH verandert de ladingen en dus de vorm: elk enzym heeft een pH-optimum dat past bij waar het werkt.
- Meer substraat geeft meer snelheid tot het enzym verzadigd is: Vmax is de grootste snelheid, Km de concentratie bij de halve Vmax.
- Stofwisseling bestaat uit enzymketens; het eindproduct remt vaak het eerste enzym (negatieve terugkoppeling).
- Fotosynthese ($6\ CO_2 + 6\ H_2O \rightarrow C_6H_{12}O_6 + 6\ O_2$) neemt 2803 kJ per mol glucose op; de celademhaling geeft ze vrij, gisting maar een klein deel.

- ATP is de oplaadbare batterij van de cel: ± 30 ATP per glucose, ongeveer een derde van de energie; een mens zet per dag ongeveer zijn lichaamsgewicht aan ATP om.
- Voor de sla ligt het optimum in het model rond 21 °C overdag; 's nachts koeler bespaart ademhaling en warmte.

BEGRIPPEN

Wat je nu kent

enzym
Eiwit dat een bepaalde reactie in een levend wezen versnelt zonder zelf verbruikt te worden: een biologische katalysator.

katalysator
Stof die een reactie versnelt door de activeringsenergie te verlagen, en na de reactie onveranderd overblijft.

actief centrum
De plek van een enzym waar het substraat precies in past en de reactie gebeurt.

substraat
De stof waarop een enzym inwerkt.

denaturatie
Het verlies van de ruimtelijke vorm van een eiwit, door hitte, een extreme pH of bepaalde stoffen; meestal onomkeerbaar.

temperatuuroptimum
De temperatuur waarbij een enzym (of een organisme) het snelst werkt: sneller door meer botsingen, niet zo warm dat de vorm verloren gaat.

Q10
Factor waarmee een reactie sneller gaat als de temperatuur 10 °C stijgt; voor enzymreacties ongeveer 2 (onder het optimum).

Vmax
De grootste snelheid van een enzymreactie: alle actieve centra zijn bezet (verzadiging).

Km
De substraatconcentratie waarbij een enzym op de helft van zijn grootste snelheid werkt; een kleine Km betekent dat het enzym zijn substraat al bij lage concentratie goed bindt.

stofwisseling
Alle chemische reacties in een levend wezen samen: afbreken (energie vrijmaken) en opbouwen (energie gebruiken).

celademhaling
Het afbreken van glucose met zuurstof tot koolstofdioxide en water in de cellen, waarbij ATP gemaakt wordt.

gisting
Afbraak van suiker zonder zuurstof, tot alcohol en koolstofdioxide (gist) of melkzuur; levert maar 2 ATP per glucose.

ATP
Adenosinetrifosfaat: de energiedrager van de cel. Het afsplitsen van één fosfaatgroep levert energie voor arbeid, transport en opbouw.

ademhalingsquotiënt
De verhouding tussen het aantal mol koolstofdioxide dat je uitademt en het aantal mol zuurstof dat je verbruikt.

HANDBOEK · HOOFDSTUK 35

Micro-organismen en biotechnologie

Onzichtbare helpers en belagers: van gist in je brood tot insuline uit een bacterie, en een compostvat voor Habitat

Er leven meer bacteriën in je darmen dan er mensen op aarde zijn, en toch zie je er geen enkele. Micro-organismen maken ons brood, onze yoghurt en ons biogas, ze ruimen afval op en ze maken geneesmiddelen. Sommige maken ons ziek. In dit hoofdstuk leer je ze kennen, kweken zonder risico, hun groei beschrijven met de groeicurve, en benutten in fermentatie en biotechnologie. Voor Habitat ontwerp je een compostvat en een hygiëneplan.

10× uitleg · 5× uitgewerkt voorbeeld · 3× proef · 1× ontleed · 1× naslag · 2× verhaal · 1× het geheel · 1× code · ± 10 lestijden

B Leven en systemen · O Onderzoeken en meten · W Wiskundig modelleren

Elk stipje is een kolonie van miljoenen cellen, gegroeid uit één enkele bacterie of spore.

Na dit hoofdstuk

- Je kan bacteriën, gisten, schimmels en virussen onderscheiden naar bouw, grootte en manier van vermeerderen.
- Je kan uitleggen hoe je steriel werkt, en een kweekproef veilig opzetten en afronden.
- Je kan de vier fasen van een groeicurve benoemen en verklaren, op een lineaire en op een logaritmische as.
- Je kan met een verdunningsreeks het aantal bacteriën per mL berekenen, en een generatietijd bepalen uit metingen.
- Je kan uitleggen hoe temperatuur de groei van bacteriën beïnvloedt, en waarom koelen, pasteuriseren en steriliseren werken.
- Je kan fermentaties (alcohol, melkzuur, azijn, methaan) als reactievergelijking schrijven en er mee rekenen.
- Je kan uitleggen hoe bacteriën menselijke insuline maken.
- Je kan een compostvat en hygiëneregels voor de kweekruimte van Habitat onderbouwen.

UITLEG · 35.1

Wat je al weet, en wat nu komt

1/24

In F26 bekeek je eencelligen onder de microscoop, in F28 zag je hoe reducenten dood materiaal afbreken, en in hoofdstuk 5 rekende je met bacteriën die elke 20 minuten verdubbelen. In hoofdstuk 34 zag je dat gist zonder zuurstof suiker kan vergisten. Nu brengen we het samen.

wat je al weet	waar	wat er nu bij komt
eencelligen, de microscoop, vergroting	F26	bacteriën, gisten, schimmels en virussen: bouw en grootte
reducenten breken dood materiaal af	F28	compost en biogas: reducenten aan het werk
bacteriën verdubbelen elke 20 minuten: exponentiële groei	hoofdstuk 5	de volledige groeicurve, met aanloop, plafond en afsterven
een gistfles maakt CO_2 uit suiker	hoofdstuk 14	fermentatie in brood, yoghurt en biogas
enzymen, temperatuuroptimum, gisting levert 2 ATP	hoofdstuk 34	waarom de koelkast werkt en koken ontsmet

In dit hoofdstuk

- leer je de vier grote groepen kennen: **bacteriën**, **gisten**, **schimmels** en **virussen**;
- leer je **steriel werken** en kweek je veilig kiemen van je handen en je smartphone;
- beschrijf je de volledige **groeicurve**, tel je bacteriën met een verdunningsreeks en bepaal je een generatietijd;
- zie je waarom de koelkast, pasteuriseren en koken werken;
- maak je zelf CO_2 met gist en yoghurt met melkzuurbacteriën, en meet je wat er gebeurt;
- leer je hoe **biogas** en **insuline** uit micro-organismen komen;
- ontwerp je voor Habitat een **compostvat** en een **hygiëneplan** voor de kweekruimte.

Brood, yoghurt, zuurkool, kaas en azijn: allemaal het werk van micro-organismen.

VERHAAL · 35.2

Delft: gist, spiritus, een virus en een geheime penicilline

In 1869 stichtten **Jacques van Marken** en zijn vrouw **Agneta Matthes** in Delft de *Nederlandsche Gist- en Spiritusfabriek*. Ze maakten bakkersgist, en uit het gistingsproces ook spiritus (alcohol). Gist kweken is kiemen kweken: één besmetting met de verkeerde bacterie en een hele productie is bedorven. Daarom haalde Van Marken in 1885 een jonge bioloog naar zijn fabriek, **Martinus Willem Beijerinck**, en richtte hij voor hem een laboratorium in.

Beijerinck bleef tien jaar. In 1888 haalde hij uit de wortelknolletjes van erwten en bonen de bacteriën die stikstof uit de lucht vastleggen (hoofdstuk 36). In 1895 werd hij hoogleraar aan de school die vandaag de TU Delft is. Daar onderzocht hij de mozaïekziekte van tabak. Hij perste het sap van zieke bladeren door een porseleinen filter dat alle bacteriën tegenhoudt. Het heldere sap maakte gezonde planten toch ziek, en de ziekte ging over van plant op plant. In 1898 besloot hij dat het ging om iets dat kleiner is dan elke bacterie en zich alleen in levende cellen vermenigvuldigt: een *contagium vivum fluidum*, een 'levende besmettelijke vloeistof'. Hij noemde het een **virus**. Beijerinck geldt daarom als een van de grondleggers van de virologie.

In 1928 zag de Schotse arts **Alexander Fleming** in Londen dat een schimmel die toevallig op een bacteriekweek gevallen was, de bacteriën eromheen doodde: penicilline. Tijdens de Duitse bezetting kreeg Albert Jan Kluyver, de opvolger van Beijerinck, van de directie van de gistfabriek de ruimte om in het geheim penicilline te leren maken. Om de bezetter te misleiden kreeg het project een schuilnaam: **Bacinol**. Op 15 november 1945 kreeg een Delftse patiënte met een levensgevaarlijke infectie als eerste Delftse penicilline toegediend; ze genas. De fabriek werd later Gist-Brocades en hoort sinds 1998 bij DSM: nog altijd een grote plek voor biotechnologie.

Waarom dit verhaal ertoe doet

Een fabriek die gist maakte, leverde de ontdekking van het virus, stikstofbindende bacteriën en penicilline. Biotechnologie begint bij zorgvuldig kweken: steriel werken, tellen, en precies weten wat er in je vat groeit. Dat leer je in dit hoofdstuk.

Een gistfabriek aan het water: in zo'n omgeving begon de Delftse school van de microbiologie.

UITLEG · 35.3

Hoe klein? Van virus tot gistcel

3/24

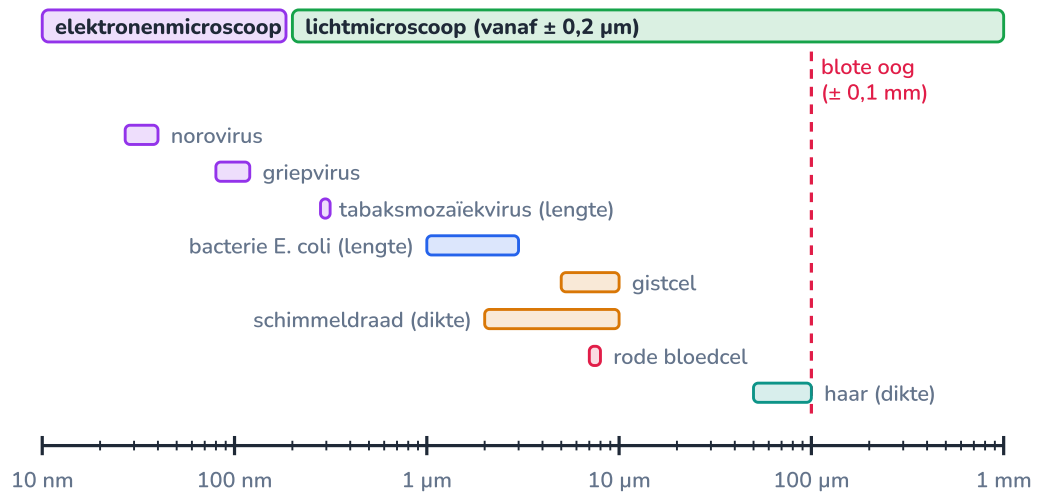
Micro-organismen zijn levende wezens die te klein zijn om met het blote oog te zien, dat tot ongeveer 0,1 mm scherp ziet. De belangrijkste groepen:

- **bacteriën** (en de verwante archaea): cellen zonder kern, meestal 1 tot 5 µm;
- **gisten**: eencellige schimmels met een kern, 5 tot 10 µm;
- **schimmels**: draden (hyfen) van enkele µm dik, die samen een zichtbaar pluis vormen;
- **algen** en **eencellige dieren** (F26): meestal groter, tot enkele tienden van een millimeter;
- **virussen**: geen cellen, maar deeltjes van 20 tot 300 nm.

Een micrometer (µm) is een duizendste millimeter; een nanometer (nm) een duizendste micrometer. Een lichtmicroscoop onderscheidt details tot ongeveer 0,2 µm: bacteriën zie je als stipjes en staafjes, virussen niet. Die werden pas in 1939 zichtbaar, met de elektronenmicroscoop.

Een logaritmische schaal

De figuur loopt van 10 nm tot 1 mm: vijf decaden. Op een gewone as zou een virus onzichtbaar klein naast een haar staan. Elke stap naar rechts is tien keer groter (hoofdstuk 5): een haar is ongeveer duizend keer zo dik als een griepvirus.



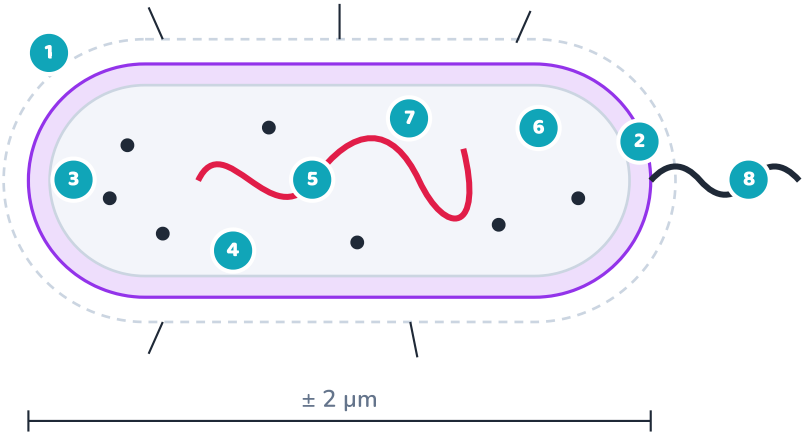
Balkjes tonen de gewone grootte. Wat rechts van ± 0,2 µm staat, zie je met een lichtmicroscoop.

ONTLEED · 35.4

De bacterie

4/24

Een bacterie is een volledige cel, maar eenvoudiger gebouwd dan een planten- of dierencel (F26): ze heeft **geen kern** en geen mitochondriën of bladgroenkorrels. Zo'n cel heet een **prokaryoot**. De darmbacterie *E. coli* is ongeveer 2 µm lang: vijfhonderd achter elkaar is een millimeter.



Schematische doorsnede van een staafvormige bacterie.

1 Kapsel
Een slijmlaag rond sommige bacteriën: bescherming tegen uitdrogen en tegen afweercellen, en een lijm om zich vast te hechten.

2 Celwand
Een stevig net dat de vorm geeft en de cel beschermt tegen barsten. Penicilline verhindert dat bacteriën hun celwand kunnen bouwen: daardoor werkt het tegen bacteriën en niet tegen jouw cellen, die geen celwand hebben.

3 Celmembraan
Het vlies dat bepaalt wat de cel in en uit gaat, zoals bij elke cel. Hier zitten ook de enzymen van de celademhaling: een bacterie heeft geen mitochondriën nodig.

4 Cytoplasma
De vloeistof in de cel, vol enzymen: hier gebeuren de meeste reacties van de stofwisseling.

5 DNA
Eén groot, ringvormig chromosoom, los in het cytoplasma (geen kern). Het bevat de informatie voor alle eiwitten van de bacterie.

6 Plasmide
Klein, extra ringetje DNA, dat bacteriën onderling kunnen doorgeven, bijvoorbeeld met genen voor resistentie tegen antibiotica. Biotechnologen gebruiken plasmiden om een gen in een bacterie te brengen (insuline, verderop).

7 Ribosomen
Duizenden kleine eiwitfabriekjes. Ze zijn anders gebouwd dan de onze, en daarom kan een antibioticum ze stilleggen zonder jouw cellen te raken.

8 Zweepdraad
Een draad die ronddraait als een scheepsschroef, aangedreven door een klein motortje in de celwand: zo zwemt de bacterie naar voedsel toe. De korte haartjes (pili) dienen om zich vast te hechten.

UITLEG · 35.5

Gisten en schimmels

5/24

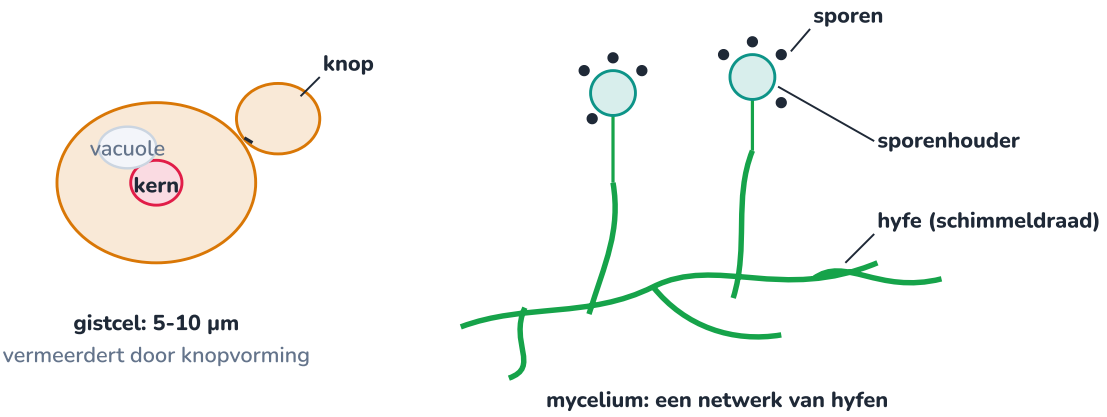
Gisten en schimmels horen samen bij de **fungi**, een eigen rijk naast planten en dieren. Hun cellen hebben een kern, zoals de jouwe, en een celwand van chitine, de stof van het pantser van insecten. Ze hebben geen bladgroen: ze leven van wat anderen maakten, als reducent of als parasiet.

Gisten zijn eencellig. Bakkersgist (*Saccharomyces cerevisiae*) is een ovale cel van 5 tot 10 µm. Ze vermeerdert door **knopvorming**: er groeit een uitstulping die een kopie van de kern krijgt en loslaat. Met zuurstof ademt gist suiker volledig af; zonder zuurstof vergist ze het tot alcohol en koolstofdioxide (hoofdstuk 34).

Schimmels groeien met lange draden van enkele µm dik, de **hyfen**. Samen vormen ze een netwerk, het **mycelium**: het pluis op oud brood, of onder de grond het netwerk waarvan een paddenstoel alleen de vrucht is. Schimmels verspreiden zich met **sporen**: piepkleine cellen die met de lucht meewaaien en jaren kunnen overleven. Daarom groeit er op elke vochtige boterham na een paar dagen schimmel.

Nuttig en schadelijk

Penicillium levert penicilline en de blauwe aders in blauwe kaas. *Aspergillus* maakt sojasaus en citroenzuur, maar sommige soorten maken giftige stoffen op beschimmelde noten. *Botrytis*, de grauwe schimmel, laat aardbeien en sla rotten als de lucht vochtig is: een belager van de kweekruimte van Habitat.



Links een gistcel met een knop. Rechts schimmeldraden met sporenhouders: elk bolletje bevat duizenden sporen.

UITLEG · 35.6

Virussen: leven lenen

6/24

Een **virus** is geen cel. Het is een pakketje erfelijk materiaal (DNA of RNA) in een mantel van eiwit, soms met een vetlaagje eromheen. Het heeft geen eigen stofwisseling en kan zich niet zelf delen. Het moet een levende cel binnendringen en die cel dwingen om nieuwe virussen te maken:

1. het virus hecht aan een cel waarvan het de buitenkant 'herkent' (elk virus heeft zijn eigen gastheren);
2. het erfelijk materiaal gaat de cel binnen;
3. de enzymen en ribosomen van de cel kopiëren het virus-DNA of -RNA en bouwen nieuwe mantels;
4. honderden nieuwe virussen komen vrij, vaak doordat de cel openbarst en sterft.

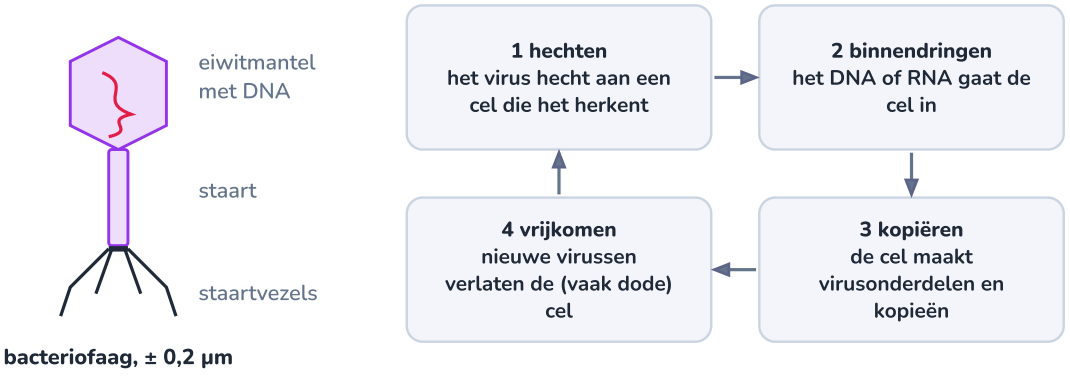
Er zijn virussen voor elke vorm van leven. **Bacteriofagen** ('bacterie-eters') infecteren bacteriën; het tabaksmozaïekvirus van Beijerinck infecteert planten; griep, verkoudheid, norovirus en het coronavirus infecteren mensen.

Antibiotica helpen tegen griep

Antibiotica werken op onderdelen die alleen bacteriën hebben: de celwand, de bacteriële ribosomen. Een virus heeft die niet. Tegen virussen helpen je afweer, vaccins, enkele antivirale middelen, en hygiëne. Wie antibiotica slikt tegen een verkoudheid, helpt alleen de bacteriën in zijn lichaam om resistent te worden.

Leeft een virus?

Volgens de levenskenmerken van F26 niet: geen stofwisseling, geen groei, geen voortplanting op eigen kracht. Maar het heeft erfelijk materiaal, het verandert door mutaties en het evolueert. Biologen noemen virussen daarom soms 'op de grens van het leven'.



Een bacteriofaag spuit zijn DNA in een bacterie. De cel maakt daarna zelf de nieuwe virussen.

UITGEWERKT VOORBEELD · 35.7

Wat zie je met welke microscoop?

7/24

Een lichtmicroscoop vergroot 1000 keer (objectief 100 ×, oculair 10 ×, F26). (a) Hoe groot lijkt een *E. coli* van 2 µm, en een gistcel van 8 µm? (b) Hoe groot zou een griepvirus van 100 nm lijken, en waarom zie je het toch niet? (c) Welke vergroting heb je nodig om het virus 5 mm groot te zien?

GEGEVEN

- vergroting $V = 1000$
- $\ell_{\text{E. coli}} = 2\,\mu\text{m}$, $\ell_{\text{gist}} = 8\,\mu\text{m}$, $\ell_{\text{virus}} = 100\,\text{nm}$
- oplossend vermogen lichtmicroscoop $\approx 0{,}2\,\mu\text{m}$

GEVRAAGD

- schijnbare grootte
- benodigde vergroting

OPLOSSING

1. (a) Schijnbare grootte = echte grootte × vergroting: $2\,\mu\text{m} \cdot 1000 = 2000\,\mu\text{m} = 2\,\text{mm}$; gist: $8\,\text{mm}$. De bacterie is een streepje, de gistcel een duidelijke ovaal met een knop.
2. (b) $100\,\text{nm} \cdot 1000 = 0{,}1\,\text{mm}$. Dat zou je net kunnen zien, maar het virus is kleiner dan de halve golflengte van licht (± 0,2 µm): de microscoop beeldt het niet scherp af. Meer vergroten geeft alleen een grotere vlek.
3. (c) $V = 5\,\text{mm}/100\,\text{nm} = 5 \cdot 10^{-3}/1 \cdot 10^{-7} = 50\,000$. Dat haalt een elektronenmicroscoop moeiteloos: elektronen hebben een veel kleinere golflengte dan licht.

ANTWOORD

De bacterie lijkt 2 mm, de gistcel 8 mm. Het virus zou 0,1 mm lijken, maar ligt onder het oplossend vermogen; met een elektronenmicroscoop die 50 000 keer vergroot, wordt het 5 mm.

Klopt dit? Vergroting en oplossend vermogen zijn twee verschillende dingen. Een foto van een bacterie kan je op je scherm eindeloos vergroten, maar je ziet er niet meer details door: dat heet 'lege vergroting'.

UITLEG · 35.8

Steriel werken en kweken

8/24

Ontstaan micro-organismen vanzelf in bedorven voedsel? Rond 1860 beslechtte **Louis Pasteur** die vraag met een slimme kolf. Hij kookte bouillon in een kolf met een lange, S-vormige hals. Lucht kon vrij in en uit, maar stof en kiemen bleven in de bocht hangen. Het bouillon bleef maandenlang helder. Brak

hij de hals af, dan werd het binnen enkele dagen troebel. Leven komt van leven, ook bij micro-organismen.

Wie micro-organismen kweekt, wil precies weten wat er groeit. Dat vraagt **steriel** (kiemvrij) en **aseptisch** werken:

- **Steriliseren**: voedingsbodems en glaswerk 15 minuten op 121 °C in een autoclaaf (een snelkookpan doet hetzelfde: stoom onder ongeveer 1 bar overdruk). Ook sporen sterven dan.
- **Een voedingsbodem**: een oplossing van voedingsstoffen, gestold met **agar** (uit zeewier). Het idee kwam in 1881 van Fanny Hesse, die in het labo van Robert Koch werkte; Julius Petri bedacht in 1887 de platte schaal met deksel. Op agar groeit elke cel uit tot een zichtbare **kolonie**.
- **Aseptisch werken**: werkblad ontsmetten met ethanol 70 %, handen wassen, het deksel zo kort en zo schuin mogelijk openen, een entoog uitgloeien in een vlam.
- **Kweken** bij een gekozen temperatuur, en **vernietigen** na afloop: opnieuw steriliseren.

Elke kweek kan een ziekteverwekker bevatten

Wat je uit de omgeving kweekt, ken je niet. Daarom gelden op school vaste regels: platen dicht met twee of vier korte stroken plakband (niet helemaal rondom: dan groeien juist de gevaarlijke bacteriën die geen zuurstof willen), kweken bij 20 à 25 °C (niet bij 37 °C, de temperatuur van je lichaam), **nooit openen** na het kweken, en de leerkracht steriliseert de gesloten platen voor ze weggegooid worden.

Links bleef het bouillon helder: de lucht kon erbij, de kiemen niet. Rechts viel stof met kiemen erin.

PROEF · 35.9

Kiemen op agar: waar zitten de meeste bacteriën?

9/24

ONDERZOEKSVRAAG

Hoeveel micro-organismen draag je mee op je handen, en helpt handen wassen? Zit er meer op je smartphone of op een deurklink?

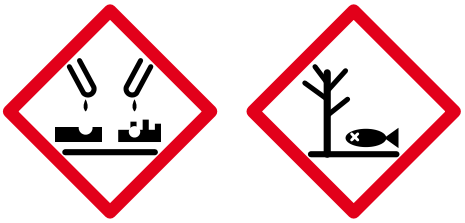
MATERIAAL

- zes steriele petrischalen met voedingsagar (kant-en-klaar gekocht)
- steriele wattenstaafjes
- plakband, watervaste stift
- zeep, ethanol 70 % voor het werkblad
- een plek bij 20-25 °C (of een broedstoof, ingesteld op 25 °C)

Veiligheid

Werkblad eerst ontsmetten met ethanol 70 % (ontvlambaar: geen vlam in de buurt). Schrijf op de bodem, niet op het deksel. Plak elke plaat na het beënten dicht met twee of vier korte stroken plakband aan tegenoverliggende kanten. Leg ze omgekeerd weg bij 20-25 °C. Na het kweken open je ze nooit meer: je telt door het deksel. De leerkracht steriliseert de gesloten platen (snelkookpan of autoclaaf, 121 °C, 15 min) of legt ze een nacht in javel voor ze in het afval gaan. Handen wassen na elke stap.

ethanol 70 %



javel (naargelang de sterkte ook GHS07 in plaats van GHS05)

Javel nooit mengen met zuur of ammoniak: dan ontstaat giftig gas.

Ethanol 70 % is ontvlambaar. Javel is bijtend of irriterend en giftig voor waterleven.

WERKWIJZE

1. Nummer de bodems 1 tot 6. Plaat 1 blijft dicht: de blanco.
2. Plaat 2: deksel 10 minuten van de plaat, open in de klas.
3. Plaat 3: druk drie ongewassen vingertoppen zacht op de agar. Plaat 4: dezelfde vingers, na 30 s wassen met zeep en water.
4. Plaat 5 en 6: wrijf een steriel wattenstaafje over een smartphonescherm en over een deurklink, en daarna zigzag over de agar.
5. Plak alle platen dicht, leg ze omgekeerd weg en tel na 2 à 3 dagen de kolonies door het deksel.

WAARNEMING

plaat	wat	kolonies na 3 dagen
1	niet geopend (blanco)	0
2	10 minuten open in de klas	6
3	vingertoppen, ongewassen	48
4	vingertoppen na 30 s wassen met zeep	9
5	wattenstaafje over een smartphonescherm	21
6	wattenstaafje over een deurklink	35

Voorbeeldresultaat van een klas, telkens één plaat.

VERKLARING

Elke kolonie groeide uit één cel of één spore: een kolonievormende eenheid (KVE). Kolonies met een andere kleur, vorm of rand zijn andere soorten: glanzende ronde stipjes zijn meestal bacteriën, pluizige vlekken schimmels. De blanco bewijst dat de platen en de werkwijze kiemvrij waren: anders kan je niets besluiten. Handen wassen verminderde het aantal hier met 80 %. Eén plaat per situatie is een steekproef van één (hoofdstuk 2): wie een besluit wil, herhaalt.

Tellen door het deksel: de platen blijven dicht.

UITLEG · 35.10

Groei: delen, verdubbelen, en de groeicurve

10/24

Een bacterie groeit tot ze ongeveer twee keer zo groot is, kopieert haar DNA en deelt zich in tweeën. Bij *E. coli* in een rijke voedingsbodem bij 37 °C duurt dat ongeveer 20 minuten. Zo lang er voedsel is, verdubbelt het aantal dus in gelijke tijden: exponentiële groei (hoofdstuk 5).

EXPONENTIËLE FASE

$$N(t) = N_0 \cdot 2^{t/t_d}$$

N_0 het aantal bij de start, t_d de **generatietijd** (verdubbelingstijd). Op een logaritmische as is dit een rechte.

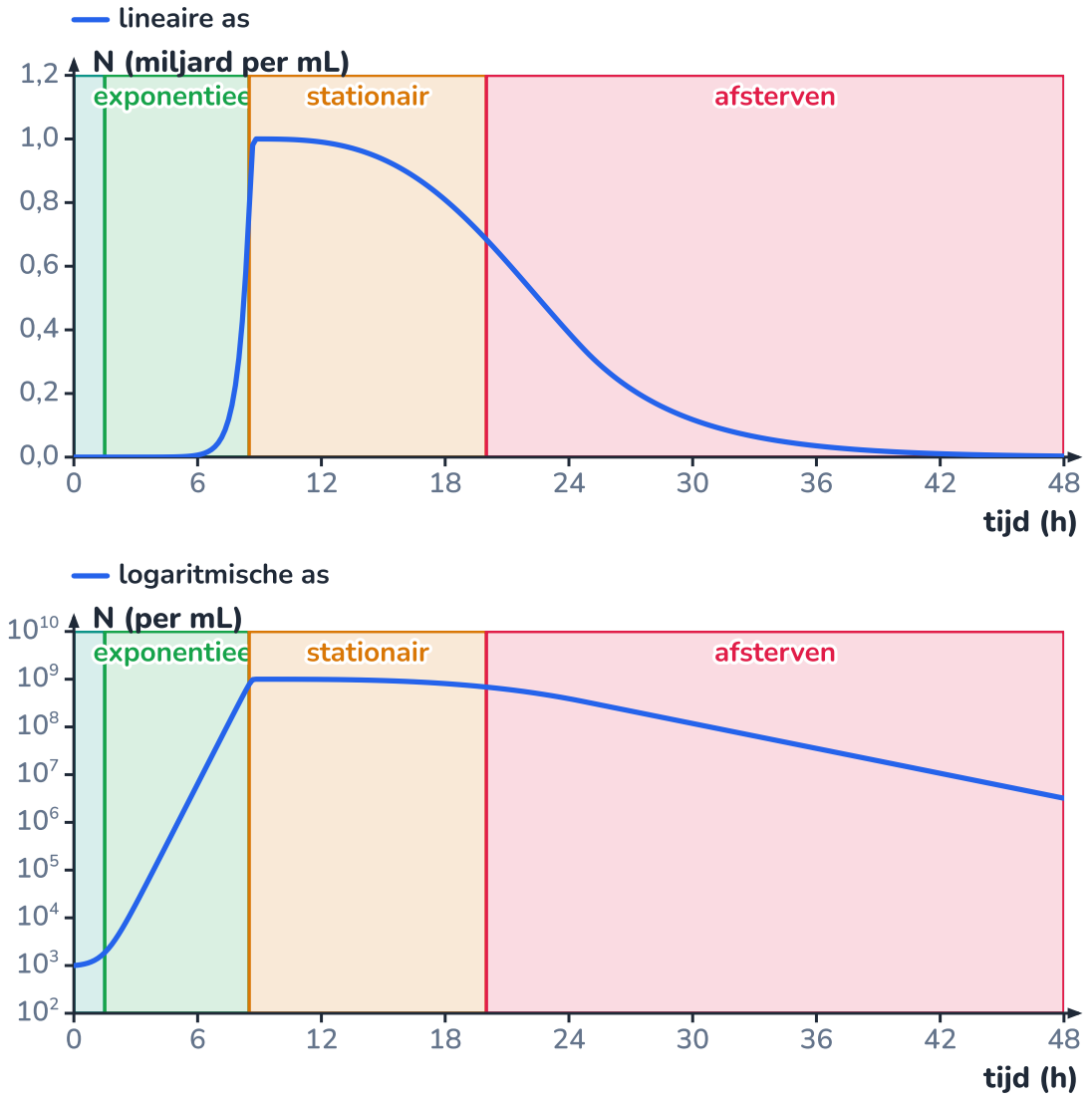
In een afgesloten cultuur, zoals een buisje bouillon, blijft het daar niet bij. Een **groeicurve** telt vier fasen:

1. **aanloopfase** (lag-fase): de bacteriën passen zich aan hun nieuwe omgeving aan en maken de enzymen die ze nodig hebben. Het aantal verandert nog nauwelijks;
2. **exponentiële fase** (log-fase): elke generatietijd een verdubbeling;
3. **stationaire fase**: het voedsel raakt op en afvalstoffen hopen zich op. Er komen evenveel bacteriën bij als er sterven;
4. **afstervingsfase**: er sterven er meer dan er bijkomen, en het aantal daalt, vaak ook exponentieel.

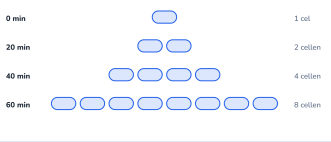
De figuur toont één en dezelfde groeicurve twee keer, berekend met het programma op het codeblad. Op de lineaire as lijkt er acht uur lang niets te gebeuren en dan plots alles. Op de logaritmische as zie je de aanloop, de rechte van de exponentiële fase, het plafond van ongeveer 10^9 per mL en de trage daling.

Een model met een geldigheidsgebied

Het model gebruikt gekozen waarden die passen bij *E. coli* bij 37 °C in een buisje bouillon. Echte curven verschillen per soort, per voedingsbodem en per temperatuur. In hoofdstuk E31 leer je het Monod-model, waarmee bioprocesingenieurs zo'n curve voor een bioreactor voorspellen.



Boven lineair, onder logaritmisch: de exponentiële fase is onderaan een rechte (model uit groei.py).



Na één uur zijn er uit één bacterie acht; na acht uur, zonder rem, meer dan tien miljoen.

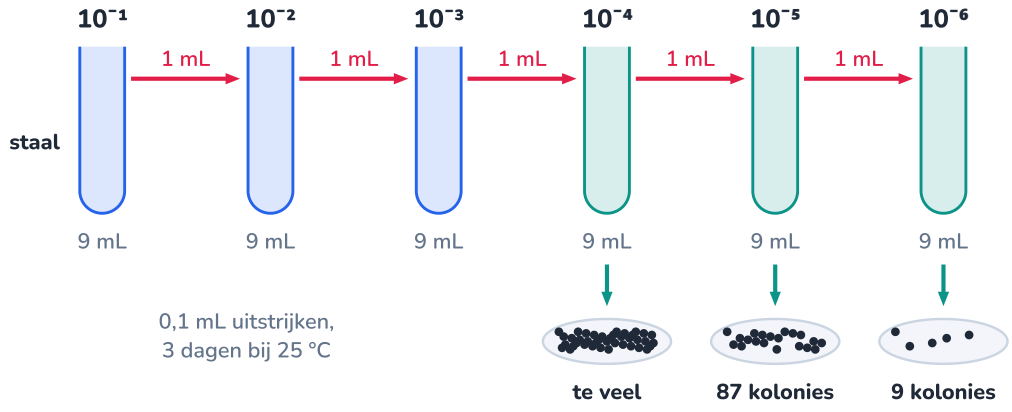
UITGEWERKT VOORBEELD · 35.11

Bacteriën tellen met een verdunningsreeks

11/24

Hoeveel bacteriën zitten er in een staal yoghurtdrank? Je verdunt telkens 1 mL in 9 mL steriel water, zes keer na elkaar. Van de laatste drie buisjes strijk je 0,1 mL uit op een plaat. Na drie dagen tel je: te veel

om te tellen, 87 kolonies en 9 kolonies. Hoeveel KVE per mL bevat het staal?



Zes keer tien keer verdunnen; alleen de middelste van de drie platen is goed te tellen.

GEGEVEN

- verdunningen 10^{-1} tot 10^{-6}
- 0,1 mL per plaat
- 10^{-4}: te veel; 10^{-5}: 87 kolonies; 10^{-6}: 9 kolonies

GEVRAAGD

- KVE per mL in het staal

OPLOSSING

1. Elke stap verdunt tien keer (1 mL in samen 10 mL). Na vijf stappen zit er van elke mL staal nog 10^{-5} mL in elke mL van het buisje.
2. Kies de plaat met 30 tot 300 kolonies: minder is te onnauwkeurig (toeval), meer groeit samen. Hier de plaat van 10^{-5} met 87 kolonies.
3. 87 kolonies in 0,1 mL van een 10^{-5}-verdunning: $N = \dfrac{87}{0{,}1\,\text{mL} \cdot 10^{-5}} = 8{,}7 \cdot 10^{7}\ \text{KVE/mL}$.
4. Controle met de volgende plaat: $9/(0{,}1 \cdot 10^{-6}) = 9 \cdot 10^{7}$. Dat klopt binnen de toevalsfout van een klein aantal kolonies.

ANTWOORD

Het staal bevat ongeveer $8{,}7 \cdot 10^{7}$ KVE per mL.

Klopt dit? Dat is een gewone waarde voor een gefermenteerde melkdrank: meer dan een miljard levende bacteriën per flesje van 100 mL. Een getal zonder eenheid of zonder verdunning zegt niets: schrijf altijd KVE per mL (of per g).

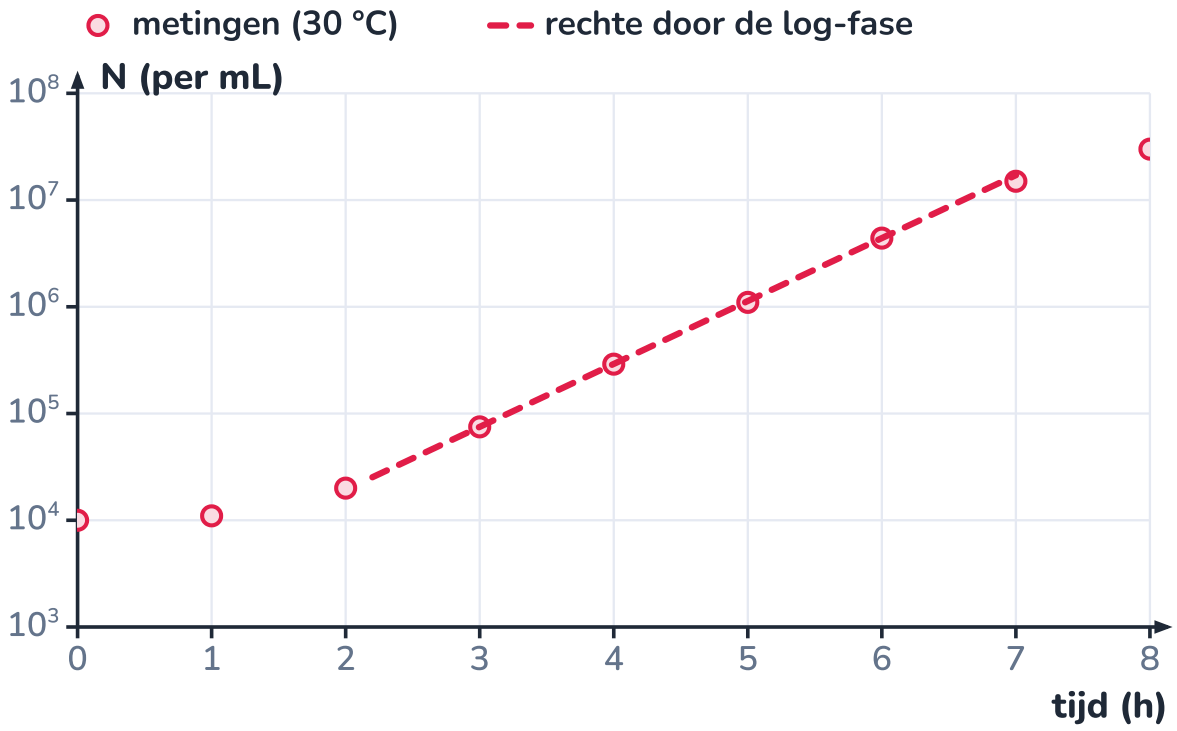
UITGEWERKT VOORBEELD · 35.12

De generatietijd uit een meetreeks

12/24

Een groep volgt een cultuur bij 30 °C met een troebelheidsmeter die ze eerst geijkt heeft met plaattellingen. Bepaal de generatietijd in de exponentiële fase.

t (h)	0	1	2	3	4	5	6	7	8
N (per mL)	1,0 · 10^4	1,1 · 10^4	2,0 · 10^4	7,5 · 10^4	2,9 · 10^5	1,1 · 10^6	4,4 · 10^6	1,5 · 10^7	3,0 · 10^7



Op een logaritmische as vallen de middelste punten op een rechte: dat is de exponentiële fase.

GEGEVEN

- de meetreeks
- exponentieel: $N = N_0 \cdot 2^{t/t_d}$

GEVRAAGD

- generatietijd t_d

OPLOSSING

1. Zet de metingen op een logaritmische as: tussen 3 en 6 uur liggen ze op een rechte. Daar is de groei exponentieel; daarvoor zit de aanloop, daarna begint de afvlakking.
2. Van 3 tot 6 uur: N gaat van $7{,}5 \cdot 10^4$ naar $4{,}4 \cdot 10^6$, een factor $4{,}4 \cdot 10^6 / 7{,}5 \cdot 10^4 = 58{,}7$.
3. Aantal verdubbelingen: $2^g = 58{,}7$, dus $g = \dfrac{\log 58{,}7}{\log 2} = \dfrac{1{,}769}{0{,}301} = 5{,}87$ (hoofdstuk 5).
4. $t_d = \dfrac{3\,\text{h}}{5{,}87} = 0{,}51\,\text{h} = 31\,\text{min}.$

ANTWOORD

De generatietijd bij 30 °C is ongeveer 31 minuten.

Klopt dit? Dat past bij het temperatuurmodel verderop: bij 30 °C verdubbelt E. coli ongeveer om het halfuur, bij 37 °C om de 20 minuten. Neem nooit de eerste of de laatste punten mee: daar is de groei niet exponentieel, en je generatietijd wordt te lang.

CODE · 35.13

Python: de groeicurve simuleren

13/24

Het model achter de grafiek van de groeicurve, in één functie. Het rekent in stappen van één minuut (de methode van Euler, E4): per stap komen er $\mu \cdot N \cdot \Delta t$ bacteriën bij en sterven er enkele. Drie ideeën zitten erin:

- `klaar` groeit van 0 naar 1: de **aanloop**;

- `voedsel / (0.05 + voedsel)` is de verzadigingskromme van hoofdstuk 34: bij veel voedsel volle snelheid, als het opraakt valt de groei stil;
- `honger` telt hoe lang het voedsel al op is; hoe langer, hoe meer bacteriën sterven.

Python **groei.py** 35 regels

```python
# groei.py - de groeicurve van een bacteriecultuur: een eenvoudig model (stap van 1 minuut)
import math

MU_MAX = math.log(2) / (20 / 60)   # per uur: E. coli bij 37 graden verdubbelt elke 20 minuten
N_MAX = 1e9                        # bacteriën per mL als al het voedsel op is
AANLOOP = 1.5                      # uur om zich aan te passen (aanloopfase)
STERFTE = 0.2                      # per uur, als de honger lang genoeg duurt


def groeicurve(n0=1e3, uren=48, dt=1 / 60):
    """Geeft een lijst (t in uur, N per mL) terug."""
    n, voedsel, honger = n0, 1.0, 0.0           # voedsel: 1 = vol, 0 = op
    klaar_q = 1 / (math.exp(MU_MAX * AANLOOP) - 1)
    punten = [(0.0, n)]
    for stap in range(1, round(uren / dt) + 1):
        klaar = klaar_q / (1 + klaar_q)          # 0 bij de start, 1 als de cel is aangepast
        mu = MU_MAX * klaar * voedsel / (0.05 + voedsel)
        erbij = mu * n * dt                      # nieuwe bacteriën in deze stap
        voedsel = max(0.0, voedsel - erbij / N_MAX)
        klaar_q = klaar_q * math.exp(MU_MAX * dt)
        if voedsel < 0.01:
            honger += dt                         # hoe lang is het voedsel al op?
        sterf = STERFTE * min(1.0, honger / 16) ** 2 * n * dt
        n = n + erbij - sterf
        punten.append((stap * dt, n))
    return punten


if __name__ == "__main__":
    curve = groeicurve()
    print("  t (h)    N per mL    log N")
    for t, n in curve[::180]:                    # elke 3 uur
        print(f"{t:6.0f}    {n:10.2e}   {math.log10(n):5.2f}")
    top_t, top_n = max(curve, key=lambda p: p[1])
    print(f"hoogste waarde {top_n:.2e} per mL na {top_t:.1f} h")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
  t (h)    N per mL    log N
     0    1.00e+03    3.00
     3    1.93e+04    4.29
     6    6.42e+06    6.81
     9    1.00e+09    9.00
    12    9.90e+08    9.00
    15    9.36e+08    8.97
    18    8.09e+08    8.91
    21    6.13e+08    8.79
    24    3.90e+08    8.59
    27    2.15e+08    8.33
    30    1.18e+08    8.07
    33    6.46e+07    7.81
    36    3.54e+07    7.55
    39    1.94e+07    7.29
    42    1.07e+07    7.03
    45    5.84e+06    6.77
```

```
    48     3.20e+06    6.51
hoogste waarde 1.00e+09 per mL na 8.8 h
```

Na 8,8 uur is het plafond bereikt; na 48 uur zijn er nog 3,2 · 10^6 per mL over. In opdracht 35.12 laat je de bacteriën koeler groeien en start je met meer of minder bacteriën.

UITLEG · 35.14

Temperatuur en hygiëne: waarom de koelkast werkt

14/24

Bacteriën groeien met enzymen, en dus heeft hun groei een temperatuuroptimum (hoofdstuk 34). Elke soort heeft een minimum, een optimum en een maximum. De meeste ziekteverwekkers zijn aangepast aan ons lichaam: ze groeien het snelst tussen 30 en 40 °C. De figuur toont een eenvoudig **model** (van de Australische voedingsmicrobioloog David Ratkowsky) voor twee soorten.

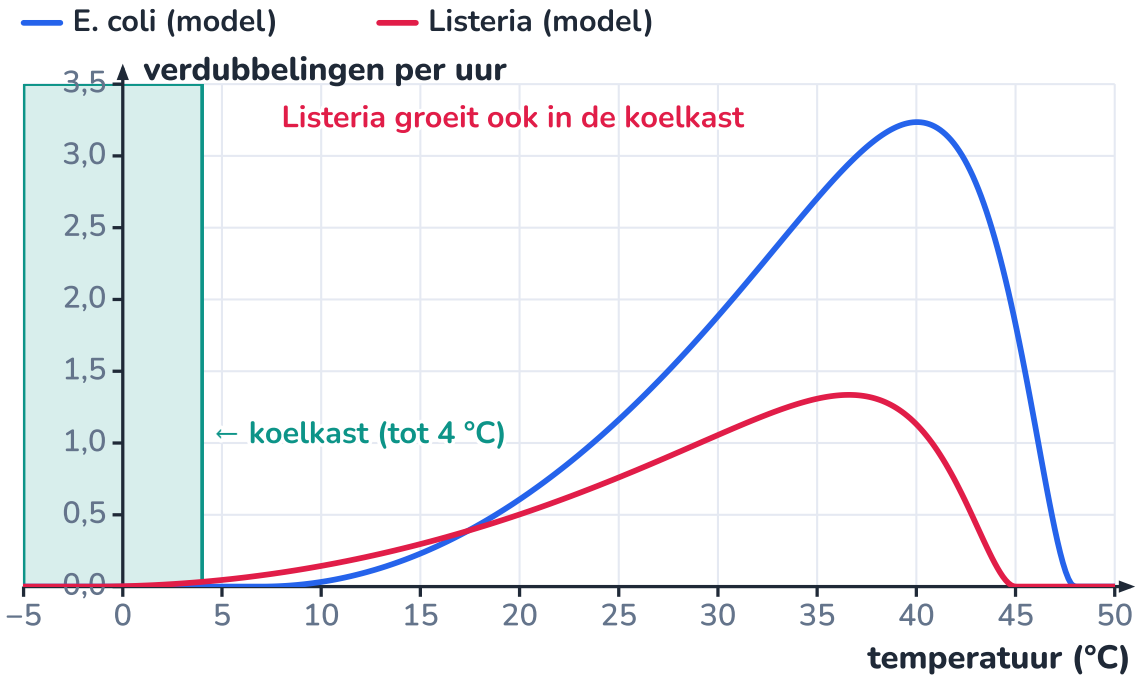
- *E. coli* groeit niet onder ongeveer 7 °C. In de koelkast (4 °C) staat ze stil, bij 20 °C verdubbelt ze om de 1,6 uur, bij 37 °C om de 20 minuten.
- *Listeria monocytogenes* groeit ook in de koelkast, traag: bij 4 °C verdubbelt ze ongeveer elke 30 uur. Daarom is ze gevaarlijk in kant-en-klaar voedsel dat lang koel bewaard wordt: gerookte vis, paté, voorgesneden sla.

Wat de voedselveiligheid daaruit haalt:

- **koelen** (ten hoogste 4 à 7 °C) vertraagt de groei sterk, maar doodt niets; **invriezen** legt de groei stil, maar de meeste bacteriën overleven het;
- tussen ongeveer 7 en 60 °C groeien ziekteverwekkers vlot: daarom houd je voedsel niet uren lauw;
- **pasteuriseren**: kort verhitten, bijvoorbeeld melk 15 seconden op 72 °C, doodt de meeste ziekteverwekkers (hun enzymen denatureren) zonder de melk te koken;
- **steriliseren** doodt ook sporen: 15 minuten op 121 °C (conserven, autoclaaf);
- zuur (pH onder ongeveer 4,5), veel zout of veel suiker remmen de groei: yoghurt, zuurkool, pekelvlees, confituur.

De koelketen

Voedsel dat van de winkel naar huis anderhalf uur in een warme auto ligt, is niet meteen gevaarlijk, maar het heeft een deel van zijn 'groeitijd' opgebruikt. Houd de **koelketen** zo ononderbroken mogelijk: koelwaren als laatste in het winkelwagentje en thuis meteen in de koelkast. Een koelkast die op 7 °C staat in plaats van 4 °C, laat *Listeria* in het model meer dan twee keer zo snel groeien.



Model: E. coli staat stil in de koelkast, Listeria groeit er traag door. Beide hebben hun optimum rond 37-40 °C.

UITLEG · 35.15

Fermentatie: brood, bier, yoghurt en azijn

15/24

Zonder zuurstof halen micro-organismen een klein beetje energie uit suiker met **fermentatie**: 2 ATP per glucose in plaats van ongeveer 30 (hoofdstuk 34). Wat voor hen afval is, is voor ons voedsel.

ALCOHOLGISTING (GIST)

$$C_6H_{12}O_6 \longrightarrow 2\,C_2H_5OH + 2\,CO_2$$

In **brood** blaast de koolstofdioxide het deeg op; de alcohol verdampt in de oven. In **bier** en **wijn** blijft de alcohol.

MELKZUURGISTING (MELKZUURBACTERIËN)

$$C_6H_{12}O_6 \longrightarrow 2\,C_3H_6O_3$$

In yoghurt, kaas, zuurkool en zuurdesem. Melkzuur verlaagt de pH, en dat houdt andere bacteriën tegen: een oude manier om voedsel te bewaren.

AZIJNZUURVORMING (AZIJNZUURBACTERIËN, MET ZUURSTOF)

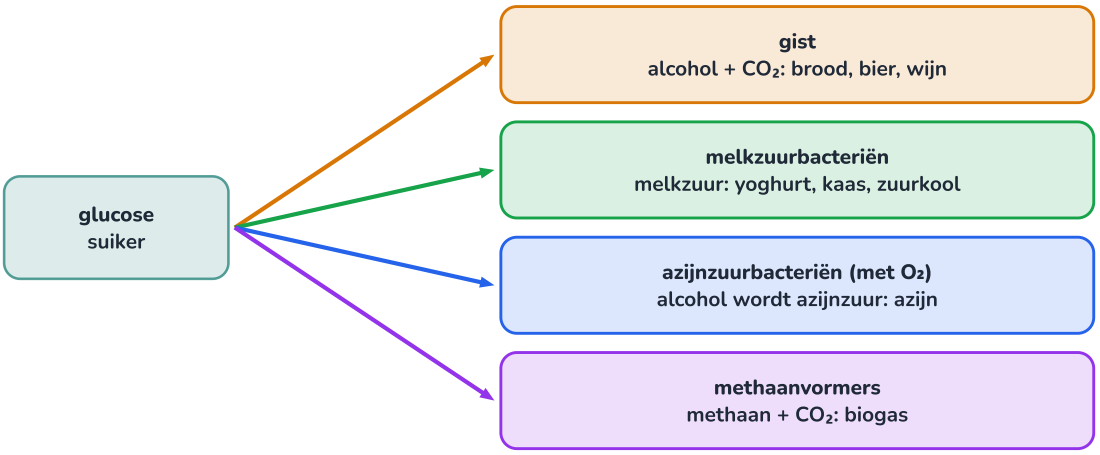
$$C_2H_5OH + O_2 \longrightarrow CH_3COOH + H_2O$$

Wijn of bier die open blijft staan, wordt azijn.

Melk bevat melksuiker (lactose, $C_{12}H_{22}O_{11}$). Melkzuurbacteriën splitsen die eerst met een enzym in twee kleinere suikers en vergisten die tot melkzuur. Samen: $C_{12}H_{22}O_{11} + H_2O \longrightarrow 4\,C_3H_6O_3$.

Lambiek: bier van de lucht

In het Pajottenland en de Zennevallei, rond Brussel, brouwt men **lambiek** met spontane gisting. Na het koken laat de brouwer het wort een nacht afkoelen in een brede, ondiepe bak, het koelschip. Gisten en bacteriën uit de lucht vallen erin. Het bier gist en rijpt daarna een tot drie jaar in houten vaten; geuze is een mengsel van jonge en oude lambiek. Het is het omgekeerde van steriel werken, en het werkt omdat de brouwer weet welke micro-organismen in zijn brouwerij leven.



Dezelfde glucose, vier soorten micro-organismen, vier producten.

PROEF · 35.16

Gist en suiker: koolstofdioxide in een ballon

16/24

ONDERZOEKSVRAAG

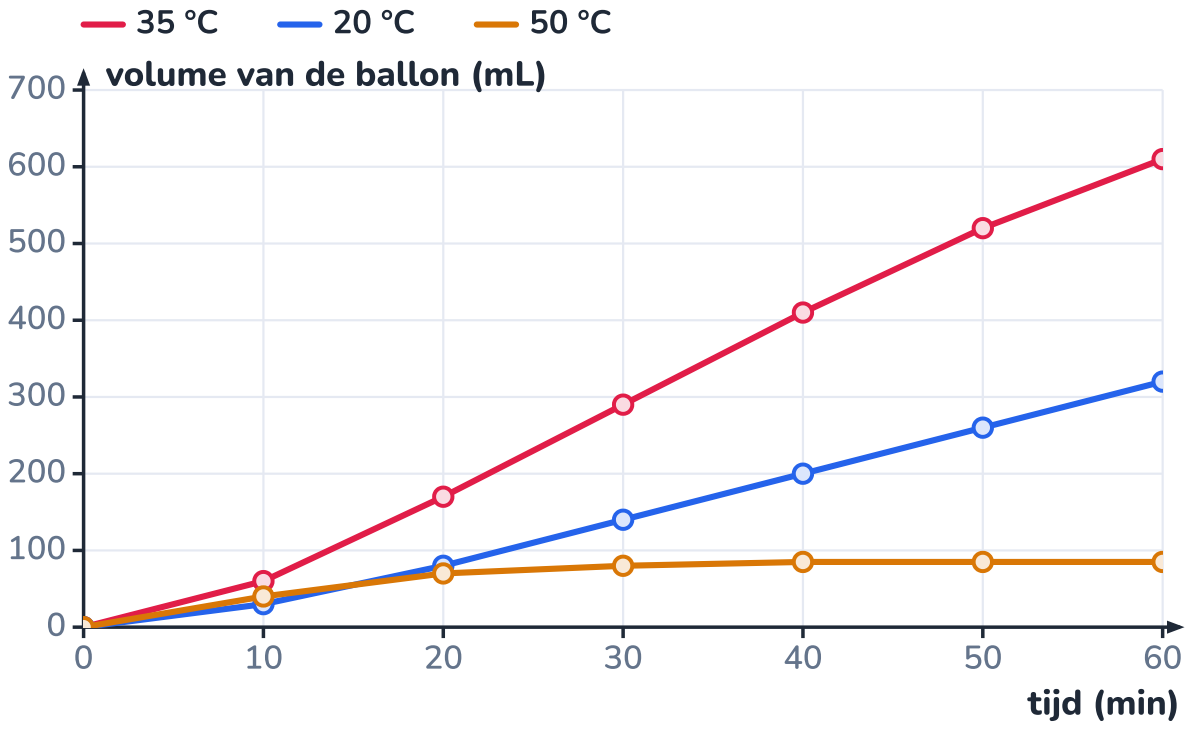
Hoe snel maakt gist koolstofdioxide uit glucose, en hoe hangt dat af van de temperatuur?

MATERIAAL

- drie flesjes of erlenmeyers van 250 mL
- droge bakkersgist (3 × 5 g), glucose (3 × 5,0 g)
- lauw water, drie ballonnen, touwtje en lintmeter
- waterbaden van 20, 35 en 50 °C (bakken met water en een thermometer)

Veiligheid

Geen gevaarlijke stoffen. Maak geen flesjes met een vaste dop dicht: de druk kan oplopen. De ballon is de veiligheidsklep.



Snelst bij 35 °C; bij 50 °C houdt de gisting na een kwartier op.

WERKWIJZE

1. Los in elk flesje 5,0 g glucose op in 100 mL water. Zet de flesjes 5 minuten in hun waterbad.
2. Voeg 5 g gist toe, schud kort en trek er meteen een lege ballon over.
3. Meet om de 10 minuten met het touwtje de omtrek C van de ballon op zijn dikste plek.
4. Bereken het volume als een bol: $V = C^3/(6\pi^2)$. Zet het volume uit tegen de tijd.

WAARNEMING

t (min)	0	10	20	30	40	50	60
20 °C: V (mL)	0	30	80	140	200	260	320
35 °C: V (mL)	0	60	170	290	410	520	610
50 °C: V (mL)	0	40	70	80	85	85	85
35 °C: C (cm)	0,0	15,3	21,6	25,8	29,0	31,3	33,1

Voorbeeldmeting. De volumes zijn uit de omtrek berekend; de ballon begint pas merkbaar op te zwellen na een paar minuten.

VERKLARING

Bij 35 °C werkt gist het best: haar enzymen zitten dicht bij hun optimum. Bij 20 °C gaat het ongeveer half zo snel. Bij 50 °C zwelt de ballon eerst op, maar na een kwartier stopt de gisting: de enzymen denatureren en de gistcellen sterven (hoofdstuk 34). Het volgende voorbeeld rekent uit hoeveel koolstofdioxide er maximaal kan ontstaan.

De ballon vangt de koolstofdioxide op; een touwtje meet hoe dik hij wordt.

UITGEWERKT VOORBEELD · 35.17

Hoeveel koolstofdioxide past er in de ballon?

17/24

In de proef vergist 5,0 g glucose. Hoeveel gram en hoeveel liter koolstofdioxide kan er maximaal ontstaan bij 35 °C? Welk deel is er na 60 minuten gevormd?

GEGEVEN

- $\mathrm{C_6H_{12}O_6} \longrightarrow 2\,\mathrm{C_2H_5OH} + 2\,\mathrm{CO_2}$
- $m = 5{,}0\,\mathrm{g}$, $M(\mathrm{C_6H_{12}O_6}) = 180{,}16\,\mathrm{g/mol}$
- $V_m = 25{,}3\,\mathrm{L/mol}$ bij 35 °C (hoofdstuk 14: evenredig met de absolute temperatuur)
- gemeten: 610 mL na 60 min

GEVRAAGD

- $m(\mathrm{CO_2})$, $V(\mathrm{CO_2})$, het deel na 60 min

OPLOSSING

1. $n(\text{glucose}) = 5{,}0/180{,}16 = 0{,}0278\,\text{mol}$; molverhouding 1 : 2, dus $n(\mathrm{CO_2}) = 0{,}0555\,\text{mol}$.
2. $m = 0{,}0555 \cdot 44{,}01 = 2{,}44\,\text{g}$; $V = 0{,}0555 \cdot 25{,}3 = 1{,}40\,\text{L}$.
3. Na 60 minuten: $0{,}61/1{,}40 = 0{,}43$, ongeveer 43 % van het maximum.

ANTWOORD

Maximaal 2,4 g of 1,4 L koolstofdioxide. Na een uur is er ongeveer 43 % van gevormd.

Klopt dit? Het echte volume is kleiner dan wat de reactie maakt: een deel van de koolstofdioxide lost op in het water (in hoofdstuk 15 zag je dat er zo koolzuur ontstaat), en de ballon zelf drukt het gas een beetje samen. De gist gebruikt ook wat glucose om te groeien. Wie het hele maximum wil halen, moet uren wachten.

PROEF · 35.18

Yoghurt maken en de pH volgen

18/24

ONDERZOEKSVRAAG

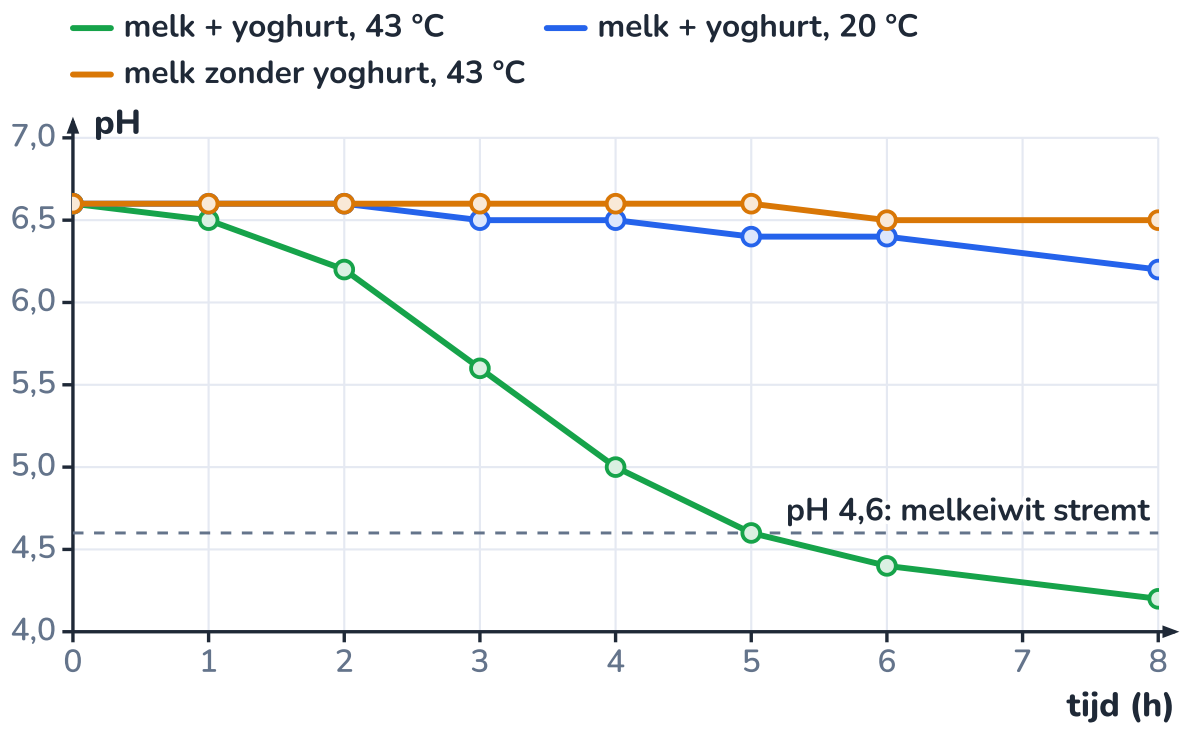
Wat doen yoghurtbacteriën met melk, en hoe snel? Wat is de rol van de temperatuur?

MATERIAAL

- 1 L volle melk, 3 eetlepels natuuryoghurt (met levende culturen)
- drie schone potjes met deksel
- thermosfles of yoghurtmaker (43 °C), thermometer
- pH-meter (geijkt, hoofdstuk 15) of pH-papier

Veiligheid

Werk met schone potjes en lepels. Proef alleen van de yoghurt als alles in de keuken of een voedselveilig lokaal gebeurde, niet in het labo. Wat raar ruikt of kleurt, gaat weg.



Met yoghurtbacteriën bij 43 °C zakt de pH in 5 uur onder 4,6: de melk wordt dik.

WERKWIJZE

1. Verwarm de melk tot net onder de kook en laat ze afkoelen tot 43 °C: zo sterven de meeste bacteriën die er al in zaten.
2. Potje A: melk + 1 eetlepel yoghurt, bij 43 °C. Potje B: hetzelfde, bij kamertemperatuur (20 °C). Potje C: melk zonder yoghurt, bij 43 °C.
3. Meet elk uur de pH, en kijk of de melk dik wordt (kantel het potje).

WAARNEMING

t (h)	0	1	2	3	4	5	6	8
A: 43 °C	6,6	6,5	6,2	5,6	5,0	4,6	4,4	4,2
B: 20 °C	6,6	6,6	6,6	6,5	6,5	6,4	6,4	6,2
C: zonder	6,6	6,6	6,6	6,6	6,6	6,6	6,5	6,5

Voorbeeldmeting. Potje A is na ongeveer 5 uur dik en ruikt fris zuur.

VERKLARING

De yoghurtbacteriën (*Lactobacillus delbrueckii* subsp. *bulgaricus* en *Streptococcus thermophilus*) vergisten melksuiker tot melkzuur. De pH daalt eerst traag (aanloop, weinig bacteriën), dan snel (exponentiële groei), en vlakt af rond 4,2 als het zuur de bacteriën zelf afremt. Rond **pH 4,6** verliest het belangrijkste melkeiwit, caseïne, zijn lading: de eiwitten klonteren samen tot een gel en de melk wordt dik. Bij 20 °C werken de bacteriën, die van warmte houden, veel trager. Zonder yoghurt gebeurt er in 8 uur bijna niets: de verwarmde melk bevatte weinig melkzuurbacteriën.

De pH-meter toont het zuur worden; kantel het potje om te zien of het al gestremd is.

UITLEG · 35.19

Biogas: methaan uit afval

19/24

Als organisch afval (mest, groenteresten, slib) zonder zuurstof in een gesloten tank ligt, breekt een gemeenschap van bacteriën en archaea het af tot **biogas**: 50 tot 70 % methaan, de rest vooral koolstofdioxide. Dat gebeurt in vier stappen, telkens door andere micro-organismen:

METHAANVORMING, SAMENGEVAT VOOR SUIKER

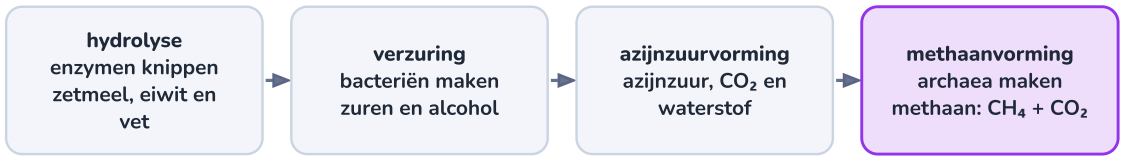
$$\mathrm{C_6H_{12}O_6} \longrightarrow 3\,\mathrm{CH_4} + 3\,\mathrm{CO_2}$$

Anaeroob. Methaan verbrandt later met zuurstof: 890 kJ per mol (hoofdstuk 14).

Opvallend: in de 3 mol methaan zit $3 \cdot 890 = 2670\,\text{kJ}$, dat is 95 % van de 2803 kJ in de glucose. De micro-organismen houden er zelf maar een paar procent van over (ze groeien traag), bijna alle energie blijft in het gas. Daarom is vergisten zo'n goede manier om energie uit nat afval te halen, beter dan verbranden, dat eerst al het water moet verdampen.

In Vlaanderen staan vergisters bij boerderijen, afvalverwerkers en waterzuiveringen. Het gas drijft een motor met een generator aan (warmtekrachtkoppeling) of wordt gezuiverd tot aardgaskwaliteit. Wat overblijft, het **digestaat**, bevat de mineralen en gaat als meststof terug naar het land.

> ⚠ **Methaan mag niet ontsnappen**
>
> Methaan is een sterk broeikasgas: per kilogram warmt het de aarde over twintig jaar tientallen keren sterker op dan koolstofdioxide. Een lekkende vergister of een mesthoop zonder dak doet meer kwaad dan goed. Verbranden maakt er weer koolstofdioxide van, die de plant eerder uit de lucht haalde.



Vier groepen micro-organismen werken na elkaar; alleen de laatste maakt methaan.

Een vergister op een boerderij: de groene koepels vangen het biogas op.

VERHAAL · 35.20

Insuline: van alvleesklieren naar bacteriën

20/24

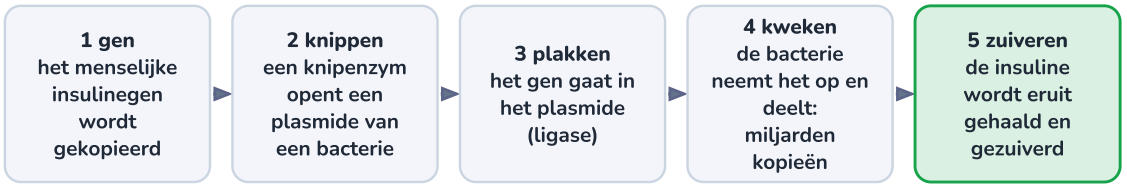
Wie diabetes type 1 heeft, maakt geen **insuline**: het hormoon dat de cellen glucose laat opnemen (E33). Tot 1922 betekende dat een zekere dood. In 1921 haalden de Canadezen **Frederick Banting** en **Charles Best** in Toronto een werkzaam extract uit de alvleesklier van honden. In januari 1922 kreeg de veertienjarige Leonard Thompson, die al bijna dood was, als eerste mens een gezuiverd extract; hij herstelde. Banting en zijn labochef John Macleod kregen in 1923 de Nobelprijs.

Zestig jaar lang kwam insuline uit alvleesklieren van varkens en runderen uit slachthuizen. Dat werkte, maar dierlijke insuline verschilt een of enkele bouwstenen van de onze, en de voorraad hing af van het vlees dat gegeten werd. In 1978 lukte het de Amerikaanse firma Genentech om het menselijke insulinegen in de darmbacterie *E. coli* te brengen. In 1982 werd die insuline goedgekeurd: het eerste geneesmiddel dat door genetisch gewijzigde bacteriën gemaakt wordt. Vandaag maken bacteriën en gisten in grote fermentatietanks alle insuline, en ook groeihormoon, stollingsfactoren, vaccins en het stremsel (chymosine) voor de meeste kaas.

Insuline is een klein eiwit van 51 aminozuren. Een patiënt gebruikt ongeveer 40 eenheden per dag, ongeveer 1,4 mg: ongeveer een halve gram per jaar.

Waarom een bacterie het kan

De genetische code is voor alle levende wezens dezelfde (E29). Een bacterie leest het menselijke gen dus correct en maakt precies hetzelfde eiwit. Omdat bacteriën elke twintig minuten verdubbelen, groeit een fermentatietank op een paar dagen van een buisje tot duizenden liter.



Het gen in een plasmide, het plasmide in een bacterie, de bacterie in een tank: recombinant DNA (E30).

In zulke tanks maken genetisch gewijzigde bacteriën of gisten insuline.

UITLEG · 35.21

Habitat: ziekteverwekkers en hygiëne in de kweekruimte

21/24

In een station wonen mensen, planten en micro-organismen dicht op elkaar, met water dat rondgepompt wordt en lucht die gerecycleerd wordt. Een ziekteverwekker die binnenraakt, vindt er veel kansen, en hulp van buiten komt pas na dertig dagen. Welke gevaren zijn er?

soort	voorbeeld	waar in Habitat?	wat helpt
bacterie	*Salmonella*, *E. coli* (sommige stammen), *Listeria*	op sla via handen, water, insecten; *Listeria* groeit in de koelkast	handen wassen, schoon water, oogst koel en snel opeten
bacterie	*Legionella*	lauw water (25-45 °C) in vaten en leidingen; besmet via fijne druppels	water koud (onder 25 °C) of heet (boven 60 °C), geen nevel, vaten doorspoelen
virus	norovirus (buikgriep), griep	van bewoner tot bewoner, via handen en oppervlakken	handen wassen met zeep (handgel werkt slecht tegen norovirus), zieken van de oogst weg
schimmel	grauwe schimmel (*Botrytis*)	op bladeren bij hoge luchtvochtigheid	ventileren, geen condens op de bladeren, zieke planten verwijderen

soort	voorbeeld	waar in Habitat?	wat helpt
waterschimmel	*Pythium* (wortelrot)	warme voedingsoplossing	reservoir onder 24 °C, ontsmetten tussen twee teelten

Pythium lijkt op een schimmel, maar is er eigenlijk geen: het hoort bij de oömyceten, die nauwer verwant zijn aan bruinwieren. In de tuinbouw heet het toch een waterschimmel.

Voor Habitat leggen we acht hygiëneregels vast voor de kweekruimte:

1. Handen wassen met zeep (30 s) voor je de kweekruimte binnengaat en na het compostvat.
2. Geen eten of drinken in de kweekruimte; wie ziek is (braken, diarree, koorts), oogst niet.
3. Scharen en messen na gebruik reinigen en ontsmetten met ethanol 70 %.
4. Het reservoir blijft lichtdicht en onder 24 °C. Na elke oogst wordt de goot met haar netpotjes geschrobd, ontsmet volgens het etiket van het middel en nagespoeld; om de vijf weken wordt het reservoir helemaal ververst en gereinigd (hoofdstuk 36).
5. Zieke planten meteen verwijderen, in een gesloten zak, niet in het compostvat.
6. Geen water vernevelen en geen lauw water laten staan (*Legionella*); vaten en leidingen die stilstaan, doorspoelen.
7. Oogsten met schone handen of handschoenen, de sla meteen koelen (4 °C) en binnen enkele dagen opeten.
8. Het compostvat staat buiten de kweekruimte, gesloten. Compost die enkele dagen boven 55 °C was, gaat onder de potgrond van de meetkas; nooit iets van het compostvat in het NFT-systeem.

> **Waarom hydroponie hier voordelen heeft**
>
> Zonder aarde komen er geen bodembacteriën en -schimmels mee met de planten, en de oogst is proper. Het nadeel: alles hangt aan hetzelfde water. Eén besmetting in het reservoir bereikt binnen één rondgang van de pomp (12,5 minuten pompen, hoofdstuk 11) alle 48 planten.

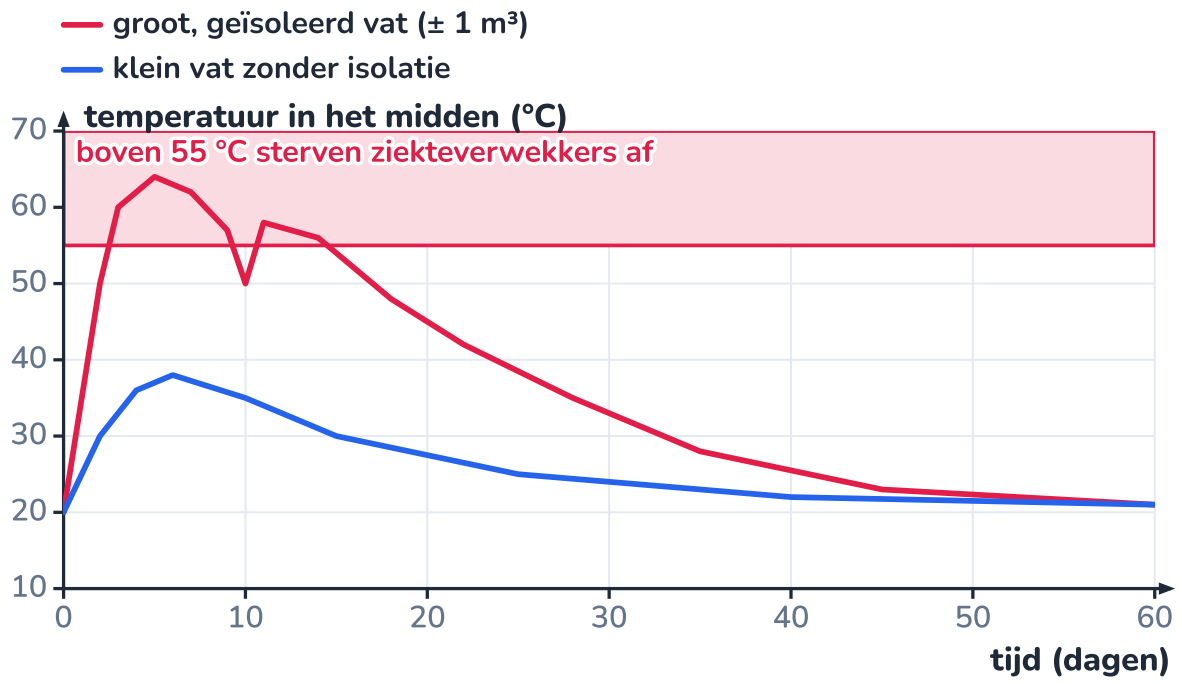
Een wasbak bij de deur: de eenvoudigste bescherming van de kweekruimte.

UITGEWERKT VOORBEELD · 35.22

Habitat: een compostvat voor de plantenresten

22/24

Per teelt blijft er van elke krop ongeveer 50 g over: wortels en buitenbladeren, samen 2,4 kg. Dat is voedsel voor reducenten (F28). Compost werkt het best met een **C/N-verhouding** (massa koolstof gedeeld door massa stikstof) rond 30 en een vochtgehalte van 50 à 60 %. Slaresten zijn nat en stikstofrijk; karton van de bevoorrading is droog en koolstofrijk. (a) Wat is de C/N van slaresten? (b) Hoeveel karton per kg slaresten geeft C/N = 30? (c) Is het vochtgehalte dan goed? (d) Hoeveel koolstofdioxide geeft het composteren van één teelt, als de helft van de koolstof als CO_2 vrijkomt?



Een groot, geïsoleerd vat blijft dagen boven 55 °C (de dip op dag 10: omgeschept); een klein vat zonder isolatie niet.

GEGEVEN

- slaresten: droge stof 5 %; daarvan 40 % koolstof en 4,0 % stikstof
- karton: droge stof 90 %; daarvan 42 % koolstof en 0,14 % stikstof
- rekenwaarden, typische grootte

GEVRAAGD

- C/N van slaresten
- kg karton per kg slaresten
- vochtgehalte
- g CO_2 per teelt

OPLOSSING

1 (a) In 1 kg slaresten: 50 g droge stof, met 20 g C en 2,0 g N. $C/N = 20/2{,}0 = 10$: te veel stikstof. Het teveel ontsnapt als ammoniak (het stinkt).

2 (b) 1 kg karton bevat $900 \cdot 0{,}42 = 378\,\text{g}$ C en $900 \cdot 0{,}0014 = 1{,}26\,\text{g}$ N. Met m kg karton: $\frac{20 + 378\,m}{2{,}0 + 1{,}26\,m} = 30$, dus $20 + 378\,m = 60 + 37{,}8\,m$ en $m = 40/340{,}2 = 0{,}118\,\text{kg}$: ongeveer 118 g karton per kg slaresten.

3 (c) Water: $950 + 0{,}1 \cdot 118 = 962\,\text{g}$ in $1118\,\text{g}$: 86 %. Veel te nat: het vat wordt zuurstofloos en begint te rotten in plaats van te composteren.

4 (d) Koolstof in de slaresten van één teelt: $2{,}4 \cdot 50\,\text{g} \cdot 0{,}40 = 48\,\text{g}$; de helft wordt CO_2: $24/12{,}01 \cdot 44{,}01 = 88\,\text{g}$ per teelt, 2,5 g per dag.

ANTWOORD

Slaresten hebben C/N = 10. Met 118 g karton per kg haal je C/N = 30, maar het mengsel bevat dan nog 86 % water. Je hebt dus ook droog materiaal nodig, of je laat de resten eerst drogen. Het composteren van een teelt geeft maar 2,5 g CO_2 per dag.

Klopt dit? Vergelijk: één bewoner ademt 1000 g CO_2 per dag uit, de sla neemt er 24 g op (hoofdstuk 14). Het compostvat is geen CO_2-bron van betekenis. Wel verbruikt het zuurstof: in Biosphere 2 (F28) was te veel compost in de bodem een oorzaak van het zuurstoftekort. De stikstof in de slaresten (4,8 g per teelt) is maar een vijfde van wat de sla opnam: de rest eten de bewoners op (hoofdstuk 36).

Overzicht: micro-organismen en biotechnologie

exponentiële fase

$$N(t) = N_0 \cdot 2^{t/t_d}$$

symbool	betekenis	eenheid
N_0	aantal bij de start	per mL
t_d	generatietijd	min of h

generatietijd uit twee metingen in de log-fase

$$t_d = \frac{\Delta t}{\log(N_2/N_1)/\log 2}$$

tellen met een verdunningsreeks (plaat met 30-300 kolonies)

$$N_{\text{staal}} = \frac{\text{aantal kolonies}}{V_{\text{uitgestreken}} \cdot \text{verdunning}}$$

alcoholgisting (gist)

$$\mathrm{C_6H_{12}O_6} \longrightarrow 2\,\mathrm{C_2H_5OH} + 2\,\mathrm{CO_2}$$

melkzuurgisting

$$\mathrm{C_6H_{12}O_6} \longrightarrow 2\,\mathrm{C_3H_6O_3}$$

methaanvorming (biogas)

$$\mathrm{C_6H_{12}O_6} \longrightarrow 3\,\mathrm{CH_4} + 3\,\mathrm{CO_2}$$

C/N-verhouding: compost ± 25-30

$$C/N = \frac{m_{\mathrm{C}}}{m_{\mathrm{N}}}$$

	bacterie	gist	schimmel	virus
cel?	ja, zonder kern	ja, met kern	ja, met kern, in draden	nee
grootte	1-5 µm	5-10 µm	draden 2-10 µm dik	20-300 nm
vermeerdering	deling	knopvorming	sporen, groei van hyfen	alleen in een gastheercel
nuttig	yoghurt, compost, insuline	brood, bier	penicilline, kaas	fagen tegen bacteriën, vectoren in de geneeskunde
schadelijk	Salmonella, Legionella	schimmelinfecties	grauwe schimmel	griep, norovirus
bestrijding	antibiotica, hitte	schimmelwerende middelen	droog houden, ventileren	vaccins, hygiëne

behandeling	wat	effect
koelen	4 à 7 °C	vertraagt; Listeria groeit traag door
invriezen	−18 °C	groei staat stil, de meeste bacteriën overleven
pasteuriseren	bv. 15 s op 72 °C	doodt de meeste ziekteverwekkers, niet alle sporen
steriliseren	15 min op 121 °C	doodt alles, ook sporen
verzuren	pH onder ± 4,5	remt de meeste ziekteverwekkers

Habitat: rekenwaarden uit dit hoofdstuk	
plantenresten	± 50 g per krop: 2,4 kg per teelt (0,12 kg droge stof, 4,8 g stikstof)
keukenafval	± 0,25 kg per dag voor twee bewoners (rekenwaarde)
compostvat	120 L, geïsoleerd, buiten de kweekruimte; C/N 25-35, vocht 50-65 %; compost (na enkele dagen boven 55 °C) onder de potgrond van de meetkas
CO_2 uit het compostvat	± 2,5 g per dag voor de slaresten: verwaarloosbaar
hygiëne	acht regels voor de kweekruimte (blad Habitat: ziekteverwekkers en hygiëne)

HET GEHEEL · 35.24

Waar micro-organismen terugkomen

24/24

Micro-organismen verbinden de biologie met de chemie (fermentatie is stoichiometrie), met de wiskunde (exponentiële groei en logaritmen) en met de techniek (steriel werken, bioreactoren). In Habitat zijn ze tegelijk helpers (compost, straks algen en nitrificerende bacteriën) en belagers (ziekteverwekkers in water, op de oogst en bij de bewoners).

F26 Cellen en organismen
Eencelligen, de microscoop en de levenskenmerken.

F28 Ecosystemen en kringlopen
Reducenten en kringlopen; Biosphere 2 en de compost die zuurstof opat.

I2 Experimenten ontwerpen en rapporteren
Controlegroep, blanco en herhalingen bij een kweekproef.

I5 Groei en verval: exponentiële modellen
Verdubbelingstijd, logaritmen en de logaritmische as.

I14 Reacties en stoichiometrie
Rekenen met gisting, CO2-volumes en het molair volume.

I15 Zuren, basen en pH
De pH van yoghurt meten; zuur als bewaarmiddel.

I34 Enzymen en stofwisseling
Gisting levert 2 ATP; enzymen verklaren het temperatuuroptimum van de groei.

I36 Hydroponie en voedselproductie
Nitrificerende bacteriën in aquaponie en de stikstofkringloop.

E4 Differentiaalvergelijkingen en simulatie
Logistische groei en de methode van Euler, zoals in groei.py.

E25 Thermochemie en reactiesnelheid
Waarom compost in de winter stilvalt: reactiesnelheid en temperatuur.

E30 Biotechnologie en genetische technieken
Recombinant DNA, PCR en CRISPR: de technieken achter insuline uit bacteriën.

E31 Bioprocestechniek
Het Monod-model, batch en continu, opschalen: de groeicurve in een bioreactor.

E33 Het lichaam als regelsysteem
Insuline en de regeling van de bloedsuiker.

Habitat
Voor Habitat leggen we vast: de plantenresten (± 50 g per krop, 2,4 kg per teelt) en het keukenafval gaan naar een geïsoleerd compostvat van 120 L buiten de kweekruimte, met karton van de bevoorrading als droog, koolstofrijk materiaal; compost die enkele dagen boven 55 °C was, gaat onder de potgrond van de meetkas, nooit in het NFT-systeem. Voor de kweekruimte gelden acht hygiëneregels. In opdracht 35.13 ontwerp je het compostvat, in 35.14 werk je het hygiëneplan uit.

SAMENVATTING

In het kort

- Micro-organismen zijn te klein voor het blote oog: bacteriën (1-5 µm, geen kern), gisten (5-10 µm), schimmels (hyfen, sporen) en virussen (20-300 nm, geen cel).
- Een virus heeft geen eigen stofwisseling en laat zich vermenigvuldigen door een gastheercel; antibiotica werken er niet tegen.
- Steriel werken: steriliseren (121 °C, 15 min), aseptisch werken, gesloten platen bij 20-25 °C, nooit openen, vernietigen na gebruik.
- Een groeicurve heeft vier fasen: aanloop, exponentieel (een rechte op een logaritmische as), stationair en afsterven.
- Met een verdunningsreeks tel je bacteriën: kolonies gedeeld door (uitgestreken volume × verdunning), op een plaat met 30-300

kolonies.

- Bacteriën groeien het snelst rond hun optimum; koelen vertraagt, pasteuriseren en steriliseren doden. Listeria groeit ook in de koelkast.
- Fermentatie maakt brood en bier (alcohol + CO_2), yoghurt en zuurkool (melkzuur), azijn en biogas (methaan, dat bijna alle energie van de suiker bewaart).
- Genetisch gewijzigde bacteriën en gisten maken menselijke insuline sinds 1982.
- Habitat composteert plantenresten met karton (C/N ± 30, niet te nat) en volgt acht hygiëneregels in de kweekruimte.

BEGRIPPEN

Wat je nu kent

micro-organisme
Levend wezen dat te klein is om met het blote oog te zien: bacteriën, archaea, gisten, veel schimmels, algen en eencellige dieren.

bacterie
Eencellig organisme zonder celkern (prokaryoot), meestal 1 tot 5 µm groot; vermeerdert door zich in tweeën te delen.

plasmide
Klein ringvormig stukje DNA in een bacterie, los van het grote chromosoom; in de biotechnologie gebruikt om een gen binnen te brengen.

gist
Eencellige schimmel die zich vermeerdert door knopvorming en suiker kan vergisten tot alcohol en koolstofdioxide.

schimmel
Organisme met een celkern en een celwand van chitine, zonder bladgroen; groeit met draden (hyfen) en verspreidt zich met sporen.

virus
Deeltje van DNA of RNA in een eiwitmantel, zonder eigen stofwisseling; kan zich alleen in een levende gastheercel laten vermenigvuldigen.

steriel
Zonder levende micro-organismen of sporen.

kolonie
Hoopje micro-organismen op een voedingsbodem, gegroeid uit één cel of spore; zichtbaar met het blote oog.

KVE
Kolonievormende eenheid: een cel of klompje cellen dat één kolonie oplevert. Aantallen bacteriën worden in KVE per mL of per g opgegeven.

groeicurve
Het aantal micro-organismen in een cultuur in functie van de tijd: aanloop-, exponentiële, stationaire en afstervingsfase.

generatietijd
De tijd waarin een populatie in de exponentiële fase verdubbelt.

pasteuriseren
Voedsel kort verhitten (bv. melk 15 s op 72 °C) zodat de meeste ziekteverwekkers sterven, zonder het voedsel te koken.

steriliseren
Alle micro-organismen en sporen doden, bv. 15 minuten op 121 °C in een autoclaaf.

fermentatie
Omzetting van stoffen door micro-organismen, vaak zonder zuurstof: gisting, melkzuurvorming, methaanvorming.

C/N-verhouding
De massa koolstof gedeeld door de massa stikstof in organisch materiaal; voor goede compost ongeveer 25 tot 30.

recombinant DNA
DNA waarin een stuk van een ander organisme is ingebouwd, bv. het menselijke insulinegen in een plasmide van een bacterie.

HANDBOEK · HOOFDSTUK 36

Hydroponie en voedselproductie

Sla zonder aarde onder ledlampen: hoeveel per vierkante meter, per kWh en per liter, en wat dat betekent voor het voedsel van een station

De klimaatmodule heeft een kweekbed van 2,88 m² met 48 kroppen sla, een reservoir van 60 L, een pomp, twee groeilampen, en sensoren voor EC, pH en temperatuur. In dit hoofdstuk breng je alles samen wat je leerde over voedingsoplossingen, licht, CO_2 en enzymen, en je rekent eerlijk uit wat het oplevert: hoeveel kilogram sla per jaar, per kWh en per liter water, en welk deel van het voedsel van een bewoner dat is. Het antwoord is klein, en juist daarom leerrijk.

9× uitleg · 6× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 10 lestijden

B Leven en systemen · M Materie en chemie · E Energie

Het kweekbed van de klimaatmodule: vier goten, 48 kroppen, twee lampen.

Na dit hoofdstuk

- Je kan uitleggen wat een voedingsoplossing bevat, waarom er twee stamoplossingen zijn, en wat de sla per teelt opneemt.
- Je kan EC, pH en volume van een voedingsoplossing samen interpreteren en een proef met metingen uitvoeren.
- Je kan licht voor planten beschrijven met PAR, µmol/(m²·s) en de daglichtsom, en de energie van een foton berekenen.
- Je kan groeiledlampen vergelijken op hun spectrum en hun rendement in µmol per joule.
- Je kan de systemen NFT, eb en vloed, druppel, diepwater en Kratky vergelijken en het NFT-rek van Habitat uitleggen.
- Je kan de stikstofkringloop tekenen en uitleggen hoe aquaponie vissen en planten verbindt.
- Je kan de opbrengst van een teeltsysteem berekenen per m², per kWh en per liter water, en vergelijken met andere systemen.
- Je kan een teeltschema en een onderbouwd voedselplan voor Habitat opstellen, met de grenzen van wat de module kan.

UITLEG · 36.1

Wat je al weet, en wat nu komt

In F27 kweekte je sla in een pot zonder aarde. In de klimaatmodule werd dat een systeem: een reservoir met een recept (hoofdstuk 12), een EC- en een pH-meting (hoofdstuk 13 en 15), CO_2 uit de lucht (hoofdstuk 14), lampen met een lichtdoel (hoofdstuk 20 en 25), een temperatuur die past bij de enzymen (hoofdstuk 34) en hygiëneregels (hoofdstuk 35). Nu komt het samen.

wat je al weet	waar	wat er nu bij komt
telen zonder aarde, de Kratky-pot	F27	de systemen van de tuinbouw, en het NFT-rek van Habitat
het recept van de voedingsoplossing (N, P, K, Ca, Mg, S)	hoofdstuk 12	wat de sla ervan opneemt, per teelt
EC: 1,4 mS/cm, stamoplossingen A en B	hoofdstuk 13	EC en volume samen lezen
CO_2-opname van de sla, ventilatie	hoofdstuk 14	licht en CO_2 als beperkende factoren
pH 5,8 en pH-min	hoofdstuk 15	pH en de beschikbaarheid van voedingsstoffen
lampen 2 × 95 W, 200 µmol/(m²·s), 16 uur	hoofdstuk 20, 25	wat µmol betekent, en welke lamp het zuinigst is
temperatuur, enzymen, ATP	hoofdstuk 34	van kWh tot kJ op je bord
bacteriën, hygiëne	hoofdstuk 35	nitrificerende bacteriën en aquaponie

In dit hoofdstuk

- kijk je opnieuw naar de voedingsoplossing, nu met wat de sla ervan **opneemt**;
- meet je zelf EC en pH in een Kratky-pot;
- leer je licht tellen in **µmol**, en groeilampen vergelijken;
- vergelijk je teeltsystemen en ontleed je het **NFT-rek** van Habitat;
- volg je de stikstof door de **stikstofkringloop** en zie je hoe **aquaponie** vissen en planten verbindt;
- reken je de **opbrengst** uit per m², per kWh en per liter water, en stel je een eerlijk **voedselplan** op.

Een krop uit de goot: 200 g blad, en een wortelmat die in een filmpje water hing.

VERHAAL · 36.2

Van een tomaat van zeven meter tot een serre in Antarctica

Dat planten kunnen groeien in water met opgeloste zouten, wisten plantenfysiologen zoals Julius von Sachs al rond 1860. Maar het was labowerk. In 1929 begon **William Frederick Gericke** van de universiteit van Californië in Berkeley te verkondigen dat je zo ook echte gewassen kan telen. Hij kweekte in zijn tuin tomatenplanten van 7,6 meter hoog in bakken met voedingsoplossing, en in 1937 gaf hij de techniek een naam: **hydroponics**, 'werk met water'.

De eerste die er echt op rekende, was een luchtvaartmaatschappij. Pan American Airways had op Wake Island, een koraaleiland midden in de Grote Oceaan, een tussenstop voor zijn watervliegtuigen, met een hotel, maar zonder vruchtbare grond. Eind 1937 trok een student van Gericke erheen om er groenten te telen zonder aarde, voor de passagiers en het personeel.

In 1965 bedacht **Allen Cooper** aan een tuinbouwonderzoekscentrum in Littlehampton, in Engeland, de **voedingsfilm-techniek** (NFT): geen bakken vol water meer, maar een dun filmpje dat door licht hellende goten stroomt. Dat systeem gebruikt de klimaatmodule. Op 10 augustus 2015 aten astronauten in het internationale ruimtestation voor het eerst sla die ze er zelf gekweekt hadden. En in 2018 kwam naast het Duitse poolstation Neumayer III in Antarctica de proefserre **EDEN ISS** te staan: 12,5 m² teeltoppervlak onder ledlampen, dat in 2018 in negen maanden 268 kg groenten opleverde, maar daarvoor ongeveer 240 kWh elektriciteit per dag verbruikte.

Waarom dit verhaal ertoe doet

Hydroponie werd groot op plaatsen zonder goede grond: een eiland, een woestijn, een poolstation, een ruimtestation. Dat zijn ook de plaatsen waar energie duur is. De vraag van dit hoofdstuk is dus niet alleen 'hoe groeit het', maar ook 'wat kost het'.

Een containerserre in de sneeuw: groenten in een poolstation, zolang er stroom is.

UITLEG · 36.3

De voedingsoplossing: wat erin zit en wat de sla opneemt

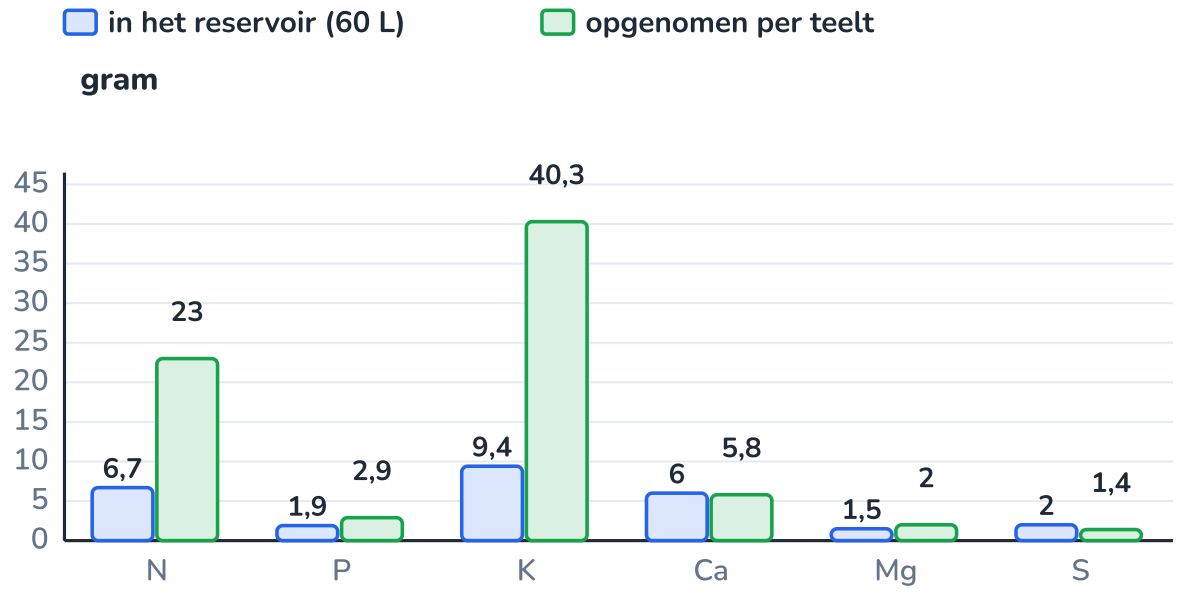
3/21

De voedingsoplossing van Habitat bevat per liter regenwater 112 mg stikstof, 32 mg fosfor, 156 mg kalium, 100 mg calcium, 25 mg magnesium en 33 mg zwavel, plus sporenelementen (hoofdstuk 12). Je maakt ze met twee **stamoplossingen**, A en B, elk 100 keer geconcentreerd (hoofdstuk 13). Waarom twee? In A zit het calciumnitraat, in B de sulfaten en fosfaten. Zo geconcentreerd zouden calcium en sulfaat of fosfaat samen neerslaan tot gips en calciumfosfaat, en dan zijn ze voor de plant verloren. Pas in het verdunde reservoir blijven ze samen opgelost.

Hoeveel neemt de sla eruit? Een krop heeft 12 g droge stof. Droge sla bevat typisch ongeveer 4 % stikstof, 0,5 % fosfor, 7 % kalium en 1 % calcium. Voor een teelt van 48 kroppen is dat ongeveer 23 g stikstof en 40 g kalium: drie à vier keer zoveel als er in het volle reservoir zit (grafiek). Het reservoir moet dus tijdens een teelt meermaals bijgestuurd worden. Ook water gaat eruit: per teelt nemen de planten ongeveer 153 L op, twee en een halve keer de inhoud van het reservoir.

Rekenwaarden

De samenstelling van droge sla verschilt per ras, per lichtsterkte en per oplossing. De percentages hier zijn typische waarden: goed genoeg om te zien welke voedingsstof het snelst op is, niet om op te doseren. Doseren doe je met de EC-meting.



Stikstof en kalium raken het snelst op: de sla neemt er per teelt 3 à 4 keer meer van op dan het reservoir bevat.

UITLEG · 36.4

EC en pH samen lezen

4/21

De **EC** meet de totale hoeveelheid ionen (hoofdstuk 13). Ze verandert om twee redenen, in tegengestelde richting:

- de planten nemen **water** op en er verdampt water: de zouten blijven achter, de EC **stijgt**;
- de planten nemen **ionen** op: de EC **daalt**.

Wie alleen de EC bekijkt, weet dus niet wat er gebeurt. Houd daarom ook het **volume** bij: EC × volume (min het regenwater) is een maat voor de totale hoeveelheid zouten in het reservoir. Die daalt altijd, ook als de EC stijgt.

De **pH** stijgt in een oplossing met nitraat, omdat de wortels voor elk opgenomen nitraation een negatief ion teruggeven (hoofdstuk 15). En de pH bepaalt welke voedingsstoffen opgelost blijven:

pH van de oplossing	wat er gebeurt	wat je ziet
onder 5,5	wortels kunnen beschadigen; calcium en magnesium worden minder goed opgenomen; mangaan en ijzer soms te veel	bruine wortelpunten, vlekken op de bladeren
5,5 tot 6,5	de meeste voedingsstoffen blijven opgelost en opneembaar	gezonde groei (streefwaarde Habitat: 5,8)
boven 6,5	ijzer, mangaan en fosfaat slaan neer of worden slecht opgenomen	jonge bladeren geel met groene nerven (ijzertekort)

De dagelijkse ronde

Eerst het niveau (hoeveel water is er weg?), dan de pH (bijsturen met pH-min), dan de EC (bijsturen met regenwater of stamoplossing), en alles in het logboek. Een plotse verandering zegt meer dan één waarde: een EC die ineens hard stijgt, kan een lek zijn, of een ziekte die de planten doet stoppen met opnemen.

Niveau, pH, EC, logboek: elke dag dezelfde volgorde.

PROEF · 36.5

Sla in Kratky-potten: EC, pH en groei

5/21

ONDERZOEKSVRAAG

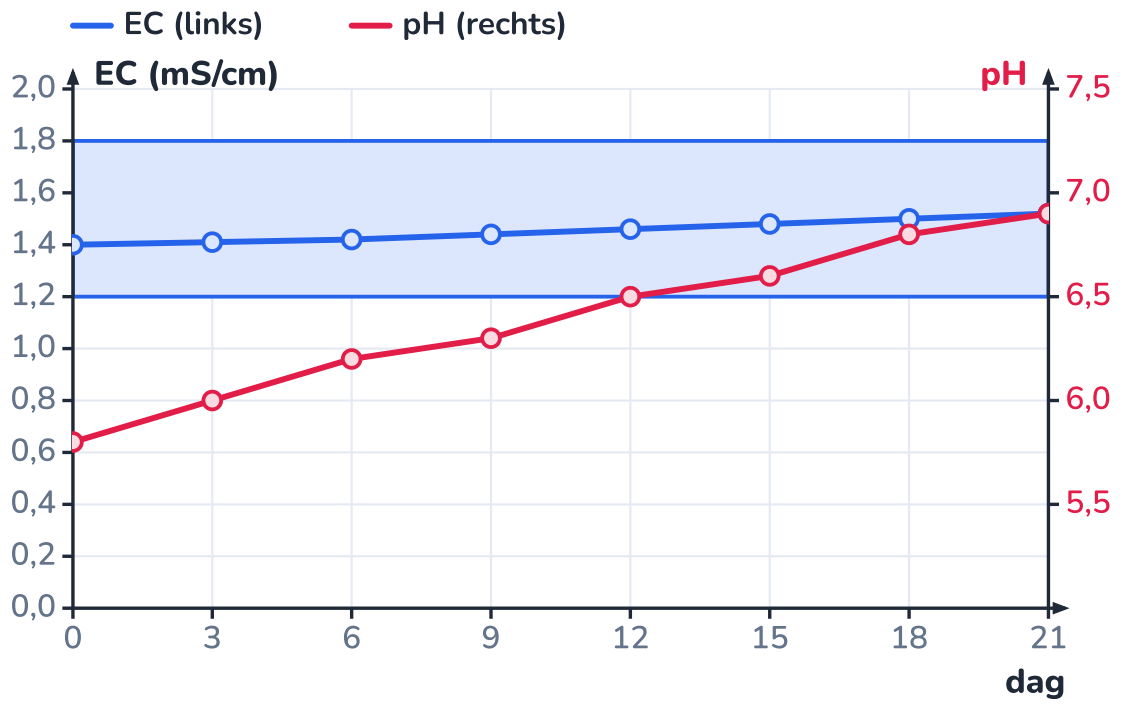
Hoe sterk moet de voedingsoplossing zijn voor de beste groei, en hoe veranderen EC, pH en volume in een pot zonder pomp?

MATERIAAL

- vier ondoorzichtige potten van 4,0 L met deksel en netpotje, kleikorrels
- vier slaplantjes van twee weken, even groot
- stamoplossingen A en B, regenwater
- EC-meter, pH-meter (geijkt), maatbeker van 1 L, keukenweegschaal
- lamp of vensterbank met veel licht

Veiligheid

Stamoplossingen zijn geconcentreerde zouten: handschoenen en bril bij het aanmaken, niet drinken, gemorst meteen opnemen. Etiket op elke pot. Restjes verdund als plantenvoeding gebruiken, niet in de gootsteen gieten.



De EC blijft in de streefzone (1,2-1,8, blauw), de pH kruipt omhoog tot 6,9.

WERKWIJZE

1. Vul de potten met 4,0 L: regenwater (EC 0,05), en oplossingen met EC 0,7, 1,4 en 2,8 mS/cm (half, gewoon en dubbel recept).
2. Zet in elke pot een plantje, zo dat de onderste wortels net in de oplossing hangen. Zet de potten naast elkaar bij hetzelfde licht.
3. Meet om de drie dagen in de pot van EC 1,4: de EC, de pH en het volume (til het netpotje op en lees de maatstreepjes af, of weeg de pot).
4. Oogst na 21 dagen alle planten, laat de wortels uitdruipen, snij ze eraf en weeg het blad.

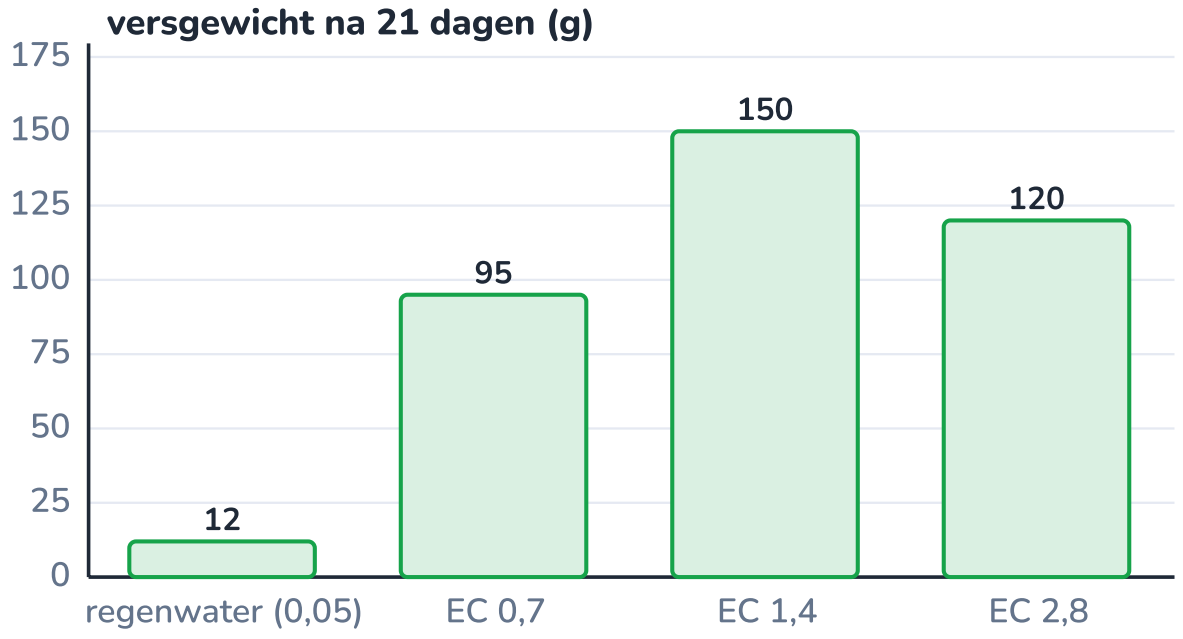
WAARNEMING

dag	0	3	6	9	12	15	18	21
EC (mS/cm)	1,40	1,41	1,42	1,44	1,46	1,48	1,50	1,5
pH	5,8	6,0	6,2	6,3	6,5	6,6	6,8	6,9
volume (L)	4,0	3,8	3,5	3,1	2,6	2,1	1,7	1,2

Voorbeeldmeting in de pot met EC 1,4. Versgewicht na 21 dagen: regenwater (0,05): 12 g, EC 0,7: 95 g, EC 1,4: 150 g, EC 2,8: 120 g.

VERKLARING

De EC stijgt een beetje, terwijl het volume daalt tot minder dan een derde: de plant neemt verhoudingsgewijs meer water dan zouten op. Toch zijn er zouten verdwenen (volgend voorbeeld). De pH kruipt van 5,8 naar 6,9, zoals in het reservoir van de module (hoofdstuk 15): in een Kratky-pot kan je niet bijsturen, en na drie weken komt het ijzer in de gevarenzone. De groei is het best bij EC 1,4. Met alleen regenwater groeit er bijna niets (geen voeding), met het dubbele recept is de oplossing te zout: de wortels krijgen moeilijker water binnen.



Te weinig en te veel voeding remmen de groei; EC 1,4 is hier het best.

UITGEWERKT VOORBEELD · 36.6

Wat zeggen EC en volume over wat de plant opnam?

6/21

Gebruik de metingen van de Kratky-pot met EC 1,4 en de ijklijn van hoofdstuk 13: EC $= 0{,}05 + 1{,}0 \cdot c$, met c de zouten in g/L. (a) Hoeveel gram zouten zat er bij de start en na 21 dagen in de pot? (b) Hoeveel nam de plant op? (c) Hoeveel stikstof is dat, als het recept 112 mg N per 1,30 g zouten bevat? (d) Past dat bij een plant van 150 g vers?

GEGEVEN

- start: EC 1,40, 4,0 L; na 21 dagen: EC 1,52, 1,2 L
- EC $= 0{,}05 + 1{,}0 \cdot c$
- 112 mg N per 1,30 g zouten; droge sla ± 6 % van het versgewicht (met wortels), 4 % daarvan is N

GEVRAAGD

- zouten begin en eind
- opname zouten en stikstof
- controle met de plant

OPLOSSING

1. (a) Start: $c = 1{,}40 - 0{,}05 = 1{,}35\,\text{g/L}$, maal 4,0 L: $5{,}40\,\text{g}$. Eind: $c = 1{,}47\,\text{g/L}$, maal 1,2 L: $1{,}76\,\text{g}$.
2. (b) Opgenomen: $5{,}40 - 1{,}76 = 3{,}64\,\text{g}$ zouten, ook al steeg de EC.
3. (c) $3{,}64 \cdot 112/1{,}30 = 313\,\text{mg}$ stikstof.
4. (d) 150 g blad plus wortels, ± 6 % droge stof: ± 10 g droog, daarvan 4 % N: ± 0,40 g. Dezelfde orde van grootte als de 313 mg uit de oplossing.

ANTWOORD

De pot bevatte 5,40 g zouten en na 21 dagen nog 1,76 g: de plant nam 3,64 g op, waarvan ongeveer 313 mg stikstof, ook al steeg de EC.

Klopt dit? De EC alleen zou je doen denken dat er voeding bijkwam. Het volume erbij toont het omgekeerde. Een systeem dat alleen op de EC stuurt en onbeperkt bijvult met stamoplossing, verzamelt na verloop van tijd de ionen die de plant níet opneemt (zoals natrium en sulfaat): daarom ververs je het reservoir om de paar weken helemaal.

UITLEG · 36.7

Licht voor planten: fotonen tellen

7/21

Voor de fotosynthese telt niet hoeveel watt licht er op een blad valt, maar hoeveel **lichtdeeltjes** (fotonen): elke CO_2-molecule vraagt er minstens acht, of ze nu blauw of rood zijn (hoofdstuk 34). Planten gebruiken licht van 400 tot 700 nm: de **PAR** (fotosynthetisch actieve straling). Tuinbouwers tellen de fotonen in micromol: 1 µmol is $6{,}022 \cdot 10^{17}$ fotonen.

DAGLICHTSOM EN ENERGIE VAN EEN FOTON

$$\mathrm{DLI} = \mathrm{PPFD} \cdot t \qquad E_{\text{foton}} = \frac{h\,c}{\lambda}$$

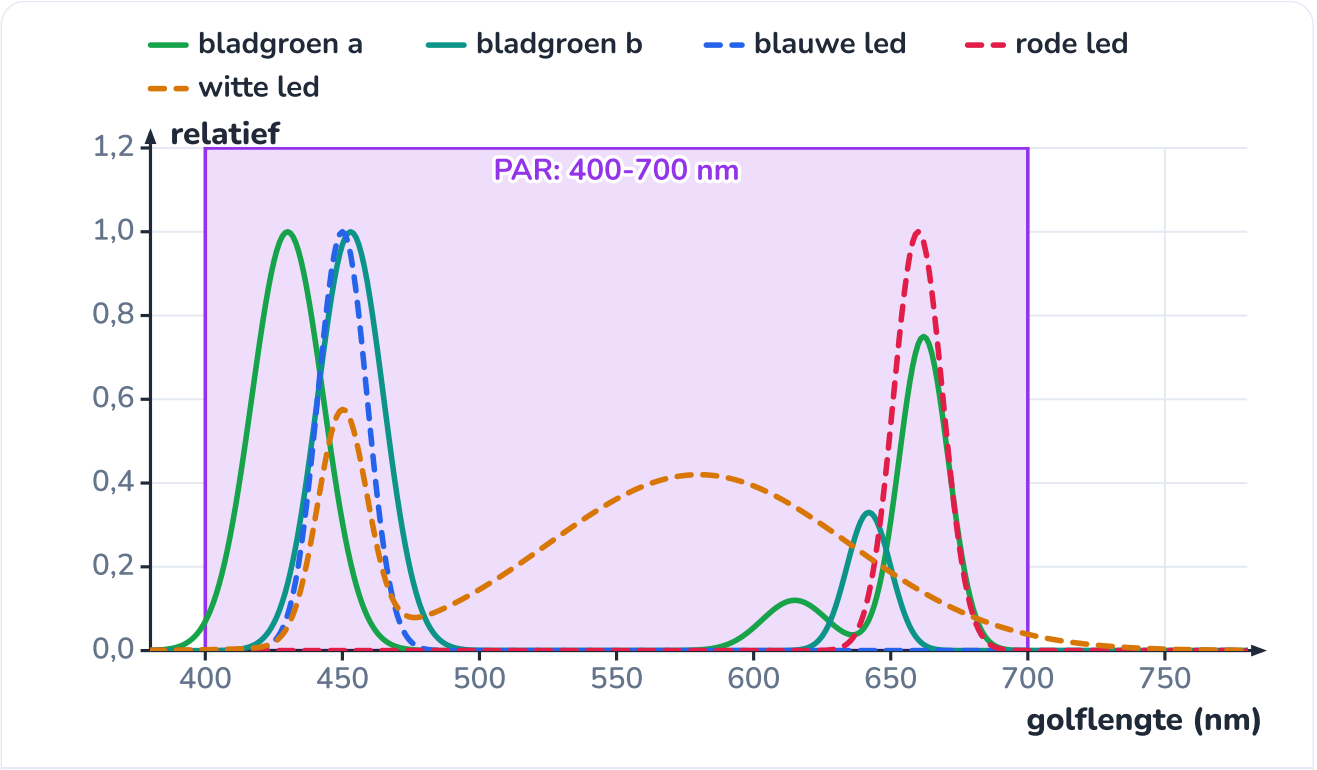
PPFD in µmol/(m²·s), t in s, DLI in mol/(m² · dag). $h = 6{,}626 \cdot 10^{-34}\,\mathrm{J\,s}$, $c = 3{,}00 \cdot 10^{8}\,\mathrm{m/s}$, λ de golflengte in m.

De kweekbedden van Habitat krijgen 200 µmol/(m²·s) gedurende 16 uur: een **daglichtsom** van $200 \cdot 10^{-6} \cdot 16 \cdot 3600 = 11{,}5\,\mathrm{mol/(m^2 \cdot dag)}$. Sla groeit goed vanaf ongeveer 12 à 17 mol per dag. Buiten in België valt er in juni gemiddeld ongeveer 40 mol per m² per dag, in december maar 4 à 5: daarom groeit er in de winter zo weinig zonder lampen.

Bladgroen absorbeert vooral blauw en rood licht; groen wordt deels teruggekaatst (daarom zie je bladeren groen). Toch gaat groen licht niet verloren: het dringt dieper in het blad en in het gewas, en wordt daar alsnog gebruikt. Blauw licht houdt planten compact en maakt rode sla roder; verrood licht (boven 700 nm) laat ze strekken, zoals in de schaduw.

Paars licht is het beste voor planten

Rode en blauwe leds samen geven paars-roze licht, en rode fotonen zijn goedkoop (volgend voorbeeld). Maar planten groeien even goed of beter onder wit licht met veel rood. En in een werkruimte zoals de klimaatmodule is wit licht nodig: onder paars licht zie je geen gele bladeren of schimmelvlekken.



Schematische vorm: bladgroen absorbeert blauw en rood; leds leveren licht waar het nodig is.

UITGEWERKT VOORBEELD · 36.8

Blauwe, rode en witte fotonen: welke lamp is het zuinigst?

8/21

(a) Bereken de energie van één mol fotonen van 450 nm (blauw) en van 660 nm (rood). (b) Stel dat twee leds elk 60 % van hun elektrische energie omzetten in licht. Hoeveel µmol fotonen maken ze per joule? (c) De lampen van Habitat (2 × 95 W) geven 200 µmol/(m²·s) op het kweekbed van 2,88 m². Hoeveel µmol per joule komt er op het bed aan?

GEGEVEN

- $h = 6{,}626 \cdot 10^{-34}\,\mathrm{J\,s}$, $c = 2{,}998 \cdot 10^{8}\,\mathrm{m/s}$, $N_A = 6{,}022 \cdot 10^{23}\,\mathrm{mol^{-1}}$
- rendement led 60 %
- 190 W, 200 µmol/(m²·s), 2,88 m²

GEVRAAGD

- energie per mol fotonen
- µmol per joule

OPLOSSING

1. (a) Per foton $E = hc/\lambda$; per mol maal N_A: $E = \dfrac{6{,}022 \cdot 10^{23} \cdot 6{,}626 \cdot 10^{-34} \cdot 2{,}998 \cdot 10^{8}}{450 \cdot 10^{-9}} = 266\,\mathrm{kJ/mol}$ voor blauw, en 181 kJ/mol voor rood.
2. (b) 60 % van 1 J is 0,60 J licht. Blauw: $0{,}60/266 \cdot 10^{-3}\,\mathrm{J}/\mu\mathrm{mol} = 2{,}26\,\mu\mathrm{mol}$; rood: $0{,}60/181 \cdot 10^{-3} = 3{,}31\,\mu\mathrm{mol}$ per joule.
3. (c) Op het bed: $200 \cdot 2{,}88 = 576\,\mu\mathrm{mol/s}$ voor $190\,\mathrm{W} = 190\,\mathrm{J/s}$: $3{,}03\,\mu\mathrm{mol/J}$.

ANTWOORD

Een mol blauwe fotonen draagt 266 kJ, een mol rode 181 kJ. Bij hetzelfde rendement maakt een rode led dus 1,47 keer zoveel fotonen per joule. Op het kweekbed komt 3,0 µmol per joule aan.

Klopt dit? Een rode foton levert voor de fotosynthese evenveel op als een blauwe: het teveel aan energie van blauw wordt warmte. Daarom bevatten zuinige groeilampen veel rood. De zuinigste lampen halen vandaag iets meer dan 3 µmol/J aan de lamp; de

rekenwaarde van Habitat (3,0 op het bed) veronderstelt dus een topzuinige lamp en wanden die het licht terugkaatsen. Met een gewone lamp van 2,5 µmol/J zou het bed maar 165 µmol/(m²·s) krijgen.

UITLEG · 36.9

Systemen: NFT, eb en vloed, druppel

9/21

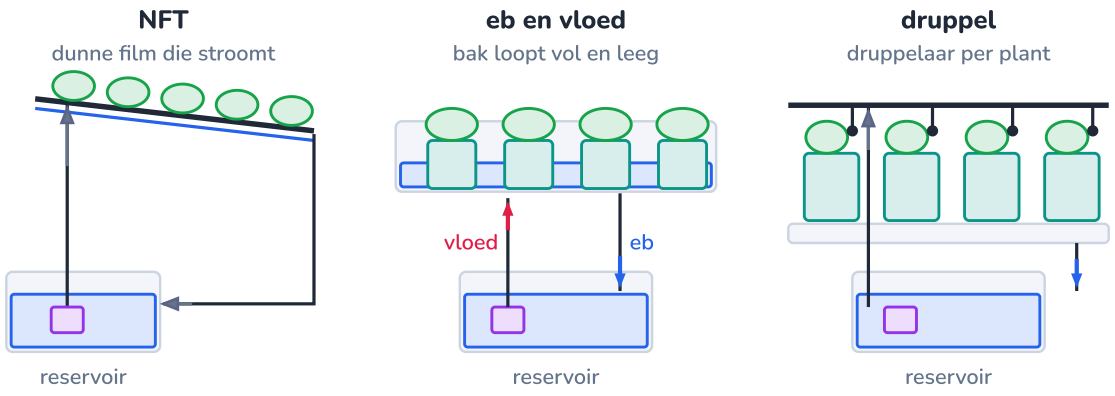
Elk hydroponiesysteem moet vier dingen doen die de aarde anders doet (F27): de plant steunen, water geven, voeding geven, en de wortels zuurstof geven. De systemen verschillen in hoe ze het water en de lucht bij de wortels brengen.

systeem	hoe	goed voor	let op
NFT (voedingsfilm)	dunne film stroomt door licht hellende goten	sla, kruiden: licht, weinig water in het systeem	valt de pomp lang uit, dan drogen de wortels
eb en vloed	een bak loopt enkele keren per dag vol en weer leeg	potplanten, kruiden, opkweek	groot reservoir nodig (de hele bak vullen)
druppel	een druppelaar per plant in steenwol of kokos	tomaat, komkommer, paprika	druppelaars verstoppen; open systemen lozen water
diepwater (DWC)	wortels in een bak met belucht water	sla, snel en eenvoudig	luchtpomp nodig; veel water
Kratky (F27)	pot zonder pomp; de luchtlaag groeit mee	proeven, kleine teelt	niet bij te sturen, alleen voor snelle gewassen

Bij een **gesloten** systeem vloeit het water terug naar het reservoir; bij een **open** druppelsysteem loopt de overmaat weg. Gesloten systemen zijn zuiniger met water en zouten, maar een ziekte verspreidt zich er sneller (hoofdstuk 35). Habitat kiest een gesloten **NFT**-systeem: licht, weinig water in het systeem (snel op temperatuur, snel bijgestuurd), en geschikt voor sla en kruiden.

Waarom niet gewoon aarde?

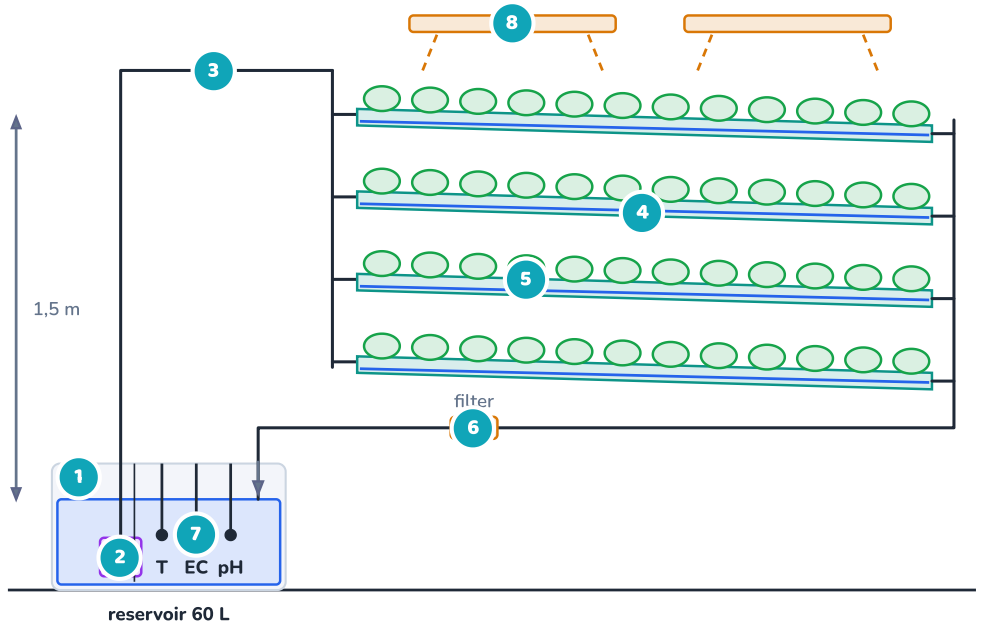
Aarde weegt veel (een volle plantenbak van de meetkas: 4 kg, F11), houdt ziektes vast en is moeilijk te regelen. In een station, waar elke kilogram aangevoerd wordt, is een systeem van goten, een reservoir en een zak zouten lichter en beter te sturen. Wel moeten pomp en sensoren betrouwbaar zijn: in een NFT-goot is de wortel binnen enkele uren droog als de pomp uitvalt.



Links een stromende film, midden een bak die volloopt en leegloopt, rechts een druppelaar per plant.

Het NFT-rek van de klimaatmodule

Het kweekbed is een getrapt rek tegen de wand: in bovenaanzicht 2,4 m bij 1,2 m. We leggen hier de goten vast: **vier goten** van 2,4 m, elk met **12 plantplaatsen** om de 20 cm, 30 cm uit elkaar. De schets toont het rek van voren, niet op schaal: de goten lijken boven elkaar te hangen, maar lopen in trappen naar achteren op.



Schema van de waterkringloop: reservoir, pomp, vier goten, retour met filter.

1 Reservoir
60 L, lichtdicht (algen, hoofdstuk 5), op de vloer in de schaduw van het rek: de oplossing blijft tussen 18 en 22 °C (hoofdstuk 34). Druksensor en vlotter bewaken het niveau (hoofdstuk 11).

2 Pomp
12 V, 4,8 L/min bij 1,56 m opvoerhoogte (hoofdstuk 11), via een MOSFET op D9 (hoofdstuk 23). Ze draait 10 minuten per uur: genoeg om de wortelmat vochtig en de oplossing zuurstofrijk te houden, en 48 Wh per dag.

3 Aanvoer
Een slang van 16 mm naar de bovenkant van het rek, 1,5 m boven het reservoir, met een aftakking naar het begin van elke goot. Een kraantje per goot verdeelt het debiet: 1,2 L/min per goot.

4 Goot
Een witte, lichtdichte goot van 2,4 m met een helling van 1 op 50 (4,8 cm hoogteverschil; op de tekening overdreven). Het water stroomt als een film van enkele mm langs de wortels.

5 Plantplaatsen
12 per goot, om de 20 cm, in een netpotje met een steenwolblokje. Samen 48 kroppen: 16,7 per m².

6 Retour en filter
Aan het einde van elke goot valt het water in een verzamelbuis die terugloopt naar het reservoir. Een zeefje houdt losgekomen wortels en steenwol tegen. Het klateren in het reservoir brengt zuurstof in het water.

7 Sensoren
NTC voor de temperatuur (A0), EC (A6) en pH (A2) op de Mega (hoofdstuk 22), plus de druksensor voor het niveau (A5). Eerst de pH, dan de EC (hoofdstuk 15).

8 Lampen
Twee ledlampen van 95 W, PWM-dimbaar (hoofdstuk 25), die het daglicht aanvullen tot 200 µmol/(m²·s), 16 uur per dag.

UITGEWERKT VOORBEELD · 36.11

Debiet en helling in de goten

11/21

De pomp levert 4,8 L/min, verdeeld over vier goten. Een goot is 10 cm breed; de film is ongeveer 3 mm dik. (a) Hoeveel water stroomt er per goot, in mL/s? (b) Hoe snel stroomt de film, en hoe lang doet het water over een goot van 2,4 m? (c) Hoeveel hoogteverschil geeft een helling van 1 op 50?

GEGEVEN

- $Q = 4{,}8\,\mathrm{L/min}$ over 4 goten
- film: $b = 10\,\mathrm{cm}$, $d = 3\,\mathrm{mm}$
- $\ell = 2{,}4\,\mathrm{m}$, helling $1:50$

GEVRAAGD

- Q_{goot}, v, t, Δh

OPLOSSING

1. (a) $Q_{\text{goot}} = 4{,}8/4 = 1{,}2\,\mathrm{L/min} = 1200\,\mathrm{mL}/60\,\mathrm{s} = 20\,\mathrm{mL/s} = 20\,\mathrm{cm^3/s}$.
2. (b) Doorsnede van de film: $10\,\mathrm{cm} \cdot 0{,}3\,\mathrm{cm} = 3{,}0\,\mathrm{cm^2}$. $v = Q/A = 20/3{,}0 = 6{,}7\,\mathrm{cm/s}$ (continuïteit, E18). Over 240 cm: $t = 240/6{,}7 = 36\,\mathrm{s}$.
3. (c) $\Delta h = 240\,\mathrm{cm}/50 = 4{,}8\,\mathrm{cm}$.

ANTWOORD

1,2 L/min of 20 mL/s per goot; de film stroomt ongeveer 6,7 cm/s en doet 36 s over een goot; het hoogteverschil is 4,8 cm.

Klopt dit? NFT-telers mikken op 1 à 2 L/min per goot: genoeg om ook de laatste plant verse, zuurstofrijke oplossing te geven. Een helling van minder dan ongeveer 1 op 100 laat plassen staan (weinig zuurstof), een steilere laat de film te snel en te dun lopen.

UITLEG · 36.12

Aquaponie en de stikstofkringloop

12/21

Stikstof is het element dat de sla het meest nodig heeft na koolstof, zuurstof en waterstof. Het draait in een kringloop waarin bacteriën het meeste werk doen:

- **stikstofbinding**: stikstofgas uit de lucht wordt ammonium, door bacteriën in de wortelknolletjes van vlinderbloemigen (Beijerinck, hoofdstuk 35), of in de fabriek met het proces van Haber en Bosch: $\mathrm{N_2 + 3\,H_2 \longrightarrow 2\,NH_3}$. Bijna alle stikstofkunstmest, ook het calcium- en kaliumnitraat van Habitat, begint hier;
- **nitrificatie**: bacteriën (*Nitrosomonas*, *Nitrobacter*) zetten ammonium met zuurstof om in nitriet en daarna nitraat:

NITRIFICATIE (IN TWEE STAPPEN, SAMENGEVAT)

$$\mathrm{NH_4^+ + 2\,O_2 \longrightarrow NO_3^- + 2\,H^+ + H_2O}$$

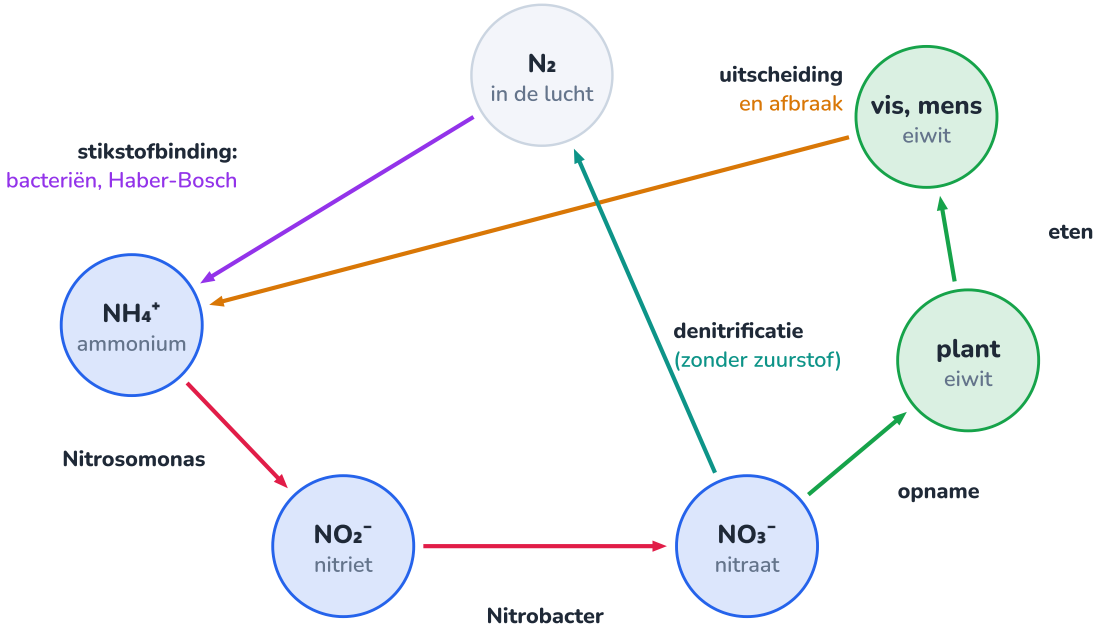
Er komen protonen vrij: nitrificatie maakt het water zuurder. In aquaponie daalt de pH daardoor, in de hydroponie van Habitat stijgt ze juist (nitraatopname).

- **opname** door planten, die er eiwitten van maken; dieren en mensen eten die eiwitten en scheiden de stikstof uit als ammonium of ureum;
- **denitrificatie**: zonder zuurstof maken andere bacteriën van nitraat weer stikstofgas.

Aquaponie maakt van die kringloop een teeltsysteem: vissen scheiden ammonium uit, nitrificerende bacteriën op een filter maken er nitraat van, planten nemen het nitraat op en zuiveren zo het water voor de vissen. De vissen zelf moeten wel gevoerd worden: de stikstof komt uit het visvoer.

Nitrificerende bacteriën in de ruimtevaart

Het Europese project MELiSSA (ESA, sinds 1989) bouwt een kringloop voor ruimtereizen in vijf compartimenten: bacteriën die afval afbreken, nitrificerende bacteriën (*Nitrosomonas* en *Nitrobacter*), de blauwalg *Arthrospira* (spirulina), hogere planten, en de bemanning. Het Belgische SCK CEN in Mol levert er onderzoek en bacteriestammen voor. De proefinstallatie staat in Barcelona.



Stikstof draait rond: bacteriën binden, nitrificeren en denitrificeren; planten en dieren maken er eiwit van.

UITGEWERKT VOORBEELD · 36.13

Stikstof bijhouden: reservoir, urine en visvoer

13/21

(a) Hoeveel stikstof nemen de 48 kroppen per dag op (12 g droge stof per krop, 4 % stikstof, 35 dagen)? (b) Hoe lang volstaat de stikstof in een vol reservoir (60 L met 112 mg/L)? (c) Een mens scheidt met zijn urine ongeveer 4 kg stikstof per jaar uit. Hoeveel keer de behoefte van het kweekbed is dat? (d) Hoeveel visvoer (32 % eiwit, eiwit bevat 16 % stikstof) is nodig als de helft van de stikstof in het voer als nitraat bij de planten komt?

GEGEVEN

- 48 kroppen, 12 g droge stof, 4 % N, 35 dagen
- 60 L met 112 mg N per L
- urine: ± 4 kg N per persoon per jaar
- visvoer: 32 % eiwit, eiwit 16 % N; 50 % benut

GEVRAAGD

- N per dag
- dagen
- factor urine
- visvoer per dag

OPLOSSING

1. (a) Per krop $12 \cdot 0{,}04 = 0{,}48\,\text{g}$ N; per teelt $48 \cdot 0{,}48 = 23{,}0\,\text{g}$; per dag $23{,}0/35 = 0{,}66\,\text{g}$.
2. (b) In het reservoir: $60 \cdot 0{,}112 = 6{,}72\,\text{g}$ N. Dat volstaat $6{,}72/0{,}66 = 10$ dagen.

3 (c) $4000/365 = 11$ g N per dag: 17 keer wat het kweekbed nodig heeft.

4 (d) Visvoer bevat $0{,}32 \cdot 0{,}16 = 0{,}051$ of 5,1 % N, waarvan de helft bij de planten komt: 2,6 %. Voor 0,66 g N per dag: $0{,}66/0{,}026 = 26$ g voer per dag.

ANTWOORD

De sla neemt 0,66 g stikstof per dag op; een vol reservoir volstaat ongeveer 10 dagen. Eén bewoner scheidt 17 keer meer stikstof uit dan de sla nodig heeft. In aquaponie zou ongeveer 26 g visvoer per dag de stikstof leveren.

Klopt dit? De vuistregel van de FAO voor aquaponie is ruimer: 40 à 50 g voer per m² per dag, voor het kweekbed 115 à 144 g. Een deel van de stikstof blijft in de vissen en het slib of gaat verloren, en de planten moeten ook kalium, ijzer en de andere mineralen uit het voer halen. Wie zoveel voer geeft, heeft 6 à 13 kg vis nodig (vissen eten 1 à 2 % van hun gewicht per dag), en bij 20 kg vis per 1000 L een bak van minstens een halve kubieke meter: veel meer dan het reservoir van 60 L. En het visvoer moet aangevoerd worden.

UITLEG · 36.14

Opbrengst per vierkante meter, per kWh en per liter

14/21

Hoe goed is een teeltsysteem? Drie maten tellen, en ze vertellen elk iets anders:

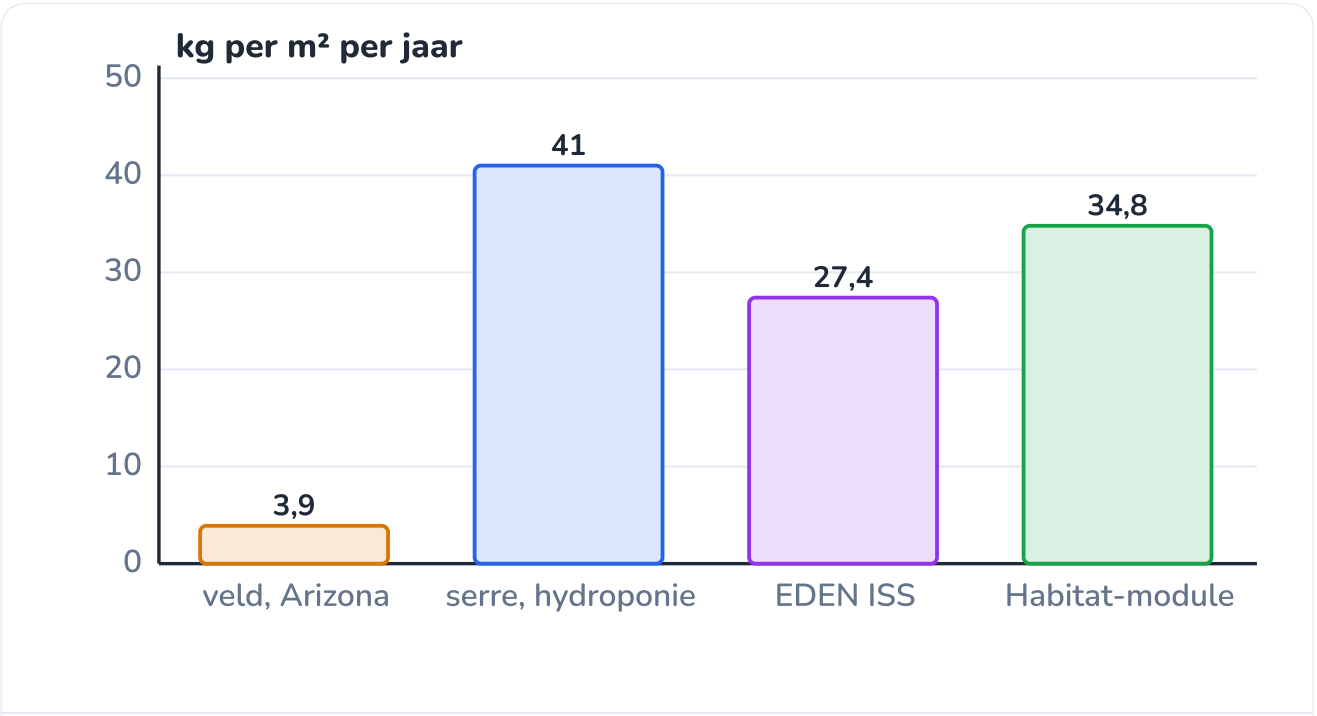
- **kg per m² per jaar**: hoeveel ruimte heb je nodig? In een station is ruimte duur.
- **kWh per kg**: hoeveel energie kost elke kilogram? Voor een module op zonne-energie de belangrijkste maat.
- **liter water per kg**: hoeveel water gaat er verloren? In een woestijn, of in een station met 98 % waterterugwinning (F29), even belangrijk.

	kg per m² per jaar	water per kg	energie per kg
sla in het veld, Arizona (Barbosa e.a., 2015)	3,9	250 L	0,3 kWh
sla in een serre met hydroponie, Arizona (idem)	41	20 L	± 25 kWh (verwarmen, koelen, belichten)
volledig belichte teeltruimtes (hoofdstuk 20)			10 tot 18 kWh
EDEN ISS, Antarctica, 2018 (26 gewassen)	27,4		± 205 kWh (alles meegerekend)
Habitat-module (rekenmodel)	34,8	16 L (2,5 L met terugwinning)	11,1 kWh (lampen)

Hydroponie met lampen haalt tien keer meer per m² dan het veld, met een tiende van het water, maar vraagt tientallen keren meer energie. Dat verschil is geen toeval: het veld krijgt zijn licht gratis van de zon. De cijfers van EDEN ISS liggen veel hoger dan die van de module, omdat daar alles meetelt (verwarmen in Antarctica, lucht behandelen, pompen, computers) en omdat er ook trage gewassen als komkommer en tomaat groeiden.

Vergelijk alleen wat vergelijkbaar is

Een getal als 'kWh per kg' zegt niets zonder de grenzen: alleen de lampen, of ook verwarming, koeling en ventilatie? Vers gewicht of droog? Sla of tomaat? De module rekent hier met de lampen alleen; de verwarming komt van het station (hoofdstuk 19).



Per m² haalt belichte hydroponie 7 à 10 keer meer dan het veld.

UITGEWERKT VOORBEELD · 36.15

Habitat: wat levert het kweekbed op?

15/21

Het kweekbed van 2,88 m² levert 48 kroppen van 200 g per 35 dagen. De lampen verbruiken 3,04 kWh per dag. Per krop nemen de planten 3,19 L water op: 3,0 L die verdampt (rekenwaarde) en 0,19 L in de krop zelf. Bereken de opbrengst per jaar, per m², per kWh lampenergie en per liter water. Wat verandert er als het verdampte water op het koude kasdak condenseert en voor 90 % teruggewonnen wordt?

GEGEVEN

- 48 kroppen × 200 g per 35 dagen; 2,88 m²
- lampen 3,04 kWh per dag
- 3,19 L water per krop, waarvan 3,0 L verdampt

GEVRAAGD

- kg per jaar en per m²
- kWh per kg, g per kWh
- L per kg, met en zonder terugwinning

OPLOSSING

1. Teelten per jaar: $365/35 = 10{,}43$. Oogst: $48 \cdot 10{,}43 \cdot 0{,}200 = 100{,}1\,\text{kg}$ per jaar, of $100{,}1/2{,}88 = 34{,}8\,\text{kg}/(\text{m}^2 \cdot \text{jaar})$.
2. Lampen: $3{,}04 \cdot 365 = 1110\,\text{kWh}$ per jaar. $1110/100{,}1 = 11{,}1\,\text{kWh/kg}$, of $90\,\text{g}$ sla per kWh.
3. Water: $3{,}19\,\text{L}/0{,}200\,\text{kg} = 15{,}9\,\text{L/kg}$.
4. Met 90 % terugwinning gaat er per krop nog $0{,}1 \cdot 3{,}0 + 0{,}19 = 0{,}49\,\text{L}$ verloren: $2{,}5\,\text{L/kg}$.

ANTWOORD

Ongeveer 100 kg sla per jaar, 35 kg per m²; 11,1 kWh lampenergie per kg (90 g per kWh); 16 L water per kg, of 2,5 L met terugwinning.

Klopt dit? Dat ligt binnen de 10 tot 18 kWh per kg van volledig belichte teeltruimtes (hoofdstuk 20), en de water- en ruimtecijfers zijn die van een goede serre. Twee kanttekeningen. In de winter leveren de panelen te weinig voor 16 lichturen (hoofdstuk 20): dan komt

de stroom van het station, of groeit de sla trager. En de hele module verbruikt 3,27 kWh per dag: 11,9 kWh per kg, zonder de verwarming.

UITLEG · 36.16

Een teeltschema: elke negen dagen oogsten

16/21

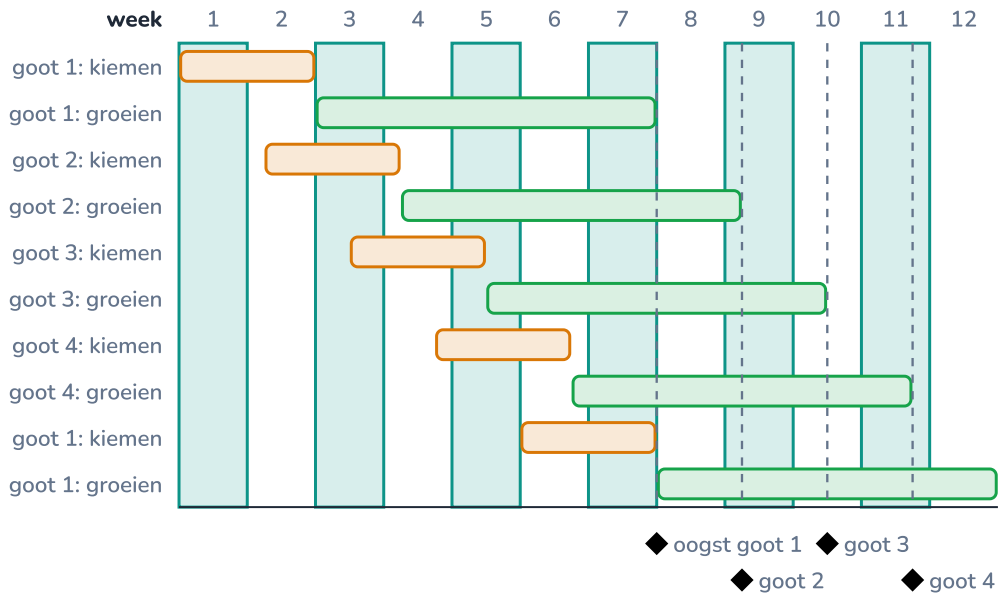
Zou je alle 48 plantplaatsen tegelijk beplanten, dan heb je 34 dagen niets en dan in één keer 9,6 kg sla, meer dan twee bewoners vers kunnen eten. Daarom werkt Habitat met een **verspringend teeltschema**: elke goot heeft haar eigen ritme.

1. **Kiemen** (2 weken): zaadjes in steenwolblokjes op een kiemplank aan de rand van het rek, onder hetzelfde licht.
2. **Uitplanten**: 12 plantjes in een lege goot.
3. **Groeien** (5 weken, 35 dagen) tot een krop van 200 g.
4. **Oogsten**, de goot schoonmaken (hoofdstuk 35) en meteen opnieuw beplanten.

Met vier goten en 35 dagen per teelt oogst je om de $35/4 = 8{,}75$ dagen één goot: 12 kroppen, ongeveer elke negen dagen. Gemiddeld is dat 1,4 kroppen per dag, of 247 g eetbare sla per dag voor twee bewoners. Elke 12 kroppen worden bijna gelijktijdig oogstklaar; wie elke dag verse sla wil, oogst van de rijpste goot enkele kroppen per dag en laat de rest nog een paar dagen staan.

Het reservoir blijft in gebruik

Omdat er altijd planten in de goten staan, kan het reservoir nooit helemaal droog. Om de vijf weken ververs je de oplossing in één keer (leegpompen, spoelen, nieuw vullen: een halfuur), terwijl de goten even droog staan. Goten ontsmetten gebeurt per goot, bij het oogsten.



Oranje: kiemen, groen: groeien in de goot. Na de opstart komt er elke 8,75 dagen een goot klaar.

UITLEG · 36.17

Van kWh tot kJ op je bord

17/21

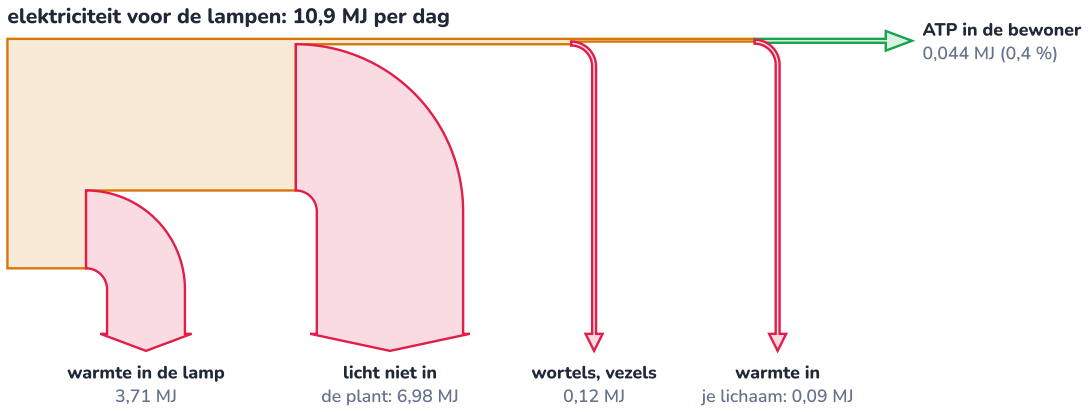
Volg één dag lamplicht door de hele keten (hoofdstuk 34 en F29):

- de lampen nemen $3{,}04\,\mathrm{kWh} = 10{,}9\,\mathrm{MJ}$ elektrische energie op;
- daarvan valt 7,2 MJ als licht op het kweekbed (66 %): de rest wordt warmte in de lamp;
- de sla legt 256 kJ vast als chemische energie in haar droge stof (3,5 % van het licht);
- van de oogst is 133 kJ eetbare voedingsenergie: wortels, buitenbladeren en vezels tellen niet mee;
- de bewoner zet ongeveer een derde daarvan om in ATP: 44 kJ.

Van elke 100 J uit het stopcontact komt 1,2 J als voedsel op het bord. De sla is daarom geen energiebron: ze is een bron van **verse vitaminen** (100 g kropsla bevat ongeveer de hele dagbehoefte aan vitamine K en een vijfde van die aan vitamine A), vezels, smaak, en van iets wat bewoners van poolstations en ruimtestations hoog inschatten: zelf iets zien groeien.

Alle energie van de lampen gaat verloren

Voor de voeding wel, maar niet voor de module: de warmte van de lampen verwarmt de kweekruimte (hoofdstuk 20). In de winter spaart dat stookenergie; in de zomer moet ze weggeventileerd worden. Ook de energie in je voedsel wordt uiteindelijk warmte: de 116 W van een bewoner (F29).



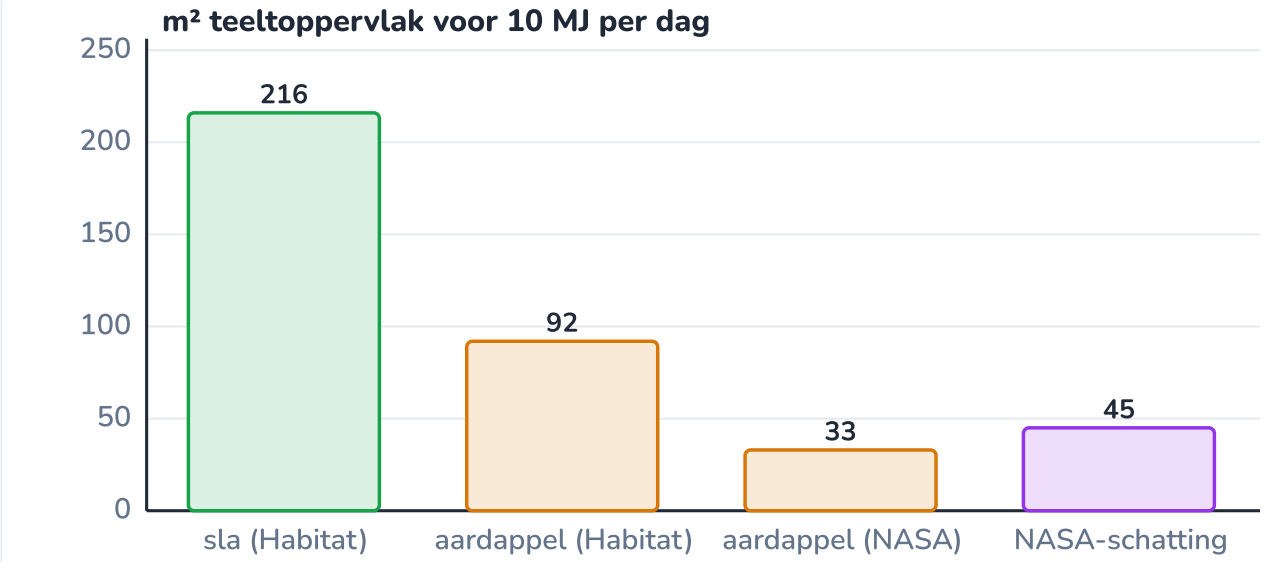
De banden zijn op schaal: van 10,9 MJ elektriciteit blijft 0,13 MJ voedsel over, en 0,04 MJ ATP.

UITGEWERKT VOORBEELD · 36.18

Habitat: een eerlijk voedselplan

18/21

Eén bewoner heeft ongeveer 10 MJ voedsel per dag nodig (F29). (a) Welk deel levert het kweekbed? (b) Hoeveel m² sla onder de lampen van Habitat zou 10 MJ per dag leveren? (c) Aardappelen leveren meer energie per m²: NASA haalde in een kweekkamer ongeveer 20 g droge knollen per m² per dag bij ongeveer 32 mol licht per m² per dag, en droge knol levert ongeveer 15,3 kJ per g. Hoeveel m² aardappel onder Habitat-licht, en hoeveel lampenergie per dag? (d) Vergelijk met de zonnepanelen van de module.



Hoe meer energie een gewas per m² vastlegt en hoe meer licht, hoe kleiner het oppervlak. De energie per MJ voedsel blijft groot.

GEGEVEN

- sla: 97 kJ eetbaar per krop, 48 kroppen per 35 dagen op 2,88 m²
- aardappel: 0,62 g droge knol per mol licht; 15,3 kJ per g
- Habitat-licht: 11,5 mol/(m²·dag), 1,06 kWh lampenergie per m² per dag
- zonnepanelen module: ± 1400 kWh per jaar (hoofdstuk 20)

GEVRAAGD

- deel van 10 MJ
- m² sla, m² aardappel
- kWh per dag
- vergelijking met de panelen

OPLOSSING

1. (a) $48 \cdot 97\,\mathrm{kJ}/35 = 133\,\mathrm{kJ}$ per dag: 1,3 % van 10 MJ (voor twee bewoners 0,7 %).
2. (b) Per m²: $133/2{,}88 = 46\,\mathrm{kJ}$ per dag; voor 10 MJ: $10\,000/46 = 216\,\mathrm{m}^2$.
3. (c) Onder Habitat-licht: $0{,}62 \cdot 11{,}5 = 7{,}1\,\mathrm{g}$ droge knol per m² per dag, $\cdot 15{,}3 = 109\,\mathrm{kJ}$. Voor 10 MJ: $92\,\mathrm{m}^2$. Lampen: $92 \cdot 1{,}06 = 97\,\mathrm{kWh}$ per dag.
4. (d) De panelen leveren gemiddeld $1400/365 = 3{,}8\,\mathrm{kWh}$ per dag, voor de hele module. Het voedsel van één bewoner onder lampen zou 25 keer zoveel vragen.

ANTWOORD

Het kweekbed levert 1,3 % van het voedsel van één bewoner. Voor alles zou je 216 m² sla nodig hebben, of 92 m² aardappel, met ongeveer 97 kWh lampenergie per dag: 25 keer de hele zonneopbrengst van de module.

Klopt dit? Ook met veel meer licht verandert de energierekening niet: NASA's aardappelen op 32 mol per dag vragen maar 33 m², maar evenveel lampenergie, omdat licht per m² nu eenmaal energie kost. NASA schat dat ongeveer 40 à 50 m² gewassen onder veel licht al het voedsel van één persoon kunnen leveren. Wat dat betekent voor Habitat: het voedsel komt met de bevoorrading (39 kg per bewoner per 30 dagen, F29); de module levert verse groenten; wie meer wil, bouwt een serre met daglicht (gratis licht), teelt gewassen met meer energie per m² (aardappel, bonen, graan), en gebruikt de lampen alleen om aan te vullen.

CODE · 36.19

Python: de opbrengst doorrekenen

19/21

Het rekenmodel van de voorbeelden in één functie. Alle waarden staan als constanten bovenaan, zodat je meteen ziet waarmee gerekend wordt; de functie geeft een woordenboek terug (hoofdstuk 28). Dezelfde functie berekent de getallen in de tabellen en grafieken van deze pagina.

Python **opbrengst.py** 27 regels

```python
# opbrengst.py - wat levert het kweekbed van de klimaatmodule op? (rekenmodel)
PLAATSEN = 48          # kroppen op het kweekbed
OPPERVLAK = 2.88       # m2
TEELT = 35             # dagen van uitplanten tot oogst
KROP = 0.200           # kg per krop bij de oogst
EETBAAR = 0.180        # kg die je echt eet (zonder buitenbladeren)
LAMPEN = 3.04          # kWh per dag (2 x 95 W, 16 uur)
WATER = 3.19           # liter per krop: verdampt water + water in de krop
KJ_PER_KG = 540        # voedingsenergie van sla (13 kcal per 100 g)


def opbrengst(plaatsen=PLAATSEN, teelt=TEELT, krop=KROP, lampen=LAMPEN):
    teelten = 365 / teelt
    kg = plaatsen * teelten * krop
    return {"kg per jaar": kg,
            "kg per m2 per jaar": kg / OPPERVLAK,
            "kWh per kg": lampen * 365 / kg,
            "liter water per kg": WATER / krop,
            "kJ voedsel per dag": plaatsen * teelten * EETBAAR * KJ_PER_KG / 365}


if __name__ == "__main__":
    r = opbrengst()
    for naam, waarde in r.items():
        print(f"{naam:22s} {waarde:8.1f}")
    deel = r["kJ voedsel per dag"] / 10000 * 100
    print(f"deel van 10 MJ per dag {deel:8.1f} %")
```

Uitvoer (gedraaid bij het bouwen van deze pagina):

```
kg per jaar              100.1
kg per m2 per jaar        34.8
kWh per kg                11.1
liter water per kg        15.9
kJ voedsel per dag       133.3
deel van 10 MJ per dag     1.3 %
```

In opdracht 36.11 gebruik je de functie om te zien wat een kortere teelt, meer licht of een ander gewas oplevert.

NASLAG · 36.20

Overzicht: hydroponie en voedselproductie

20/21

daglichtsom

$$\mathrm{DLI} = \mathrm{PPFD} \cdot t$$

symbool	betekenis	eenheid
PPFD	fotonenstroomdichtheid	µmol/(m²·s)
t	belichtingstijd	s
DLI	daglichtsom	mol/(m²·dag)

energie van een mol fotonen: 266 kJ (450 nm), 181 kJ (660 nm)

$$E = \frac{N_A\, h\, c}{\lambda}$$

µmol per joule

$$\text{rendement lamp} = \frac{\mathrm{PPFD} \cdot A}{P}$$

symbool	betekenis	eenheid
A	belichte oppervlakte	m²
P	elektrisch vermogen	W

nitrificatie (maakt het water zuurder)

$$NH_4^+ + 2\,O_2 \longrightarrow NO_3^- + 2\,H^+ + H_2O$$

stikstofbinding in de fabriek (Haber-Bosch)

$$N_2 + 3\,H_2 \longrightarrow 2\,NH_3$$

kg per m² per jaar

$$\text{opbrengst} = \frac{\text{plaatsen} \cdot m_{\text{krop}} \cdot 365}{t_{\text{teelt}} \cdot A}$$

Habitat: het hydroponiesysteem	
kweekbed	getrapt NFT-rek, 2,4 × 1,2 m (2,88 m²); 4 goten van 2,4 m, 12 plaatsen om de 20 cm, goten 30 cm uit elkaar, helling 1 : 50
reservoir en pomp	60 L, lichtdicht, 18-22 °C; pomp 4,8 L/min, 10 min per uur, 1,2 L/min per goot
oplossing	EC 1,4 mS/cm (1,2-1,8), pH 5,8 (5,5-6,5); stikstof in het volle reservoir volstaat ± 10 dagen
licht	2 × 95 W, 200 µmol/(m²·s), 16 h: 11,5 mol/(m²·dag); 3,0 µmol/J op het bed
teeltschema	kiemen 2 weken, groeien 35 dagen; om de 8,75 dagen één goot oogsten (12 kroppen)
opbrengst	± 100 kg per jaar (35 kg/m²); 11,1 kWh lampenergie per kg; 16 L water per kg (2,5 L met terugwinning)
voeding	± 247 g eetbare sla en 133 kJ per dag: 1,3 % van 10 MJ

HET GEHEEL · 36.21

Waar hydroponie en voedselproductie samenkomen

21/21

Dit hoofdstuk is een kruispunt: chemie (voedingsoplossing, EC, pH, stikstof), energie (lampen, zonnepanelen, rendement), biologie (fotosynthese, enzymen, bacteriën), elektronica en programmeren (sensoren, pomp, dimmen, logger) en ontwerpen (het rek, het teeltschema). Het integratieproject van Intermediate (hoofdstuk 40) bouwt er de volledige klimaatmodule mee.

F27 Planten: fotosynthese en groei
Telen zonder aarde en de Kratky-pot; beperkende factoren.

F29 Het lichaam als systeem
Wat een mens nodig heeft: 10 MJ voedsel per dag.

I12 Atomen en het periodiek systeem
Het recept van de voedingsoplossing, element per element.

I13 Chemische bindingen en stofeigenschappen
EC meten en bijsturen met stamoplossingen.

I14 Reacties en stoichiometrie
CO2-opname van de sla en de ventilatie van de kweekruimte.

I15 Zuren, basen en pH
De pH bijsturen met pH-min; waarom de pH stijgt.

I20 Zonne-energie dimensioneren
Het energiebudget: de lampen zijn 93 % van het verbruik.

I25 Van analoog naar digitaal
Lampen dimmen met PWM tot 200 µmol/(m²·s).

I34 Enzymen en stofwisseling
Temperatuuroptimum en de energieketen tot ATP.

I35 Micro-organismen en biotechnologie
Nitrificerende bacteriën, ziekteverwekkers, compost.

I38 Duurzaam ontwerpen
De levenscyclus en de CO2-voetafdruk van een krop sla uit de module.

I40 Integratieproject: de klimaatmodule
De volledige module ontwerpen, bouwen en verdedigen.

E26 Chemisch evenwicht
Een buffer die de pH van de oplossing stabiel houdt.

E31 Bioprocestechniek
Een algenreactor die CO2 omzet in zuurstof en voedsel.

E32 Ecologische modellen
Hoe stabiel is een gesloten kringloop met planten, bacteriën en mensen?

Habitat

Voor Habitat leggen we vast: vier NFT-goten van 2,4 m met 12 plaatsen, helling 1 : 50, 1,2 L/min per goot; een verspringend teeltschema met om de 8,75 dagen één goot oogsten; ± 100 kg sla per jaar, 11,1 kWh lampenergie per kg, 16 L water per kg. De module levert verse groenten, geen voedselautonomie: 1,3 % van de energie van één bewoner. In opdracht 36.13 stel je het voedselplan op, in 36.14 de fiche van het systeem.

SAMENVATTING

In het kort

- Een voedingsoplossing levert alle mineralen; A (calcium) en B (sulfaat, fosfaat) blijven apart tot ze verdund zijn. Per teelt neemt de sla 3 à 4 keer meer N en K op dan het reservoir bevat.
- De EC stijgt als er water verdwijnt en daalt als er ionen opgenomen worden: lees EC altijd samen met het volume. De pH stijgt door nitraatopname; tussen 5,5 en 6,5 blijft alles opneembaar.
- Planten tellen fotonen van 400-700 nm (PAR): PPFD in µmol/(m²·s), daglichtsom in mol/(m²·dag). Habitat: 200 µmol, 16 h, 11,5 mol per dag.
- Een rode foton draagt minder energie dan een blauwe maar doet evenveel voor de fotosynthese: rode leds maken meer fotonen per joule.
- NFT, eb en vloed, druppel, diepwater en Kratky verschillen in hoe water en lucht bij de wortels komen; Habitat gebruikt een gesloten NFT-rek met vier goten.
- Bacteriën binden stikstof, nitrificeren ammonium tot nitraat en denitrificeren; aquaponie laat vissen en planten dezelfde kringloop delen, maar het visvoer komt van buiten.
- Het kweekbed levert ± 100 kg sla per jaar (35 kg/m²), 90 g per kWh lampenergie, ± 16 L water per kg (2,5 L met terugwinning).
- Dat is 1,3 % van de voedingsenergie van één bewoner: de module levert verse groenten, geen voedselautonomie. Voedsel onder lampen kost ± 10 kWh per MJ.

BEGRIPPEN

Wat je nu kent

hydroponie
Planten telen zonder aarde, met de wortels in (of in contact met) water met opgeloste voedingszouten.

NFT
Nutrient film technique: een dunne film voedingsoplossing stroomt door licht hellende goten langs de wortels.

stamoplossing
Geconcentreerde oplossing van voedingszouten die je verdunt; A (met calcium) en B (met sulfaat en fosfaat) blijven gescheiden omdat ze samen neerslaan.

PAR
Fotosynthetisch actieve straling: licht van 400 tot 700 nm, het deel dat planten voor fotosynthese gebruiken.

PPFD
Fotonenstroomdichtheid: het aantal lichtdeeltjes (in µmol) dat per seconde op een vierkante meter valt, in µmol/(m²·s).

daglichtsom
Het aantal mol lichtdeeltjes (PAR) dat op één dag op een vierkante meter valt; DLI = PPFD · tijd.

µmol per joule
Rendement van een groeilamp: hoeveel micromol lichtdeeltjes (PAR) ze maakt per joule elektrische energie.

aquaponie
Vissen en planten in één waterkringloop: bacteriën zetten de ammonium van de vissen om in nitraat voor de planten.

nitrificatie

stikstofbinding

Omzetting van ammonium via nitriet naar nitraat door bacteriën (Nitrosomonas, Nitrobacter), met zuurstof; maakt het water zuurder.

Omzetting van stikstofgas uit de lucht in ammonium, door bacteriën (bv. in wortelknolletjes) of industrieel (Haber-Bosch).

teeltschema

Planning van zaaien, uitplanten en oogsten, zodat er regelmatig geoogst kan worden.

HANDBOEK · HOOFDSTUK 37

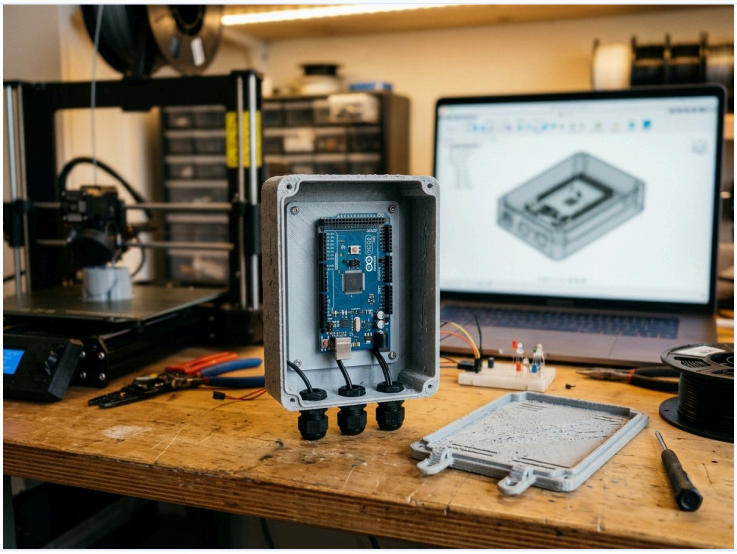
Parametrisch ontwerpen

Een ontwerp als functie: verander één maat en de rest volgt

In Fundamental tekende je de sensorhouder met vaste maten: wie een andere sensor kiest, begint opnieuw. Een **parametrisch** ontwerp legt vast hoe de maten van elkaar afhangen. Je verandert de printplaat, en de doos, het deksel, de gaten, de massa en de werktekening passen zich aan. Je leert schetsen met beperkingen, bewerkingen die op elkaar bouwen, toleranties en verbindingen die werken op een 3D-printer, en je ontwerpt de regelkast van de klimaatmodule als een functie, in Python en in OpenSCAD.

9× uitleg · 6× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 2× code · ± 10 lestijden

D Ontwerpen en maken · W Wiskundig modelleren



De regelkast van de klimaatmodule: één model, en elke maat volgt uit een paar invoermaten.

Na dit hoofdstuk

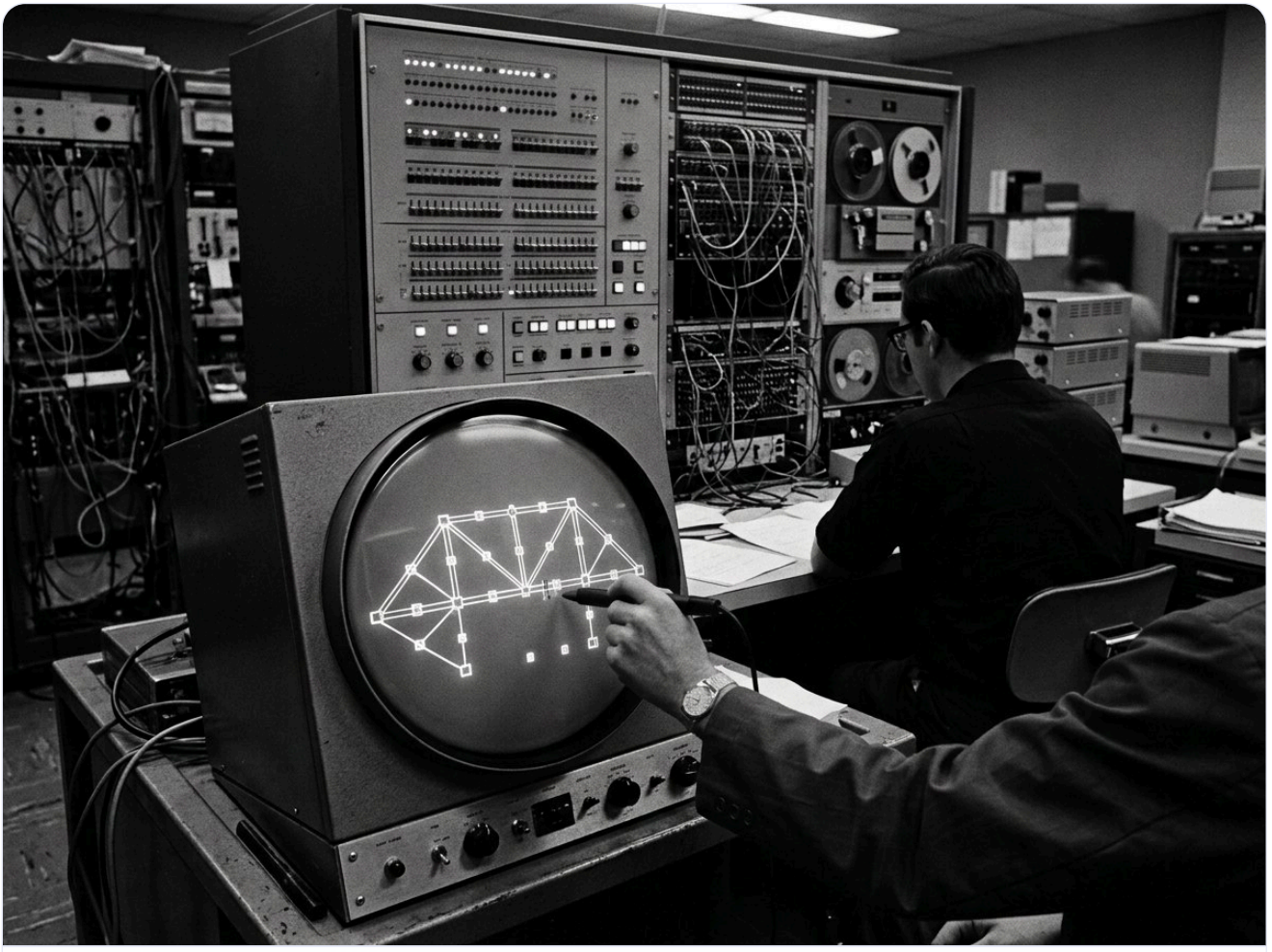
- Je kan invoermaten en afgeleide maten onderscheiden en hun afhankelijkheden als formules schrijven.
- Je kan een schets volledig bepalen met geometrische beperkingen en maten, en de vrijheidsgraden tellen.
- Je kan extruderen, uitsparen, draaien, afronden, afschuinen, uithollen en patronen gebruiken, en met de regel van Pappus het volume van een draaideel berekenen.
- Je kan speling-, overgangs- en perspassing uitleggen, gaten maatvoeren voor je eigen printer en een tolerantiestapeling in het slechtste geval doorrekenen.
- Je kan een schroef-, klik- of persverbinding kiezen en een klikvinger dimensioneren met de toegelaten rek.
- Je kan een parametrisch model in Python of OpenSCAD schrijven dat maten, massa, printtijd en prijs berekent en ontwerpregels controleert.
- Je kan een ontwerpruimte verkennen, varianten vergelijken en een werktekening afleiden uit dezelfde parameters.

Een lichtpen die meedenkt

In 1963 beschreef Ivan Sutherland in zijn doctoraat aan het MIT een programma dat hij **Sketchpad** noemde. Hij tekende met een lichtpen rechtstreeks op het scherm van de TX-2, een grote experimentele computer van het Lincoln Laboratory. Nieuw was niet dat je kon tekenen, maar wat de computer met de tekening deed. Je kon zeggen dat twee lijnen evenwijdig moesten zijn, dat een hoek recht was of dat twee lijnen even lang waren, en Sketchpad zocht zelf een tekening die aan al die voorwaarden voldeed. Je kon ook één 'moedertekening' maken en er kopieën van plaatsen: veranderde je de moeder, dan veranderden alle kopieën mee. Sutherland kreeg er in 1988 de Turingprijs voor, de hoogste onderscheiding in de informatica.

Het duurde nog een kwarteeuw voor het idee in de fabrieken aankwam. Eind jaren 1980 bracht het Amerikaanse bedrijf PTC het programma Pro/ENGINEER uit: een onderdeel is er een reeks bewerkingen met maten en beperkingen, en wie één maat verandert, laat het hele model opnieuw opbouwen. Vandaag werkt bijna elk CAD-programma zo, ook de gratis programma's **FreeCAD** (sinds 2002) en **OpenSCAD** (sinds 2010), waarin je een onderdeel schrijft als een programma.

> **Waarom dit hoofdstuk ertoe doet**
>
> De klimaatmodule heeft kasten, klemmen, houders en kapjes nodig, en elk ervan verandert nog: een andere printplaat, een extra kabel, een dikkere wand. Wie parametrisch ontwerpt, past één getal aan in plaats van alles opnieuw te tekenen, en maakt minder fouten. In dit hoofdstuk wordt de regelkast RK-1 zo'n model; in het integratieproject (hoofdstuk 40) bouw je de module ermee.



Tekenen met voorwaarden in plaats van met vaste lijnen: het begin van parametrisch ontwerpen.

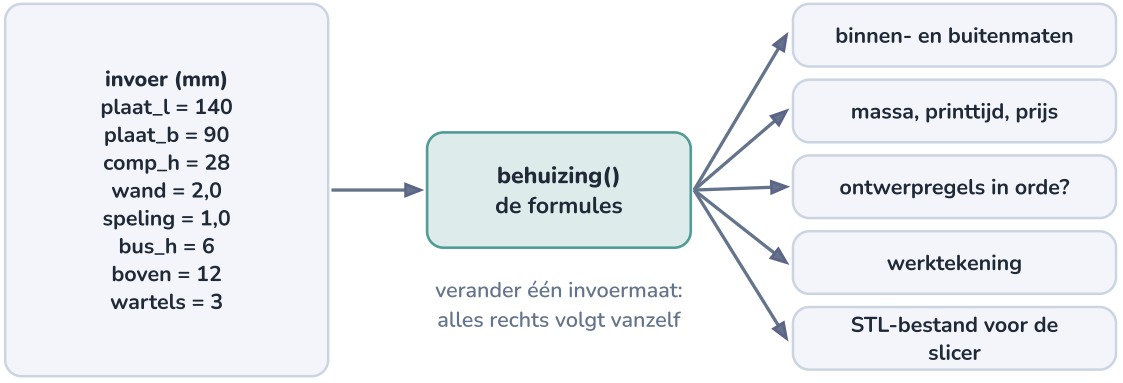
UITLEG · 37.2

Wat je al weet, en wat parametrisch verandert

Je kent de ontwerpcyclus, je kan tekenen in aanzichten en je hebt in CAD een sensorhouder gemaakt. Wat er nu bij komt, is één idee: een ontwerp is een **functie** (hoofdstuk 4). De invoer is een handvol **invoermaten** of **parameters**: hoe groot is de printplaat, hoe hoog het hoogste onderdeel, hoe dik de wand. De uitvoer is al de rest: de binnen- en buitenmaten, de plaats van de gaten, de massa, de printtijd, en of het ontwerp aan de regels voldoet.

wat je al kan	waar	wat er nu bij komt
een programma van eisen: moet, wens, meetbaar	F30	eisen worden regels die het model zelf controleert
aanzichten, maatvoering, stuklijst	F31	een werktekening die volgt uit de parameters
CAD in de browser, 45°-regel, speling 0,2 mm, printtijd uit het volume	F32	beperkingen, bewerkingsboom, tolerantiestapeling
een functie: de uitvoer hangt af van de invoer	I4	een ontwerp als functie van een paar maten
zwaartepunt van samengestelde vormen	I10	het volume van een draaideel (regel van Pappus)
PETG in de kas, Tg minstens 15 °C boven de hoogste temperatuur	I17	het materiaal van de regelkast
Python-functies met standaardwaarden, woordenboeken	I28	behuizing() geeft alle maten in één woordenboek

Het voorbeeld van dit hoofdstuk is de **regelkast RK-1** van de klimaatmodule: een doos voor het stuurdeel, de Arduino Mega 2560 met een shield, de klok en de SD-kaart (hoofdstuk 30) op een montageplaat. Het vermogensdeel met de MOSFETs, de H-bruggen en de zekeringen zit in een aparte kast. De regelkast hangt in de kweekruimte, waar het warm en vochtig is.

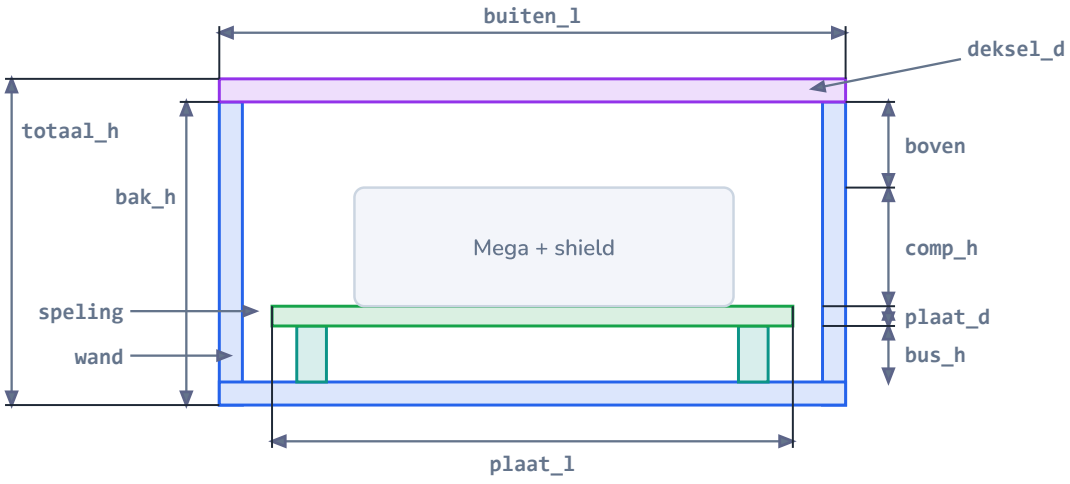


Links wat je kiest, rechts wat volgt. Je ontwerpt de pijlen, niet één doos.

UITGEWERKT VOORBEELD · 37.3

Van invoermaten naar alle maten van de regelkast

De montageplaat van de regelkast is 140 × 90 × 3 mm. Het hoogste onderdeel (Mega met shield) staat 28 mm boven de plaat. De plaat rust op afstandsbussen van 6 mm; rond de plaat blijft 1,0 mm speling; boven de onderdelen 12 mm ruimte voor kabels. Wand en bodem zijn 2,0 mm, het deksel 2,0 mm. Schrijf de afhankelijkheden als formules en bereken de binnen- en buitenmaten.



De ketting van maten: elke hoogte en breedte is een som van parameters.

GEGEVEN

- plaat_l = 140, plaat_b = 90, plaat_d = 3, comp_h = 28 (mm)
- bus_h = 6, speling = 1,0, boven = 12, wand = 2,0, deksel_d = 2,0 (mm)

GEVRAAGD

- formules en waarden voor de binnen- en buitenmaten, de hoogte van de bak en de totale hoogte

OPLOSSING

1. **Lengte en breedte.** Binnen komt de plaat met aan elke kant de speling: $\text{binnen}_l = \text{plaat}_l + 2 \cdot \text{speling} = 140 + 2 = 142\,\text{mm}$ en $\text{binnen}_b = 92\,\text{mm}$. Buiten komt er aan elke kant een wand bij: $\text{buiten}_l = \text{binnen}_l + 2 \cdot \text{wand} = 146\,\text{mm}$, $\text{buiten}_b = 96\,\text{mm}$.

2. **Hoogte.** De binnenhoogte is een ketting van maten van onder naar boven: $\text{binnen}_h = \text{bus}_h + \text{plaat}_d + \text{comp}_h + \text{boven} = 6 + 3 + 28 + 12 = 49\,\text{mm}$.

3. **Bak en totaal.** De bodem is even dik als de wand: $\text{bak}_h = 49 + 2 = 51\,\text{mm}$; met het deksel erbij $\text{totaal}_h = 53\,\text{mm}$.

4. **Welke maten zijn invoer?** Alleen de negen gegeven maten. Elke andere maat is een **afgeleide maat**: je typt ze nooit in, je berekent ze.

ANTWOORD

Buitenmaten 146 × 96 × 53 mm (bak 51 mm hoog), binnenmaten 142 × 92 × 49 mm.

Klopt dit? Verander de plaat naar 100 × 60 mm: dan wordt de kast 106 × 66 mm, zonder één lijn opnieuw te tekenen. En de eenheden kloppen: je telt altijd millimeters bij millimeters.

UITLEG · 37.4

Schetsen met beperkingen

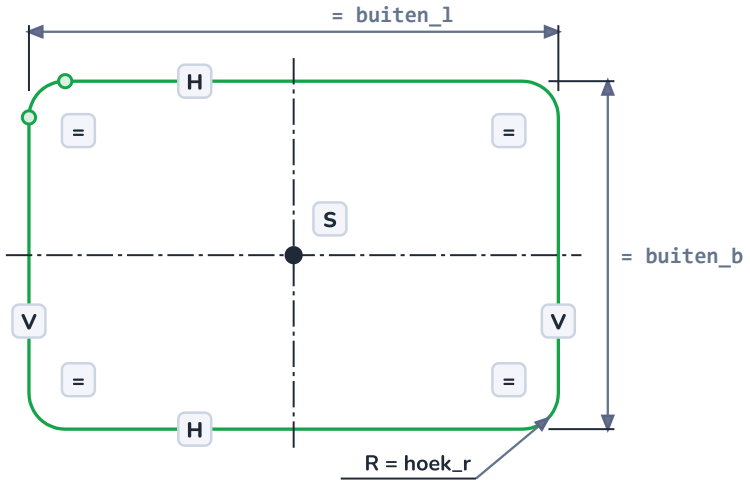
Elk CAD-model begint met een **schets**: een vlakke tekening die je daarna optrekt of draait. In een parametrisch programma teken je de schets eerst ruw, en leg je daarna vast wat je bedoelt met **beperkingen** (*constraints*). Er zijn twee soorten:

- **Geometrische beperkingen** leggen de vorm vast zonder getallen: deze lijn is horizontaal, die twee lijnen zijn evenwijdig, deze boog raakt aan die lijn, de vier stralen zijn gelijk. Ze vangen de **ontwerpintentie**: wat moet er altijd waar blijven, welke maat je ook kiest?
- **Maatbeperkingen** leggen getallen vast: een afstand, een straal, een hoek. In een goed model verwijst zo'n maat naar een parameter (`buiten_l`) in plaats van naar een getal (146).

beperking	betekenis	neemt weg
samenvallend	twee punten liggen op elkaar (het eind van een lijn op het begin van de volgende)	2
horizontaal / verticaal	een lijn ligt waterpas of loodrecht	1
evenwijdig / loodrecht	twee lijnen maken 0° of 90°	1
raaklijn	een lijn en een boog gaan zonder knik in elkaar over	1
gelijk	twee lengtes of twee stralen zijn gelijk	1
symmetrisch	twee punten liggen gespiegeld rond een lijn	2
maat (afstand, straal, hoek)	een getal, of beter: een parameter	1
vastzetten	een punt ligt op een vaste plaats (de oorsprong)	2

Een schets die nog kan bewegen, heeft **vrijheidsgraden**. Elke beperking neemt er één of meer weg. Het doel is een **volledig bepaalde** schets: nul vrijheidsgraden. FreeCAD kleurt zo'n schets groen. Te weinig beperkingen en de schets verspringt als je een parameter verandert; te veel of tegenstrijdige (**overbepaald**) en het programma weigert.

> **Eerst de vorm, dan de maten**
>
> Zet eerst de geometrische beperkingen (horizontaal, symmetrisch, raaklijn), dan de maten. Leg een symmetrische schets rond de oorsprong: dan blijft het midden in het midden, welke lengte je ook kiest, en kan een latere bewerking (een gat in het midden) daarop steunen.



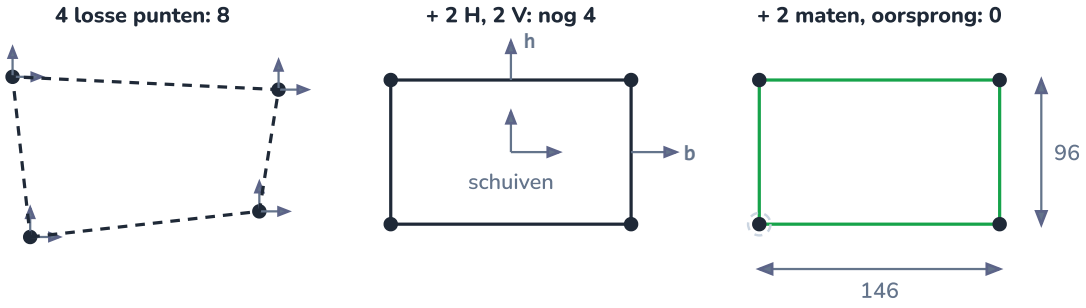
De buitenomtrek met zijn beperkingen: twee maten verwijzen naar parameters, de rest is vorm.

UITGEWERKT VOORBEELD · 37.5

Vrijheidsgraden tellen: is de schets bepaald?

5/22

(a) Een rechthoek wordt getekend als vier punten met vier lijnen ertussen. Hoeveel vrijheidsgraden heeft hij, en welke beperkingen maken hem volledig bepaald? (b) De buitenomtrek van de regelkast bestaat uit vier lijnen en vier bogen. Tel de vrijheidsgraden en de beperkingen.



Van acht vrijheidsgraden naar nul: eerst de richtingen, dan de maten en de plaats.

GEGEVEN

- een punt in het vlak: 2 vrijheidsgraden (x en y)
- een lijn: 4 (twee eindpunten); een boog: 5 (middelpunt 2, straal 1, begin- en eindhoek 2)
- de tabel met beperkingen

GEVRAAGD

- het aantal vrijheidsgraden vóór en na de beperkingen

OPLOSSING

1. **(a) De rechthoek als vier punten**: $4 \times 2 = 8$ vrijheidsgraden. Twee horizontale en twee verticale lijnen nemen er 4 weg: over blijven 4, namelijk verschuiven in x en in y, breedte en hoogte (figuur). Twee maten (146 en 96) nemen er 2 weg, de hoek op de oorsprong vastzetten nog 2: $8 - 4 - 2 - 2 = 0$.

2. **(b) Vier lijnen en vier bogen**: $4 \times 4 + 4 \times 5 = 36$ vrijheidsgraden. De acht plaatsen waar een lijn en een boog elkaar raken, krijgen elk 'samenvallend' (2) en 'raaklijn' (1): $8 \times 3 = 24$. Over: 12.

3 Twee horizontale en twee verticale lijnen: nog 8 over (plaats 2, breedte, hoogte en vier stralen). 'Gelijk' tussen de vier stralen: 3 weg, nog 5.

4 Drie maten (buiten_l, buiten_b, hoek_r) en het middelpunt vast op de oorsprong (symmetrisch rond de twee assen, of één punt vastzetten): $5 - 3 - 2 = 0$.

ANTWOORD

(a) 8 vrijheidsgraden, weggenomen door 4 richtingsbeperkingen, 2 maten en 2 voor de plaats. (b) 36 vrijheidsgraden, weggenomen door $24 + 4 + 3 + 3 + 2 = 36$ beperkingen: volledig bepaald.

Klopt dit? Een vijfde maat, bv. de diagonaal, zou overbepaald zijn: die volgt al uit breedte en hoogte. Het programma meldt dan een overbodige of tegenstrijdige beperking. Tellen voorkomt die fout.

UITLEG · 37.6

Bewerkingen: extruderen, draaien, afronden

6/22

Uit een schets maak je een lichaam met een **bewerking** (*feature*). De twee basisbewerkingen zijn **extruderen** (de schets optrekken over een afstand, zoals een koekjesvorm door het deeg) en **draaien** (een profiel rond een as draaien, zoals op een draaibank). Daarna werk je het lichaam af: materiaal weghalen, ribben afronden of afschuinen, uithollen tot een wand, herhalen in een patroon.

bewerking	wat ze doet	in de regelkast
extruderen (pad)	een schets over een afstand optrekken tot een lichaam	de buitenvorm, 51 mm hoog
uitsparen (pocket)	een schets weghalen uit een lichaam	de wartelgaten, het venster
draaien (revolve)	een profiel rond een as draaien	de afstandsbussen, een wartel
afronden (fillet)	een rib vervangen door een boog	de verticale hoeken, R4
afschuinen (chamfer)	een rib schuin afsnijden	de onderrand op het printbed
uithollen (shell)	een massief lichaam hol maken met een vaste wanddikte	de bak, wand 2 mm
patroon (array)	een bewerking herhalen op een lijn, een raster of een cirkel	drie wartelgaten, vier bussen
spiegelen	een bewerking spiegelen rond een vlak	de twee klikvingers

Het volume van een extrusie is eenvoudig: oppervlakte van de schets maal de hoogte. Voor een afgeronde rechthoek met straal r gaat er in elke hoek een stukje af, samen een vierkant van $2r$ min een cirkel:

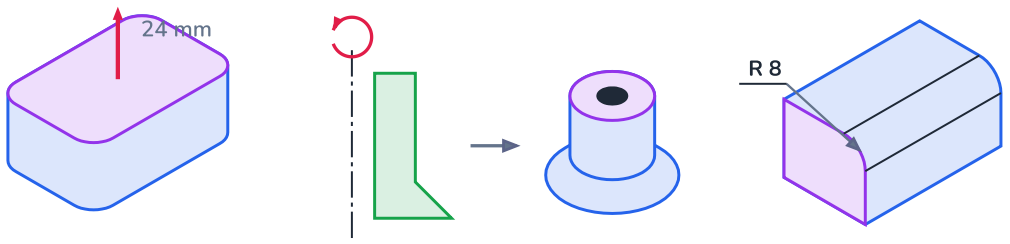
OPPERVLAKTE VAN EEN AFGERONDE RECHTHOEK

$$A = l \cdot b - (4 - \pi)\, r^2$$

Voor de buitenomtrek van RK-1: $146 \cdot 96 - (4 - \pi) \cdot 4^2 = 14\,016 - 13{,}7 = 14\,002\ \text{mm}^2$.

Afronden of afschuinen op het printbed?

Een afronding langs de onderste rand begint op het bed als een bijna horizontale overhang: de eerste lagen hangen in de lucht. Een afschuining van 45° print zonder problemen (F32). Rond dus de verticale ribben en de bovenrand af, en schuin de onderrand af.



extruderen
schets + hoogte

draaien (revolve)
profiel + as

afronden (fillet)
een rib rond maken

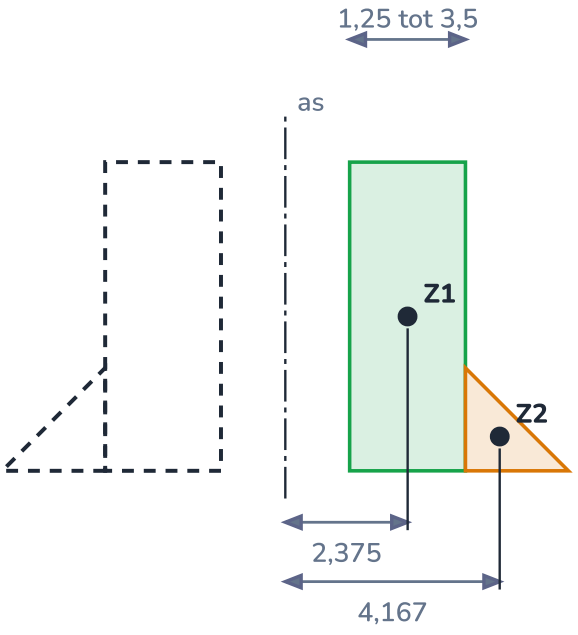
Een schets die opgetrokken wordt, een profiel dat rond een as draait, een rib die rond wordt.

UITGEWERKT VOORBEELD · 37.7

Het volume van een draaideel: de regel van Pappus

7/22

Een afstandsbus van de regelkast is een draaideel: een buisje Ø 7 mm met een gat Ø 2,5 mm, 6 mm hoog, met onderaan een voetje van 45° (2 mm breed en 2 mm hoog) dat de bus steviger op de bodem zet. Bereken het volume met de regel van Pappus en controleer met de formules van een cilinder.



rechthoek: A1 = 2,25 × 6 = 13,5 mm²
driehoek: A2 = ½ × 2 × 2 = 2 mm²

elk stukje beschrijft
een ring rond de as

Het profiel (rechts van de as) draait rond; de stippellijn is de gespiegelde helft.

GEGEVEN

- profiel: rechthoek van $r = 1{,}25$ tot $3{,}5\,\text{mm}$, hoogte $6\,\text{mm}$
- driehoek met rechthoekszijden 2 mm aan de voet
- regel van Pappus: $V = 2\pi \cdot \bar{x} \cdot A$

GEVRAAGD

- V in mm³

OPLOSSING

1. **Waarom de regel werkt.** Elk stukje ΔA van het profiel op afstand x van de as beschrijft een ring met volume $2\pi x \cdot \Delta A$. Tel je alle ringen op, dan krijg je $2\pi \sum x\,\Delta A = 2\pi\,\bar{x}\,A$, met $\bar{x}$ de afstand van het zwaartepunt tot de as (hoofdstuk 10).

2. **Rechthoek.** $A_1 = 2{,}25 \cdot 6 = 13{,}5\,\text{mm}^2$, $\bar{x}_1 = (1{,}25 + 3{,}5)/2 = 2{,}375\,\text{mm}$.

3. **Driehoek.** $A_2 = \tfrac{1}{2} \cdot 2 \cdot 2 = 2\,\text{mm}^2$; het zwaartepunt ligt op een derde van de rechthoekszijde: $\bar{x}_2 = 3{,}5 + \tfrac{2}{3} = 4{,}167\,\text{mm}$.

4. **Samen.** $V = 2\pi\,(A_1\bar{x}_1 + A_2\bar{x}_2) = 2\pi\,(32{,}06 + 8{,}33) = 2\pi \cdot 40{,}40 = 253{,}8\,\text{mm}^3$.

ANTWOORD

$V \approx 254\,\text{mm}^3$ **(0,25 cm³, 0,32 g PETG per bus).**

Klopt dit? Het buisje alleen: $\pi\,(3{,}5^2 - 1{,}25^2) \cdot 6 = 201{,}5\,\text{mm}^3$. Het voetje: $2\pi \cdot 4{,}167 \cdot 2 = 52{,}4\,\text{mm}^3$. Samen 253,8 mm³: dezelfde uitkomst. Pappus is vooral handig als het profiel geen rechthoek is en een cilinderformule dus niet volstaat.

UITLEG · 37.8

De bewerkingsboom en de spreadsheet

8/22

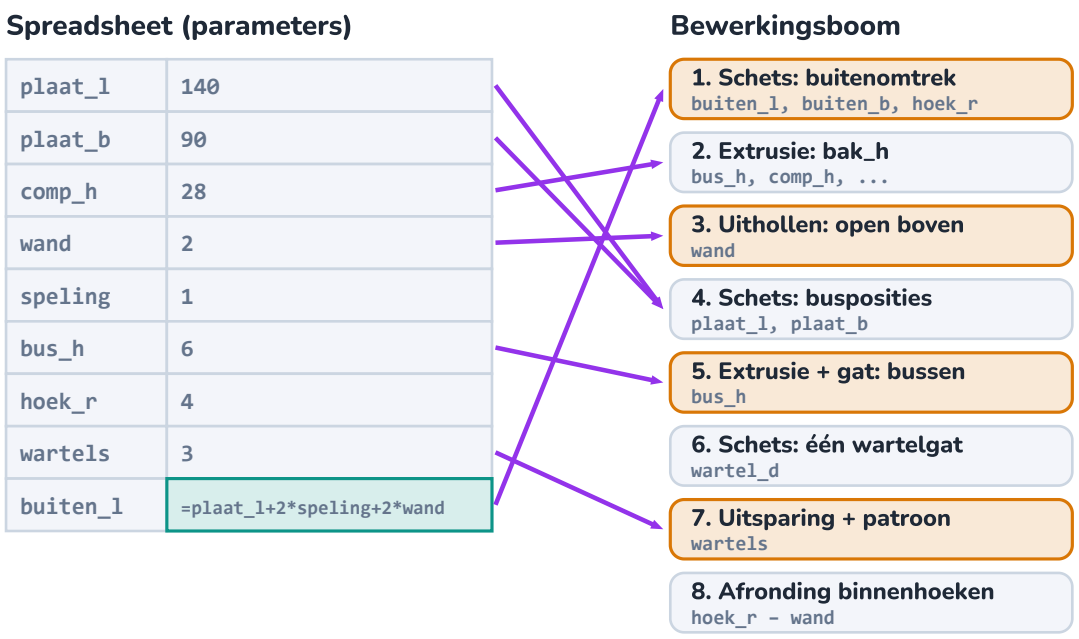
Een CAD-model onthoudt niet de vorm, maar de **weg ernaartoe**: de bewerkingsboom. Elke bewerking bouwt op de vorige. Verander je een parameter, dan voert het programma de hele boom opnieuw uit, van boven naar onder. Zo werkt het in FreeCAD:

1. Maak een **spreadsheet** met de parameters, en geef elke cel een naam (*alias*), bv. `plaat_l`. Afgeleide maten zijn formules in de spreadsheet: `=plaat_l + 2*speling + 2*wand`.
2. Teken de schets van de buitenomtrek, symmetrisch rond de oorsprong, en typ in de maatbeperking `=Spreadsheet.buiten_l` in plaats van 146.
3. Extrudeer tot `bak_h`, hol uit met `wand`, en zet de bussen en de wartelgaten met schetsen die ook naar de spreadsheet verwijzen.
4. Test het model: zet `plaat_l` op 100 en op 200. Klopt alles nog? Dan zit de ontwerpintentie in het model.

De volgorde doet ertoe. Rond je de hoeken af vóór het uithollen, dan krijgt de binnenkant vanzelf de juiste straal ($r - \text{wand}$). Zet je de wartelgaten op 30, 60 en 90 mm van de rand, dan staan ze bij een kleinere kast buiten de wand; zet je ze op $i \cdot \text{buiten}_b/(n+1)$, dan verdelen ze zich altijd gelijk.

"Als het model er goed uitziet, is het parametrisch"

Een model met ingetypte getallen ziet er even goed uit, tot iemand een maat verandert. Dan blijft een gat op 60 mm staan terwijl de kast korter werd, of verwijst een afronding naar een rib die niet meer bestaat. Dat laatste heet het *topologische naamprobleem*: een latere bewerking hangt aan een vlak of een rib van een eerdere, en die krijgt na de wijziging een ander nummer. FreeCAD heeft dat sinds versie 1.0 (2024) grotendeels opgelost, maar de regel blijft: verwijs naar schetsen, vlakken van de oorsprong en parameters, niet naar losse ribben.



De pijlen tonen welke parameter welke bewerking stuurt; buiten_l is zelf een formule.

CODE · 37.9

Het model in Python: behuizing.py

9/22

Een parametrisch model is een functie, dus je kan het ook gewoon programmeren (hoofdstuk 28). De functie `behuizing()` krijgt de invoermaten, met standaardwaarden voor alles wat meestal niet verandert, en geeft een woordenboek terug met alle afgeleide maten, de volumes, de massa, de printtijd en de prijs. Een tweede functie, `controleer()`, toetst de ontwerpregels. Onderaan rekent het programma vier varianten door.

Python **behuizing.py** 76 regels

```python
# behuizing.py - een parametrische behuizing: alle maten volgen uit een handvol invoermaten
import math

RHO = 1.27              # g/cm³, PETG (hoofdstuk 17)
DEBIET = 3.0            # mm³/s, gemiddeld debiet van de printer (rekenwaarde)
PRIJS = 20.0            # euro per kg filament
BED = (220, 220, 250)   # bouwvolume van de printer in mm (rekenwaarde)


def hoekvlak(l, b, r):
    """Oppervlakte (mm²) van een rechthoek l x b met vier afgeronde hoeken met straal r."""
    return l * b - (4 - math.pi) * r ** 2


def ring(d_buiten, d_binnen, h):
    """Volume (mm³) van een buisje: een cilinder met een gat erin."""
    return math.pi / 4 * (d_buiten ** 2 - d_binnen ** 2) * h


def behuizing(plaat_l, plaat_b, comp_h, wand=2.0, speling=1.0, bus_h=6.0, plaat_d=3.0, boven=12.0,
              hoek_r=4.0, deksel_d=2.0, lip_h=5.0, lip_b=1.6, lip_speling=0.3, wartels=3, wartel_d=12.5):
    """Alle maten (mm), de volumes (mm³), de massa (g), de printtijd (h) en de prijs (euro)."""
    m = {}
    m["binnen_l"] = plaat_l + 2 * speling
    m["binnen_b"] = plaat_b + 2 * speling
    m["binnen_h"] = bus_h + plaat_d + comp_h + boven
    m["buiten_l"] = m["binnen_l"] + 2 * wand
    m["buiten_b"] = m["binnen_b"] + 2 * wand
```

```
    m["bak_h"] = m["binnen_h"] + wand                # de bodem is even dik als de wand
    m["totaal_h"] = m["bak_h"] + deksel_d
    r_binnen = max(hoek_r - wand, 0.0)
    bak = (hoekvlak(m["buiten_l"], m["buiten_b"], hoek_r) * m["bak_h"]
           - hoekvlak(m["binnen_l"], m["binnen_b"], r_binnen) * m["binnen_h"])
    bak += 4 * ring(7.0, 2.5, bus_h)                 # vier afstandsbussen, gat voor een schroef M3
    bak -= wartels * ring(wartel_d, 0, wand)         # gaten voor de kabelwartels
    lip_l = m["binnen_l"] - 2 * lip_speling
    lip_bb = m["binnen_b"] - 2 * lip_speling
    r_lip = max(r_binnen - lip_speling, 0.0)
    deksel = hoekvlak(m["buiten_l"], m["buiten_b"], hoek_r) * deksel_d
    deksel += (hoekvlak(lip_l, lip_bb, r_lip)
               - hoekvlak(lip_l - 2 * lip_b, lip_bb - 2 * lip_b, max(r_lip - lip_b, 0.0))) * lip_h
    m["volume_bak"], m["volume_deksel"] = bak, deksel
    m["massa"] = (bak + deksel) / 1000 * RHO
    m["printtijd"] = (bak + deksel) / DEBIET / 3600
    m["prijs"] = m["massa"] / 1000 * PRIJS
    return m


def controleer(m, wand=2.0, wartels=3, wartel_d=12.5, lip_speling=0.3):
    """De ontwerpregels: geeft een lijst met wat niet klopt (een lege lijst = in orde)."""
    fout = []
    if wand < 1.2:
        fout.append("wand dunner dan drie lijnen van 0,4 mm")
    if max(m["buiten_l"], m["buiten_b"]) > BED[0] or m["bak_h"] > BED[2]:
        fout.append("past niet op de printer")
    if wartel_d + 10 > m["bak_h"]:
        fout.append("wand te laag voor het wartelgat")
    if wartels * (wartel_d + 10) > m["buiten_b"]:
        fout.append("te veel wartels voor de korte zijde")
    if lip_speling < 0.3:
        fout.append("deksel kan klemmen")
    return fout


varianten = {                       # naam: (plaat_l, plaat_b, comp_h, wartels)
    "meetkas (Uno)": (68.6, 53.4, 15, 2),
    "regelkast RK-1": (140, 90, 28, 3),
    "relaiskast": (100, 60, 20, 4),
    "alles samen": (220, 150, 30, 5),
}
print(f"{'variant':15} {'buiten (mm)':>17} {'massa':>7} {'print':>6} {'prijs':>6}  regels")
for naam, (l, b, h, n) in varianten.items():
    m = behuizing(l, b, h, wartels=n)
    fouten = controleer(m, wartels=n)
    maten = f"{m['buiten_l']:.0f} x {m['buiten_b']:.0f} x {m['totaal_h']:.0f}"
    print(f"{naam:15} {maten:>17} {m['massa']:5.0f} g {m['printtijd']:4.1f} h {m['prijs']:5.2f}  {'; '.join(fouten) or
'in orde'}")
```

Uitvoer:

```
variant              buiten (mm)   massa  print  prijs  regels
meetkas (Uno)         75 x 59 x 40    49 g  3.5 h  0.97  in orde
regelkast RK-1      146 x 96 x 53   134 g  9.8 h  2.69  in orde
relaiskast          106 x 66 x 45    73 g  5.3 h  1.46  te veel wartels voor de korte zijde
alles samen        226 x 156 x 55   283 g 20.7 h  5.67  past niet op de printer
```

Eén functie, vier kasten. De meetkas-Uno en RK-1 zijn in orde. De relaiskast met vier wartels faalt: vier gaten van 12,5 mm met elk 5 mm materiaal ernaast vragen 90 mm, en de korte zijde is maar 66 mm. De kast 'alles samen' is 226 mm lang en past niet op een bed van 220 mm. Het programma vindt die fouten *voor* je iets print. De massa volgt uit het volume maal de dichtheid: de wanden van 2 mm print de slicer massief (vijf lijnen van 0,4 mm), dus vulling telt hier niet mee. RK-1: bak 94 g, deksel 40 g.

Eén bron van waarheid

De werktekening, de stuklijst en de figuren van dit hoofdstuk zijn met dezelfde functie gemaakt. Zo kunnen tekening en berekening nooit uit elkaar lopen: verander een parameter en alles volgt.

UITLEG · 37.10

Ontwerpregels: wat de printer en de montage vragen

10/22

Een parametrisch model kan elke maat aan, ook onzin: een wand van 0,3 mm of een kast van een halve meter. Daarom hoort bij het model een lijst **ontwerpregels**: grenzen die uit de printer, het materiaal en de montage komen. Een deel ken je uit hoofdstuk 32 van Fundamental; andere volgen verder in dit hoofdstuk.

regel	waarde	waarom
wanddikte	minstens 1,2 mm (3 lijnen van 0,4 mm); een kast die je vastschroeft en vastpakt: 2,0 mm	F32: minstens twee lijnen; stijfheid groeit met de dikte tot de derde
bodem	even dik als de wand	gelijke afkoeling, geen kromtrekken
speling rond een printplaat	1,0 mm per kant	printplaten zijn niet exact; de plaat moet er schuin in kunnen
gat voor een pen of schroef	0,2 mm groter tekenen dan nodig	de printer maakt gaten 0,2 mm kleiner (F32)
deksel in bak	0,3 mm speling per kant	twee geprinte delen: stapeling van afwijkingen (verder in dit hoofdstuk)
overhang, bruggen	hoogstens 45° van de verticale; bruggen tot ±10 mm	anders support (F32)
rand op het printbed	afschuinen, niet afronden	een afronding onderaan begint met een bijna horizontale overhang
wartelgat	Ø 12,5 voor een wartel M12; minstens 5 mm wand rondom	de moer en de dichting moeten plat aanliggen
schroefbus voor een zelftappende M3	gat Ø 2,5, bus Ø 7	de schroef snijdt zelf draad; genoeg materiaal rond het gat
klikvinger	rek hoogstens 2 %	geprint PETG breekt eerst tussen de lagen
bouwvolume	220 × 220 × 250 mm (rekenwaarde voor de schoolprinter)	groter: in delen ontwerpen
materiaal	PETG, lichte kleur	vochtige, warme kweekruimte (I17)

Sommige regels zijn harde grenzen (het bouwvolume), andere vuistregels (2 mm voor een kast die je vastpakt). Schrijf bij elke regel waar ze vandaan komt: zo weet de volgende ontwerper of hij ze mag aanpassen. In `controleer()` staan de regels die je met de maten alleen kan toetsen; de printrichting en de keuze van de verbinding blijven een beslissing van de ontwerper.

> ⚠ **Printrichting en klikvingers**
>
> De bak print je met de bodem op het bed, het deksel met de buitenkant op het bed. Dan staan de klikvingers van het deksel rechtop: de lagen liggen dwars op de vinger, de zwakke richting (F32, I17). Daarom rekenen we met een voorzichtige toegelaten rek en een afronding aan de voet van de vinger.

UITLEG · 37.11

Toleranties en passingen

11/22

Geen enkele maat wordt precies zoals getekend. Een **tolerantie** zegt hoeveel ze mag afwijken. Bij twee delen die in elkaar passen, bepaalt de combinatie van hun toleranties de **passing**:

- **spelingspassing**: het gat is altijd groter dan de as; ze draaien of schuiven;
- **overgangspassing**: soms speling, soms een beetje overmaat; nauwkeurig centreren;

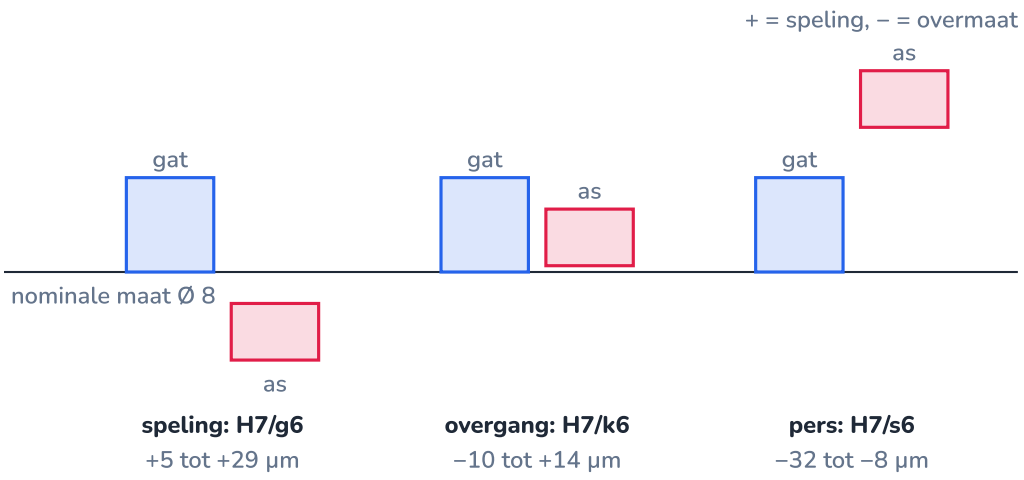
- **perspassing**: de as is altijd dikker dan het gat; ze zitten vast zonder schroef of lijm.

In de metaalbewerking beschrijft de ISO-norm passingen met een letter en een cijfer: een hoofdletter voor het gat, een kleine letter voor de as. **H7/g6** voor Ø 8 mm betekent: het gat ligt tussen 8,000 en 8,015 mm, de as tussen 7,986 en 7,995 mm, dus 5 tot 29 µm speling. Dat is honderd keer fijner dan wat een FDM-printer haalt: daar werk je met tienden van een millimeter.

Voor je eigen printer maak je daarom een eigen **passingstabel**, met een testprint (F32). Met de metingen van dat testplaatje:

passing	teken het gat	testplaatje F32 (pen 5,00 mm)	voorbeeld
pers (vast, niet meer los)	$d + 0{,}2\,\text{mm}$	getekend 5,2: gemeten 5,01, past met duwen	een metalen pen in een scharnier
glij (draait of schuift, weinig speling)	$d + 0{,}4\,\text{mm}$	getekend 5,4: gemeten 5,19, schuift	een as in een lager van kunststof
ruim (vrije doorgang)	$d + 0{,}6\,\text{mm}$ of meer	getekend 5,6: gemeten 5,40, schuift ruim	een schroef door een deksel

Die tabel geldt voor een *metalen* pen in een *geprint* gat. Passen twee geprinte delen in elkaar, dan wijken ze allebei af: dat rekent het volgende voorbeeld door.



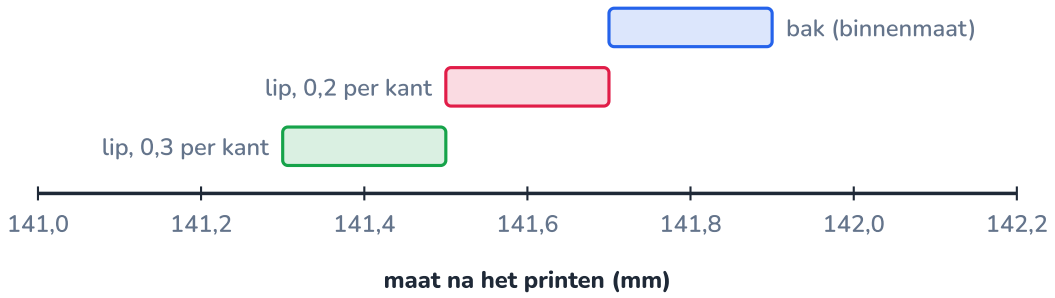
Blauw het gat, rood de as. Speling: de as ligt altijd onder het gat; pers: altijd erboven.

UITGEWERKT VOORBEELD · 37.12

Past het deksel? Een tolerantiestapeling

12/22

De lip van het deksel moet in de bak passen, zonder te klemmen en zonder veel te rammelen. De printer maakt binnenmaten (een holte, een gat) gemiddeld 0,2 mm kleiner; buitenmaten komen gemiddeld juist uit. Daarbovenop wijkt elke maat toevallig tot 0,1 mm af (rekenwaarden). Welke speling per kant moet je tekenen: 0,2 of 0,3 mm?



Met 0,2 mm per kant raken de strook van de bak en die van de lip elkaar: het deksel kan klemmen.

GEGEVEN

- binnenmaat van de bak getekend: 142,0 mm
- lip getekend: $142{,}0 - 2s$ met s de speling per kant
- holte: −0,2 mm ± 0,1 mm; buitenmaat: ± 0,1 mm

GEVRAAGD

- de kleinste en de grootste totale speling voor $s = 0{,}2$ en $s = 0{,}3\,\text{mm}$

OPLOSSING

1. **Bak.** Na het printen ligt de binnenmaat tussen $142{,}0 - 0{,}2 - 0{,}1 = 141{,}7$ en $142{,}0 - 0{,}2 + 0{,}1 = 141{,}9\,\text{mm}$.
2. **Lip met $s = 0{,}2$.** Getekend 141,6 mm, geprint tussen 141,5 en 141,7 mm.
3. **Slechtste gevallen.** Kleinste speling: kleinste bak min grootste lip $= 141{,}7 - 141{,}7 = 0{,}0\,\text{mm}$. Grootste speling: $141{,}9 - 141{,}5 = 0{,}4\,\text{mm}$. Bij een ongelukkige print klemt het deksel.
4. **Lip met $s = 0{,}3$.** Getekend 141,4 mm, geprint 141,3 tot 141,5 mm: speling $141{,}7 - 141{,}5 = 0{,}2$ tot $141{,}9 - 141{,}3 = 0{,}6\,\text{mm}$, dus 0,1 tot 0,3 mm per kant.

ANTWOORD

Teken 0,3 mm per kant: de speling blijft dan in het slechtste geval 0,2 mm (samen), en hoogstens 0,6 mm.

Klopt dit? Algemeen: de kleinste speling is de getekende speling min de som van alle afwijkingen die haar kleiner maken. Hier: $2 \cdot 0{,}3 - 0{,}2 - 0{,}1 - 0{,}1 = 0{,}2$ mm. De afwijkingen van 0,2 en 0,1 mm zijn aannames: meet ze voor je eigen printer met een testprint van drie lippen (0,2 / 0,3 / 0,4 mm) voor je de echte kast print.

UITLEG · 37.13

Verbindingen: schroef, klik en pers

13/22

Een kast bestaat uit delen, en de verbinding bepaalt hoe je ze monteert, hoe vaak je ze kan openen en of je ze later nog kan herstellen of scheiden voor recyclage (hoofdstuk 38).

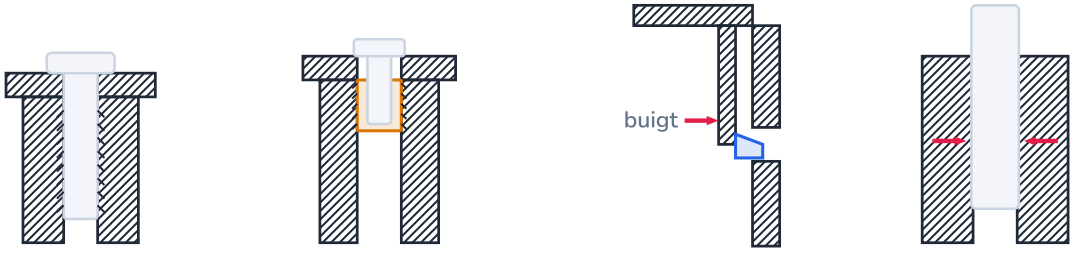
verbinding	los en vast	extra onderdeel	sterk genoeg voor	let op
zelftappende schroef in een bus	enkele keren; daarna slijt de draad	schroef	montageplaat, deksels die zelden open gaan	bus Ø ±2,5 × de schroef
inzetmoer (messing, warm ingedrukt)	honderden keren	inzetmoer + bout	een deksel dat vaak open moet	gat volgens de fabrikant, +0,2 voor de printer

verbinding	los en vast	extra onderdeel	sterk genoeg voor	let op
moer in een zeskantige holte	vaak	moer + bout	zware delen	holte net iets ruimer dan de moer
klikvinger (snap-fit)	tientallen keren, als de rek klein blijft	geen	lichte deksels, kapjes	vinger liefst in de lengte van de lagen
perspassing	eigenlijk niet	pen of lager	assen, scharnierpennen	overmaat klein houden: een print scheurt
lijmen	niet	lijm	delen die nooit meer uit elkaar moeten	slecht voor herstel en recyclage (hoofdstuk 38)

In RK-1 zit de montageplaat met vier zelftappende schroeven M3 in de afstandsbussen: die gaat zelden los. Het deksel moet vaak open (een sensor vervangen, een SD-kaart wisselen) en klikt vast met twee **klikvingers**: geen schroeven die in de kweekruimte verloren gaan, geen gereedschap nodig. Moet de kast later waterdicht worden, dan komen er vier inzetmoeren en een rubberen dichting.

Ontwerp voor montage

Hoe minder losse onderdelen, hoe sneller de montage en hoe minder er misloopt. Vraag bij elke schroef: kan dit ook met een klik, een uitsparing of een vorm die zichzelf op zijn plaats houdt? En zorg dat een onderdeel maar op één manier past (een schuine hoek, een nokje): dan kan niemand het verkeerd monteren.



Gearceerd is het geprinte materiaal in doorsnede; de schroef, de inzetmoer en de pen zijn aparte delen.

UITGEWERKT VOORBEELD · 37.14

Een klikvinger dimensioneren

14/22

Het deksel van RK-1 krijgt twee klikvingers: 1,6 mm dik (zoals de lip) en 8 mm breed, met een haak van 0,8 mm. Bij het monteren buigt de vinger 0,8 mm door. Hoe lang moet de vinger minstens zijn als de rek hoogstens 2 % mag zijn?

GEGEVEN

- dikte $t = 1{,}6\,\text{mm}$, doorbuiging $y = 0{,}8\,\text{mm}$
- toegelaten rek $\varepsilon = 2\,\% = 0{,}02$
- klikvinger met constante dikte: $\varepsilon = \dfrac{1{,}5\,t\,y}{L^2}$

GEVRAAGD

- $L_{\min}$, de gekozen lengte en haar rek

OPLOSSING

1 **De formule.** Een vinger die aan één kant vastzit en aan het uiteinde y doorbuigt, wordt aan de voet het sterkst uitgerekt. Die rek groeit met de dikte en de doorbuiging, en daalt met het kwadraat van de lengte: een twee keer zo lange vinger rekt vier keer minder.

2 **Omvormen naar L.** $L = \sqrt{\dfrac{1{,}5\,t\,y}{\varepsilon}} = \sqrt{\dfrac{1{,}5 \cdot 1{,}6 \cdot 0{,}8}{0{,}02}} = \sqrt{96} = 9{,}8\,\text{mm}.$

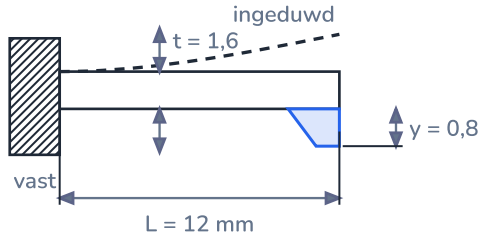
3 **Kiezen met marge.** We nemen $L = 12\,\text{mm}$: $\varepsilon = \dfrac{1{,}5 \cdot 1{,}6 \cdot 0{,}8}{12^2} = 1{,}33\,\%.$

4 **De prijs van een lange vinger.** De kracht om de vinger 0,8 mm te buigen (en dus de houdkracht) daalt met L^3: van 10 naar 12 mm is $(10/12)^3 = 0{,}58$, 42 % minder. Een klik die te makkelijk gaat, houdt ook slecht.

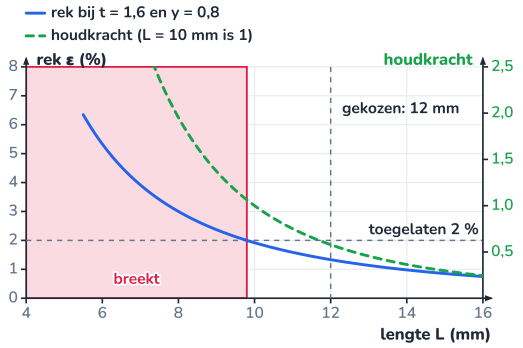
ANTWOORD

Minstens 9,8 mm; gekozen 12 mm, met een rek van 1,3 %, ruim onder de 2 %.

Klopt dit? Eenheden: t·y/L² is mm·mm/mm², zonder eenheid, zoals een rek hoort te zijn. En het klopt met de proef hierna: vingers korter dan 10 mm worden wit of breken.



De stippellijn toont de vinger tijdens het monteren; de haak is vergroot getekend.



Korter dan 9,8 mm breekt de vinger; langer houdt hij slechter vast.

PROEF · 37.15

Hoe lang moet een klikvinger zijn?

15/22

ONDERZOEKSVRAAG

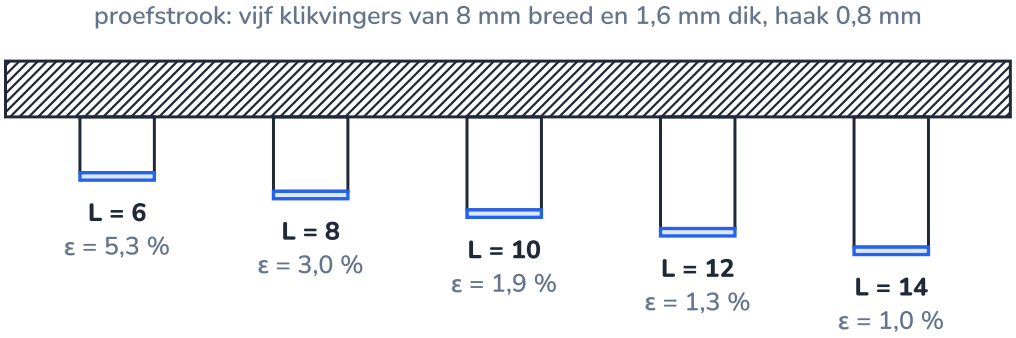
Vanaf welke lengte overleeft een geprinte klikvinger van 1,6 mm dikte twintig klikken, en hoe verandert de houdkracht?

MATERIAAL

- 3D-printer met PETG
- proefstrook met vijf klikvingers (6, 8, 10, 12 en 14 mm lang, 8 mm breed, 1,6 mm dik, haak 0,8 mm) en een raampje met vijf vensters
- keukenweegschaal (of een veerunster tot 50 N)
- loep

Veiligheid

Een vinger kan afbreken en wegschieten: draag een veiligheidsbril.



De proefstrook: vijf vingers, één parameter die verandert.

WERKWIJZE

1. Print de proefstrook (proefstrook.scad) met de voet plat op het bed: de vingers staan dan rechtop, zoals de klikvingers van het deksel. Print het raampje apart.
2. Schuif het raampje langs de vingers tot elke haak in zijn venster klikt, en duw de haken weer los; doe dat twintig keer. Kijk na elke vijf klikken met de loep naar de voet van de vinger: wordt hij wit (verstrekt)? Breekt hij?
3. Duw na twintig klikken met de weegschaal zijdelings tegen de haak tot hij uit zijn venster loskomt (0,8 mm opzij), en noteer de grootste kracht (1 kg op de weegschaal is ongeveer 9,8 N).
4. Bereken voor elke vinger de rek met de formule van het vorige blad en vergelijk.

WAARNEMING

Een voorbeeld van een meetreeks (PETG, lagen van 0,2 mm):

L (mm)	rek ε	na 20 klikken	kracht om los te duwen
6	5,3 %	breekt bij de eerste klik	–
8	3,0 %	wit na 3 klikken, breekt na 9	–
10	1,9 %	heel licht wit aan de voet	±14 N
12	1,3 %	geen sporen	±8 N
14	1,0 %	geen sporen	±5 N

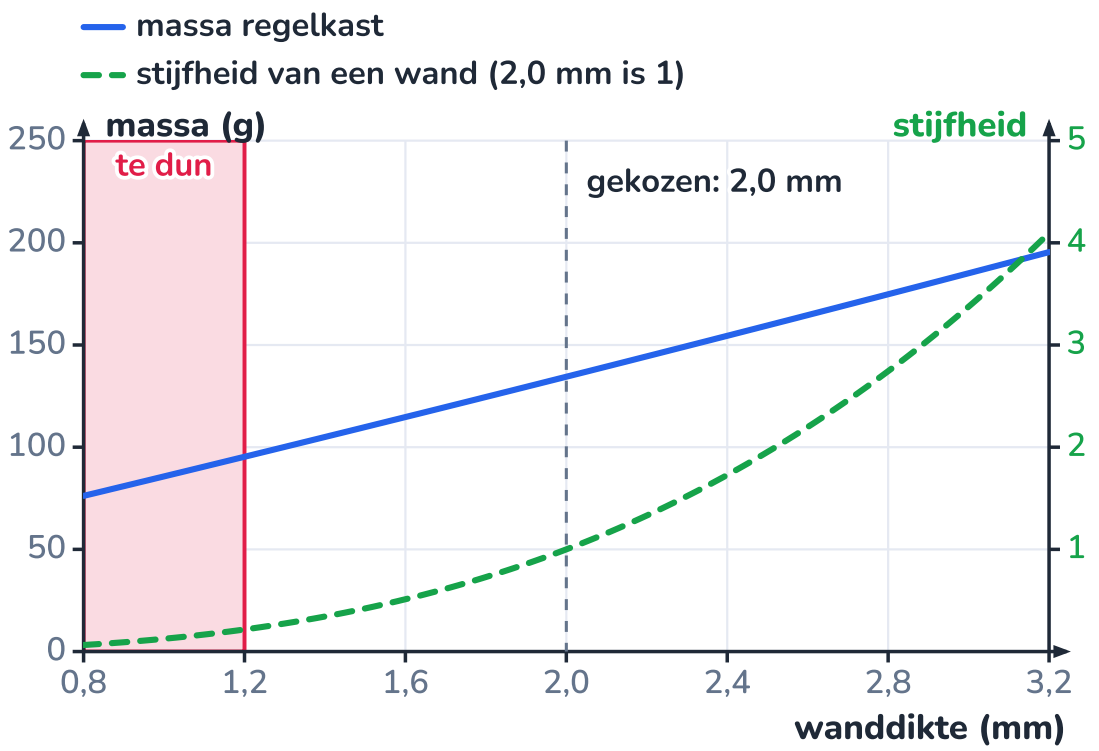
VERKLARING

De rek aan de voet groeit met $1/L^2$: bij 6 en 8 mm ligt ze ver boven wat de laaghechting verdraagt, en de vinger scheurt tussen twee lagen. Rond 2 % begint het materiaal te verstrekken (het wordt wit), daaronder blijft het elastisch. De kracht om de haak 0,8 mm opzij te duwen daalt met $1/L^3$: van 10 naar 14 mm valt ze terug tot ongeveer $(10/14)^3 \approx 0{,}36$ van de kracht bij 10 mm, wat de meting bevestigt. Omdat de bovenkant van de haak vlak is, gaat het deksel niet los door eraan te trekken: je opent het door de haak door het venster naar binnen te duwen. Voor RK-1 is 12 mm een goede keuze: veilig, en de haak blijft stevig genoeg in zijn venster zitten.

OOK TE DOWNLOADEN

Hoe dik moet de wand? De ontwerpruimte verkennen

De regelkast moet stevig aanvoelen als je ze vastpakt en tegen de wand schroeft, maar zo licht en snel te printen als redelijk is. Vergelijk wanddiktes van 1,0 tot 3,2 mm met behuizing.py. De buigstijfheid van een wand is evenredig met haar dikte tot de derde (waarom: Expert, hoofdstuk 12).



De massa (volle lijn) stijgt recht, de stijfheid (stippellijn, rechts) met de derde macht.

GEGEVEN

- behuizing.py
- stijfheid $\sim t^3$; referentie $t = 2{,}0\,\text{mm}$
- regel: wand minstens 1,2 mm

GEVRAAGD

- massa, printtijd en relatieve stijfheid per wanddikte, en een keuze

OPLOSSING

1 **Programma.** Een lus over de wanddiktes, met dezelfde functie:

```python
# verken.py - de wanddikte van de regelkast: massa, printtijd en stijfheid naast elkaar
from behuizing import behuizing, controleer

print("wand   massa  printtijd  stijfheid  regels")
for wand in [1.0, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2]:
    m = behuizing(140, 90, 28, wand=wand)
    stijf = (wand / 2.0) ** 3                # buigstijfheid van een wand: evenredig met de dikte tot de derde
    fout = controleer(m, wand=wand)
    print(f"{wand:4.1f} {m['massa']:6.0f} g {m['printtijd']:7.1f} h {stijf:9.2f}   {'; '.join(fout) or 'in orde'}")
```

```
wand   massa  printtijd  stijfheid  regels
 1.0     86 g     6.2 h      0.12   wand dunner dan drie lijnen van 0,4 mm
 1.2     95 g     6.9 h      0.22   in orde
 1.6    115 g     8.4 h      0.51   in orde
 2.0    134 g     9.8 h      1.00   in orde
 2.4    155 g    11.3 h      1.73   in orde
 2.8    175 g    12.7 h      2.74   in orde
 3.2    195 g    14.3 h      4.10   in orde
```

2 **Massa en tijd** groeien ongeveer evenredig met de dikte: elke 0,4 mm extra kost ±20 g en ±1,5 h.

3 **Stijfheid** groeit met de derde macht: 1,6 mm is maar half zo stijf als 2,0 mm ($0{,}8^3 = 0{,}51$); 2,4 mm is 1,73 keer zo stijf. Onder 1,2 mm faalt de regel.

4 **Keuze.** 2,0 mm: stevig genoeg om vast te pakken, 134 g en 9,8 h. Moet het stijver, dan helpt een ribbe langs de lange wand meer dan een dikkere wand overal.

ANTWOORD

Wand 2,0 mm: tweemaal zo stijf als 1,6 mm voor 20 g en 1,4 h extra.

Klopt dit? Dat de massa niet precies evenredig is met de dikte, komt doordat de buitenmaten meegroeien (de binnenmaten liggen vast door de printplaat).

UITLEG · 37.17

Varianten en de ontwerpruimte

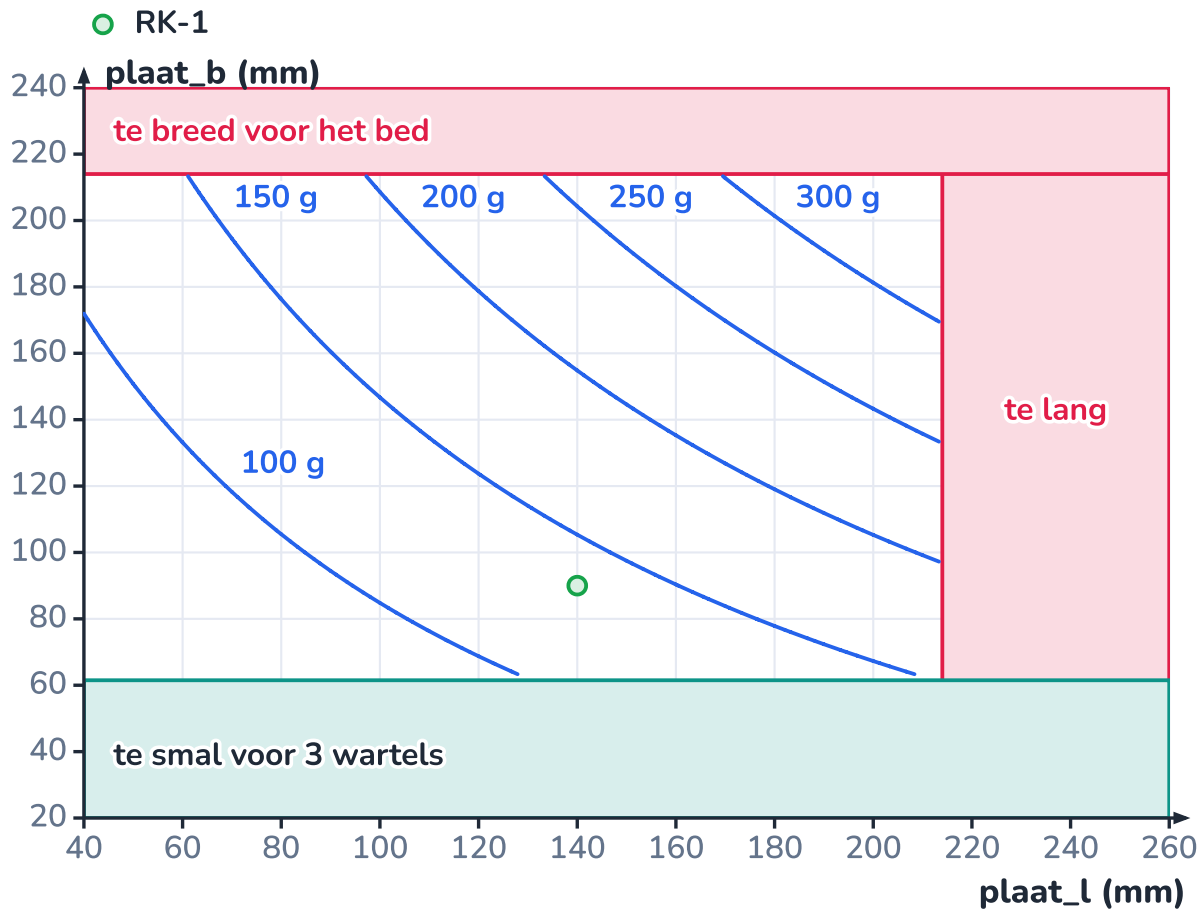
17/22

De **ontwerpruimte** is de verzameling van alle parametercombinaties die je kan kiezen. De regels snijden er stukken af. In de figuur staan de twee belangrijkste invoermaten van de regelkast, lengte en breedte van de montageplaat, tegen elkaar uit. De lijnen verbinden kasten met dezelfde massa, zoals hoogtelijnen op een kaart. De gekleurde stroken zijn verboden: te lang of te breed voor het bed, of te smal voor drie wartels naast elkaar.

Een **variant** is één punt in die ruimte. Met één model maak je een hele familie:

variant	plaat (mm)	buiten (mm)	massa	printtijd	regels
meetkas (Uno)	68,6 × 53,4	75 × 59 × 40	49 g	3,5 h	in orde
regelkast RK-1	140 × 90	146 × 96 × 53	134 g	9,8 h	in orde
relaiskast	100 × 60	106 × 66 × 45	73 g	5,3 h	te veel wartels voor de korte zijde
alles samen	220 × 150	226 × 156 × 55	283 g	20,7 h	past niet op de printer

Zo werken ook fabrikanten van behuizingen, profielen of kasten: één parametrisch model, een tabel met varianten (in FreeCAD een spreadsheet met een kolom per variant, in OpenSCAD een lijst met voorinstellingen), en een tekening per variant die automatisch volgt. Vast leggen welke variant waar in het project zit, heet **configuratiebeheer** (Expert, hoofdstuk 45).



Hoogtelijnen van de massa; RK-1 (groen) ligt ruim binnen de toegelaten ruimte.

CODE · 37.18

Hetzelfde model in OpenSCAD

18/22

In OpenSCAD schrijf je het onderdeel als een programma: bovenaan de parameters, dan de afgeleide maten, dan de vorm als bewerkingen op lichamen. `difference()` haalt weg (uitsparen, uithollen), `hull()` spant een vorm rond andere vormen (hier vier cilinders: een afgeronde rechthoek), een `for`-lus maakt een patroon. Met F6 berekent OpenSCAD het model, met F7 bewaar je een STL voor de slicer. Het venster *Customizer* toont de parameters als schuifjes.

OpenSCAD **rk1_bak.scad** 39 regels

```
// rk1_bak.scad - regelkast RK-1 van de klimaatmodule: de bak, parametrisch
plaat_l  = 140;    // montageplaat (mm)
plaat_b  = 90;
comp_h   = 28;     // hoogste onderdeel boven de montageplaat
wand     = 2.0;
speling  = 1.0;    // rond de montageplaat, per kant
bus_h    = 6;      // afstandsbussen onder de montageplaat
plaat_d  = 3;
boven    = 12;     // vrije ruimte voor kabels
hoek_r   = 4;
wartels  = 3;
wartel_d = 12.5;   // gat voor een wartel M12

```

```
binnen_l = plaat_l + 2 * speling;
binnen_b = plaat_b + 2 * speling;
binnen_h = bus_h + plaat_d + comp_h + boven;
buiten_l = binnen_l + 2 * wand;
buiten_b = binnen_b + 2 * wand;
bak_h    = binnen_h + wand;
e        = wand + speling + 5;     // middelpunt van een bus vanaf de buitenrand

module afgerond(l, b, h, r) {      // blok l x b x h met afgeronde verticale ribben
    hull() for (x = [r, l - r], y = [r, b - r])
        translate([x, y, 0]) cylinder(r = r, h = h, $fn = 48);
}

difference() {
    afgerond(buiten_l, buiten_b, bak_h, hoek_r);
    translate([wand, wand, wand])                          // uithollen: open aan de bovenkant
        afgerond(binnen_l, binnen_b, bak_h, max(hoek_r - wand, 0.01));
    for (i = [1 : wartels])                                // wartelgaten in de korte zijde x = 0
        translate([-1, i * buiten_b / (wartels + 1), wand + binnen_h / 2])
            rotate([0, 90, 0]) cylinder(d = wartel_d, h = wand + 2, $fn = 64);
}
for (x = [e, buiten_l - e], y = [e, buiten_b - e])         // vier afstandsbussen op de bodem
    translate([x, y, wand]) difference() {
        cylinder(d = 7, h = bus_h, $fn = 32);
        cylinder(d = 2.5, h = bus_h + 1, $fn = 24);
    }
```

Vergelijk met behuizing.py: dezelfde parameters, dezelfde formules voor de afgeleide maten, dezelfde wartelposities $i \cdot \text{buiten}_b/(n+1)$. Het verschil: OpenSCAD maakt de vorm, Python rekent en controleert. Veel ontwerpers doen beide: een rekenprogramma dat de regels toetst en de parameters kiest, en een CAD-model dat ze gebruikt. Let op de kleine truc bij het uithollen: het binnenblok is even hoog als de bak maar begint op hoogte `wand`, dus steekt het er bovenaan uit en is de bak open.

FreeCAD of OpenSCAD?

OpenSCAD is ideaal voor onderdelen die je met formules beschrijft (kasten, klemmen, houders) en om varianten te maken. FreeCAD werkt met schetsen en muis, en is sterker voor vrije vormen en samenstellingen. Het idee is hetzelfde: parameters, afhankelijkheden, een boom van bewerkingen.

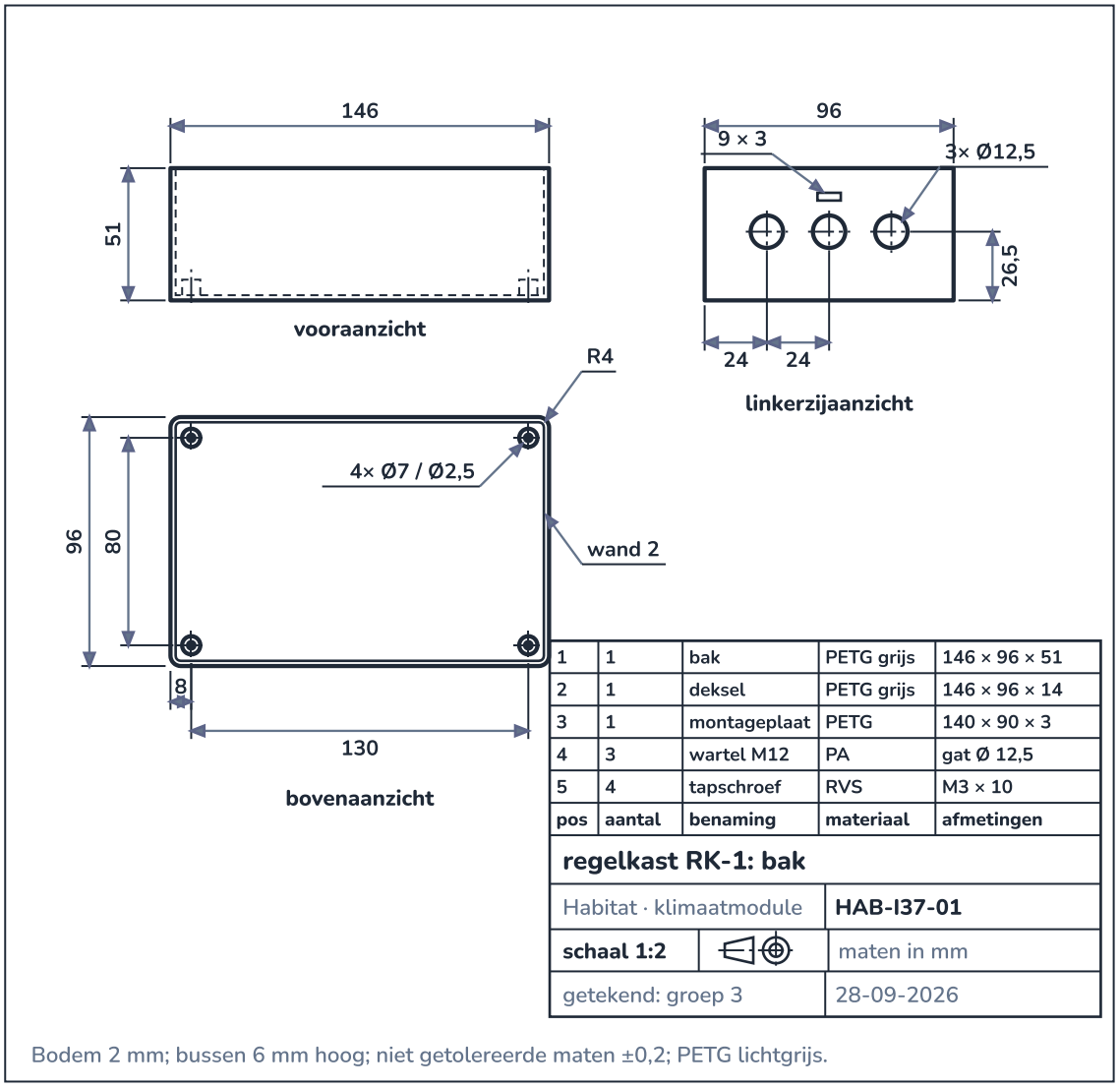
UITLEG · 37.19

De werktekening uit dezelfde parameters

19/22

Een werktekening (F31) zegt de maker wat hij moet maken. Bij een parametrisch ontwerp volgt ze uit het model: de maten op de tekening zijn de uitkomsten van dezelfde formules. Toch kies je bewust welke maten je op de tekening zet: **functionele maten**, die er voor het gebruik toe doen (de afstand tussen de bussen, die de montageplaat moet volgen), in plaats van maten die toevallig uit de constructie volgen. Elke maat staat één keer op de tekening; wat volgt uit andere maten, laat je weg.

In de tekening van RK-1 staan de buitenmaten (146 × 96 × 51), de plaats van de bussen gemeten van de rand (8) en tussen de middelpunten (130 en 80), de wartelgaten met hun steek (24) en hoogte (26,5), het venster voor de klikvinger en de afronding. Een algemene tolerantie (±0,2 mm) staat onderaan; alleen de passende maten krijgen een eigen tolerantie. Het nummer HAB-I37-01 volgt de afspraak van F31.



pos	aantal	benaming	materiaal	afmetingen
1	1	bak	PETG grijs	146 × 96 × 51
2	1	deksel	PETG grijs	146 × 96 × 14
3	1	montageplaat	PETG	140 × 90 × 3
4	3	wartel M12	PA	gat Ø 12,5
5	4	tapschroef	RVS	M3 × 10

regelkast RK-1: bak

Habitat · klimaatmodule	**HAB-I37-01**
schaal 1:2	maten in mm
getekend: groep 3	28-09-2026

Bodem 2 mm; bussen 6 mm hoog; niet getolereerde maten ±0,2; PETG lichtgrijs.

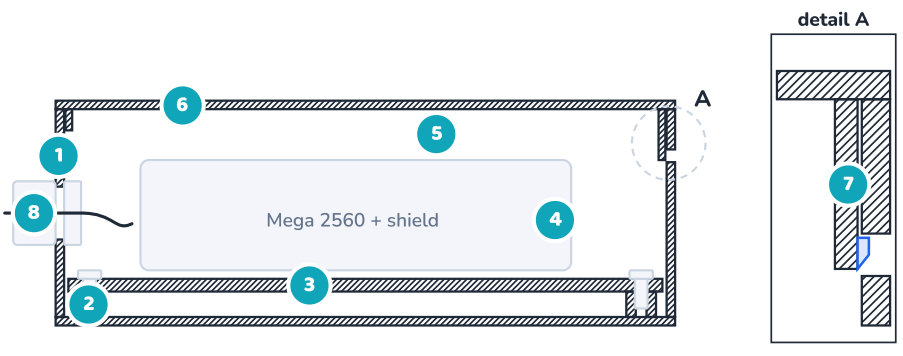
Werktekening van de bak (Europese projectie). Alle maten komen uit behuizing(140, 90, 28).

ONTLEED · 37.20

De regelkast RK-1 in doorsnede

20/22

De gesloten kast, schematisch in doorsnede: bussen, wartel en klikvinger liggen in werkelijkheid niet in één vlak, maar staan hier samen om te tonen hoe alles in elkaar past.



Gearceerd: doorgesneden PETG. Detail A toont de haak in het venster.

1 Wand en bodem, 2,0 mm
Vijf lijnen van 0,4 mm, massief geprint. Stijf genoeg om de kast vast te pakken; de bodem is even dik, zodat alles gelijk afkoelt.

2 Afstandsbus en schroef
Bus Ø 7 mm, 6 mm hoog, met een gat Ø 2,5 mm: de zelftappende schroef M3 snijdt er zelf draad in. De bussen staan 8 mm van de rand, 5 mm van de hoek van de montageplaat.

3 Montageplaat
140 × 90 × 3 mm, met de Mega, de klok en de SD-kaart erop. Rondom 1,0 mm speling: de plaat moet er schuin in kunnen.

4 Mega met shield
Het hoogste onderdeel staat 28 mm boven de plaat: de parameter comp_h.

5 Ruimte voor kabels
12 mm vrij boven de onderdelen (boven): voor de draden naar de klemmen en een lusje in elke kabel.

6 Deksel met lip
2 mm dik, met een lip van 5 mm die 0,3 mm per kant ruimte heeft in de bak: de uitkomst van de tolerantiestapeling.

7 Klikvinger
12 mm lang, 1,6 mm dik, met een haak van 0,8 mm die in een venster van 9 × 3 mm valt. Rek 1,3 %: veilig, ook voor een print.

8 Kabelwartel M12
In een gat van Ø 12,5 mm; de wartel klemt de kabel en dicht het gat af. Drie wartels in de korte zijde, die naar onder hangt: druppels lopen van de kast weg.

NASLAG · 37.21

Formules, regels en passingen

21/22

afgeleide maat (voorbeeld)

$$\text{buiten}_l = \text{plaat}_l + 2\,\text{speling} + 2\,\text{wand}$$

symbool	betekenis	eenheid
plaat_l	invoermaat: lengte van de montageplaat	mm
$\text{speling}, \text{wand}$	invoermaten	mm

afgeronde rechthoek

$$A = l\,b - (4 - \pi)\,r^2$$

symbool	betekenis	eenheid
l, b	lengte en breedte	mm
r	hoekstraal	mm

regel van Pappus (draaideel)

$$V = 2\pi\,\bar{x}\,A$$

symbool	betekenis	eenheid
$\bar{x}$	afstand zwaartepunt van het profiel tot de as	mm
A	oppervlakte van het profiel	mm²

rek aan de voet van een klikvinger

$$\varepsilon = \frac{1{,}5\,t\,y}{L^2}$$

symbool	betekenis	eenheid
t	dikte	mm
y	doorbuiging (haak)	mm
L	lengte	mm

tolerantiestapeling (slechtste geval)

$$s_{\min} = s_{\text{getekend}} - \sum \text{afwijkingen}$$

symbool	betekenis	eenheid
s	totale speling	mm

vrijheidsgraden van een schets

$$F = \sum \text{vrijheidsgraden} - \sum \text{beperkingen}$$

symbool	betekenis	eenheid
F	0 = volledig bepaald	–

passing	teken het gat	testplaatje F32 (pen 5,00 mm)	voorbeeld
pers (vast, niet meer los)	$d + 0{,}2\,\text{mm}$	getekend 5,2: gemeten 5,01, past met duwen	een metalen pen in een scharnier
glij (draait of schuift, weinig speling)	$d + 0{,}4\,\text{mm}$	getekend 5,4: gemeten 5,19, schuift	een as in een lager van kunststof
ruim (vrije doorgang)	$d + 0{,}6\,\text{mm}$ of meer	getekend 5,6: gemeten 5,40, schuift ruim	een schroef door een deksel

regel	waarde	waarom
wanddikte	minstens 1,2 mm (3 lijnen van 0,4 mm); een kast die je vastschroeft en vastpakt: 2,0 mm	F32: minstens twee lijnen; stijfheid groeit met de dikte tot de derde

regel	waarde	waarom
bodem	even dik als de wand	gelijke afkoeling, geen kromtrekken
speling rond een printplaat	1,0 mm per kant	printplaten zijn niet exact; de plaat moet er schuin in kunnen
gat voor een pen of schroef	0,2 mm groter tekenen dan nodig	de printer maakt gaten 0,2 mm kleiner (F32)
deksel in bak	0,3 mm speling per kant	twee geprinte delen: stapeling van afwijkingen (verder in dit hoofdstuk)
overhang, bruggen	hoogstens 45° van de verticale; bruggen tot ±10 mm	anders support (F32)
rand op het printbed	afschuinen, niet afronden	een afronding onderaan begint met een bijna horizontale overhang
wartelgat	Ø 12,5 voor een wartel M12; minstens 5 mm wand rondom	de moer en de dichting moeten plat aanliggen
schroefbus voor een zelftappende M3	gat Ø 2,5, bus Ø 7	de schroef snijdt zelf draad; genoeg materiaal rond het gat
klikvinger	rek hoogstens 2 %	geprint PETG breekt eerst tussen de lagen
bouwvolume	220 × 220 × 250 mm (rekenwaarde voor de schoolprinter)	groter: in delen ontwerpen
materiaal	PETG, lichte kleur	vochtige, warme kweekruimte (I17)

HET GEHEEL · 37.22

Waar parametrisch ontwerpen vandaan komt en naartoe gaat

22/22

Parametrisch ontwerpen is waar wiskunde en maken elkaar raken: een ontwerp wordt een functie, een eis wordt een ongelijkheid, een tekening wordt een uitkomst. Het steunt op technisch tekenen en CAD uit Fundamental, op functies en zwaartepunten uit Intermediate, en op de materiaalkennis van hoofdstuk 17. In Expert wordt het systematisch: optimaliseren (de beste parameters zoeken in plaats van proberen), sterkteleer (hoe stijf is een wand echt) en configuratiebeheer in grote projecten.

F30 De ontwerpcyclus
Het programma van eisen: hier worden eisen regels die het model zelf toetst.

F31 Technisch tekenen
Aanzichten, maatvoering en stuklijst van de werktekening.

F32 Van idee tot object: CAD en 3D-printen
De 45°-regel, speling, de maatafwijking van de printer en de printtijd.

I4 Functies als modellen
Een functie met invoer en uitvoer: hier is het ontwerp zelf de functie.

I10 Moment en evenwicht
Het zwaartepunt van samengestelde vormen, nodig voor de regel van Pappus.

I17 Polymeren en composieten
Waarom PETG in de kweekruimte, en waarom een print tussen de lagen breekt.

I28 Functies, lijsten en data in Python
Functies met standaardwaarden en woordenboeken: de bouwstenen van behuizing.py.

I38 Duurzaam ontwerpen
Klikken of lijmen, hoeveel materiaal: de keuzes van dit hoofdstuk op hun milieu-impact gewogen.

I40 Integratieproject: de klimaatmodule
De regelkast en de andere kasten van de module worden gebouwd.

E2 Optimaliseren
De beste parameters berekenen in plaats van een tabel doorlopen.

E12 Sterkteleer
Waarom de stijfheid met de dikte tot de derde groeit, en hoe ver een wand doorbuigt.

E45 Systems engineering en het V-model
Eisen, varianten en configuratiebeheer in een groot project.

Habitat

De klimaatmodule heeft nu een parametrische **regelkast RK-1** voor het stuurdeel: montageplaat 140 × 90 × 3 mm, bak 146 × 96 × 51 mm met wand en bodem van 2 mm, vier afstandsbussen, drie wartels M12 in de korte zijde die naar onder hangt, een deksel met lip (0,3 mm speling per kant) en twee klikvingers van 12 mm. In PETG lichtgrijs: 134 g, 9,8 h printen, € 2,69. De werktekening is HAB-I37-01. Opdracht 37.15 legt de sensorkast vast met hetzelfde model; in hoofdstuk 38 weeg je de materiaalkeuze af, en in het integratieproject monteer je de kasten.

SAMENVATTING

In het kort

- Een parametrisch ontwerp is een functie: een paar invoermaten erin, alle afgeleide maten, de massa en de controles eruit.
- Een schets krijgt geometrische beperkingen (de vorm, de ontwerpintentie) en maatbeperkingen (liefst gekoppeld aan parameters) tot ze volledig bepaald is: 0 vrijheidsgraden.
- Extruderen en draaien maken een lichaam; uitsparen, afronden, afschuinen, uithollen en patronen werken het af. De volgorde in de bewerkingsboom doet ertoe.
- Een draaideel heeft volume $V = 2\pi\,\bar{x}\,A$ (regel van Pappus).
- Passingen: speling, overgang, pers. Maak een passingstabel voor je eigen printer, en reken bij twee geprinte delen de stapeling van de afwijkingen door (deksel: 0,3 mm per kant).
- Een klikvinger rekt $\varepsilon = 1{,}5\,t\,y/L^2$ aan de voet; langer is veiliger maar houdt slechter vast.
- Ontwerpregels (wand, speling, bed, wartels) horen bij het model; een programma kan ze toetsen voor je iets print.
- Werktekening, stuklijst en berekening komen uit dezelfde parameters: één bron van waarheid.

BEGRIPPEN

Wat je nu kent

parametrisch ontwerp
Ontwerp waarin de maten afhangen van een paar invoermaten (parameters) via formules en beperkingen, zodat een wijziging overal doorwerkt.

ontwerpintentie
Wat de ontwerper bedoelt met een maat of een plaats (het gat staat in het midden, de bus volgt de printplaat), vastgelegd zodat het bij elke wijziging blijft kloppen.

beperking (constraint)
Voorwaarde in een schets: geometrisch (horizontaal, evenwijdig, raaklijn, gelijk) of een maat.

vrijheidsgraad
Een manier waarop een schets nog kan bewegen of vervormen; een volledig bepaalde schets heeft er nul.

extruderen
Een tweedimensionale schets over een afstand optrekken tot een lichaam.

draaien (revolve)
Een profiel rond een as draaien tot een rond lichaam.

bewerkingsboom
De reeks bewerkingen waaruit een CAD-model is opgebouwd; een latere bewerking bouwt op een eerdere.

passing
Hoe twee passende delen samengaan: met speling, als overgang of met overmaat (pers).

tolerantiestapeling

Het optellen van de maatafwijkingen van meerdere delen om de kleinste en de grootste speling te vinden.

klikvinger (snap-fit)

Een buigzaam lipje met een haak dat bij het monteren doorbuigt en daarna vastklikt.

regel van Pappus

Het volume van een draaideel is de oppervlakte van het profiel maal de omtrek die het zwaartepunt van het profiel beschrijft.

ontwerpruimte

Alle combinaties van parameters die je kan kiezen, met de grenzen die de regels opleggen.

HANDBOEK · HOOFDSTUK 38

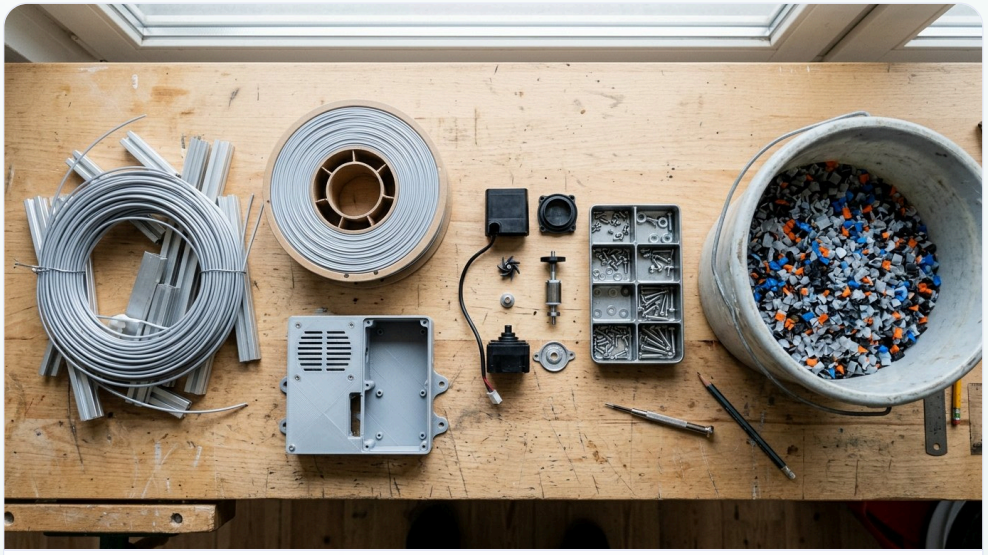
Duurzaam ontwerpen

Van grondstof tot afval en terug: rekenen aan de voetafdruk van wat je maakt

Een ontwerp is pas af als je weet wat het kost aan grondstoffen, energie en uitstoot, van de mijn tot de vuilnisbak. In dit hoofdstuk volg je de levenscyclus van een product, reken je met CO_2-equivalenten, maak je een eenvoudige levenscyclusanalyse in Python, en leer je ontwerpen voor herstel, demontage en een tweede leven. Je weegt materialen af met een keuzematrix, en je lichten de klimaatmodule door: waar zit haar voetafdruk echt, en welke keuze helpt het meest?

7× uitleg · 5× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 1× naslag · 1× verhaal · 1× het geheel · 1× code · ± 9 lestijden

D Ontwerpen en maken · E Energie · M Materie en chemie



Grondstof, onderdeel, herstel en een nieuw begin: elke fase van een product telt mee.

Na dit hoofdstuk

- Je kan de fasen van de levenscyclus van een product benoemen en de systeemgrens van een vergelijking aangeven.
- Je kan uitstoot van verschillende broeikasgassen omrekenen naar CO_2-equivalent en de ecologische voetafdruk uitleggen.
- Je kan een eenvoudige LCA opstellen met een functionele eenheid, rekenwaarden en fasen, en de uitkomst interpreteren.
- Je kan cradle to cradle en de R-ladder gebruiken om een product circulair te maken.
- Je kan een product beoordelen en herontwerpen op demontage en herstel.
- Je kan een materiaalkeuzematrix met gewichten opstellen en nagaan hoe gevoelig de keuze is voor de gewichten.
- Je kan een eerlijke vergelijking van een oneerlijke onderscheiden en een milieuclaim kritisch lezen.

VERHAAL · 38.1

Het draagtasje dat won

1/18

Welk boodschappentasje is het best voor het klimaat: het dunne plastic tasje, een papieren zak of een katoenen tas? De meeste mensen kiezen katoen. In 2011 publiceerde het Britse Environment Agency

een levenscyclusanalyse van de tasjes die in 2006 in Britse supermarkten werden gebruikt. Het rekende alles mee: de grondstoffen, het maken, het vervoer en wat er met het tasje gebeurt als het op is.

De uitkomst verraste. Om per keer boodschappen doen minder bij te dragen aan de opwarming dan een dun plastic tasje dat je maar één keer gebruikt, moet je een papieren zak minstens **3 keer** gebruiken, een stevige plastic tas **4 keer**, een tas uit niet-geweven polypropeen **11 keer** en een katoenen tas **131 keer**. Katoen vraagt veel land, water, meststof en energie; een dun plastic tasje weegt maar een paar gram.

Betekent dat: koop plastic tasjes? Nee. Het rapport keek naar de opwarming van de aarde, niet naar zwerfvuil in zee, en de katoenen tas wint wel als je ze jarenlang gebruikt. De les is een andere: je intuïtie is geen meetinstrument. Wie wil weten wat het duurzaamst is, moet rekenen, over de hele levensloop, per keer dat het product zijn werk doet.

Waarom dit hoofdstuk ertoe doet

Habitat moet jarenlang werken met weinig middelen: elk kilo dat bevoorraad moet worden, elke kilowattuur en elk onderdeel dat stuk gaat, telt. In dit hoofdstuk reken je na wat de regelkast, de batterijen en het kweekrek van de klimaatmodule kosten aan CO_2, en je beslist welke keuzes echt verschil maken.

Welk tasje belast het klimaat het minst? Het hangt af van hoe vaak je het gebruikt.

UITLEG · 38.2

Wat je al weet, en de levenscyclus van een product

2/18

Duurzaam ontwerpen bouwt op veel wat je al kan:

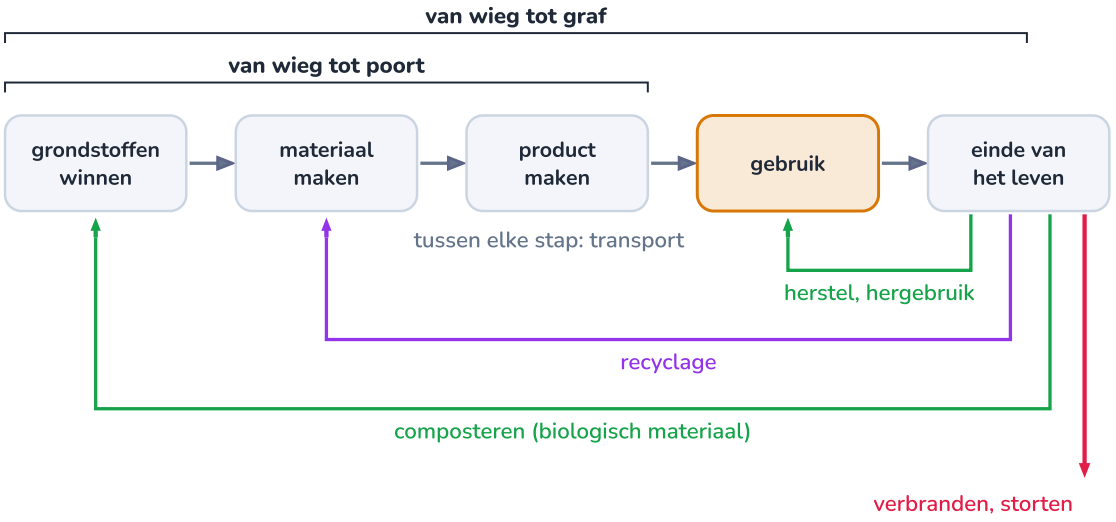
wat je al kan	waar	wat er nu bij komt
materiaalfamilies, grondstoffen, de materialenkringloop	F10	de hele levensloop van een product in cijfers
CO_2 en klimaat, CO_2 per kWh van een energiebron	F17	emissiefactoren voor materialen en processen
kringlopen in een ecosysteem	F28	biologische en technische kringlopen van materialen
CO_2 uit een verbranding berekenen	I14	het einde van een kunststof onderdeel
batterijtypes, cycli, rendement	I16	de batterijkeuze van de module op CO_2 gewogen
kunststoffen, bioplastics, recyclagecodes	I17	welk filament over 15 jaar het minst belast
energiebudget en zonnepanelen van de module	I20	waar de CO_2 van 'gratis' zonnestroom zit
de regelkast als parametrisch model	I37	klikken of lijmen, hoeveel materiaal

Elk product doorloopt een **levenscyclus**. Eerst worden **grondstoffen** gewonnen (erts, aardolie, hout, maïs), dan wordt er **materiaal** van gemaakt (aluminium, filament), dan het **product**. Tussen elke stap is er **transport**. Daarna volgt het **gebruik**, vaak de langste fase: een lamp verbruikt stroom, een pomp slijt. Aan het **einde van het leven** wordt het product hersteld, hergebruikt, gerecycleerd, verbrand of gestort.

Een eerlijke vergelijking legt eerst vast welke fasen ze meetelt: de **systeemgrens**. *Van wieg tot poort* (cradle to gate) stopt bij de fabriekspoort: zo geven fabrikanten hun waarden op. *Van wieg tot graf* (cradle to grave) neemt ook het gebruik en het einde mee. *Van wieg tot wieg* (cradle to cradle) vraagt dat het einde van het ene product het begin van een ander wordt.

"Het maken is het probleem, het gebruik niet"

Voor een lamp, een pomp of een auto is het gebruik vaak de grootste fase: een groeilamp van 95 W die 16 uur per dag brandt, gebruikt in één jaar 555 kWh. Voor een geprint kapje of een tafel is het omgekeerd: daar telt het maken. Waar de voetafdruk zit, weet je pas na het rekenen.



De vijf fasen van links naar rechts; de lussen onderaan houden materiaal in gebruik.

CO_2-equivalent en ecologische voetafdruk

CO_2 is niet het enige broeikasgas. Methaan en lachgas houden per kilogram veel meer warmte vast, en sommige koelmiddelen nog duizend keer meer. Om ze op te tellen, reken je elk gas om naar de hoeveelheid CO_2 die over honderd jaar evenveel opwarming geeft: het **CO_2-equivalent**. De omrekenfactor heet het **GWP** (*global warming potential*).

CO_2-EQUIVALENT

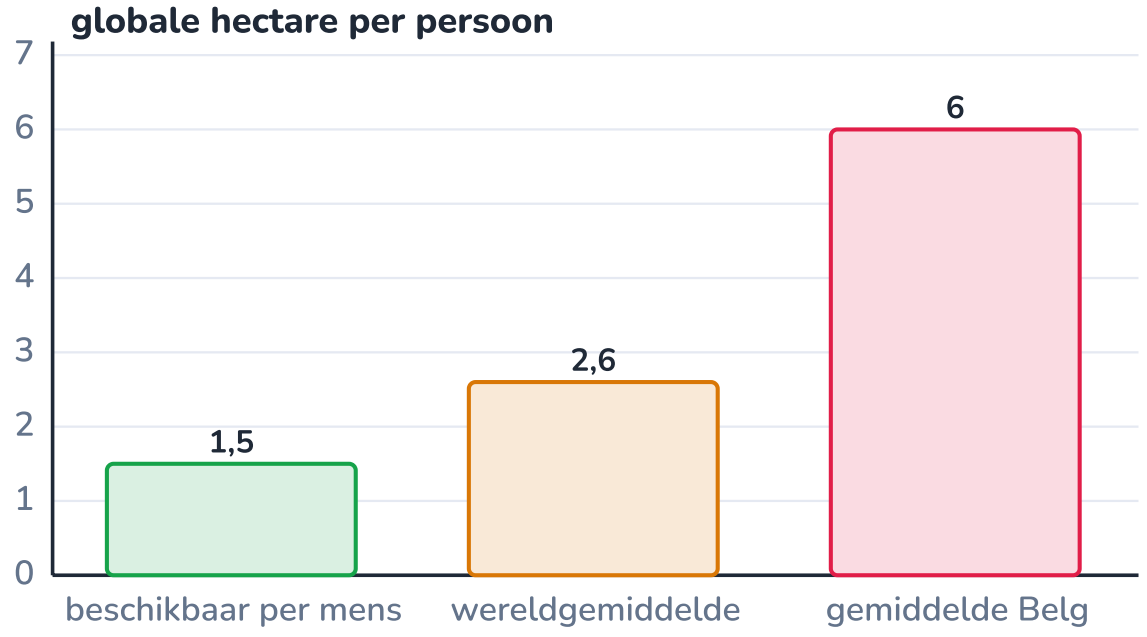
$$m_{\mathrm{CO_2\text{-}eq}} = \sum_i m_i \cdot \mathrm{GWP}_i$$

m_i = massa van gas i (kg); $\mathrm{GWP}_{\mathrm{CO_2}} = 1$.

gas	waar het vandaan komt	GWP (100 jaar)
koolstofdioxide CO_2	verbranding, cement, ontbossing	1
methaan CH_4 (niet-fossiel)	vee, rijst, stortplaatsen, compost zonder lucht	27
methaan CH_4 (fossiel)	lekken bij aardgas en steenkool	29,8
lachgas N_2O	kunstmest, mest, verbranding	273
HFK-134a (koelmiddel)	oudere koelkasten en autoairco's	1530

Waarden van het IPCC (2021). Methaan werkt sterk maar kort (ongeveer 12 jaar in de lucht), CO_2 zwakker maar eeuwenlang: daarom hangt het getal af van de periode die je kiest. Honderd jaar is de afspraak.

De **ecologische voetafdruk** kijkt breder: hoeveel vruchtbaar land en water is er nodig om alles te leveren wat je gebruikt (voedsel, hout, vezels) en om je CO_2 op te nemen? De eenheid is de **globale hectare** (gha): een hectare met de gemiddelde vruchtbaarheid van de aarde. Volgens het Global Footprint Network heeft de aarde ongeveer 1,5 gha per mens, maar gebruiken we gemiddeld ongeveer 2,6 gha: alsof we 1,7 aardes hadden. Ongeveer 60 % van die voetafdruk is koolstof: het bos dat nodig zou zijn om onze CO_2 op te nemen. Een gemiddelde Belg zit rond 6 gha: leefde iedereen zo, dan waren er ongeveer vier aardes nodig.



Wat de aarde per mens kan leveren, tegenover wat we gebruiken (orde van grootte).

UITGEWERKT VOORBEELD · 38.4

Een lek en een compostvat in CO_2-equivalent

4/18

(a) Uit de airco van een oude auto lekt in een jaar de hele vulling weg: 0,60 kg HFK-134a. Hoeveel CO_2-equivalent is dat, en hoeveel kilometer met een benzineauto (5,5 L per 100 km, 2,30 kg CO_2 per liter, hoofdstuk 14) komt daarmee overeen? (b) Het compostvat van het station (hoofdstuk 35) krijgt te weinig lucht en stoot 0,50 kg methaan per jaar uit, plus 0,02 kg lachgas. Hoeveel CO_2-equivalent?

GEGEVEN

- GWP: HFK-134a 1530, CH_4 (niet-fossiel) 27, N_2O 273
- auto: 5,5 L per 100 km, 2,30 kg CO_2 per liter

GEVRAAGD

- kg CO_2-eq voor (a) en (b); kilometers voor (a)

OPLOSSING

1. (a) $0{,}60 \cdot 1530 = 918\,\text{kg}$ CO_2-eq.
2. De auto stoot $5{,}5 \cdot 2{,}30/100 = 0{,}1265\,\text{kg}$ CO_2 per km uit: $918/0{,}1265 \approx 7300\,\text{km}$.
3. (b) $0{,}50 \cdot 27 + 0{,}02 \cdot 273 = 13{,}5 + 5{,}5 = 19\,\text{kg}$ CO_2-eq per jaar.
4. Ter vergelijking: bij goede compostering (genoeg lucht, hoofdstuk 35) wordt de koolstof CO_2 die kort geleden door planten werd opgenomen: die telt niet als extra opwarming. Het methaan telt wel.

ANTWOORD

(a) 918 kg CO_2-eq, zoveel als ±7300 km rijden; (b) ±19 kg CO_2-eq per jaar.

Klopt dit? Een halve kilo koelmiddel weegt zwaarder door dan een jaar methaan uit een slecht compostvat: daarom zijn nieuwe koelkasten in Europa met isobutaan of propaan gevuld (GWP bijna nul), en recupereert een vakman het koelmiddel bij het afdanken.

UITLEG · 38.5

Een levenscyclusanalyse in vier stappen

5/18

Een **levenscyclusanalyse** (LCA) rekent de milieu-effecten van een product uit over zijn hele levenscyclus. De methode ligt vast in de normen ISO 14040 en 14044 en heeft vier stappen:

1. **Doel en reikwijdte.** Welke vraag beantwoord je? Wat is de **functionele eenheid**: de dienst die je vergelijkt? Niet 'een kast in PLA of in PETG', maar '15 jaar een behuizing voor het stuurdeel in de kweekruimte'. En welke fasen tellen mee (de systeemgrens)?
2. **Inventaris.** Alle stromen in en uit: kilogram materiaal, kilowattuur, kilometer transport, afval.
3. **Effectbeoordeling.** De stromen omrekenen naar effecten met **emissiefactoren**: kg CO_2-eq per kg aluminium, per kWh stroom. Een volledige LCA kijkt ook naar water, verzuring, giftigheid en landgebruik; wij beperken ons tot CO_2-eq.
4. **Interpretatie.** Wat is de grootste post (de **hotspot**)? Verandert de conclusie als een rekenwaarde anders is (gevoeligheid)? Is de vraag goed gesteld?

DE KERN VAN ELKE LCA-REKENING

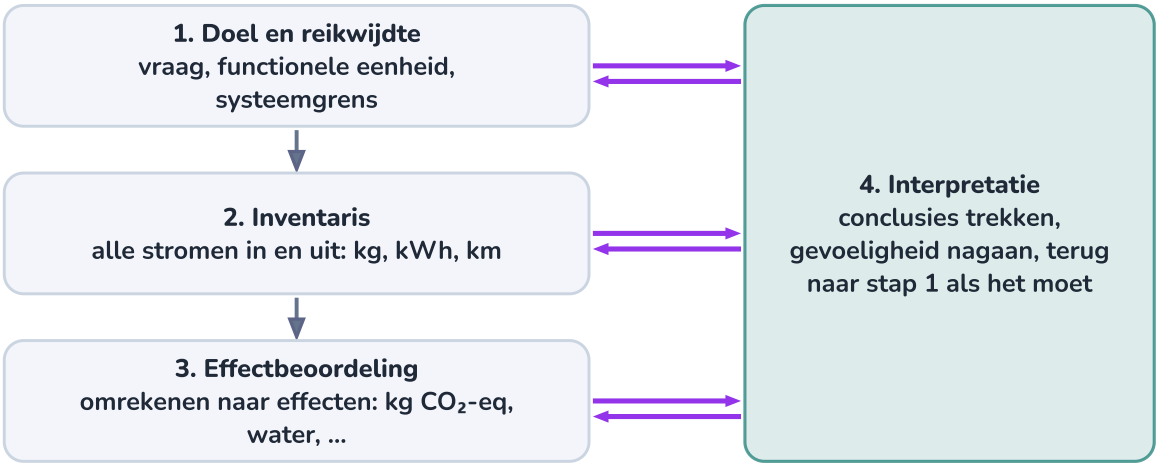
$$\text{uitstoot} = \sum_{\text{fasen}} \text{hoeveelheid} \times \text{emissiefactor}$$

Bv. 0,134 kg PETG × 5,85 kg CO_2-eq per kg = 0,79 kg CO_2-eq.

De emissiefactoren komen uit databanken en studies. Ze zijn onzeker: primair aluminium uit Europa geeft ongeveer 7 kg CO_2 per kg, het wereldgemiddelde ongeveer 15, gerecycleerd minder dan 1. In dit hoofdstuk gebruiken we afgeronde **rekenwaarden** (naslagblad), altijd met hun bron.

De eerste LCA

In 1969 liet Coca-Cola het Midwest Research Institute uitzoeken welke drankverpakking het minst grondstoffen en energie vroeg over haar hele levensloop. Die studie geldt als het begin van de levenscyclusanalyse. Pas in de jaren 1990 kwam er een internationale methode, en in 2006 de huidige ISO-normen.



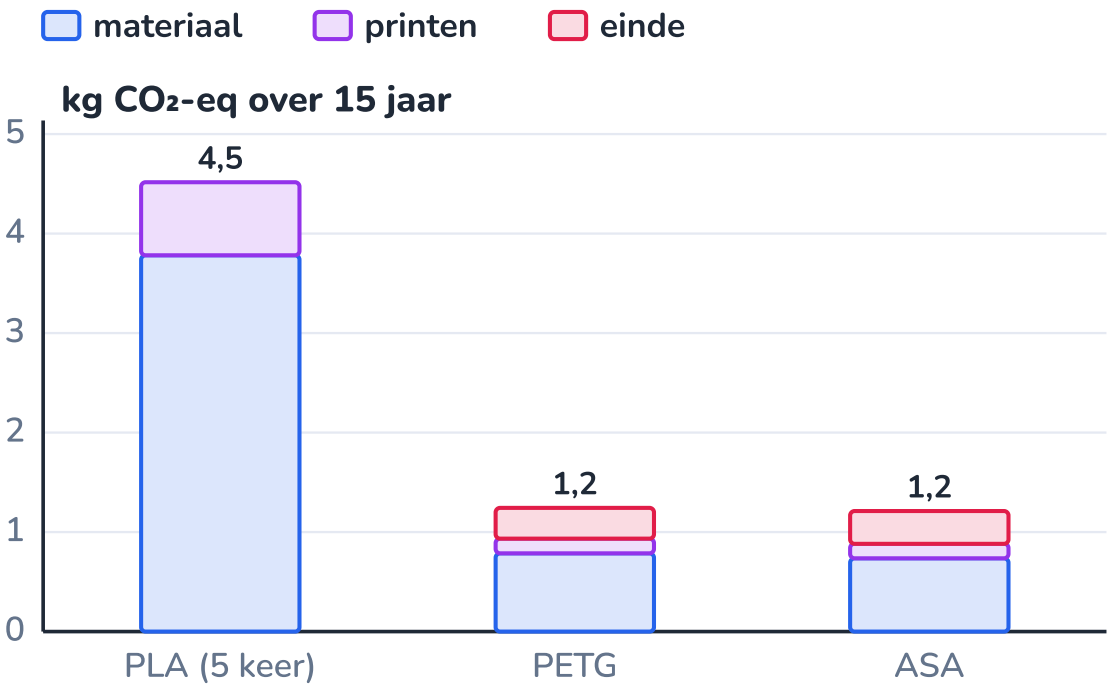
De interpretatie praat met alle andere stappen: een LCA is een lus, geen rechte weg.

UITGEWERKT VOORBEELD · 38.6

Een eenvoudige LCA: de regelkast in PLA, PETG of ASA

6/18

De regelkast RK-1 (hoofdstuk 37) is 105,9 cm³ en print 9,8 uur. Welk filament belast het klimaat het minst? Functionele eenheid: 15 jaar een behuizing voor het stuurdeel in de kweekruimte. PLA wordt in de warme, vochtige ruimte na ongeveer 3 jaar bros door vocht en warmte (hoofdstuk 17); PETG en ASA gaan 15 jaar mee (rekenwaarden).



Gestapeld per fase: bij PLA weegt het vijf keer opnieuw maken, niet het materiaal.

GEGEVEN

- filament: PLA 5,76, PETG 5,85, ASA ±6,5 kg CO_2-eq per kg; dichtheid 1,24 / 1,27 / 1,07 g/cm³
- printer 0,10 kW, stroom 0,15 kg CO_2-eq per kWh
- einde: verbranden; PETG 2,3 en ASA 2,9 kg CO_2 per kg; PLA telt niet (koolstof uit de plant)

GEVRAAGD

- kg CO_2-eq per fase en in totaal, over 15 jaar

OPLOSSING

1. **Hoeveel keer maken?** PLA: $15/3 = 5$ kasten; PETG en ASA: 1.
2. **Materiaal.** PETG: $105{,}9\,\text{cm}^3 \cdot 1{,}27 = 134{,}5\,\text{g}$; $0{,}1345 \cdot 5{,}85 = 0{,}79\,\text{kg}$. PLA: $5 \cdot 0{,}1313 \cdot 5{,}76 = 3{,}78\,\text{kg}$. ASA: $0{,}1133 \cdot 6{,}5 = 0{,}74\,\text{kg}$.
3. **Printen.** $9{,}8\,\text{h} \cdot 0{,}10\,\text{kW} = 0{,}98\,\text{kWh}$, maal 0,15: $0{,}15\,\text{kg}$ per kast (PLA: 5 keer: 0,73 kg).
4. **Einde.** PETG: $0{,}1345 \cdot 2{,}3 = 0{,}31\,\text{kg}$; ASA: $0{,}1133 \cdot 2{,}9 = 0{,}33\,\text{kg}$; PLA: 0.
5. **Totaal.** PLA 4,52 kg, PETG 1,24 kg, ASA 1,21 kg.

ANTWOORD

PETG en ASA zijn gelijkwaardig (±1,2 kg CO_2-eq over 15 jaar); PLA is bijna vier keer slechter omdat je het vijf keer moet maken. In de kweekruimte: PETG (licht van kleur); buiten in de zon: ASA.

Klopt dit? Het verschil tussen PETG en ASA (0,03 kg) is veel kleiner dan de onzekerheid op de rekenwaarden: noem ze gelijk. Het verschil met PLA komt niet van het materiaal maar van de levensduur: een typische uitkomst. En in perspectief: 1,2 kg is wat de groeilampen in ongeveer drie dagen aan netstroom zouden kosten (2 × 95 W × 16 h = 3,0 kWh per dag = 0,46 kg). De kast is geen hotspot.

CODE · 38.7

lca.py: de rekening in Python

Een LCA is een som van hoeveelheid maal factor, per fase. Met functies (hoofdstuk 28) reken je elke variant door zonder rekenfouten, en verander je een rekenwaarde op één plaats. Het programma berekent de regelkast uit het vorige voorbeeld en vergelijkt twee batterijen voor de klimaatmodule (volgend blad). `math.ceil` rondt naar boven af: 2,1 pakketten betekent 3 kopen.

Python **lca.py** 43 regels

```python
# lca.py - een eenvoudige levenscyclusanalyse (rekenwaarden, geen metingen)
import math

NET = 0.15        # kg CO2-eq per kWh uit het Belgische net (rekenwaarde)
JAREN = 15        # functionele eenheid: 15 jaar in de kweekruimte

# kg CO2-eq per kg filament (van grondstof tot rol), dichtheid (g/cm³), levensduur in de kweekruimte (jaar),
# kg CO2 bij verbranden (PLA: de koolstof komt uit de maïsplant, die telt hier niet mee)
FILAMENT = {"PLA": (5.76, 1.24, 3, 0.0), "PETG": (5.85, 1.27, 15, 2.3), "ASA": (6.5, 1.07, 15, 2.9)}


def onderdeel(volume_cm3, printuren, materiaal, printer_kw=0.10):
    """kg CO2-eq per fase voor een geprint onderdeel dat JAREN lang moet meegaan."""
    factor, rho, levensduur, verbranden = FILAMENT[materiaal]
    stuks = math.ceil(JAREN / levensduur)            # hoeveel keer je het moet maken
    massa = volume_cm3 * rho / 1000                  # kg per stuk
    return {"stuks": stuks,
            "materiaal": stuks * massa * factor,
            "printen": stuks * printuren * printer_kw * NET,
            "einde": stuks * massa * verbranden}


def batterij(kwh, per_kwh, cycli, rendement, massa, nacht_kwh=1.3):
    """Batterij die elke nacht nacht_kwh levert: productie van alle pakketten en het verlies bij laden en ontladen."""
    pakketten = math.ceil(JAREN * 365 / cycli)       # één cyclus per dag
    verlies = nacht_kwh * (1 / rendement - 1) * 365 * JAREN        # kWh die extra opgewekt moet worden
    return {"pakketten": pakketten, "productie": pakketten * kwh * per_kwh,
            "verlies": verlies * NET, "massa": pakketten * massa}


print("regelkast RK-1 (105,9 cm³, 9,8 h printen), 15 jaar:")
for naam in FILAMENT:
    r = onderdeel(105.9, 9.8, naam)
    totaal = r["materiaal"] + r["printen"] + r["einde"]
    print(f"  {naam:5} {r['stuks']} keer: materiaal {r['materiaal']:.2f} + printen {r['printen']:.2f}"
          f" + einde {r['einde']:.2f} = {totaal:.2f} kg CO2-eq")

print("batterij, 1,3 kWh per nacht, 15 jaar:")
lfp = batterij(3.07, 100, 3000, 0.95, 28)
lood = batterij(4.8, 80, 500, 0.80, 138)
for naam, r in (("LiFePO4", lfp), ("lood-zuur", lood)):
    print(f"  {naam:9} {r['pakketten']:2d} pakketten ({r['massa']:4d} kg): productie {r['productie']:5.0f}"
          f" + verlies {r['verlies']:3.0f} = {r['productie'] + r['verlies']:5.0f} kg CO2-eq")
```

Uitvoer:

```
regelkast RK-1 (105,9 cm³, 9,8 h printen), 15 jaar:
  PLA   5 keer: materiaal 3.78 + printen 0.73 + einde 0.00 = 4.52 kg CO2-eq
  PETG  1 keer: materiaal 0.79 + printen 0.15 + einde 0.31 = 1.24 kg CO2-eq
  ASA   1 keer: materiaal 0.74 + printen 0.15 + einde 0.33 = 1.21 kg CO2-eq
batterij, 1,3 kWh per nacht, 15 jaar:
  LiFePO4    2 pakketten (  56 kg): productie   614 + verlies  56 =   670 kg CO2-eq
  lood-zuur 11 pakketten (1518 kg): productie  4224 + verlies 267 =  4491 kg CO2-eq
```

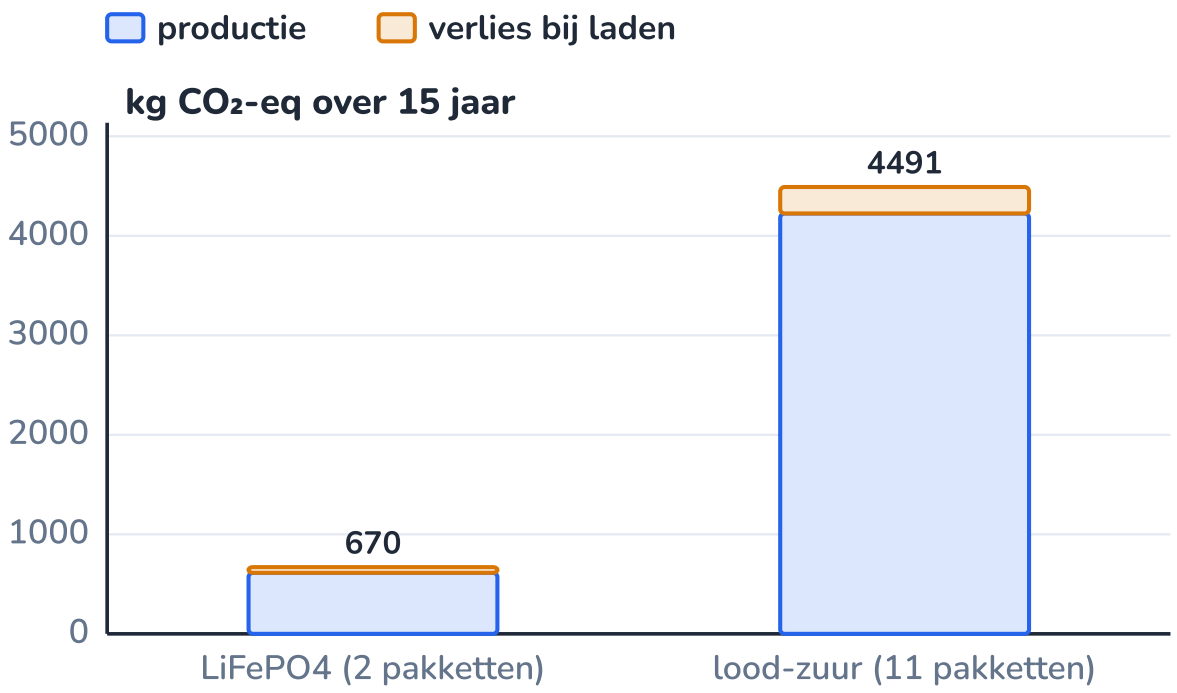
Een eerlijke LCA zegt ook hoe zeker ze is. Zet de rekenwaarden eens anders: met lood-zuur van 1000 in plaats van 500 cycli wordt het 2571 kg, nog altijd 3,8 keer meer dan LiFePO4. Een conclusie die blijft staan als je de onzekere getallen verandert, heet **robuust**. Die van de batterij is robuust; het verschil tussen PETG en ASA is dat niet.

UITGEWERKT VOORBEELD · 38.8

Batterij van de klimaatmodule: LiFePO4 of lood-zuur?

8/18

In hoofdstuk 16 en 20 koos Habitat twee LiFePO4-batterijen van 12,8 V en 120 Ah (samen 3,07 kWh, 2,46 kWh bruikbaar). Klopt die keuze ook voor het klimaat? Het alternatief is lood-zuur: twee batterijen van 12 V en 200 Ah (4,8 kWh, bij 50 % ontlaaddiepte 2,4 kWh bruikbaar). Elke nacht haalt de module ongeveer 1,3 kWh uit de batterij. Vergelijk over 15 jaar.



Elf keer een nieuw loodpakket weegt veel zwaarder dan het hogere rendement van twee LiFePO4-pakketten.

GEGEVEN

- LiFePO4: 100 kg CO_2-eq per kWh capaciteit, 3000 cycli, rendement 95 %, 28 kg
- lood-zuur: 80 kg CO_2-eq per kWh, 500 cycli, rendement 80 %, 138 kg
- één cyclus per dag; verloren energie: 0,15 kg CO_2-eq per kWh

GEVRAAGD

- aantal pakketten, CO_2 van productie en van verliezen, totaal

OPLOSSING

1. **Pakketten.** 15 jaar is 5475 cycli. LiFePO4: $5475/3000 = 1{,}8$: 2 pakketten. Lood-zuur: $5475/500 = 11{,}0$: 11 pakketten.
2. **Productie.** LiFePO4: $2 \cdot 3{,}07 \cdot 100 = 614\,\text{kg}$. Lood-zuur: $11 \cdot 4{,}8 \cdot 80 = 4224\,\text{kg}$.
3. **Verliezen.** Om 1,3 kWh terug te krijgen, moet er $1{,}3/\eta$ in. LiFePO4: $1{,}3 \cdot (1/0{,}95 - 1) = 0{,}068\,\text{kWh}$ per dag, over 15 jaar 375 kWh: 56 kg. Lood-zuur: $0{,}325\,\text{kWh}$ per dag, 1779 kWh: 267 kg.

4 **Totaal.** LiFePO4 670 kg, lood-zuur 4491 kg: 6,7 keer meer. En lood-zuur betekent 1518 kg batterijen aanvoeren en afvoeren.

ANTWOORD

LiFePO4, met ruime marge: ±670 kg CO_2-eq tegenover ±4500 kg over 15 jaar. De keuze van hoofdstuk 16 en 20 blijft staan.

Klopt dit? Wat telt hier niet mee? De recyclage. Loodbatterijen worden in Europa bijna volledig (±99 %) gerecycleerd, en die winst zit al deels in de rekenwaarde. LiFePO4-cellen recycleren gebeurt nog weinig, omdat er weinig waardevol metaal in zit: dat is een zwak punt van de gekozen batterij. Eerst hergebruiken als reserve (R7), dan pas recycleren.

UITLEG · 38.9

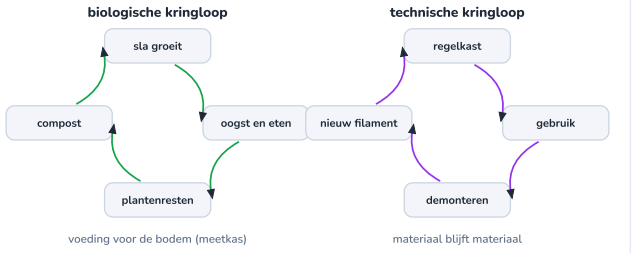
Cradle to cradle en circulair ontwerpen

9/18

In een **lineaire economie** gaat materiaal van de mijn naar de vuilnisbak. In een **circulaire economie** blijft het in gebruik. De architect William McDonough en de chemicus Michael Braungart beschreven in 2002 het idee **cradle to cradle** (van wieg tot wieg): ontwerp zo dat elk materiaal na gebruik in een van twee kringlopen terechtkomt:

- de **biologische kringloop**: materiaal dat veilig composteert en weer voeding wordt (plantenresten, papier, onbehandeld hout);
- de **technische kringloop**: materiaal dat zijn kwaliteit behoudt en opnieuw een product wordt (aluminium, staal, zuivere kunststof).

Een mengsel van beide (hout met kunststof verlijmd, een composiet) past in geen van de twee: het wordt verbrand. Welke strategie het best is, zegt de **R-ladder**: hoe hoger op de ladder, hoe minder materiaal en energie het kost. Recycleren staat laag, want omsmelten en opnieuw maken kost energie en verliest kwaliteit (een kunststofketen breekt bij elke smeltbeurt, hoofdstuk 17).



Links: wat composteert, keert terug als voeding. Rechts: wat niet composteert, blijft materiaal.

Hoe hoger op de ladder, hoe meer waarde je behoudt; recycleren is een laatste redmiddel.

UITLEG · 38.10

Ontwerpen voor demontage en herstel

10/18

Een product dat je niet kan openen, kan je niet herstellen, niet opwaarderen en niet netjes recycleren. Wie voor herstel ontwerpt, volgt een paar regels:

- **Losneembare verbindingen**: schroeven, klikken en stekkers in plaats van lijm, lassen en soldeer (hoofdstuk 37).

- **Weinig soorten gereedschap**, liefst standaard (één kruiskop), of geen.
- **Slijtdelen vooraan**: wat het eerst stuk gaat (pomp, batterij, ventilator), bereik je in weinig stappen, zonder de rest los te maken.
- **Modules** met duidelijke grenzen en stekkers: je vervangt een module, niet het hele toestel.
- **Herkenbare materialen**: een recyclagecode op elk kunststof deel (hoofdstuk 17), geen mengsels die niet te scheiden zijn.
- **Onderdelen en uitleg** beschikbaar: een stuklijst, een schema, reserveonderdelen.

Een goed voorbeeld is de **Fairphone 4** (2021): een smartphone die je met één kruiskopschroevendraaier uit elkaar haalt, met een batterij die je zonder gereedschap wisselt. De herstelwebsite iFixit gaf hem 10 op 10 voor herstelbaarheid. De Europese batterijverordening van 2023 maakt het deels verplicht: vanaf februari 2027 moet je de batterij van draagbare toestellen zelf kunnen vervangen.

De figuur vergelijkt twee versies van de pompunit van de klimaatmodule. In versie 1 is de pomp verlijmd en de kabel gesoldeerd: een nieuwe pomp kost 40 minuten, drie soorten gereedschap en een uur wachten op lijm. Versie 2 zet de pomp op een plaatje dat in een geleider klikt, met een snelkoppeling op de slang en een stekker in de kabel.

De demontagevraag

Stel bij elk ontwerp de vraag: *hoeveel stappen en welk gereedschap heb ik nodig om het onderdeel te vervangen dat het eerst stuk gaat?* Tel ze. Minder dan vijf stappen en geen speciaal gereedschap is een goed doel.

Rood: stappen die het herstel traag maken of iets onherstelbaar beschadigen.

PROEF · 38.11

Hoe herstelbaar is een toestel?

11/18

ONDERZOEKSVRAAG

Hoe snel en hoe netjes kan je een afgedankt toestel demonteren tot bij het onderdeel dat het eerst stuk gaat, en hoe goed kan je de materialen scheiden?

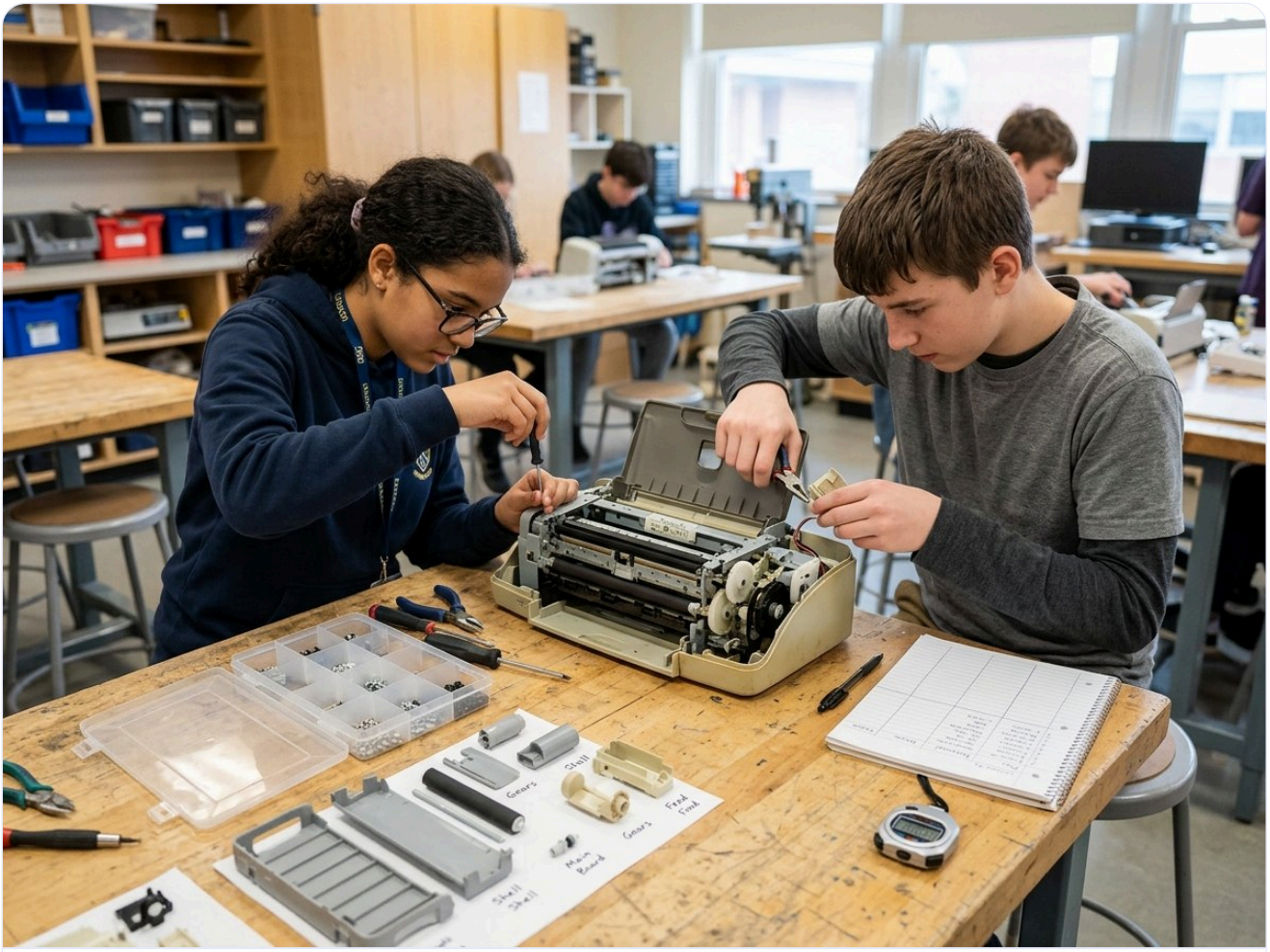
MATERIAAL

een afgedankt toestel (inkjetprinter, speelgoed, radio, handmixer) · schroevendraaiers (kruis, plat, torx), tang, spudger of plastic plectrum

bakjes en etiketten voor de onderdelen | stopwatch, keukenweegschaal | fototoestel of smartphone

Veiligheid

Alleen toestellen zonder stekker in het stopcontact en zonder ingebouwde lithiumbatterij. Geen magnetron, beeldbuis of toestel met grote condensatoren (voeding van een tv): die kunnen na het uittrekken nog gevaarlijke spanning geven. Handschoenen tegen scherpe randen.



Demonteren met een stopwatch en een sorteerbak: zo meet je herstelbaarheid.

WERKWIJZE

1. Kies vooraf het slijtdeel: de motor, de pomp, de knop, de batterij. Voorspel hoeveel stappen je nodig hebt.
2. Demonteer tot bij dat onderdeel. Noteer bij elke stap: wat, welk gereedschap, welke verbinding (schroef, klik, lijm, soldeer), tijd. Maak een foto per stap.
3. Demonteer daarna verder, zover het zonder breken kan. Sorteer de onderdelen per materiaal en weeg elke groep. Zoek de recyclagecodes.
4. Geef het toestel een score op vijf criteria (0, 1 of 2): verbindingen, gereedschap, aantal stappen tot het slijtdeel, herkenbare materialen, onderdelen en uitleg te vinden.

WAARNEMING

Typisch voor een kleine inkjetprinter: rond de dertig schroeven van drie of vier soorten, een paar klikken die breken bij het openen, zes tot tien stappen tot bij de motor, een printplaat met gesoldeerde draden, en kunststof delen waarvan maar een deel een code heeft (vaak ABS, PS of PP). Scores tussen 3 en 6 op 10 zijn gewoon.

VERKLARING

Veel toestellen zijn ontworpen om snel gemonteerd te worden, niet om hersteld te worden: klikken die alleen in één richting werken, lijm die trillen tegengaat, en onderdelen die in een vaste volgorde op elkaar liggen. Elke stap die het slijtdeel verder weg legt, maakt herstel duurder dan een nieuw toestel: dat is de reden waarom veel toestellen weggegooid worden terwijl maar één onderdeel stuk is. Criteria zoals deze gebruikt ook de Europese norm EN 45554 om herstelbaarheid te beoordelen.

UITLEG · 38.12

Een materiaalkeuzematrix met gewichten

12/18

Duurzaamheid is zelden het enige criterium. Een onderdeel moet ook sterk genoeg, veilig, betaalbaar en maakbaar zijn. Met een **materiaalkeuzematrix** maak je de afweging zichtbaar:

1. Kies de **criteria** uit het programma van eisen (F30). Harde eisen (moet) staan er niet in: een optie die een moet-eis niet haalt, valt vooraf af.
2. Geef elk criterium een **gewicht** in procent; samen 100 %. Beslis de gewichten *voor* je scoort, anders stuur je de uitkomst.
3. Geef elke optie per criterium een **score**, bv. van 1 tot 5, met een vooraf vastgelegde schaal (bv. CO_2: minder dan 20 kg = 5, 20 tot 50 = 4, 50 tot 100 = 3, 100 tot 200 = 2, meer = 1).
4. Bereken de **gewogen score**: de som van gewicht maal score.
5. Ga na hoe **gevoelig** de uitkomst is: verandert de winnaar als je een gewicht wat anders kiest?

GEWOGEN SCORE

$$S_{\text{optie}} = \sum_{k} \frac{w_k}{100\,\%} \cdot s_k$$

w_k = gewicht van criterium k (%), s_k = score van de optie op dat criterium (1 tot 5).

Een matrix beslist niet voor jou

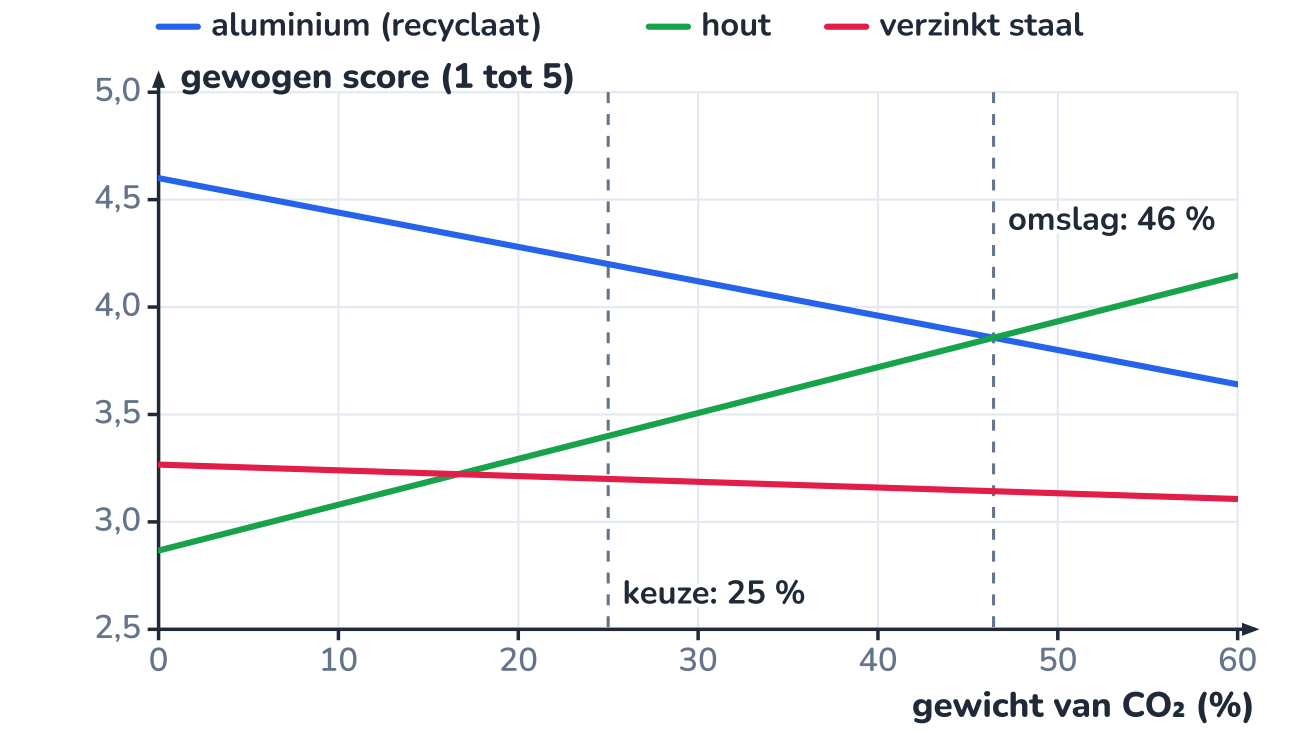
De getallen zien er objectief uit, maar de gewichten en de scores zijn keuzes. Schrijf bij elke score waarom (een getal, een meting, een bron), en toon de gevoeligheid. Liggen twee opties dicht bij elkaar, dan is de matrix geen reden om de ene te kiezen: kijk dan naar het criterium dat het zwaarst weegt voor de gebruiker.

UITGEWERKT VOORBEELD · 38.13

Het kweekrek: aluminium, hout of staal?

13/18

Het getrapte NFT-rek (2,4 × 1,2 m, vier goten) moet nog gebouwd worden. Uit een schets volgt ongeveer 24 m profiel. Drie opties: aluminium systeemprofiel 30 × 30 mm (met 75 % recyclaat), geschaafd vurenhout 44 × 44 mm, verzinkte stalen koker 25 × 25 × 1,5 mm. De kweekruimte is warm, vochtig en moet ontsmet kunnen worden (hoofdstuk 35). Maak een materiaalkeuzematrix.



Links van de omslag wint aluminium, rechts hout; staal wint nergens.

GEGEVEN

REK (24 M PROFIEL)	MASSA	KG CO_2-EQ PER KG	LEVENSDUUR	OVER 15 JAAR
aluminium	19,2 kg	4,1	30 jaar	
aluminium (primair)	19,2 kg	15	30 jaar	
hout	20,9 kg	0,26	8 jaar	
staal	26,4 kg	2,5	15 jaar	
- gewichten: CO_2 25 %, hygiëne 25 %, levensduur 15 %, aanpasbaar 15 %, kost 10 %, massa 10 %

GEVRAAGD

- de matrix met gewogen scores, en hoe gevoelig de keuze is voor het gewicht van CO_2

OPLOSSING

1 **CO_2 over 15 jaar.** Aluminium met recyclaat: $19{,}2\,\text{kg} \cdot 4{,}1 = 79\,\text{kg}$ (score 3; primair zou het 288 kg zijn, score 1). Hout: $20{,}9 \cdot 0{,}26 = 5{,}4\,\text{kg}$ per rek, twee rekken in 15 jaar: 10,9 kg (score 5). Staal: $26{,}4 \cdot 2{,}5 = 66\,\text{kg}$ (score 3).

2 **De andere criteria.** Hygiëne: aluminium en staal zijn glad en te ontsmetten, hout is poreus en kan schimmelen (2). Levensduur: hout verweert in vochtige lucht met druppels voedingsoplossing (2), zink tast aan door zouten (3). Aanpasbaar: systeemprofielen met gleuven verschuif je (5), hout schroef je (3), staal bout of las je (2). Kost: hout is het goedkoopst (5), aluminium het duurst (2). Massa: aluminium ±19 kg (5), hout ±21 kg (4), staal ±26 kg (3).

3 **Matrix.**

CRITERIUM	GEWICHT	ALUMINIUM	HOUT	STAAL
CO_2 over 15 jaar	25 %	3	5	3
hygiëne	25 %	5	2	4
levensduur	15 %	5	2	3
aanpasbaar	15 %	5	3	2
kost	10 %	2	5	4
massa	10 %	5	4	3
gewogen score	100 %	**4,20**	**3,40**	**3,20**

4 **Gevoeligheid.** Laat je het gewicht van CO_2 stijgen (de andere schalen mee), dan haalt hout aluminium in bij ongeveer 46 %. Bij 25 % is de keuze duidelijk; wie CO_2 het belangrijkste criterium vindt, kiest hout, en moet dan de hygiëne op een andere manier oplossen (lak, vervangbare latten).

ANTWOORD

Aluminium systeemprofiel met recyclaat (gewogen score 4,20, tegenover 3,40 voor hout en 3,20 voor staal), en dan liefst met een recyclaatgehalte op papier: primair aluminium zou de CO_2-score van 3 naar 1 brengen.

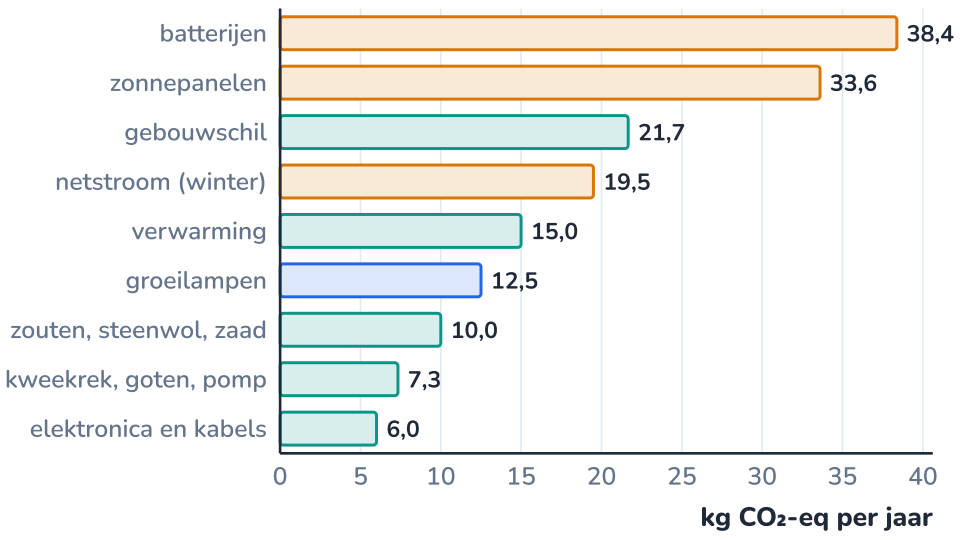
Klopt dit? De matrix bevestigt wat de hygiëneregels van hoofdstuk 35 al suggereerden. Circulair plus: het systeemprofiel kan na 30 jaar opnieuw gebruikt of omgesmolten worden (R3, R8), zolang je het niet lijmt.

UITGEWERKT VOORBEELD · 38.14

Habitat: de klimaatmodule doorgelicht

14/18

Waar zit de voetafdruk van de klimaatmodule? Reken per jaar: verdeel de productie van elk onderdeel over zijn levensduur en tel het gebruik erbij. Welke onderdelen zijn de hotspots, en wat zegt dat over de keuzes die er nog te maken zijn?



Geel: de energievoorziening; blauw: de lampen. Samen het grootste deel van de voetafdruk.

GEGEVEN

- de rekenwaarden van het naslagblad
- levensduur: panelen 25, gebouw 30, rek 15, elektronica 10, batterijen en lampen 8 jaar
- netstroom in de winter ±130 kWh per jaar (tekort van november tot februari, hoofdstuk 20)
- verwarming door het station: ±100 kWh elektriciteit per jaar (warmtepomp); opbrengst ±100 kg sla per jaar

GEVRAAGD

- kg CO_2-eq per jaar per onderdeel, het totaal en de hotspots

OPLOSSING

1 **Rekenen** met een woordenboek, zoals lca.py:

```
# doorlichting.py - de klimaatmodule per jaar: productie gespreid over de levensduur, plus het gebruik
NET = 0.15                    # kg CO2-eq per kWh (rekenwaarde)
ONDERDELEN = {                # naam: (kg CO2-eq om te maken, levensduur in jaar, kWh per jaar uit het net)
    "zonnepanelen 1,40 kWp": (1.40 * 600, 25, 0),
    "batterijen 2 x LiFePO4": (3.07 * 100, 8, 0),
    "groeilampen 2 x 95 W": (2 * 50, 8, 0),
    "gebouwschil": (650, 30, 0),
    "kweekrek, goten, reservoir, pomp": (110, 15, 0),
    "elektronica en kabels": (60, 10, 0),
    "netstroom in de winter": (0, 1, 130),
    "verwarming (warmtepomp station)": (0, 1, 100),
    "voedingszouten, steenwol, zaad": (10, 1, 0),
}
per_jaar = {naam: maken / jaren + kwh * NET for naam, (maken, jaren, kwh) in ONDERDELEN.items()}
totaal = sum(per_jaar.values())
for naam, kg in sorted(per_jaar.items(), key=lambda x: -x[1]):
    print(f"{naam:34} {kg:5.1f} kg/jaar  {kg / totaal:5.1%}")
print(f"{'totaal':34} {totaal:5.1f} kg/jaar, per kg sla (100 kg per jaar): {totaal / 100:.2f} kg CO2-eq")
```

```
batterijen 2 x LiFePO4              38.4 kg/jaar  23.4%
zonnepanelen 1,40 kWp               33.6 kg/jaar  20.5%
gebouwschil                         21.7 kg/jaar  13.2%
netstroom in de winter              19.5 kg/jaar  11.9%
verwarming (warmtepomp station)     15.0 kg/jaar   9.1%
groeilampen 2 x 95 W                12.5 kg/jaar   7.6%
voedingszouten, steenwol, zaad      10.0 kg/jaar   6.1%
kweekrek, goten, reservoir, pomp     7.3 kg/jaar   4.5%
elektronica en kabels                6.0 kg/jaar   3.7%
totaal                             164.0 kg/jaar, per kg sla (100 kg per jaar): 1.64 kg CO2-eq
```

2 **Hotspots.** Batterijen en zonnepanelen samen maken 44 % uit, met de netstroom erbij 56 %: de voetafdruk zit in de energievoorziening, niet in de kasten of het rek.

3 **En waarom is die energie nodig?** 93 % van het energiebudget gaat naar de groeilampen (hoofdstuk 20). Tel 93 % van panelen, batterijen en netstroom bij de lampen zelf: dan hangt ongeveer 60 % van de voetafdruk aan het licht. Zonnestroom is dus niet gratis: de CO_2 zit in de panelen en de batterijen.

4 **Per kg sla** is het ±1,6 kg CO_2-eq; het hele jaar samen ±164 kg, zoveel als ±1300 km met de benzineauto van hoofdstuk 14.

ANTWOORD

±164 kg CO_2-eq per jaar. Hotspots: batterijen, zonnepanelen, gebouwschil; indirect de lampen. Wie de module duurzamer wil, begint bij het licht: zuinige lampen, dimmen met daglicht, reflecterende wanden, en de batterijen lang laten meegaan.

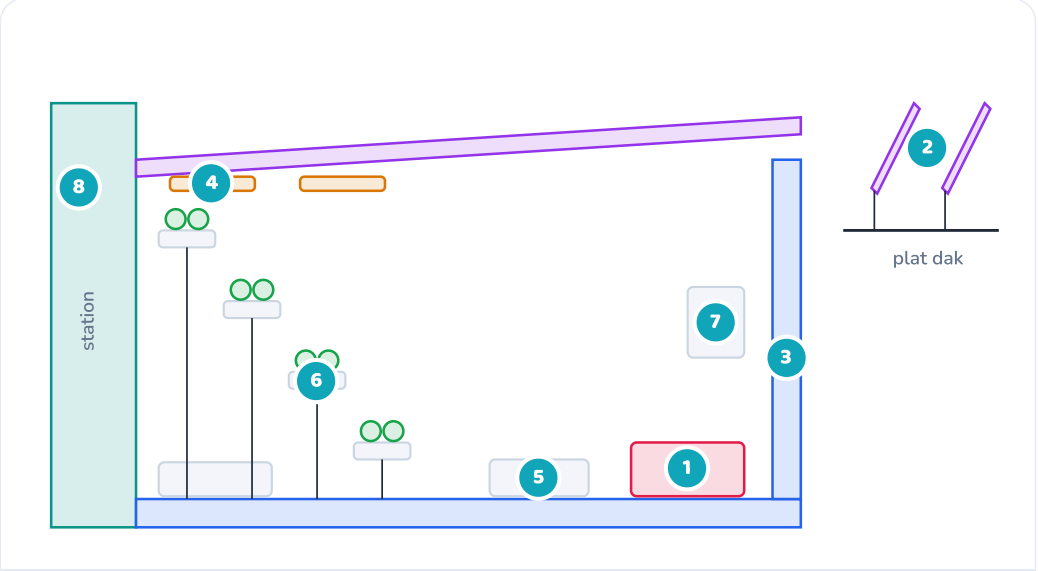
Klopt dit? Wat telt niet mee? In de zomer levert de module meer zonnestroom dan ze gebruikt (±300 kWh per jaar). Wie dat overschot aan het station geeft, mag het met 0,15 kg per kWh aftrekken: ±45 kg per jaar. Of je dat doet, is een keuze van systeemgrens, en die moet je vermelden. De rekenwaarden zijn orde van grootte: de rangorde van de hotspots is betrouwbaarder dan de getallen zelf.

ONTLEED · 38.15

Waar zit de voetafdruk van de klimaatmodule?

15/18

De onderdelen van de doorlichting in een schematische doorsnede van de module (niet op schaal), met hun aandeel per jaar.



De module tegen het station; de panelen staan op het platte dak.

1 Batterijen (23 %)
Twee LiFePO4-pakketten van 1,54 kWh, elke 8 jaar nieuw. Langer laten meegaan (niet dieper dan nodig ontladen, niet onder 0 °C laden) is de grootste winst.

2 Zonnepanelen (20 %)
Vier panelen van 350 Wp, 25 jaar. Hun CO_2 zit in het maken van het silicium; ze verdienen die terug door stroom te leveren (Expert, hoofdstuk 47).

3 Gebouwschil (13 %)
Houtskeletwanden, vloer, polycarbonaat dak en aluminium kader, 30 jaar. Hout slaat koolstof op zolang het niet verbrandt.

4 Groeilampen (8 % zelf, en de oorzaak van het meeste energieverbruik)
Twee lampen van 95 W, 16 uur per dag: 93 % van het energiebudget. Elke procent minder licht verbruik spaart panelen, batterijen en netstroom.

5 Verwarming (9 %)
Warm water van het station via een convector. Een nachtscherm (hoofdstuk 19) verlaagt het warmteverlies van 20,2 naar 16,3 W/K.

6 Kweekrek, goten, reservoir, pomp (4 %)
Klein aandeel, maar hier beslis je over hygiëne en herstel: aluminium systeemprofiel, pompunit zonder lijm.

7 Regelkast en elektronica (4 %)
Mega, sensoren, laadregelaar en kabels. De geprinte kast zelf weegt nauwelijks door: ±1,2 kg over 15 jaar.

8 Netstroom in de winter (12 %)
Van november tot februari leveren de panelen te weinig; het station vult aan. In de zomer is er een overschot.

UITLEG · 38.16

Eerlijk vergelijken en greenwashing herkennen

16/18

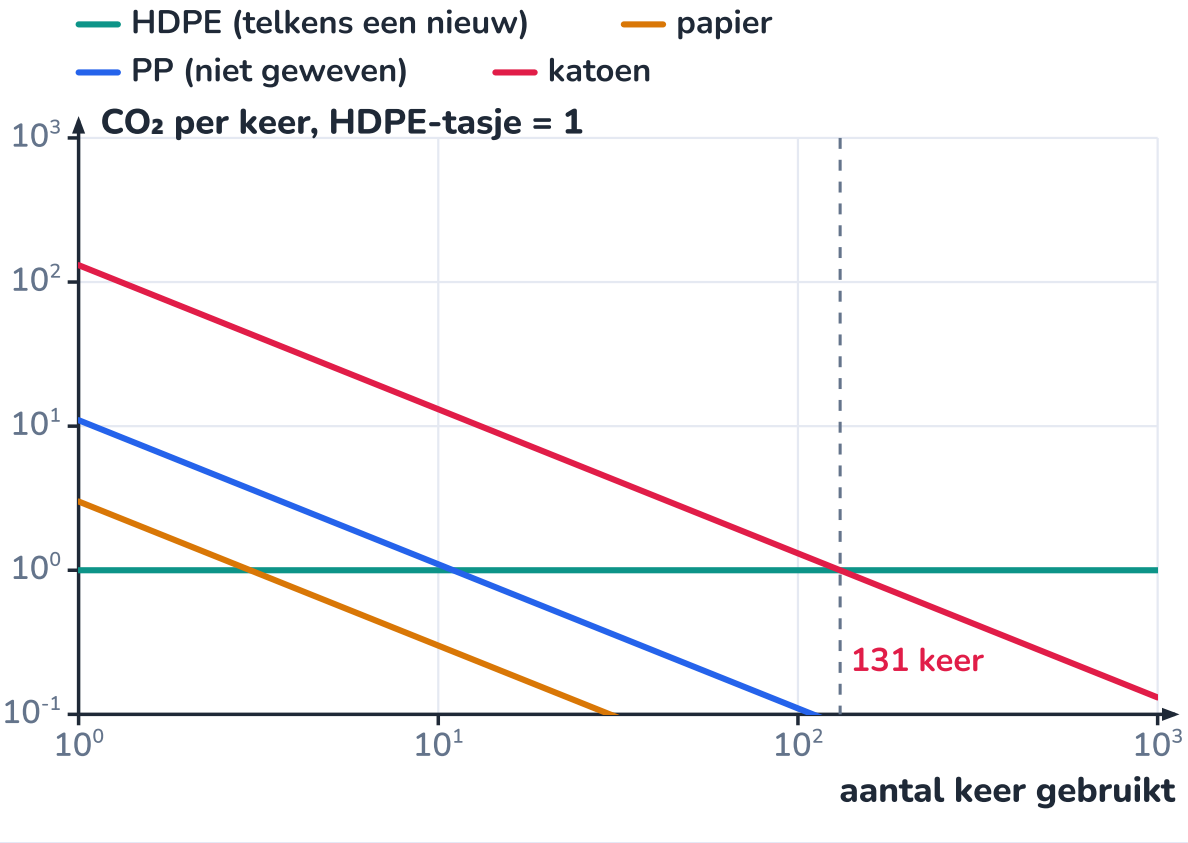
Een vergelijking is pas eerlijk als ze aan een paar voorwaarden voldoet:

- **Dezelfde functionele eenheid**: een tas per keer boodschappen doen, een kast per 15 jaar, niet per stuk.
- **Dezelfde systeemgrens**: een waarde 'van wieg tot poort' vergelijk je niet met een waarde 'van wieg tot graf'.
- **Het gebruik en de vervangingen** tellen mee, ook al zijn ze moeilijk te schatten.
- **Meer dan één effect**: een katoenen tas scoort slecht op CO_2 en goed op zwerfvuil. Wie maar één getal toont, kiest.
- **Onzekerheid**: rekenwaarden met een bereik, en een gevoeligheidsanalyse.

Greenwashing is een product groener voorstellen dan het is. Typische trucs: vage woorden ('groen', 'eco', 'natuurlijk'), één goed punt in de verf zetten (een gerecycleerde verpakking rond een wegwerpproduct), 'recycleerbaar' zeggen terwijl het bijna nergens gerecycleerd wordt, of 'CO_2-neutraal' zijn door elders bomen te laten planten. Sinds 27 september 2026 verbiedt een Europese richtlijn (2024/825) algemene milieuclaims zonder bewijs en claims als 'klimaatneutraal' die alleen op compensatie steunen.

Vijf vragen bij een milieuclaim

Wat precies (welk deel van het product, welk effect)? Vergeleken met wat? Over welke levensloop? Wie heeft het gemeten of nagekeken? Wat staat er níet?



Per keer boodschappen doen: een tas die veel kost om te maken, wint pas na veel gebruik (logaritmische assen).

NASLAG · 38.17

Rekenwaarden, formules en ladders

17/18

CO₂-equivalent

$$m_{\mathrm{CO_2\text{-}eq}} = \sum_i m_i \cdot \mathrm{GWP}_i$$

symbool	betekenis	eenheid
m_i	massa van gas i	kg
GWP_i	opwarmingsvermogen over 100 jaar	–

LCA-rekening

$$E = \sum_{\text{fasen}} Q \cdot f$$

symbool	betekenis	eenheid
Q	hoeveelheid (kg, kWh, t·km)	–
f	emissiefactor	kg CO_2-eq per eenheid

per jaar gespreid

$$E_{\text{per jaar}} = \frac{E_{\text{maken}}}{\text{levensduur}} + E_{\text{gebruik per jaar}}$$

symbool	betekenis	eenheid
E	uitstoot	kg CO_2-eq

gewogen score

$$S = \sum_{k} \frac{w_k}{100\,\%} \cdot s_k$$

symbool	betekenis	eenheid
w_k	gewicht	%
s_k	score	1 tot 5

materiaal of proces	kg CO_2-eq	per	bron (afgerond)
stroom uit het Belgische net	0,15	kWh	Ember, 2023: ±0,14
PLA-filament (nieuw)	5,76	kg	Prusa Research, 2022 (van grondstof tot rol)
PETG-filament (nieuw / gerecycleerd)	5,85 / 2,55	kg	Prusa Research, 2022
ASA-filament	±6,5	kg	schatting (geen gepubliceerde waarde)
aluminium, primair (wereldmix / Europa)	15 / 7	kg	European Aluminium, 2024: 14,8 / 6,6
aluminium, omgesmolten schroot	±0,5	kg	European Aluminium: ±0,3 plus bewerken
aluminiumprofiel met 75 % recyclaat	±4	kg	berekend: 0,25 × 15 + 0,75 × 0,5
gezaagd naaldhout (fossiel)	0,26	kg	ICE-database v3; plus ±1,6 kg biogene CO_2 opgeslagen
staal, verzinkt	±2,5	kg	schatting
LiFePO4-batterij / lood-zuur	±100 / ±80	kWh capaciteit	orde van grootte uit LCA-studies (60 tot 150)
zonnepaneel (PERC, China)	±600	kWp	IEA PVPS, 2024: 575
ledarmatuur 95 W	±50	stuk	schatting
3D-printer tijdens het printen	0,10 kW	–	rekenwaarde

gas	waar het vandaan komt	GWP (100 jaar)
koolstofdioxide CO_2	verbranding, cement, ontbossing	1
methaan CH_4 (niet-fossiel)	vee, rijst, stortplaatsen, compost zonder lucht	27
methaan CH_4 (fossiel)	lekken bij aardgas en steenkool	29,8
lachgas N_2O	kunstmest, mest, verbranding	273
HFK-134a (koelmiddel)	oudere koelkasten en autoairco's	1530

Waarden van het IPCC (2021). Methaan werkt sterk maar kort (ongeveer 12 jaar in de lucht), CO_2 zwakker maar eeuwenlang: daarom hangt het getal af van de periode die je kiest. Honderd jaar is de afspraak.

fase	vraag bij het ontwerpen
grondstof en materiaal	minder materiaal? gerecycleerd? één soort per onderdeel?
product maken	minder energie, minder afval, minder mislukte prints?
transport	lichter, dichterbij, minder volume?
gebruik	zuiniger, langer meegaan, herstelbaar?
einde	demonteerbaar, herkenbaar, hergebruik of recyclage mogelijk?

HET GEHEEL · 38.18

Waar duurzaam ontwerpen vandaan komt en naartoe gaat

18/18

Duurzaam ontwerpen verbindt drie leerlijnen: ontwerpen (welk materiaal, welke verbinding), energie (waar komt de stroom vandaan, wat kost ze) en materie (wat gebeurt er met een kunststof of een metaal aan het einde). Het rekenwerk is eenvoudig, een som van producten; de kunst zit in eerlijke keuzes over de functionele eenheid, de systeemgrens en de onzekerheid. In Expert wordt dat een volledige LCA van het station, met energieterugverdientijden en meerdere milieu-effecten.

F10 Materialen en hun eigenschappen

Materiaalfamilies, grondstoffen en de materialenkringloop: de basis van elke levenscyclus.

F17 Energie uit zon, wind en water

CO_2 per kWh van verschillende bronnen, en het klimaat.

F28 Ecosystemen en kringlopen

Kringlopen in de natuur: het model voor de biologische kringloop.

I14 Reacties en stoichiometrie

CO_2 uit een verbranding berekenen: het einde van een kunststof.

I16 Redox en batterijen

Cycli, rendement en capaciteit: de getallen achter de batterijvergelijking.

I17 Polymeren en composieten

Bioplastics, recyclagecodes en waarom composieten moeilijk te recycleren zijn.

I20 Zonne-energie dimensioneren

Het energiebudget dat de lampen tot hotspot maakt.

I35 Micro-organismen en biotechnologie

Compost en hygiëne: de biologische kringloop van de module, en waarom hout in de kweekruimte lastig is.

I37 Parametrisch ontwerpen

De regelkast en haar verbindingen: klikken in plaats van lijm.

I40 Integratieproject: de klimaatmodule

De doorlichting en de beslissingen komen in het technisch rapport van de module.

E27 Elektrochemie: batterijen, brandstofcellen en elektrolyse

Batterijen, brandstofcellen en opslag vergeleken op rendement en energiedichtheid.

E28 Materiaalkunde

Materiaalselectie met Ashby-diagrammen: de keuzematrix met fysica erachter.

E47 Duurzaamheid en levenscyclusanalyse

Een volledige LCA, systeemgrenzen en de energieterugverdientijd van een zonnepaneel.

Habitat

De klimaatmodule is doorgelicht: ±164 kg CO_2-eq per jaar (±1,6 kg per kg sla), vooral in de energievoorziening (batterijen en panelen samen 44 %), en die hangt grotendeels aan het licht. De batterijkeuze (LiFePO4) is bevestigd, de regelkast komt in PETG, het kweekrek in aluminium systeemprofiel met recyclaat, de pompunit wordt ontworpen voor herstel. Opdracht 38.15 is de beslissing voor je portfolio; het integratieproject (hoofdstuk 40) neemt ze op in het technisch rapport.

SAMENVATTING

In het kort

- De levenscyclus: grondstof, materiaal, product, gebruik, einde, met transport ertussen. De systeemgrens zegt wat je meetelt.
- CO_2-equivalent = massa × GWP (methaan ±27, lachgas 273). De ecologische voetafdruk meet in globale hectare; we gebruiken ±1,7 aardes.

- Een LCA: doel en functionele eenheid, inventaris, effecten (hoeveelheid × emissiefactor), interpretatie en gevoeligheid.
- Levensduur weegt vaak zwaarder dan materiaal: een PLA-kast die vijf keer vervangen wordt, belast het klimaat bijna vier keer meer.
- Cradle to cradle: biologische en technische kringloop. De R-ladder: eerst minder, langer en herstellen; recycleren is een laatste stap.
- Ontwerp voor demontage: losneembare verbindingen, weinig gereedschap, slijtdelen vooraan, herkenbare materialen.
- Een materiaalkeuzematrix met gewichten maakt de afweging zichtbaar; een gevoeligheidsanalyse toont hoe vast de keuze staat.
- De voetafdruk van de klimaatmodule zit in de energievoorziening, en die hangt aan het licht.

BEGRIPPEN

Wat je nu kent

levenscyclus
Alle stappen van een product: grondstoffen winnen, materiaal en product maken, transport, gebruik en het einde (hergebruik, recyclage, verbranden, storten).

levenscyclusanalyse (LCA)
Methode (ISO 14040) om de milieu-effecten van een product over zijn hele levenscyclus te berekenen.

functionele eenheid
De dienst waarop je een vergelijking baseert, bv. '15 jaar een behuizing in de kweekruimte', zodat je gelijke dingen vergelijkt.

systeemgrens
Wat je wel en niet meetelt in een LCA: van wieg tot poort, van wieg tot graf of van wieg tot wieg.

CO_2-equivalent
De hoeveelheid CO_2 die over 100 jaar evenveel opwarming veroorzaakt als een hoeveelheid van een ander broeikasgas.

GWP
Global warming potential: hoeveel keer sterker een gas opwarmt dan CO_2 over 100 jaar (methaan ±27, lachgas 273).

ecologische voetafdruk
De oppervlakte vruchtbaar land en water (in globale hectare) die nodig is om te leveren wat iemand gebruikt en zijn CO_2 op te nemen.

cradle to cradle
Ontwerpen zodat materialen na gebruik volledig terugkeren: in de biologische kringloop (composteren) of de technische (opnieuw gebruiken).

R-ladder
Rangorde van circulaire strategieën, van weigeren en verminderen via herstellen en hergebruiken tot recycleren en terugwinnen.

materiaalkeuzematrix
Tabel waarin je opties scoort op criteria met gewichten; de gewogen som helpt kiezen.

greenwashing
Een product groener voorstellen dan het is, met vage, onvolledige of onbewezen milieuclaims.

I INTERMEDIATE · DEEL I10 · ONTWERPEN EN DATA

HANDBOEK · HOOFDSTUK 39

Data visualiseren en eerste stappen in AI

Van logboek naar grafiek naar een model dat zelf zijn regels leert, en wat het niet kan

Een logboek met tienduizend regels zegt niets tot je het goed toont. In dit hoofdstuk kies je de grafiek die bij je vraag past, herken je grafieken die misleiden, en leer je waarom twee grootheden samen kunnen bewegen zonder dat de ene de andere veroorzaakt. Daarna zet je een eerste stap in kunstmatige intelligentie: je laat de computer uit voorbeelden een beslisboom leren die voorspelt wanneer de sla in de klimaatmodule water tekortkomt, je toetst hem eerlijk, en je zoekt uit wat zo'n model niet kan.

9× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× naslag · 1× verhaal · 1× het geheel · 6× code · ± 10 lestijden

A Data en AI · O Onderzoeken en meten · P Programmeren en computationeel denken

Van logboek naar grafiek naar beslissing: data zijn pas nuttig als je ze eerlijk toont en toetst.

Na dit hoofdstuk

- Je kan bij een vraag de passende grafiek kiezen en een grafiek maken met correcte assen, eenheden en bron.
- Je kan misleidende grafieken herkennen, de liegfactor berekenen en een grafiek eerlijk maken.
- Je kan uitleggen waarom correlatie geen oorzakelijk verband bewijst, en een gemeenschappelijke oorzaak opsporen door in groepen te kijken.
- Je kan uitleggen wat machine learning is en niet is, en de begrippen kenmerk, label, model, trainen en testen gebruiken.
- Je kan met de hand de beste splitsing van een beslisboom zoeken en een boom volgen om een voorspelling te maken.
- Je kan een beslisboom trainen en toetsen in Python, een verwarringsmatrix lezen en overfitting herkennen.
- Je kan bias in data herkennen en de grenzen van een model benoemen voor je het laat beslissen.

VERHAAL · 39.1

Een kaart die een pomp liet sluiten

1/23

Eind augustus 1854 brak in de Londense wijk Soho cholera uit. Binnen enkele dagen stierven honderden mensen. De meeste artsen dachten dat de ziekte zich verspreidde via 'slechte lucht'. De arts John Snow

vermoedde dat het water de boosdoener was. Hij ging van deur tot deur, noteerde waar de doden woonden, en zette ze op een kaart van de straten: een streepje per dode, bij het huis.

Op de kaart zag iedereen wat in een tabel verborgen bleef: de doden lagen dicht rond één waterpomp, in Broad Street. Snow keek ook naar de uitzonderingen. Een brouwerij vlak bij de pomp had bijna geen zieken: de arbeiders dronken bier en hadden een eigen put. Een vrouw die ver weg woonde maar wel stierf, liet water van Broad Street halen omdat ze het lekkerder vond. Op 8 september 1854 liet het bestuur van de parochie de hendel van de pomp verwijderen. De uitbraak nam toen al af, maar Snows kaart werd een van de bekendste grafieken ooit, omdat ze een oorzaak zichtbaar maakte.

Waarom dit hoofdstuk ertoe doet

Snow toonde de juiste data op de juiste manier, en keek kritisch naar wat niet paste. Dat doe je in dit hoofdstuk met het logboek van de klimaatmodule: grafieken kiezen, misleiding herkennen, en een verband pas een oorzaak noemen als je de alternatieven hebt uitgesloten. Daarna leer je de computer zelf een regel vinden, en zie je waarom ook die kritisch getoetst moet worden.

Een kaart met een streepje per overlijden: het patroon rond één pomp werd zichtbaar.

UITLEG · 39.2

Wat je al weet, en de juiste grafiek kiezen

2/23

Je kan al veel met data:

wat je al kan	waar	wat er nu bij komt
een tabel en een grafiek maken, assen met grootheid en eenheid	F4	kiezen welke grafiek bij welke vraag past
een trendlijn en de correlatie tussen twee grootheden	I3, I28	wanneer een verband geen oorzaak is
CSV inlezen, per dag samenvatten, matplotlib	I28	eerlijke en misleidende grafieken
de datalogger, opschonen, logboek.py	I30	de testweek per uur, als basis voor analyse
algoritmen, en systemen die leren uit voorbeelden (eerste kennismaking)	F22	een model dat zelf zijn regels leert: de beslisboom
if, elif, else en toestanden	I27, I32	een getrainde boom is een rij if-vragen

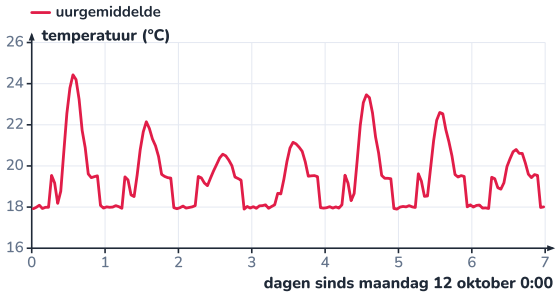
Een grafiek beantwoordt een **vraag**. Kies eerst de vraag, dan de grafiek:

vraag	grafiek	voorbeeld uit de module
hoe verandert iets in de tijd?	lijngrafiek	temperatuur per uur
hoe verhouden aparte groepen zich?	staafdiagram	verbruik en zon per dag
hangen twee grootheden samen?	spreidingsdiagram	verbruik tegenover daglicht
hoe zijn metingen verdeeld?	histogram	hoe vaak elke temperatuur voorkwam
welk deel van een geheel?	taartdiagram (weinig delen) of gestapelde staaf	het energiebudget: 93 % lampen

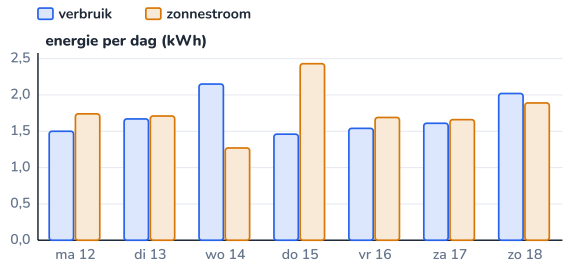
Een **lijngrafiek** verbindt metingen die in de tijd op elkaar volgen: de lijn suggereert dat er tussen twee metingen ook iets was. Voor losse groepen (dagen, materialen, bewoners) gebruik je **staven**, die altijd bij 0 beginnen omdat hun lengte de waarde is. Een **spreidingsdiagram** zet twee grootheden tegen elkaar, zonder lijn. Een **histogram** telt hoe vaak waarden in een klasse vallen. Een **taartdiagram** toont delen van een geheel, maar alleen goed met weinig delen: hoeken vergelijken is moeilijker dan lengtes.

Vijf dingen die elke grafiek heeft

Een grootheid en een eenheid op elke as, een legende als er meer dan één reeks is, een onderschrift dat zegt wat je moet zien, de bron van de data, en de periode.



Verloop in de tijd: een lijn. Elke dag een bult, het hoogst op de heldere maandag.



Vergelijken per dag: staven. Op woensdag en zondag was er te weinig zon.

CODE · 39.3

Van minuten naar uren: uurwaarden.py

Het logboek van de klimaatmodule schrijft elke minuut een regel (hoofdstuk 30): 10 079 regels voor de testweek. Voor de grafieken en de analyses van dit hoofdstuk vat je ze samen per uur. Het programma gebruikt `logboek.py` uit hoofdstuk 30 om de dagbestanden in te lezen en de onmogelijke waarden weg te halen, verzamelt de minuten per uur in een woordenboek (hoofdstuk 28), en schrijft per uur een gemiddelde (temperatuur, vocht, licht, spanning) of een som (energie).

Python `uurwaarden.py` 24 regels

```python
# uurwaarden.py - de testweek van de logger (hoofdstuk 30) per uur samenvatten in één CSV-bestand
import csv
from logboek import lees_week, opschonen          # logboek.py uit hoofdstuk 30

rijen = lees_week()
print("opgeschoond:", opschonen(rijen), "regels met onmogelijke waarden")
uren = {}                                          # "2026-10-12 09:00" -> de minuten van dat uur
for r in rijen:
    uren.setdefault(r["tijd"].strftime("%Y-%m-%d %H:00"), []).append(r)


def gemiddelde(waarden):
    geldig = [w for w in waarden if w is not None]
    return sum(geldig) / len(geldig)


with open("week_uur.csv", "w", newline="") as f:
    schrijver = csv.writer(f)
    schrijver.writerow(["uur", "temperatuur_C", "vocht_pct", "verbruik_Wh", "zon_Wh", "licht_lx", "batterij_V"])
    for uur, rs in sorted(uren.items()):
        schrijver.writerow([uur, round(gemiddelde([r["T"] for r in rs]), 2), round(gemiddelde([r["RV"] for r in rs]), 1),
                            round(sum(r["verbruik"] for r in rs), 1), round(sum(r["zon"] for r in rs), 1),
                            round(gemiddelde([r["licht"] for r in rs])), round(gemiddelde([r["bat"] for r in rs]), 2)])
print(len(uren), "uren in week_uur.csv")
```

Uitvoer, en het begin van week_uur.csv:

```
opgeschoond: 3 regels met onmogelijke waarden
168 uren in week_uur.csv

uur,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V
2026-10-12 00:00,17.92,84.1,7.4,0.0,0,13.25
2026-10-12 01:00,17.99,84.1,7.4,0.0,0,13.25
2026-10-12 02:00,18.09,83.9,7.4,0.0,0,13.25
…
```

Een gemiddelde over de geldige minuten laat de lege cellen weg; een som van energie telt elke minuut. Dinsdag om 14 uur heeft 59 minuten (de regel die ontbrak), woensdag om 11 uur 48 geldige temperaturen. Dat is goed genoeg voor een uurgemiddelde, maar vermeld het.

OOK TE DOWNLOADEN

UITGEWERKT VOORBEELD · 39.4

Vier vragen, vier grafieken

Een bewoner stelt vier vragen over de testweek. Welke grafiek beantwoordt elke vraag het best, en wat is het antwoord? (a) Was het in de module ooit te warm? (b) Welke dag had een tekort aan zonnestroom? (c) Verbruikt de module minder als er meer daglicht is? (d) Welke temperatuur kwam het vaakst voor?

GEGEVEN

- week_uur.csv (168 uren)
- de keuzetabel van het vorige blad

GEVRAAGD

- een grafiektype en een antwoord per vraag

OPLOSSING

1. **(a) Te warm?** Een verloop in de tijd: de lijngrafiek van de temperatuur (blad 2). Het warmste uur was 24,4 °C: ver onder de 27 °C waarbij het dak opengaat (hoofdstuk 31).

2. **(b) Tekort?** Aparte dagen vergelijken: staven, verbruik en zon naast elkaar. Woensdag: 2,15 kWh verbruik, 1,27 kWh zon; zondag 2,02 tegenover 1,89 kWh.

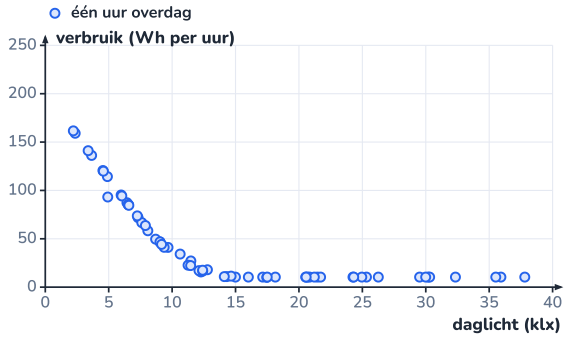
3. **(c) Samenhang?** Twee grootheden tegen elkaar: een spreidingsdiagram van het verbruik per uur tegenover het daglicht (figuur). Meer licht, minder verbruik: de lampen dimmen mee (hoofdstuk 25).

4. **(d) Verdeling?** Een histogram van de 168 uurtemperaturen (figuur). 119 van de 168 uren lagen tussen 17 en 20 °C: de nachten en de avonden, waarin het station de module op ±18 °C houdt; de klasse van 19 tot 20 °C komt het vaakst voor.

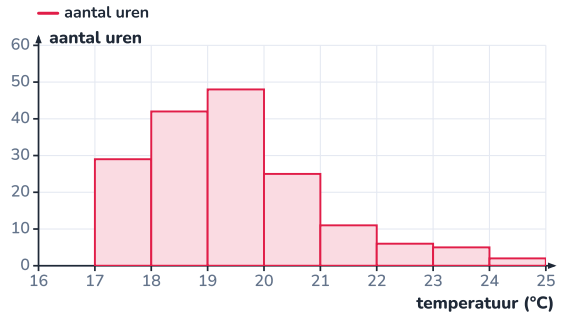
ANTWOORD

Lijn voor de tijd, staven voor de dagen, punten voor een verband, een histogram voor de verdeling.

Klopt dit? Probeer het omgekeerde: een lijngrafiek van de dagtotalen suggereert dat er tussen maandag en dinsdag iets verliep, en een taart van 168 uren zegt niets. De grafiek volgt uit de vraag.



Elk punt is één uur overdag: bij veel daglicht dimmen de lampen, het verbruik daalt.



Scheef verdeeld: veel koele uren, en alleen op zonnige middagen warmer, tot 24,4 °C.

UITLEG · 39.5

Misleidende grafieken: de as en de twee assen

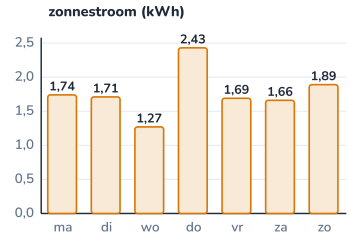
5/23

Een grafiek kan correct zijn, met juiste getallen, en toch iets verkeerds laten zien. Twee klassieke trucs:

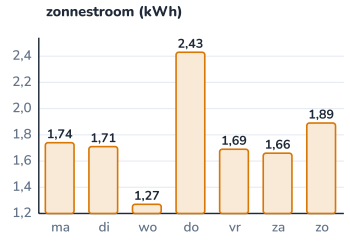
De afgeknotte as. Bij staven is de lengte de boodschap. Begint de as bij 1,2 in plaats van 0, dan is de staaf van donderdag (2,43 kWh) 17 keer zo lang als die van woensdag (1,27 kWh), terwijl het 1,9 keer zoveel is. Bij een lijngrafiek mag een as elders beginnen (je kijkt naar het verloop, niet naar de lengte), zolang de getallen op de as staan.

Twee y-assen. Met een tweede as kies je zelf hoe hoog en hoe steil een reeks lijkt. Hieronder lijken de temperatuur en de batterijspanning samen te bewegen, alsof warmte de batterij oplaadt. Maar beide volgen gewoon de zon: overdag laadt de batterij (spanning omhoog) en warmt de module op. Met een andere keuze van de rechteras lopen ze helemaal niet gelijk.

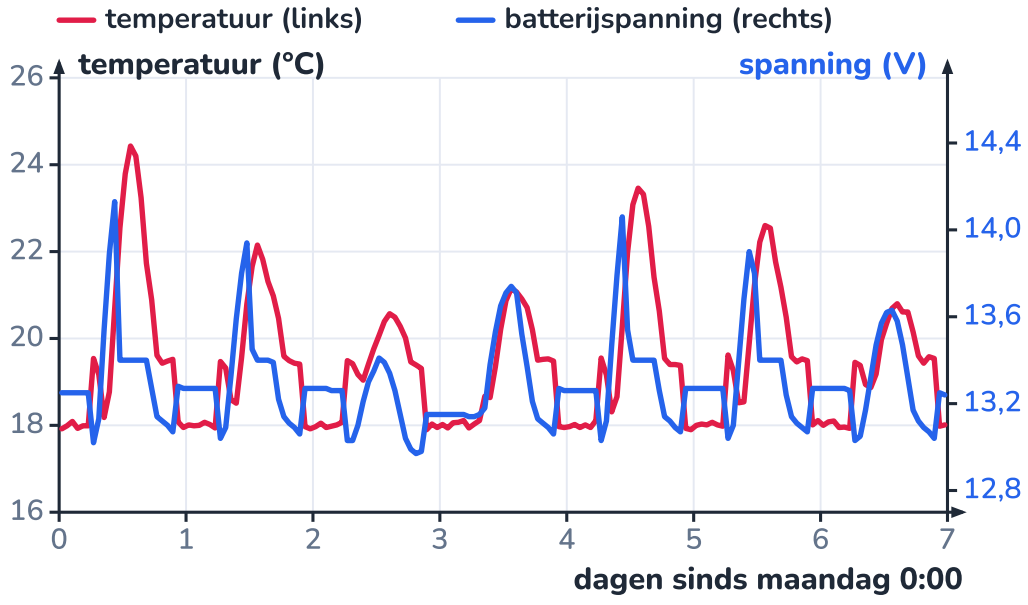
truc	wat het doet	tegengif
afgeknotte as bij staven	kleine verschillen lijken groot	staven altijd vanaf 0
twee y-assen	elke gewenste 'samenhang' is te maken door de assen te kiezen	twee grafieken onder elkaar, of één as
kersen plukken	alleen de periode tonen die het verhaal steunt	de hele reeks, of zeggen waarom een stuk
oppervlakte of volume als pictogram	een verdubbeling lijkt een verviervoudiging	lengte evenredig met de waarde (staaf)
gaten verbinden	een lijn over ontbrekende metingen verzint data	de lijn onderbreken
geen eenheid, geen bron	niemand kan nagaan wat er getoond wordt	grootheid, eenheid, bron en periode



Eerlijk: de staven beginnen bij 0.



Dezelfde getallen met een as vanaf 1,2: de verschillen lijken veel groter.



Twee assen, zo gekozen dat de lijnen samen lijken te lopen. Samenloop is hier geen oorzaak.

UITLEG · 39.6

Misleidende grafieken: kersen plukken, pictogrammen en gaten

6/23

Kersen plukken. Wie alleen donderdag toont, kan schrijven dat 'de zon 67 % meer levert dan de module verbruikt'; wie alleen woensdag toont, dat 'de module 0,88 kWh per dag tekortkomt'. Beide zijn waar

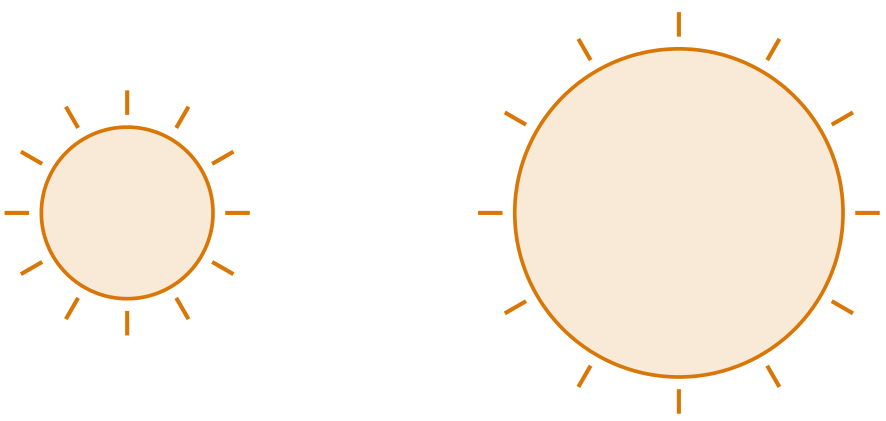
voor die dag, en beide misleiden. Over de hele week leverde de zon 12,4 kWh en verbruikte de module 12,0 kWh.

Oppervlakte en volume. Een pictogram (een zon, een batterij, een zak) dat in hoogte én breedte meegroeit, groeit in oppervlakte met het kwadraat. De statisticus Edward Tufte noemde de verhouding tussen het effect in de grafiek en het effect in de data de **liegfactor**; eerlijk is 1.

LIEGFACTOR (TUFTE, 1983)

$$\text{liegfactor} = \frac{\text{effect in de grafiek}}{\text{effect in de data}}, \qquad \text{effect} = \frac{\text{tweede} - \text{eerste}}{\text{eerste}}$$

Gaten verbinden. Woensdag van 11:47 tot 11:58 ontbreken metingen (hoofdstuk 30). Een lijn die de punten ervoor en erna verbindt, verzint een verloop. matplotlib onderbreekt de lijn als je op die plaats een lege waarde (`float('nan')`) in de lijst zet.



De diameter van de zon groeit mee met de zonnestroom; het oog kijkt naar de oppervlakte.

UITGEWERKT VOORBEELD · 39.7

De liegfactor van het zonnepictogram

7/23

Bereken de liegfactor van het pictogram hierboven: woensdag 1,27 kWh, donderdag 2,43 kWh, en de diameter van de zon is evenredig met de zonnestroom.

GEGEVEN

- woensdag $1{,}27\,\text{kWh}$, donderdag $2{,}43\,\text{kWh}$
- diameter evenredig met de waarde

GEVRAAGD

- het effect in de data, het effect in de grafiek, de liegfactor

OPLOSSING

1. **In de data:** $\frac{2{,}43 - 1{,}27}{1{,}27} = 0{,}91$: donderdag is 91 % meer.
2. **In de grafiek:** de diameter wordt $1{,}913$ keer groter, de oppervlakte dus $1{,}913^2 = 3{,}66$ keer: het effect is $3{,}66 - 1 = 2{,}66$.
3. **Liegfactor:** $2{,}66 / 0{,}91 = 2{,}91$.

ANTWOORD

Een liegfactor van ongeveer 2,9: het pictogram overdrijft het verschil bijna drie keer.

Klopt dit? Een eerlijk pictogram schaalt de oppervlakte met de waarde, dus de diameter met de vierkantswortel: √1,91 = 1,38 keer groter. Nog eenvoudiger: gebruik staven, waarvan alleen de lengte telt.

CODE · 39.8

Een eerlijke grafiek in matplotlib

8/23

matplotlib kiest zelf de assen, en dat is niet altijd eerlijk: bij staafdiagrammen begint de as bij 0, maar bij een lijngrafiek zoomt ze in op de data. Met `set_ylim` leg je de as zelf vast. Dit programma tekent de zonnestroom per dag twee keer naast elkaar: eerlijk en met een afgeknotte as, zodat je het verschil zelf ziet. `plt.subplots(1, 2)` maakt twee grafieken naast elkaar.

Python `eerlijke_grafiek.py` 19 regels

```python
# eerlijke_grafiek.py - dezelfde zonnestroom per dag, eerlijk en misleidend
import matplotlib.pyplot as plt

dagen = ["ma", "di", "wo", "do", "vr", "za", "zo"]
zon = [1.74, 1.71, 1.27, 2.43, 1.69, 1.66, 1.89]                    # kWh per dag (week_uur.csv)

figuur, (links, rechts) = plt.subplots(1, 2, figsize=(10, 4))
links.bar(dagen, zon, color="tab:orange")
links.set_ylim(0, 2.6)                                # eerlijk: de as begint bij 0
links.set_ylabel("zonnestroom (kWh per dag)")
links.set_title("eerlijk")
rechts.bar(dagen, zon, color="tab:orange")
rechts.set_ylim(1.2, 2.5)                             # misleidend: de as begint bij 1,2
rechts.set_title("misleidend: afgeknotte as")
for as_ in (links, rechts):
    as_.grid(axis="y")
figuur.tight_layout()
figuur.savefig("zon_per_dag.png", dpi=150)
print("zon_per_dag.png bewaard")
```

Uitvoer: zon_per_dag.png bewaard. De twee grafieken zien eruit zoals de staafdiagrammen op het blad over de as. Wie een grafiek in een verslag zet, kiest de as bewust en schrijft er het waarom bij.

Bewaar de code bij de grafiek

Zet het programma dat een grafiek maakte bij je verslag. Dan kan iedereen nagaan welke data en welke keuzes erachter zitten, en de grafiek opnieuw maken als er nieuwe metingen zijn.

UITLEG · 39.9

Correlatie is geen oorzakelijk verband

9/23

In hoofdstuk 28 berekende je de correlatiecoëfficiënt r: $+1$ als twee grootheden perfect samen stijgen, -1 als de ene daalt wanneer de andere stijgt, 0 als er geen rechtlijnig verband is. In de testweek is overdag $r = -0{,}40$ tussen temperatuur en verbruik: warmere uren, minder verbruik. Zet de verwarming dan hoger om stroom te sparen? Nee. Een correlatie kan vier oorzaken hebben:

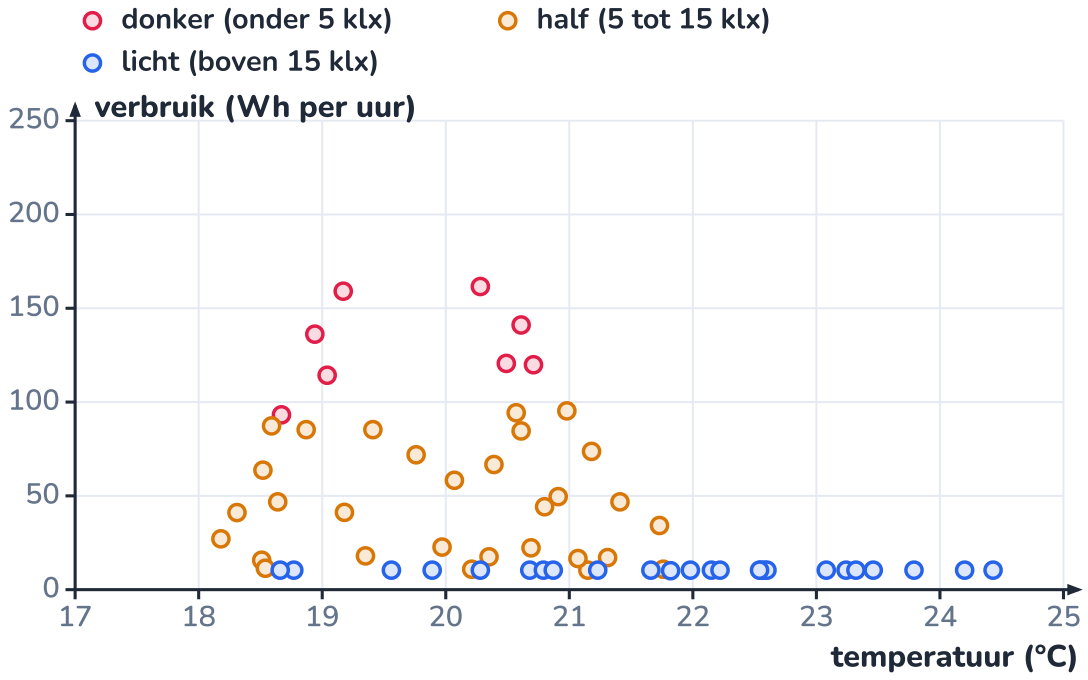
1. **De ene veroorzaakt de andere**: meer daglicht doet de lampen dimmen, dus minder verbruik ($r = -0{,}75$).
2. **Een gemeenschappelijke oorzaak**: het daglicht warmt de module op én dimt de lampen. Temperatuur en verbruik bewegen samen omdat ze allebei aan de zon hangen.

3. **De omgekeerde richting**: soms is het gevolg de oorzaak. Brandende lampen warmen de module een beetje op (hoofdstuk 30): meer verbruik, iets warmer.
4. **Toeval of selectie**: met weinig metingen of een gekozen stuk van de data vind je altijd wel een verband. Over alle uren van de week, ook 's nachts, is $r = 0{,}11$: een ander stuk, een ander getal.

Een eenvoudige test voor een gemeenschappelijke oorzaak: **houd ze constant**. Kijk alleen naar uren met ongeveer hetzelfde daglicht. Verdwijnt het verband binnen elke groep, dan liep het via het licht. In de figuur zijn de uren gekleurd naar het daglicht: binnen een kleur is er nauwelijks een helling.

"Een sterke correlatie bewijst een oorzaak"

In 2000 toonde de statisticus Robert Matthews dat het aantal ooievaarsparen in Europese landen sterk samenhangt met het aantal geboorten (p = 0,008). Ooievaars brengen geen baby's: grote landen hebben meer van beide. Een oorzaak bewijs je met een proef waarin je één ding verandert en de rest gelijk houdt (hoofdstuk 2), of met een verklaring die alle alternatieven uitsluit, zoals Snow deed.



Samen dalen de punten van links boven naar rechts onder; binnen één kleur (hetzelfde licht) bijna niet.

CODE · 39.10

Het verband binnen groepen: correlatie.py

10/23

Het programma berekent r voor alle uren overdag, en daarna opnieuw binnen drie groepen met ongeveer hetzelfde daglicht. De functie `correlatie()` is die van hoofdstuk 28.

Python correlatie.py 26 regels

```python
# correlatie.py - hangt het verbruik af van de temperatuur? (overdag, 8 tot 17 uur)
import csv


def correlatie(x, y):
    """Correlatiecoëfficiënt r van Pearson (hoofdstuk 28)."""
    mx, my = sum(x) / len(x), sum(y) / len(y)
    sxy = sum((a - mx) * (b - my) for a, b in zip(x, y))
    sxx = sum((a - mx) ** 2 for a in x)
    syy = sum((b - my) ** 2 for b in y)
```

```
    return sxy / (sxx * syy) ** 0.5


with open("week_uur.csv", newline="") as f:
    uren = [r for r in csv.DictReader(f) if 8 <= int(r["uur"][11:13]) <= 16]
T = [float(r["temperatuur_C"]) for r in uren]
V = [float(r["verbruik_Wh"]) for r in uren]
L = [float(r["licht_lx"]) for r in uren]
print(f"{len(uren)} uren overdag")
print(f"temperatuur en verbruik: r = {correlatie(T, V):+.2f}")
print(f"licht en verbruik:       r = {correlatie(L, V):+.2f}")
print(f"licht en temperatuur:    r = {correlatie(L, T):+.2f}")
for naam, laag, hoog in (("donker", 0, 5000), ("half", 5000, 15000), ("licht", 15000, 1e9)):
    groep = [i for i in range(len(uren)) if laag <= L[i] < hoog]
    r = correlatie([T[i] for i in groep], [V[i] for i in groep])
    print(f"  alleen {naam:6} ({len(groep):2d} uren): temperatuur en verbruik r = {r:+.2f}")
```

Uitvoer:

```
63 uren overdag
temperatuur en verbruik: r = -0.40
licht en verbruik:       r = -0.75
licht en temperatuur:    r = +0.67
  alleen donker ( 8 uren): temperatuur en verbruik r = +0.28
  alleen half   (30 uren): temperatuur en verbruik r = -0.08
  alleen licht  (25 uren): temperatuur en verbruik r = +0.00
```

Samen $r = -0{,}40$, binnen de groepen tussen $-0{,}08$ en $0{,}28$: het verband tussen temperatuur en verbruik verdwijnt bijna als het licht gelijk blijft. De zon is de gemeenschappelijke oorzaak. Kijk ook naar de kolom zon_Wh: die meet de *geleverde* zonnestroom, die afgeknepen wordt als de batterij vol is (hoofdstuk 30). Daarom is haar correlatie met de temperatuur zelfs negatief ($r = -0{,}53$): wie zon_Wh als maat voor zonneschijn gebruikt, trekt de verkeerde conclusie. Weet wat een kolom meet.

UITLEG · 39.11

Wat AI en machine learning zijn, en wat niet

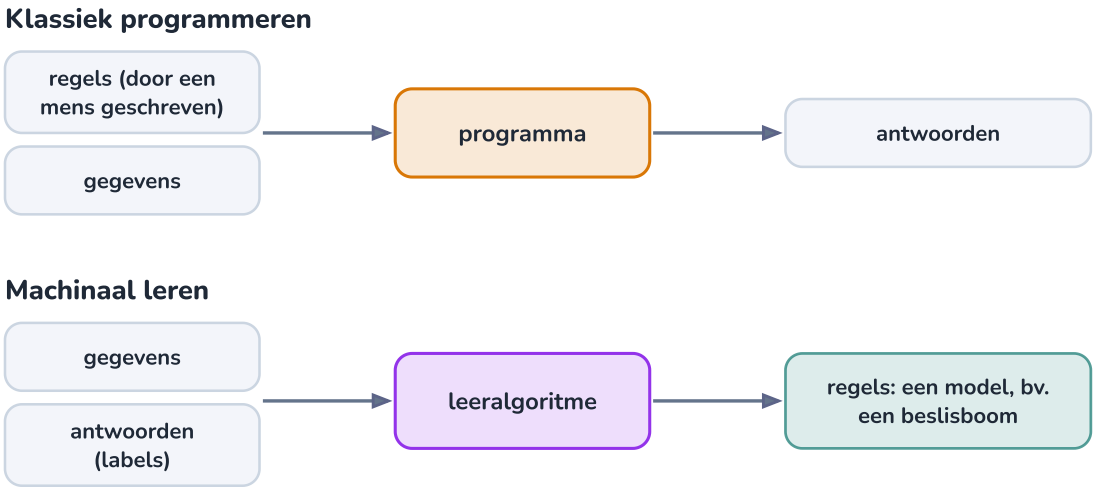
11/23

Kunstmatige intelligentie (AI) is een verzamelnaam voor computersystemen die taken uitvoeren waarvoor mensen intelligentie gebruiken: herkennen, voorspellen, plannen, taal verwerken. Het grootste deel van de huidige AI is **machine learning**: de computer leert een regel uit voorbeelden, in plaats van dat een mens de regel programmeert.

In hoofdstuk 25 van Fundamental schreef je zelf: 'water geven onder 30 % bodemvocht'. Bij machine learning geef je de computer voorbeelden: **kenmerken** (temperatuur, vocht, licht) met het juiste **label** (slap of niet slap). Een **leeralgoritme** zoekt een regel die de labels zo goed mogelijk voorspelt: het **model**. Dat heet **trainen**. Daarna gebruik je het model op nieuwe gevallen.

soort taak	wat het model leert	voorbeeld in Habitat	waar
classificeren	een categorie kiezen	slap of niet slap; zieke of gezonde plant	I39, E42
regressie	een getal voorspellen	het verbruik van morgen	E41
clusteren	groepen vinden zonder labels	soorten dagen in het logboek	E42
neurale netwerken	ingewikkelde verbanden uit veel data	de zonne-opbrengst uit de weersvoorspelling	E43
taal- en beeldmodellen	het volgende woord of beeldstukje voorspellen	een assistent die het logboek samenvat	E44

Wat AI **niet** is: een model begrijpt niets. Het vindt patronen in de voorbeelden die het kreeg, en meer niet. Het weet niet waarom sla slap hangt, het kent geen natuurkunde, en het kan niet weten wat het nooit zag. Ook een taalmodel dat vlot praat, voorspelt woord na woord wat waarschijnlijk volgt (Expert, hoofdstuk 44). 'Intelligent' zegt iets over de taak, niet over begrip.



Bij machine learning schrijf je de regels niet zelf: het algoritme haalt ze uit de voorbeelden.

UITLEG · 39.12

De beslisboom

12/23

Een **beslisboom** voorspelt met een reeks ja/nee-vragen. Je begint bovenaan (de wortel) en volgt bij elke vraag de tak die past, tot je in een **blad** komt: daar staat de voorspelling. Hij lijkt op het stroomdiagram van hoofdstuk 22 en op een rij `if`-vragen in Python of C++ (hoofdstuk 27 en 32). Het verschil: de vragen en de grenzen zijn niet door een mens gekozen, maar geleerd uit data.

De boom hieronder werd getraind op de inspecties van juni en juli. Hij stelt twee vragen. Is er weinig daglicht (hoogstens 29,5 klx), dan verdampt de sla weinig: niet slap. Is er veel licht, dan hangt het af van de lucht: droge lucht (vocht hoogstens 60,5 %) trekt water uit de bladeren, en dan verwacht de boom slappe sla.

Hoe vindt de computer de vragen? Hij probeert **elke** mogelijke vraag: elk kenmerk, en elke grens tussen twee gemeten waarden. Voor elke vraag splitst hij de voorbeelden in een ja- en een nee-groep, en meet hoe **zuiver** die groepen zijn: hoe goed ze bestaan uit één soort label. De beste vraag komt bovenaan. Daarna doet hij hetzelfde in elke groep, tot een gekozen diepte.

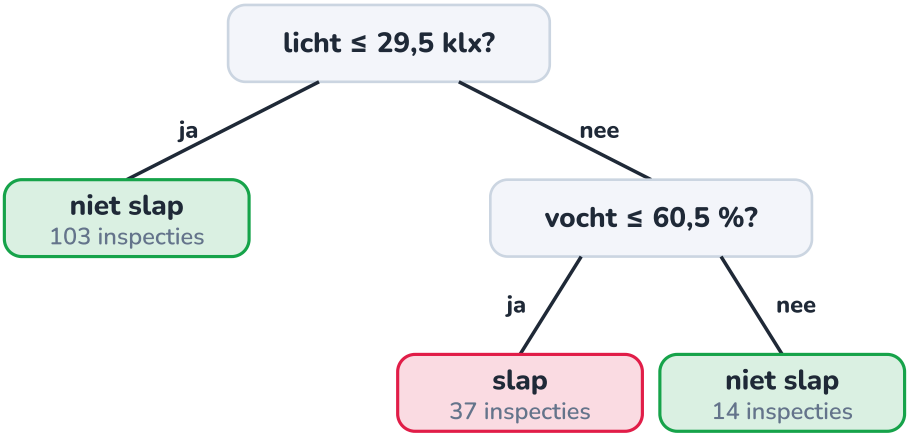
ONZUIVERHEID VAN EEN GROEP (GINI)

$$\text{onzuiverheid} = 2\,p\,(1-p)$$

p = deel van de groep met het label 'slap'. Alles slap of alles niet slap: 0. Half om half: $2 \cdot 0{,}5 \cdot 0{,}5 = 0{,}5$. Een splitsing krijgt het gewogen gemiddelde van de onzuiverheid van haar twee groepen.

Waarom niet gewoon fouten tellen?

Met de hand tel je fouten: dat is eenvoudig en meestal goed genoeg. Maar fouten tellen ziet soms geen verschil tussen twee splitsingen waarvan de ene veel nuttiger is. Voorbeeld: 400 slap en 400 niet slap. Splitsing A geeft groepen (300 slap, 100 niet) en (100, 300): 200 fouten. Splitsing B geeft (200, 400) en (200, 0): ook 200 fouten, maar B maakt een groep die al helemaal zuiver is. De onzuiverheid ziet dat: A 0,375, B 0,333. De computer kiest B.



De geleerde boom: rood de bladen 'slap', groen 'niet slap', met het aantal trainingsvoorbeelden.

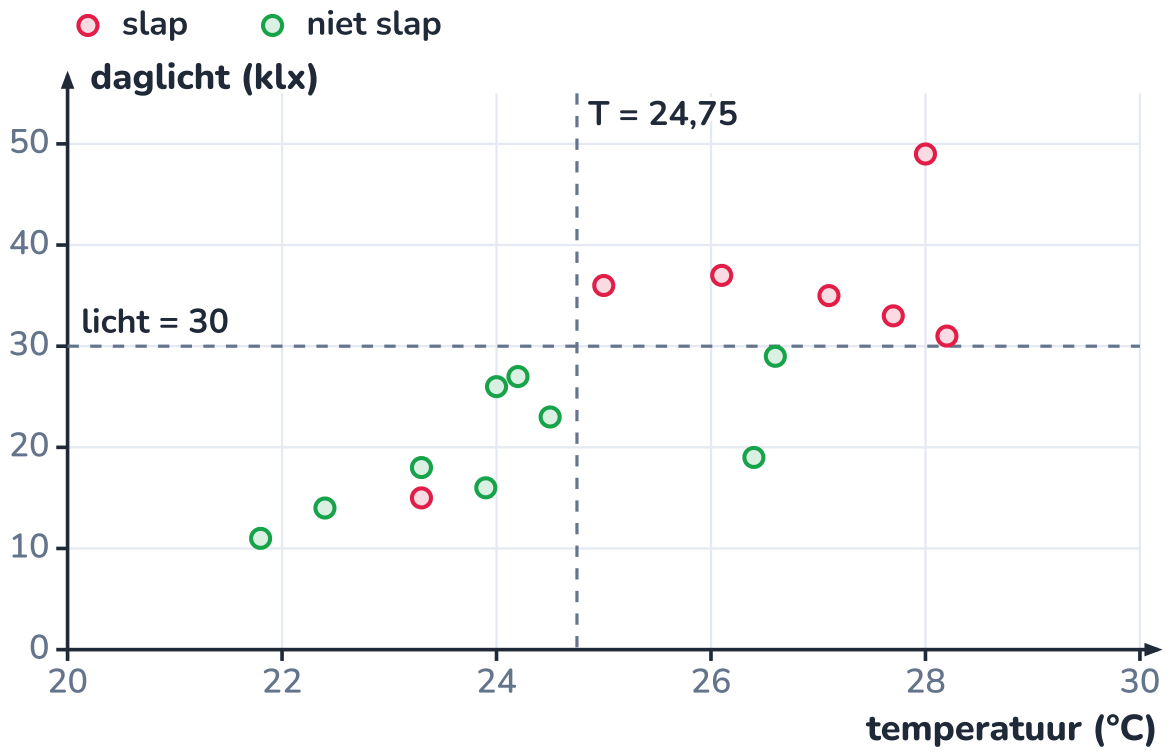
UITGEWERKT VOORBEELD · 39.13

Een beslisboom met de hand

13/23

Zestien inspecties van 10 tot 16 juni (tabel). Zoek de beste eerste vraag voor een boom, op daglicht of op temperatuur, door fouten te tellen. Toets de regel daarna op de zestien inspecties die volgen.

tijdstip	T (°C)	vocht (%)	licht (klx)	slap
10/6 14:00	28,0	52	49	ja
10/6 17:00	27,1	58	35	ja
11/6 11:00	24,0	71	26	nee
11/6 14:00	24,2	61	27	nee
11/6 17:00	23,9	68	16	nee
12/6 11:00	25,0	64	36	ja
12/6 14:00	26,1	60	37	ja
12/6 17:00	26,4	60	19	nee
13/6 11:00	27,7	57	33	ja
14/6 11:00	22,4	74	14	nee
14/6 17:00	21,8	68	11	nee
15/6 11:00	26,6	66	29	nee
15/6 14:00	28,2	47	31	ja
15/6 17:00	24,5	64	23	nee
16/6 11:00	23,3	73	18	nee
16/6 14:00	23,3	75	15	ja



Een horizontale lijn bij 30 klx scheidt de klassen beter dan een verticale bij de temperatuur.

GEGEVEN

- 16 inspecties met temperatuur, vocht, daglicht en het label
- een splitsing voorspelt in elke groep het label dat het meest voorkomt

GEVRAAGD

- de beste grens voor daglicht en voor temperatuur, met het aantal fouten
- de juistheid op de volgende 16

OPLOSSING

1 **Sorteer op daglicht** en probeer elke grens midden tussen twee waarden. Voorbeeld: grens 30 klx. Links (licht ≤ 30) liggen 10 inspecties, waarvan 1 slap: voorspel 'niet slap', 1 fout. Rechts 6, alle 6 slap: 0 fouten. Samen 1 fout. De andere grenzen:

GRENS LICHT (KLX)	FOUTEN LINKS + RECHTS
12,5	0 + 7 = 7
14,5	0 + 7 = 7
15,5	1 + 6 = 7
17,0	1 + 6 = 7
18,5	1 + 5 = 6
21,0	1 + 4 = 5
24,5	1 + 3 = 4
26,5	1 + 2 = 3
28,0	1 + 1 = 2
30,0	1 + 0 = 1
32,0	2 + 0 = 2
34,0	3 + 0 = 3
35,5	4 + 0 = 4
36,5	5 + 0 = 5
43,0	6 + 0 = 6

2 **Sorteer op temperatuur** en doe hetzelfde. De beste grens is 24,75 °C met 3 fouten: links 8 inspecties waarvan 1 slap, rechts 8 waarvan 6 slap.

3 **Kies** de vraag met de minste fouten: 'daglicht > 30 klx? dan slap'. De ene fout (16 juni om 14 uur: 23,3 °C, 75 %, 15 klx, toch slap genoteerd) kan geen enkele regel vermijden: misschien is het een vergissing bij het noteren.

4 **Toetsen** op de 16 inspecties van 16 tot 23 juni: de regel voorspelt er 13 juist, 81 %. Op de voorbeelden waaruit hij geleerd werd, was het 94 % (15 van 16).

ANTWOORD

Eerste vraag: daglicht ≤ 30 klx (1 fout op 16); op nieuwe inspecties 13 van 16 juist.

Klopt dit? Een regel scoort op zijn eigen leervoorbeelden meestal beter dan op nieuwe: daarom toets je op data die het model niet zag. En met de 154 inspecties van juni en juli kiest de computer ook daglicht als eerste vraag, met een grens van 29,5 klx: bijna dezelfde.

CODE · 39.14

De data: een zomer inspecties

14/23

Om een model te trainen, heb je voorbeelden met labels nodig. In de zomer van 2026 keek een bewoner van het station om 11, 14 en 17 uur naar de sla in de oudste goot en noteerde of de bladeren slap hingen; de logger gaf de temperatuur, de luchtvochtigheid en het daglicht op dat moment. Dit boek kan geen echte zomer meten, dus maakt het programma hieronder **testdata** met een model (zoals in hoofdstuk 30): de sla hangt slap als ze harder verdampt dan de wortels water kunnen opnemen. Hoe hard ze verdampt, hangt af van het **dampdruktekort** (hoe droog de lucht is voor die temperatuur) en van het licht.

Python **maak_slapdata.py** 33 regels

```python
# maak_slapdata.py - een zomer inspecties van de sla in de klimaatmodule (een model, geen echte metingen)
import math
import random

rng = random.Random(1)                                  # vast zaadje: iedereen krijgt dezelfde data


def dampdruk(t):
    """Verzadigde dampdruk van water (kPa) bij t graden Celsius (formule van Tetens)."""
    return 0.6108 * math.exp(17.27 * t / (t + 237.3))


regels = ["tijdstip,temperatuur_C,vocht_pct,licht_klx,bewoner,slap"]
for dag in range(92):                                   # 1 juni tot en met 31 augustus 2026
    maand, nr = (6, dag + 1) if dag < 30 else (7, dag - 29) if dag < 61 else (8, dag - 60)
    w = min(1.0, max(0.0, rng.gauss(0.55, 0.3)))        # hoe helder de dag is (0 tot 1)
    t_buiten = 17 + 11 * w + rng.gauss(0, 2) + (2 if 30 <= dag < 75 else 0)
    bewoner = "A" if (dag // 7) % 2 == 0 else "B"       # om de week kijkt een andere bewoner
    for uur in (11, 14, 17):
        if dag % 7 in (5, 6) and rng.random() < 0.7:
            continue                                    # in het weekend meestal niemand
        zon = max(0.0, math.sin(math.pi * (uur - 5.5) / 15.0)) * w
        t = 18 + (t_buiten - 17) * 0.6 + 4 * zon + rng.gauss(0, 1.0)
        rv = min(95.0, max(30.0, 82 - 1.8 * (t - 18) - 12 * zon + rng.gauss(0, 5)))
        licht = max(0.0, 55 * zon + rng.gauss(0, 3))    # daglicht op de lichtsensor (klx)
        vraag = dampdruk(t) * (1 - rv / 100) * (licht + 15) / 45   # hoe hard de plant verdampt (model)
        slap = vraag > (1.6 if bewoner == "A" else 1.25)   # B noemt de sla sneller slap
        if rng.random() < 0.05:
            slap = not slap                             # een vergissing bij het noteren
        regels.append(f"2026-{maand:02d}-{nr:02d} {uur}:00,{t:.1f},{rv:.0f},{licht:.0f},{bewoner},{int(slap)}")
with open("slap.csv", "w", newline="") as f:
    f.write("\n".join(regels) + "\n")
print(len(regels) - 1, "inspecties in slap.csv")
```

Uitvoer, en het begin van slap.csv:

```
228 inspecties in slap.csv

tijdstip,temperatuur_C,vocht_pct,licht_klx,bewoner,slap
2026-06-01 11:00,29.4,47,44,A,1
2026-06-01 14:00,29.6,51,49,A,0
2026-06-01 17:00,29.0,50,34,A,1
…
```

228 inspecties, waarvan 30 % slap. Drie dingen in dit model komen terug: om de week kijkt een andere bewoner, en bewoner B noemt de sla sneller slap dan A; in het weekend is er meestal niemand; en 5 % van de labels is een vergissing. Echte data hebben zulke eigenaardigheden altijd. Het model weet daar niets van: het krijgt alleen de kolommen.

OOK TE DOWNLOADEN

CODE · 39.15

beslisboom.py: trainen en toetsen

15/23

De boom van nul af, in een zeventigtal regels. `beste_splitsing()` probeert elke vraag en houdt de zuiverste. `bouw()` splitst en roept zichzelf op voor elke helft (een functie die zichzelf oproept, heet **recursief**: Expert, hoofdstuk 34), tot de diepte op is of een splitsing niet meer helpt. Een vraag met twee gelijke antwoorden wordt geschrapt. De boom is een woordenboek in een woordenboek. Getraind wordt op juni en juli, getoetst op augustus: data die het model nooit zag, en die *na* de trainingsdata komen, zoals in het echt.

Python **beslisboom.py** 76 regels

```python
# beslisboom.py - een beslisboom van nul af, getraind op juni en juli, getoetst op augustus
import csv

KENMERKEN = ["temperatuur_C", "vocht_pct", "licht_klx"]


def lees(naam):
    with open(naam, newline="") as f:
        return [{"x": [float(r[k]) for k in KENMERKEN], "y": int(r["slap"]), "tijd": r["tijdstip"]}
                for r in csv.DictReader(f)]


def onzuiverheid(rijen):
    """2 p (1 - p), met p het deel 'slap': 0 als alle labels gelijk zijn, 0,5 bij half om half."""
    p = sum(r["y"] for r in rijen) / len(rijen)
    return 2 * p * (1 - p)


def beste_splitsing(rijen, min_blad):
    """(onzuiverheid, kenmerk, grens) van de beste vraag 'kenmerk <= grens?', of None als er geen is."""
    beste = None
    for k in range(len(KENMERKEN)):
        waarden = sorted(set(r["x"][k] for r in rijen))
        for a, b in zip(waarden, waarden[1:]):
            grens = (a + b) / 2                          # midden tussen twee gemeten waarden
            ja = [r for r in rijen if r["x"][k] <= grens]
            nee = [r for r in rijen if r["x"][k] > grens]
            if len(ja) < min_blad or len(nee) < min_blad:
                continue
            o = (len(ja) * onzuiverheid(ja) + len(nee) * onzuiverheid(nee)) / len(rijen)
            if beste is None or o < beste[0]:
                beste = (o, k, grens)
    return beste


def bouw(rijen, diepte, min_blad=5):
```

```
    """Een boom als woordenboek: een blad {'label', 'n'} of een vraag met een ja- en een nee-tak."""
    s = beste_splitsing(rijen, min_blad)
    if diepte == 0 or s is None or s[0] >= onzuiverheid(rijen):
        ja = sum(r["y"] for r in rijen)
        return {"label": int(ja > len(rijen) - ja), "n": len(rijen)}
    _, k, grens = s
    ja = bouw([r for r in rijen if r["x"][k] <= grens], diepte - 1, min_blad)     # de functie roept zichzelf op
    nee = bouw([r for r in rijen if r["x"][k] > grens], diepte - 1, min_blad)
    if "label" in ja and "label" in nee and ja["label"] == nee["label"]:
        return {"label": ja["label"], "n": len(rijen)}      # twee gelijke bladen: de vraag is overbodig
    return {"kenmerk": k, "grens": grens, "ja": ja, "nee": nee}


def voorspel(boom, x):
    while "label" not in boom:
        boom = boom["ja"] if x[boom["kenmerk"]] <= boom["grens"] else boom["nee"]
    return boom["label"]


def toon(boom, inspring=""):
    if "label" in boom:
        print(inspring + ("slap" if boom["label"] else "niet slap"), f"({boom['n']} inspecties)")
    else:
        print(inspring + f"{KENMERKEN[boom['kenmerk']]} <= {boom['grens']:.1f}?")
        toon(boom["ja"], inspring + "  ja:  ")
        toon(boom["nee"], inspring + "  nee: ")


def juist(boom, rijen):
    return sum(voorspel(boom, r["x"]) == r["y"] for r in rijen) / len(rijen)


rijen = lees("slap.csv")
train = [r for r in rijen if r["tijd"] < "2026-08"]         # juni en juli: leren
test = [r for r in rijen if r["tijd"] >= "2026-08"]         # augustus: toetsen
if __name__ == "__main__":
    boom = bouw(train, diepte=2)
    toon(boom)
    print(f"train: {len(train)} inspecties, {juist(boom, train):.0%} juist")
    print(f"test:  {len(test)} inspecties, {juist(boom, test):.0%} juist")
```

Uitvoer:

```
licht_klx <= 29.5?
  ja:  niet slap (103 inspecties)
  nee: vocht_pct <= 60.5?
  nee:   ja:  slap (37 inspecties)
  nee:   nee: niet slap (14 inspecties)
train: 154 inspecties, 90% juist
test:  74 inspecties, 82% juist
```

Op de trainingsdata 90 % juist, op augustus 82 %. Dat verschil is normaal: het model leerde ook een stukje van de toevalligheden van juni en juli. Professionele bibliotheken zoals scikit-learn doen hetzelfde, sneller en met meer opties; wie de boom zelf schreef, weet wat er in die bibliotheek gebeurt.

UITLEG · 39.16

Trainen en testen: hoe goed is het model?

16/23

Een model beoordeel je altijd op data die het niet zag: de **testdata**. Splits bij metingen in de tijd op een datum (leren op het verleden, toetsen op de toekomst), niet willekeurig: anders test je met uren die vlak naast een trainingsuur liggen en bijna hetzelfde zijn.

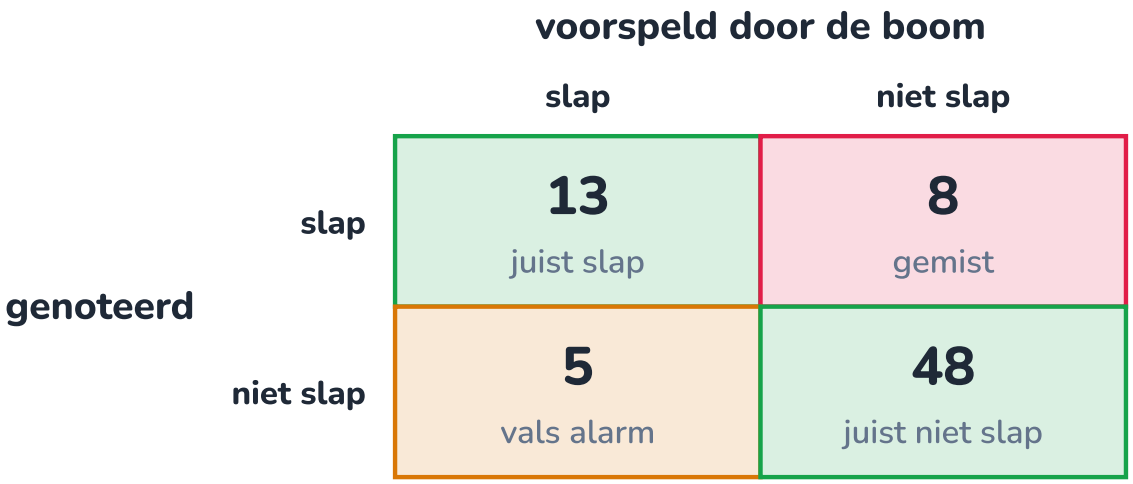
De **juistheid** (het deel juiste voorspellingen) zegt niet alles. In augustus hing de sla bij 53 van de 74 inspecties niet slap: een 'model' dat *altijd* 'niet slap' zegt, is al 72 % juist. Dat is de **basislijn**. De boom haalt 82 %: beter, maar minder indrukwekkend dan het getal alleen doet denken.

Kijk daarom naar de soorten fouten, in een **verwarringsmatrix**. De boom miste 8 keer slappe sla (**gemist**) en voorspelde 5 keer slap terwijl het niet zo was (**vals alarm**). Welke fout erger is, hangt af van wat je met de voorspelling doet. Voor een extra pompbeurt is een vals alarm goedkoop (5 minuten pompen is 1 Wh), een gemiste slappe sla niet.

JUISTHEID

$$\text{juistheid} = \frac{\text{juist slap} + \text{juist niet slap}}{\text{alle voorspellingen}}$$

Augustus: $(13 + 48)/74 = 0{,}82$.



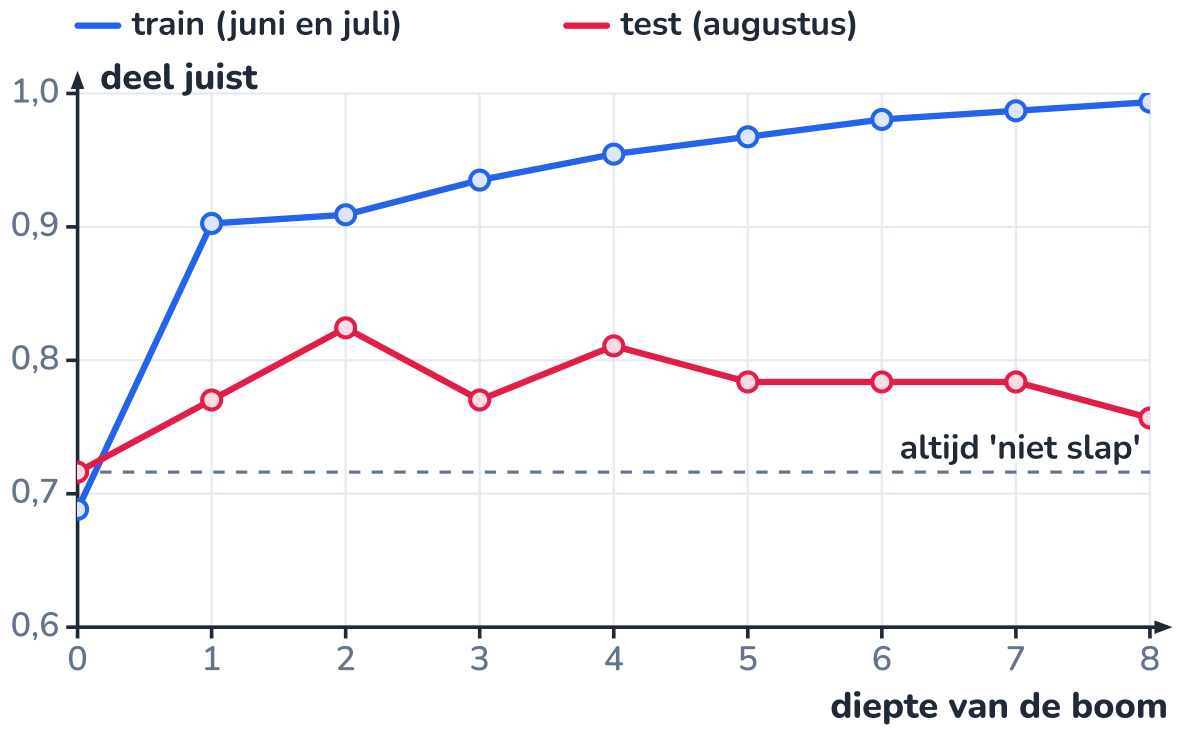
Groen: juist. Rood: gemiste slappe sla. Geel: vals alarm.

UITGEWERKT VOORBEELD · 39.17

Hoe diep mag de boom?

17/23

Een diepere boom stelt meer vragen en past de trainingsdata beter. Is dat ook beter voor augustus? Train bomen van diepte 0 tot 8, waarbij een blad mag bestaan uit één inspectie, en vergelijk.



Blauw leert steeds beter; rood (nieuwe data) is het best bij diepte 2.

GEGEVEN

- beslisboom.py als module
- min_blad = 1

GEVRAAGD

- de juistheid op train en test per diepte, en de beste diepte

OPLOSSING

1 **Programma**, dat de functies uit beslisboom.py importeert:

```
# diepte.py - hoe diep mag de boom? (min_blad = 1: de boom mag tot op één inspectie splitsen)
from beslisboom import bouw, juist, train, test

print("diepte  train   test")
for diepte in range(0, 9):
    boom = bouw(train, diepte, min_blad=1)
    print(f"{diepte:5d}   {juist(boom, train):4.0%}   {juist(boom, test):4.0%}")
```

```
diepte  train   test
    0    69%    72%
    1    90%    77%
    2    91%    82%
    3    94%    77%
    4    95%    81%
    5    97%    78%
    6    98%    78%
    7    99%    78%
    8    99%    76%
```

2 **Train** stijgt tot bijna 100 %: een diepe boom stelt zoveel vragen dat hij bijna elke trainingsinspectie apart kan zetten, ook de 5 % vergissingen.

3 **Test** is het best bij diepte 2 (82 %) en blijft daarna lager, met wat geschommel: de extra vragen leerden de toevalligheden van juni en juli, niet de sla. Dat heet **overfitting**.

ANTWOORD

Diepte 2: twee vragen, 82 % juist in augustus. Dieper is slechter voor nieuwe data.

Klopt dit? Een boom van diepte 8 is een lange lijst uitzonderingen; een boom van diepte 2 kan je uitleggen aan een bewoner. Kies de eenvoudigste die goed genoeg is. De juiste diepte zoek je op aparte data (validatie), anders gebruik je de testdata om te kiezen en is de test niet meer eerlijk (Expert, hoofdstuk 41).

UITLEG · 39.18

Bias in data

18/23

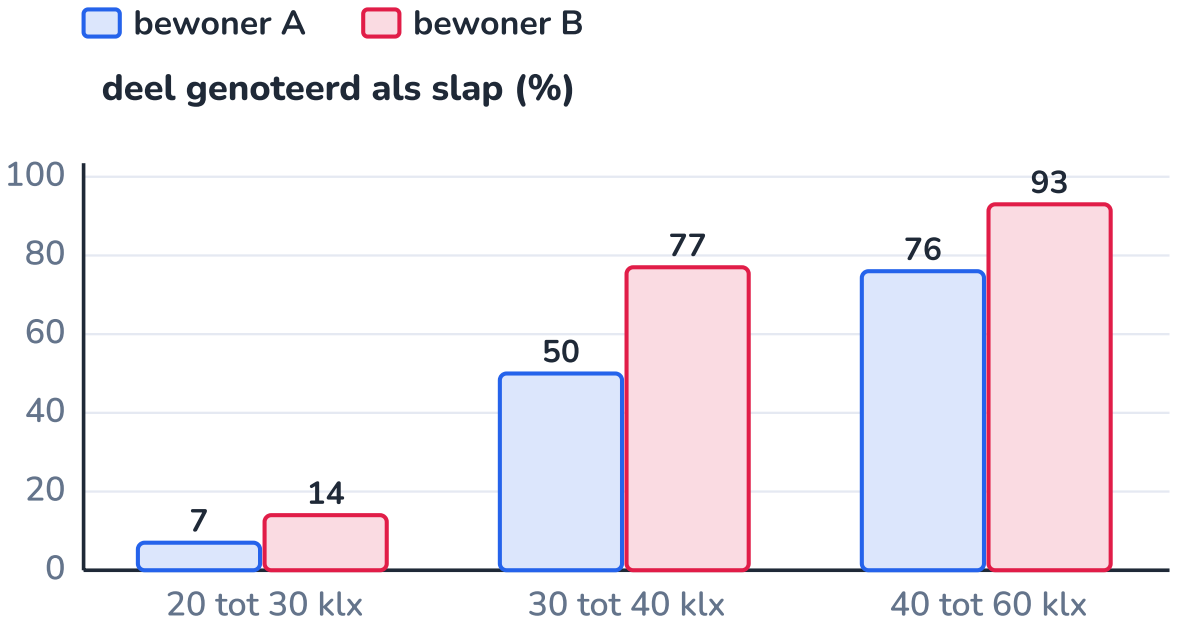
Een model leert uit de voorbeelden die het krijgt, met al hun scheefheden. Dat heet **bias in data**. In de inspecties van de zomer zitten er drie:

- **Wie labelt.** Bewoner A noteerde in 25 % van de inspecties 'slap', bewoner B in 36 %. Was het warmer in de weken van B? Vergelijk bij hetzelfde daglicht (figuur): tussen 30 en 40 klx noteerde A 50 % slap, B 77 %. B noemt de sla sneller slap. Het model leert een gemiddelde van twee meningen, en wordt strenger als er meer weken van B in de data zitten.
- **Wanneer gemeten.** Alleen om 11, 14 en 17 uur, vooral op weekdagen. Over de ochtend, de avond en de nacht weet het model niets. Gebruik je het om 7 uur, dan voorspelt het iets over een situatie die het nooit zag.
- **Welke periode.** Alleen de zomer, met één soort sla in de oudste goot. In oktober, met jonge plantjes of een ander ras, kan het verband anders zijn.

Wat helpt: vooraf afspreken wat 'slap' is (een foto met een voorbeeld, of de hoek van een blad meten), labels blind laten controleren door een tweede persoon, meten op alle uren en alle dagen, en noteren wie, wanneer en hoe gemeten werd, zodat je bias kan opsporen.

Astma die beschermt?

In de jaren 1990 trainden onderzoekers een model dat voorspelde welke patiënten met een longontsteking een hoog risico liepen om te overlijden. Het model leerde iets vreemds: astma verlaagde het risico. De data klopten: astmapatiënten met een longontsteking gingen meteen naar intensieve zorg, en overleefden daardoor vaker. Het model zag de behandeling niet, alleen de uitkomst. Had men het gebruikt om patiënten naar huis te sturen, dan waren juist de kwetsbaarste naar huis gegaan. Omdat het model leesbare regels had (zoals een beslisboom), zag men de fout. R. Caruana en collega's beschreven het geval in 2015.



Bij hetzelfde daglicht noteert B vaker 'slap': het label hangt af van wie kijkt.

PROEF · 39.19

Een beslisboom voor het klaslokaal

19/23

ONDERZOEKSVRAAG

Kan een beslisboom uit de metingen van een datalogger voorspellen of er een raam openstaat?

MATERIAAL

- de datalogger van hoofdstuk 30 (SHT31, klok, SD-kaart), op een kast in het lokaal
- een blad om bij te houden wanneer een raam open en dicht gaat
- Python met beslisboom.py

De labels komen van een leerling met een klok en een blad: zo begint elk leerproject.

WERKWIJZE

1. Laat de logger twee schooldagen lang elke minuut meten. Noteer op het blad elk tijdstip waarop een raam open- of dichtgaat: dat zijn de labels.
2. Maak in Python per minuut twee kenmerken: de verandering van de temperatuur over de laatste 5 minuten, en die van de luchtvochtigheid. Voeg het label toe (1 = een raam staat open).
3. Train een boom van diepte 1 en 2 op de eerste dag (pas KENMERKEN in beslisboom.py aan), en toets op de tweede dag.
4. Maak een verwarringsmatrix. Wanneer vergist de boom zich?

WAARNEMING

Typisch: de eerste vraag gaat over de temperatuurverandering (bv. 'daalt de temperatuur meer dan 0,3 °C in 5 minuten?'). De boom vindt een open raam in de winter vaak, maar mist het in warm weer (buiten even warm als binnen) en geeft vals alarm als de klas vertrekt of de verwarming uitvalt.

VERKLARING

Een open raam laat in de winter koude, droge lucht binnen: temperatuur en vochtigheid dalen snel. De boom leert die 'handtekening'. Hij leert niet wat een raam is: als het buiten even warm is, is er geen handtekening, en een andere oorzaak van afkoeling (een lege klas) lijkt voor hem op een open raam. Een model is zo goed als de kenmerken die je het geeft, en de periode waarin je leerde.

CODE · 39.20

Van boom naar klimaat.ino

20/23

Een getrainde beslisboom is na het trainen gewoon een paar `if`-vragen. Die passen in de Arduino Mega van de klimaatmodule, zonder bibliotheek en zonder internet. De functie hieronder is de boom van dit hoofdstuk in C++. Getest met g++ op de 74 inspecties van augustus (het testprogramma zit bij de bijlagen): dezelfde uitkomst als in Python.

C++ (Arduino) **slap.h** 11 regels

```cpp
// slap.h - de beslisboom van hoofdstuk 39, getraind op de inspecties van juni en juli 2026.
// licht_klx: daglicht op de lichtsensor (klx); vocht_pct: relatieve vochtigheid van de lucht (SHT31).
bool slapVerwacht(float licht_klx, float vocht_pct) {
  if (licht_klx <= 29.5) {
    return false;                 // weinig licht: de sla verdampt weinig
  }
  if (vocht_pct <= 60.5) {
    return true;                  // veel licht en droge lucht: slap verwacht
  }
  return false;                   // veel licht, maar vochtige lucht
}
```

Uitvoer van slap_test.cpp:

```
2026-08-02 11:00  licht   47 klx  vocht  44 %  voorspeld slap      genoteerd slap
2026-08-02 14:00  licht   51 klx  vocht  61 %  voorspeld niet slap genoteerd slap
2026-08-02 17:00  licht   33 klx  vocht  50 %  voorspeld slap      genoteerd slap
2026-08-03 11:00  licht   15 klx  vocht  61 %  voorspeld niet slap genoteerd niet slap
2026-08-03 14:00  licht   13 klx  vocht  75 %  voorspeld niet slap genoteerd niet slap
2026-08-03 17:00  licht   13 klx  vocht  79 %  voorspeld niet slap genoteerd niet slap
augustus: 61 van 74 juist
```

In klimaat.ino wordt het een taak (hoofdstuk 29): om het halve uur kijkt ze of er slappe sla verwacht wordt, en laat dan de pomp 5 minuten extra draaien. De lichtgrens is bewust lager gezet dan in de boom (25 in plaats van 29,5 klx): liever een vals alarm (1 Wh) dan een gemiste slappe sla.

```cpp
// in klimaat.ino (versie 4, hoofdstuk 32; in versie 5 van hoofdstuk 40 stuurt één taak de pomp): om het halve uur een extra p
ompbeurt van 5 minuten als de sla slap dreigt te worden
const float LICHT_GRENS = 25.0;          // klx: lager dan de 29,5 van de boom, liever vals alarm dan gemist
const float VOCHT_GRENS = 60.5;          // % relatieve vochtigheid: de tweede vraag van de boom
unsigned long extraPompTot = 0;          // millis() waarop de extra pompbeurt stopt (0 = geen)

bool slapMetMarge(float licht_klx, float vocht_pct) {
  return licht_klx > LICHT_GRENS && vocht_pct <= VOCHT_GRENS;
}

void taakExtraPomp(int minuut, float lichtLux, float vocht) {      // om de 2 s opgeroepen, zoals de meettaak
  if (minuut == 30 && extraPompTot == 0 && slapMetMarge(lichtLux / 1000.0, vocht)) {
    extraPompTot = millis() + 5UL * 60UL * 1000UL;
    Serial.println(F("# extra pompbeurt: slappe sla verwacht"));
  }
  if (extraPompTot != 0) {
    digitalWrite(POMP, (long)(millis() - extraPompTot) < 0 ? HIGH : LOW);
    if ((long)(millis() - extraPompTot) >= 0) extraPompTot = 0;
  }
}
```

> ⚠ **De boom vervangt de vlotter niet**
>
> De extra pompbeurt is een aanvulling op de pomp van minuut 0 tot 9, niet een vervanging. De beveiligingen (vlotter tegen droogdraaien, alarm bij een laag reservoir, hoofdstuk 32) blijven gewone regels die altijd voorgaan. Een geleerd model mag adviseren; een veiligheidsregel beslist.

OOK TE DOWNLOADEN

UITLEG · 39.21

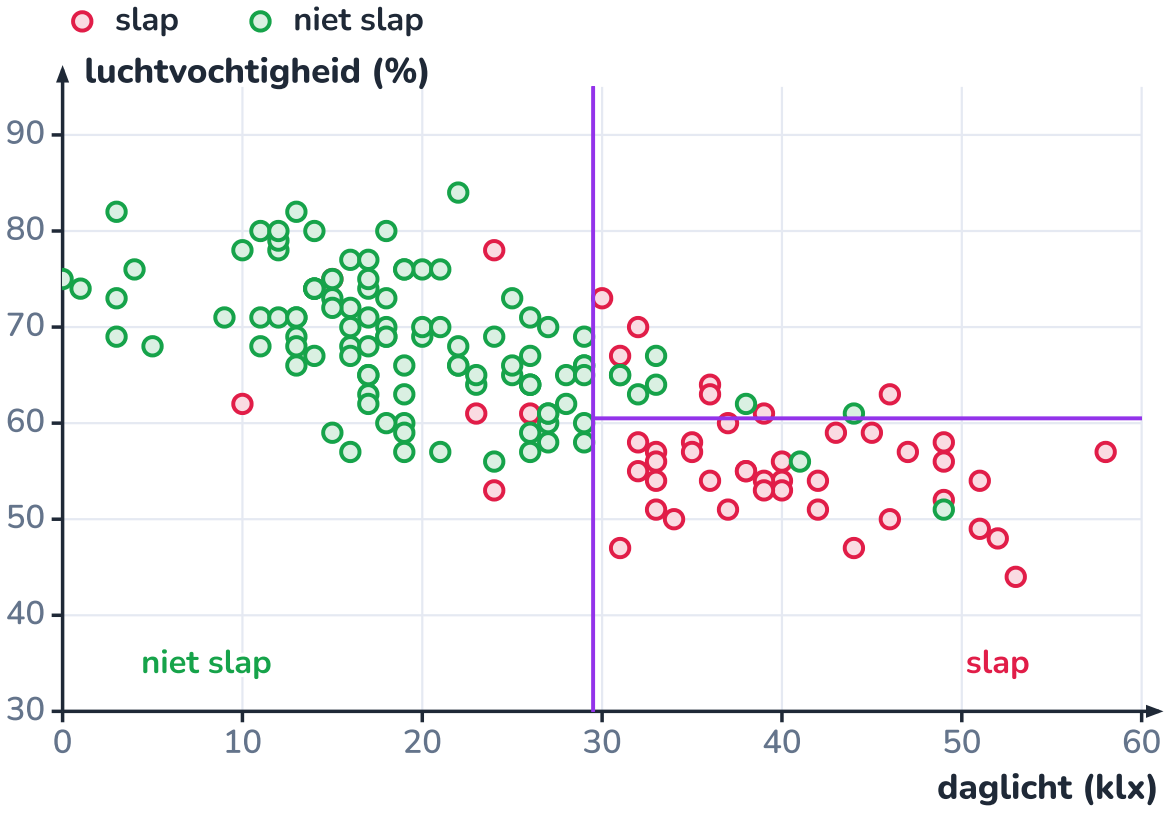
Wat een model niet kan, en wie beslist

21/23

De boom van dit hoofdstuk voorspelt slappe sla in ongeveer vier van de vijf gevallen. Daar houdt het op. Wat hij niet kan:

- **Iets herkennen wat hij nooit zag.** Valt de pomp uit, dan hangt alle sla slap, ook bij weinig licht. De boom zegt 'niet slap': in de data was er nooit een pompstoring. Daarvoor dienen sensoren en gewone regels.
- **Uitleggen waarom.** Hij weet dat veel licht en droge lucht samengaan met slappe sla, niet dat de bladeren water verliezen. Dat de regel natuurkundig klopt (dampdruktekort), is geluk en goede kenmerken.
- **Beter zijn dan zijn labels.** Als bewoners zich vergissen of anders oordelen, leert de boom dat mee.
- **Zeker zijn.** Elke voorspelling is een gok met een foutkans; die moet je kennen en aanvaarden.

Wie laat een model beslissen, en over wat? Een extra pompbeurt is goedkoop en onschuldig: die mag de module zelf nemen. Een beslissing over mensen (wie mag binnen, wie krijgt zorg) is iets anders. Stel bij elk model vier vragen: *Wat gebeurt er bij een fout, en voor wie? Kan een mens de beslissing zien, begrijpen en terugdraaien? Op welke data is het getraind, en past dat bij waar het gebruikt wordt? Wie is verantwoordelijk?* De Europese AI-verordening (in werking sinds 1 augustus 2024) legt strengere regels op naarmate een AI-systeem meer risico inhoudt voor mensen. In Expert (hoofdstuk 44) maak je er een afwegingskader van voor het brein van het station.



Elke vraag van de boom is een rechte grens: de boom kent alleen rechthoekige gebieden.

Grafieken, correlatie en beslisbomen

correlatiecoëfficiënt

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

symbool	betekenis	eenheid
r	van −1 tot +1	–

liegfactor (eerlijk: 1)

$$\text{liegfactor} = \frac{\text{effect in de grafiek}}{\text{effect in de data}}$$

Gini-onzuiverheid (twee klassen)

$$\text{onzuiverheid} = 2\,p\,(1-p)$$

symbool	betekenis	eenheid
p	deel met label 'slap'	–

juistheid; vergelijk met de basislijn

$$\text{juistheid} = \frac{\text{juist}}{\text{alle}}$$

vraag	grafiek	voorbeeld uit de module
hoe verandert iets in de tijd?	lijngrafiek	temperatuur per uur
hoe verhouden aparte groepen zich?	staafdiagram	verbruik en zon per dag
hangen twee grootheden samen?	spreidingsdiagram	verbruik tegenover daglicht
hoe zijn metingen verdeeld?	histogram	hoe vaak elke temperatuur voorkwam
welk deel van een geheel?	taartdiagram (weinig delen) of gestapelde staaf	het energiebudget: 93 % lampen

truc	wat het doet	tegengif
afgeknotte as bij staven	kleine verschillen lijken groot	staven altijd vanaf 0
twee y-assen	elke gewenste 'samenhang' is te maken door de assen te kiezen	twee grafieken onder elkaar, of één as
kersen plukken	alleen de periode tonen die het verhaal steunt	de hele reeks, of zeggen waarom een stuk
oppervlakte of volume als pictogram	een verdubbeling lijkt een vervierfoudiging	lengte evenredig met de waarde (staaf)
gaten verbinden	een lijn over ontbrekende metingen verzint data	de lijn onderbreken
geen eenheid, geen bron	niemand kan nagaan wat er getoond wordt	grootheid, eenheid, bron en periode

begrip	in dit hoofdstuk
kenmerken	temperatuur_C, vocht_pct, licht_klx
label	slap (1) of niet slap (0)
trainingsdata	juni en juli 2026 (154 inspecties)
testdata	augustus 2026 (74 inspecties)
model	beslisboom van diepte 2
basislijn	altijd 'niet slap': 72 % juist in augustus
gemist / vals alarm	slap maar niet voorspeld / voorspeld maar niet slap
overfitting	train veel beter dan test; een diepere boom helpt dan niet

HET GEHEEL · 39.23

Waar data en AI vandaan komen en naartoe gaan

23/23

Dit hoofdstuk sluit de leerlijn data en AI van Intermediate af. Grafieken en correlatie bouwen op Fundamental en op hoofdstuk 3 en 28; de logger en de CSV komen uit hoofdstuk 30; de beslisboom is een programma (P) dat uit metingen (O) leert. In Expert wordt het wiskundiger: regressie en kleinste kwadraten, classificatie met meer kenmerken, neurale netwerken, en een afwegingskader voor AI.

F4 Tabellen, grafieken en verbanden

Tabel, grafiek, schaal en de keuze van het grafiektype.

F22 Computationeel denken

De eerste kennismaking met systemen die leren uit voorbeelden.

I2 Experimenten ontwerpen en rapporteren

Een oorzaak bewijs je met een eerlijke proef, niet met een correlatie.

I3 Verbanden in data

Trendlijnen en de richtingscoëfficiënt: verbanden in data.

I28 Functies, lijsten en data in Python

CSV, woordenboeken, matplotlib en de correlatiecoëfficiënt.

I30 Datalogging en communicatie

De logger, logboek.py en de testweek die hier per uur samengevat wordt.

I32 Toestandsmachines

Veilige toestanden en alarmen: regels die voorgaan op een geleerd model.

I40 Integratieproject: de klimaatmodule

Data-analyse in Python en de extra pompbeurt in het technisch rapport van de module.

E6 Statistiek en meetonzekerheid

Hoe zeker is 82 %? Betrouwbaarheidsintervallen en toetsen.

E41 Van data naar model: regressie

Een getal voorspellen (het verbruik van morgen), overfitting en kruisvalidatie.

E42 Classificatie en clustering

Beslisbomen, k-dichtstbijzijnde buren, precisie en recall, onevenwichtige data.

E43 Neurale netwerken

Modellen die ingewikkeldere verbanden leren, en waarom ze moeilijker uit te leggen zijn.

E44 AI verantwoord inzetten

Bias, uitlegbaarheid, privacy en de Europese AI-verordening: mag het station zelf beslissen?

Habitat

De klimaatmodule heeft nu een analyse van haar logboek per uur (week_uur.csv) en een eerste lerend onderdeel: een beslisboom (daglicht ≤ 29,5 klx: niet slap; anders vocht ≤ 60,5 %: slap), 82 % juist op nieuwe data tegenover 72 % voor de basislijn. In klimaat.ino wordt hij een extra pompbeurt van 5 minuten om het halve uur, met een lagere lichtgrens (25 klx) en altijd onder de veiligheidsregels. Opdracht 39.14 legt de keuze vast voor het integratieproject (hoofdstuk 40).

SAMENVATTING

In het kort

- Kies de grafiek bij de vraag: lijn voor de tijd, staven voor groepen (vanaf 0), punten voor een verband, een histogram voor een verdeling.
- Misleiding zit vaak in de keuze: een afgeknotte as, twee assen, een gekozen periode, een pictogram dat in oppervlakte groeit (liegfactor), een lijn over een gat.
- Correlatie is geen oorzaak: zoek een gemeenschappelijke oorzaak door haar constant te houden, en denk aan de omgekeerde richting en aan toeval.

- Machine learning leert een regel (een model) uit voorbeelden met kenmerken en labels; het begrijpt niets en kent alleen wat in de data zat.
- Een beslisboom stelt ja/nee-vragen; de computer kiest telkens de vraag die de zuiverste groepen geeft (onzuiverheid $2p(1-p)$).
- Toets op data die het model niet zag, vergelijk met de basislijn en kijk naar de soorten fouten; een te diepe boom overfit.
- Bias in data (wie labelt, wanneer, welke periode) neemt het model over. Een model mag adviseren; veiligheidsregels beslissen.

BEGRIPPEN

Wat je nu kent

spreidingsdiagram
Grafiek met een punt per meting, met twee grootheden op de assen; toont of ze samenhangen.

histogram
Staafdiagram van hoe vaak metingen in elke klasse (bv. elke graad) vallen: de verdeling.

liegfactor
De grootte van een effect in de grafiek gedeeld door de grootte van het effect in de data (Tufte); eerlijk is 1.

gemeenschappelijke oorzaak
Een derde grootheid die twee andere tegelijk beïnvloedt, zodat die samen bewegen zonder elkaar te veroorzaken.

machine learning
Een computer leert regels uit voorbeelden (gegevens met de juiste antwoorden) in plaats van dat een mens ze programmeert.

kenmerk
Een gemeten grootheid die het model als invoer gebruikt, bv. temperatuur, vocht of daglicht.

label
Het juiste antwoord bij een voorbeeld, bv. 'slap' of 'niet slap'.

beslisboom
Model dat een voorspelling maakt met een reeks ja/nee-vragen over de kenmerken; de vragen en grenzen worden geleerd uit data.

onzuiverheid
Maat voor hoe gemengd de labels in een groep zijn: $2p(1-p)$, 0 als alles gelijk is, 0,5 bij half om half.

verwarringsmatrix
Tabel die telt hoe vaak een model juist en fout voorspelt, per soort fout (gemist, vals alarm).

overfitting
Een model dat de trainingsdata (met hun toevallige ruis) uit het hoofd leert en daardoor slechter voorspelt op nieuwe data.

bias in data
Een scheefheid in de gegevens (wie ze verzamelde, wanneer, hoe gelabeld) die het model overneemt.

HANDBOEK · HOOFDSTUK 40

Integratieproject: de klimaatmodule

Ontwerpen, dimensioneren, programmeren, testen en verdedigen: je team brengt 39 hoofdstukken samen in de tweede laag van Habitat

Negenendertig hoofdstukken lang leerde je telkens één stuk van de klimaat- en energiemodule berekenen: het warmteverlies, de panelen, de pomp, de pH, een programma met modi. In dit integratieproject zet je team ze samen tot één module die zichzelf van energie, water, warmte en licht voorziet, en die elke minuut noteert wat ze doet. Je legt eerst de lat (het programma van eisen), rekent elke balans na, voegt vijf versies van de code samen tot één programma, laat de module een week alleen draaien, analyseert de logbestanden in Python en verdedigt het resultaat voor een jury. Het handboek is je **projectgids**, het werkboek je **projectmap**.

12× uitleg · 4× uitgewerkt voorbeeld · 1× proef · 1× ontleed · 6× naslag · 1× verhaal · 1× het geheel · 2× code · ± 36 lestijden

D Ontwerpen en maken · E Energie · L Elektriciteit en elektronica · R Regelen en robotica · B Leven en systemen · O Onderzoeken en meten

De klimaat- en energiemodule af: vier goten sla, eigen zonnestroom, en een logger die alles bijhoudt.

Na dit hoofdstuk

- Je kan een project van twaalf weken plannen met een team, met vaste eigenaars per deelsysteem, mijlpalen, een logboek en een integrator.
- Je kan een programma van eisen voor een samengesteld systeem opstellen: meetbaar, met een vooraf vastgelegd criterium en een toets per eis, en met verwijzing naar de berekening waar de eis vandaan komt.
- Je kan de interfaces tussen deelsystemen benoemen en vastleggen (N^2-diagram, interfacetabel) en integratieproblemen opsporen door budget, schema en code naast elkaar te leggen.
- Je kan de energie-, water- en warmtebalans van de module narekenen met de waarden uit de vorige hoofdstukken, per dag, per maand en per jaar.
- Je kan meetketens van sensor tot getal narekenen, sensoren in twee punten ijken en de voedingsoplossing bijsturen.
- Je kan een programma van meer dan 500 regels lezen dat modi, regelingen, energiebeheer en een logger samenbrengt, en het uitbreiden en testen met een naspeeltabel.
- Je kan een testplan opstellen met deeltests, storingstests en een testweek, en de logbestanden in Python toetsen aan het programma van eisen.
- Je kan systematisch foutzoeken met de logbestanden en meldingen: van symptoom naar oorzaak, met een test die telkens één ding controleert.
- Je kan een technisch rapport schrijven en je ontwerp verdedigen voor een jury, en je eigen werk en dat van je teamgenoten eerlijk beoordelen.

VERHAAL · 40.1

Twee filters die niet in elkaar pasten

1/28

Op 13 april 1970, twee dagen na de lancering van Apollo 13, ontplofte een zuurstoftank in de servicemodule. De drie astronauten moesten de reis rond de maan en terug maken in de maanlander, die gebouwd was voor twee mensen en een kort verblijf. Daar liep al snel een tweede probleem op: de koolstofdioxide die ze uitademden, stapelde zich op. De maanlander had te weinig patronen met lithiumhydroxide om CO_2 te binden. In de commandomodule lagen er genoeg, maar die waren vierkant, en de houder in de maanlander was rond.

Twee deelsystemen die elk perfect werkten, gebouwd door verschillende teams, pasten niet op elkaar. In Houston zochten ingenieurs een oplossing met alleen wat er aan boord was: plastic zakken, karton, slangen van de ruimtepakken en tape. Ze bouwden de adapter eerst zelf, testten hem, en lazen de bemanning daarna de bouwinstructies voor over de radio. De CO_2 daalde weer, en op 17 april landde de capsule veilig.

Dat is **integratie**: nagaan of delen die elk werken, ook samen werken. De plaats waar ze elkaar raken, heet een **interface**: een maat, een stekker, een pin, een eenheid, een tijdzone. In de meetkas (Fundamental, hoofdstuk 33) had je er tientallen. De klimaat- en energiemodule heeft er meer, en ze zijn lastiger: een stroommeter die tot 20 A meet naast panelen die met 74 A laden, twee taken die dezelfde pomp schakelen, een energiebudget in UTC en een klok op UTC+1. In dit hoofdstuk zoek je ze op, voor ze in een testweek of in een ruimtevaartverslag opduiken.

Zo werk je met dit hoofdstuk

Het handboek is je **projectgids**: per fase wat je doet en waarom, met de waarden uit de vorige hoofdstukken, uitgewerkte berekeningen en de resultaten van een voorbeeldteam (Emma, Jonas, Sara en Wout). Het werkboek is je **projectmap**: per fase de opdrachten, met ruimte om in te vullen en een rubric. Hoofdstuknummers zonder letter verwijzen naar Intermediate; F33 is de meetkas van Fundamental, E-nummers zijn Expert.

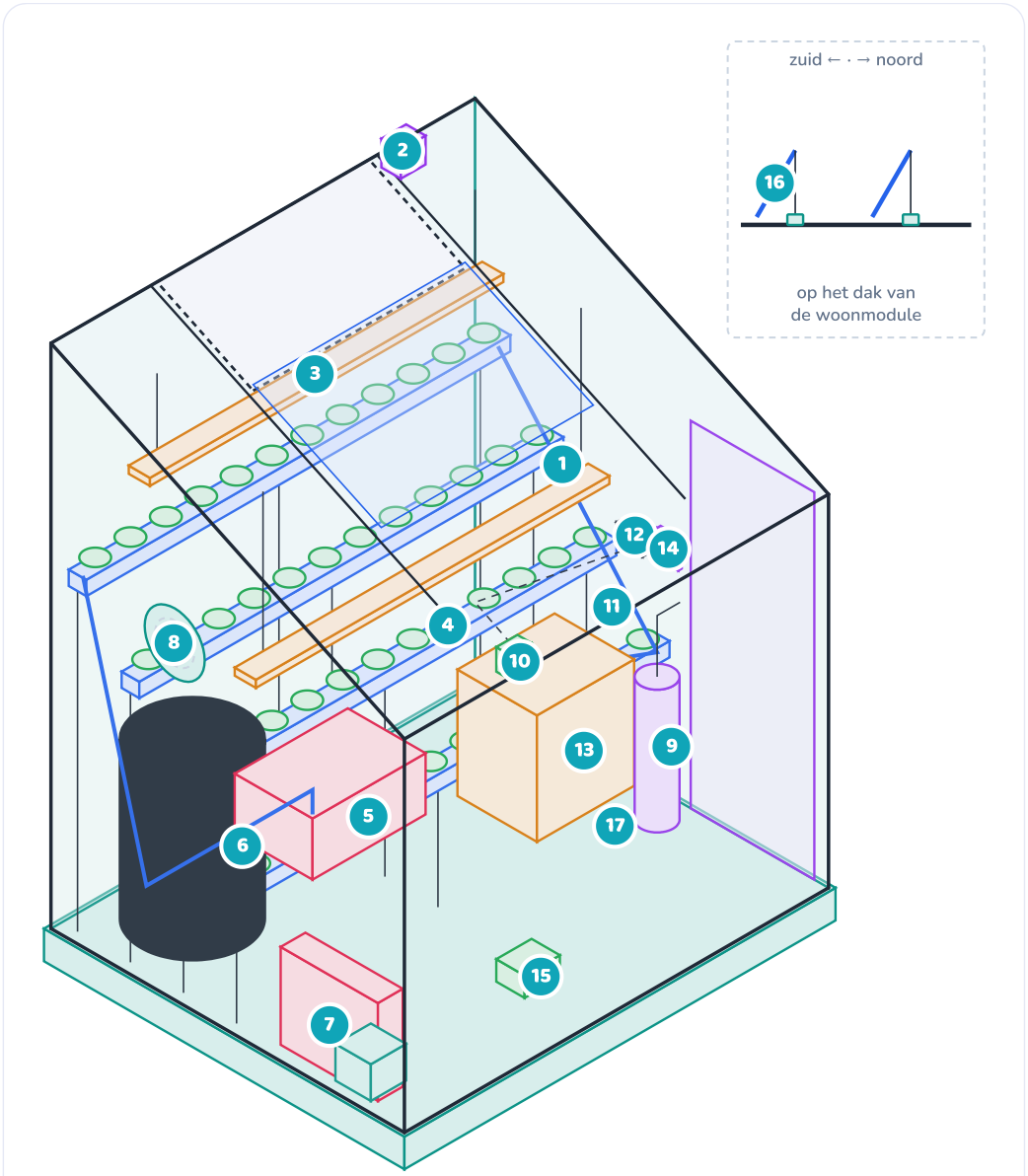
Vier eigenaars, één module: debiet meten, pH ijken, de regelkast nakijken en de logbestanden lezen.

ONTLEED · 40.2

De klimaat- en energiemodule: alle deelsystemen

2/28

Zo ziet de afgewerkte module eruit. De kweekruimte is binnen 2,40 m lang en 2,00 m diep, met een kasdak onder 25°: de lage wand (zuid) is 1,93 m hoog, de hoge wand (noord) 2,87 m (hoofdstuk 19). Het rek staat tegen de hoge wand, de gang van 0,80 m ligt langs de lage wand. De oostwand grenst aan de woonmodule van het station: daar zitten de deur, het bedieningspaneel en de warmwaterleiding. De zuid- en westwand zijn doorzichtig getekend. Bij elk nummer staat het hoofdstuk waar het onderdeel gekozen of berekend werd.



De module in isometrie, met rechtsboven de panelen op het dak van de woonmodule (zijaanzicht, niet op schaal).

1 Kasdak
Meerwandig polycarbonaat 16 mm, U = 2,0, 5,30 m²: meer dan de helft van het warmteverlies (hoofdstuk 19). Het schuifpaneel van 10 kg glijdt 0,60 m omlaag en opent de nok (hoofdstuk 7 en 8).

2 Dakmotor
Wormwielmotor 12 V met tandriem, H-brug op D6-D8, eindeloop op D23 en D24; sluit in 6,8 s, na 12 s zonder eindpunt volgt een alarm (hoofdstuk 26 en 32).

3 Groeilampen
2 × 95 W, dimbaar met PWM op D5: 200 µmol/(m²·s) op het bed, van 6 tot 22 uur. Ze vullen het daglicht aan (hoofdstuk 25 en 36).

4 NFT-rek
Vier goten van 2,4 m in trappen, elk 12 plaatsen om de 20 cm: 48 kroppen op 2,88 m². Helling 1 : 50, 1,2 L/min per goot (hoofdstuk 36).

5 Reservoir en pomp
60 L, lichtdicht, 65 kg vol, op de vloer binnen de toegelaten zone (hoofdstuk 10 en 11). Pomp 12 W, vlotter, druksensor, NTC en de pH- en EC-sonde.

6 Buffervat
Donker vat van 200 L onder het rek: samen met het reservoir 1,09 MJ per graad. Het houdt de module 's nachts warm (hoofdstuk 18 en 31).

7 Convector
Ventiloconvector op de warmwaterleiding van het station, ±800 W; klep en ventilator via D12 (hoofdstuk 31).

8 Ventilator
Afzuiging hoog in de westwand: 11 m³/h bij PWM 90 voor de CO_2 (hoofdstuk 14), vol boven 25 °C (hoofdstuk 31).

9 CO_2-fles
Fles met een klep op D10: 30 s per halfuur, alleen overdag met het dak dicht (hoofdstuk 14 en 32).

10 Sensorkast SK-1
SHT31 op I²C (0x44) aan een kabel van 3 m, midden in de ruimte op planthoogte: niet in de zon en niet boven de convector (hoofdstuk 30 en 31).

11 Lichtsensor
LDR onder een kapje op de zuidrand van het dak: meet het daglicht, niet de lampen (hoofdstuk 22 en 25).

12 Regelkast RK-1
Geprinte kast in PETG met de Mega, de klok en de SD-kaart (hoofdstuk 37). Het vermogensdeel zit apart.

13 Energiekast
Twee LiFePO4-batterijen van 120 Ah met BMS, de MPPT-regelaar, de zekeringen en de twee meetshunts (hoofdstuk 16, 20 en 30).

14 Bedieningspaneel
Knop S1 naast de deur (kort: dak, lang: onderhoud) en de statusled (hoofdstuk 24 en 32).

15 Inspectierobot
Rijdt een lus door de gang langs drie meetpunten, ±50 s per ronde, en laadt aan zijn station (hoofdstuk 33).

16 Zonnepanelen
4 × 350 Wp = 1,40 kWp onder 60° op het zuiden, in twee rijen op het platte dak van de woonmodule, met ballast (hoofdstuk 6, 10 en 20).

17 Wanden en vloer
Houtskelet met minerale wol: U = 0,22 (wand) en 0,21 (vloer). Samen met het dak en de ventilatie H = 20,2 W/K (hoofdstuk 19).

UITLEG · 40.3

De module als één systeem

3/28

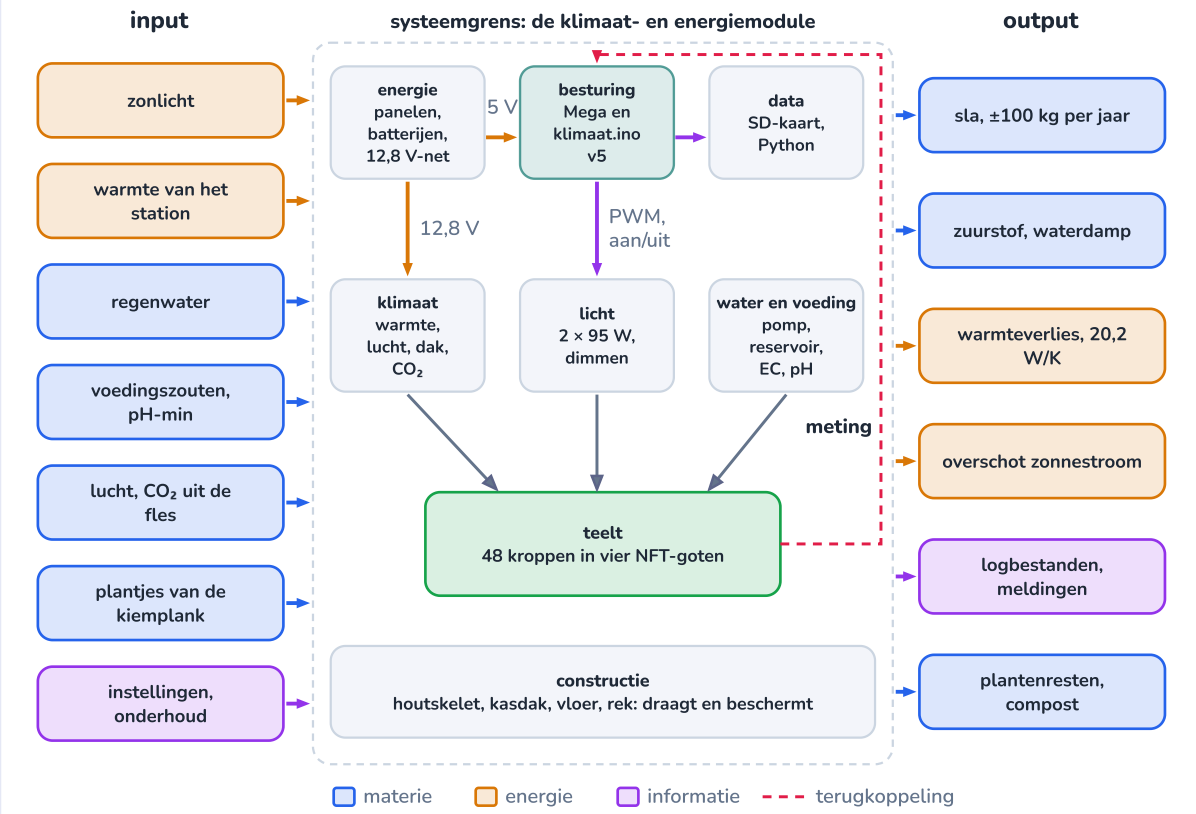
In hoofdstuk 1 van Fundamental beschreef je de meetkas als een systeem met input, proces en output. De module is hetzelfde, maar groter. Het systeemschema toont zeven soorten input (zonlicht, warmte van het station, regenwater, zouten, CO_2, plantjes en instellingen), acht deelsystemen en zes soorten output. Drie stromen houden alles samen:

- **energie**: zonnestroom via de batterijen naar lampen, pomp, ventilatoren en besturing; warmte van het station via de convector naar de lucht en de buffer;
- **materie**: water en zouten in een kringloop van reservoir, goten en planten; CO_2 uit de fles en de lucht; sla en plantenresten naar buiten;
- **informatie**: metingen naar de Mega, beslissingen naar de actuatoren, een regel per minuut naar de logger.

Een module die zichzelf in leven houdt, moet elke dag drie **balansen** sluiten. Energie: wat de panelen leveren, moet het verbruik dekken, met de batterijen als buffer voor de nacht en voor grijze dagen. Water: wat de planten verdampen, moet bijgevuld worden voor het reservoir onder 10 cm zakt. Warmte: wat door de schil verdwijnt, moet de convector, de zon of de lampen aanvullen, met de buffer voor de nacht. De fasen van dit project rekenen elke balans na, eerst op papier, daarna in een testweek.

Een schema om mee te zoeken

Loopt er iets mis, volg dan de pijlen terug. Te weinig licht op het bed? Kijk bij *licht* (lampen, dimmen), bij *besturing* (energiebeheer, grens) en bij *energie* (laadtoestand). De foutzoektabel verderop werkt precies zo.



Blauw is materie, geel energie, paars informatie; de rode stippellijn is de terugkoppeling van de sensoren naar de besturing.

UITLEG · 40.4

Het project: fasen, eigenaars en planning

4/28

Een module van 2,4 bij 2 m bouwt niet elke school. Daarom kies je bij de start een **spoor**. Beide sporen doorlopen dezelfde fasen en leveren hetzelfde op: een programma van eisen, berekeningen, een programma, logbestanden, een testrapport, een rapport en een verdediging.

	spoor A: de echte module	spoor B: tafelopstelling en simulatie
wat staat er	de kweekruimte met rek, dak, panelen en batterijen	regelkast RK-1 met de Mega, sensorkast SK-1, één goot met pomp en reservoir, één lamp, één batterij
wat je bouwt	12,8 V-net, regelkast, hydroponie, sensoren	dezelfde regelkast en sensoren, op een plank; de rest is model
testweek	de logbestanden van de module zelf	de simulatie van dit hoofdstuk, plus een week met de echte sensoren op tafel
wat je zeker leert	bouwen, keuren, alles samen laten werken	integreren, testen, data lezen: de module verandert niet van plaats

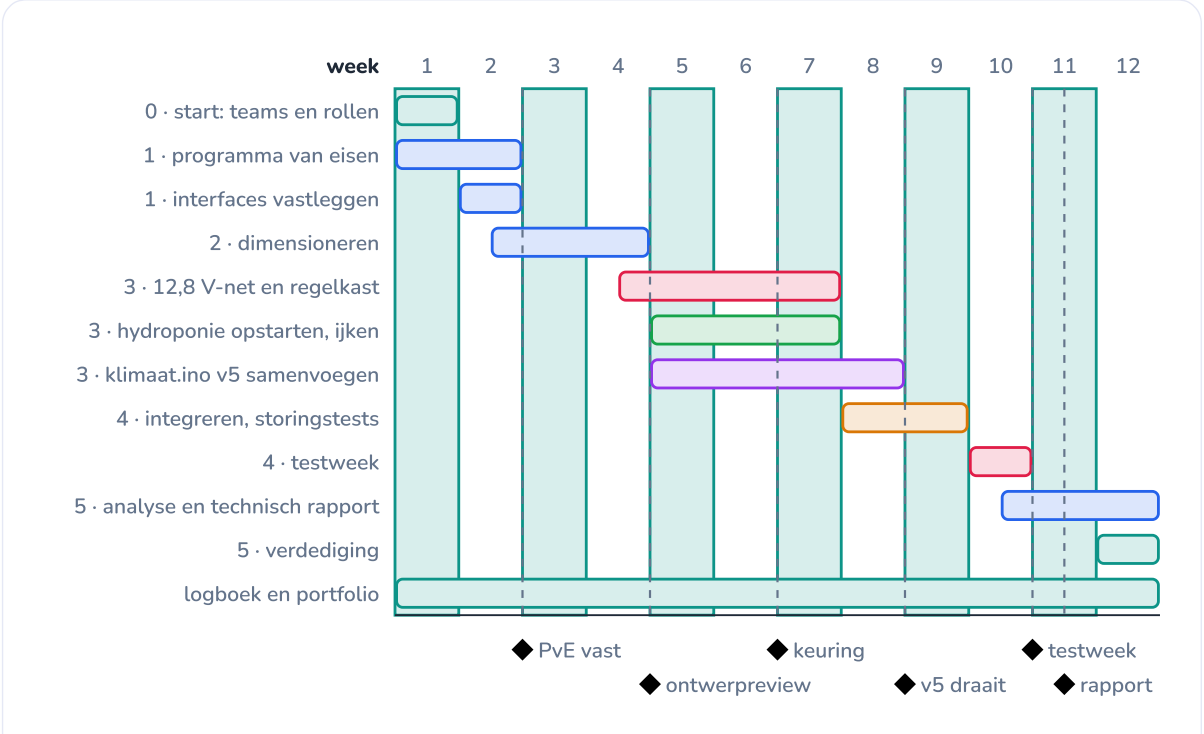
Eigenaars. Elk teamlid is **eigenaar** van één deelsysteem en van de interfaces die erbij horen: hij rekent het na, bouwt het, test het en verdedigt het. Eén van de vier is ook **integrator**: die bewaakt het programma van eisen, de interfacetabel en de planning. Iedereen soldeert, programmeert en test mee.

rol	eigenaar van	vooral in fase	steunt op hoofdstuk
energie	panelen, batterijen, 12,8 V-net, zekeringen, energiebeheer, interfaces I1, I2, I8	2, 3	16, 20, 21, 23

rol	eigenaar van	vooral in fase	steunt op hoofdstuk
water en voeding	pomp, goten, reservoir, EC en pH, bijsturen, interfaces I5, I6, I12, I13, I16	2, 3	9, 11-15, 36
klimaat en licht	warmte, ventilatie, dak, CO_2, lampen, interfaces I3, I9, I11, I14, I15, I17	2, 3	14, 18, 19, 25, 26, 31
besturing en data	klimaat.ino versie 5, logger, analyse in Python, interfaces I4, I7, I10, I18	3, 4	22, 24, 28-32, 39
integrator (één van de vier)	programma van eisen, interfacetabel, planning, testrapport, logboek	alle	2, 30 (F), 40

Planning. Het voorbeeldteam kreeg twaalf weken van drie lestijden. Fase 1 legt de lat, fase 2 rekent per deelsysteem, fase 3 bouwt en programmeert parallel, fase 4 integreert en test, fase 5 schrijft en verdedigt. Zes **mijlpalen** zijn momenten waarop iets af moet zijn en getoetst wordt: het programma van eisen ligt vast (week 2), de berekeningen doorstaan een **ontwerpreview** met de leraar (week 4), de 12,8 V-kant is gekeurd (week 6), versie 5 draait in de simulatie (week 8), de testweek is voorbij (week 10, na de integratietests van week 9) en het rapport is ingediend (week 11).

> ⚠ **Het kritieke pad**
>
> Zonder gekeurde 12,8 V-kant geen sensoren in de module, zonder werkende code geen integratietest, zonder integratietest geen testweek, zonder testweek geen testrapport. Die keten bepaalt het tempo. De hydroponie start daarom al in week 4, parallel: de planten groeien tijdens het programmeren.



Elke balk is een fase, elk ruitje een mijlpaal. De testweek (week 10) staat pas na de integratietests.

NASLAG · 40.5

Fase 1: het programma van eisen van de module

5/28

Het programma van eisen is de lat waarmee je de module beoordeelt. Voor de meetkas stelde je het op voor je rekende (Fundamental, hoofdstuk 30). Voor de module ligt al veel vast: 39 hoofdstukken berekenden en beslisten. Het werk van fase 1 is die beslissingen verzamelen, meetbaar maken en elk een toets geven. Drie regels:

1. **Meetbaar**: een grootheid, een grens met een eenheid, en hoe je het nagaat. 'Warm genoeg' wordt 'overdag minstens 17,5 °C' (K1).
2. **Vooraf**: het criterium ligt vast voor je test, anders ga je het resultaat goedpraten.
3. **Herkomst**: bij elke eis staat het hoofdstuk waar de waarde vandaan komt. Verandert daar iets, dan weet je welke eis mee verandert.

Het voorbeeldteam kwam op 54 eisen in twaalf groepen, waarvan 47 moet-eisen. De criteria zijn soms net iets ruimer dan de ontwerpwaarde: de verwarming springt aan onder 18 °C, maar de eis vraagt 17,5 °C, omdat een aan/uit-regeling altijd een beetje doorschiet (hoofdstuk 31) en de sensor ±0,2 °C meet.

De twaalf groepen

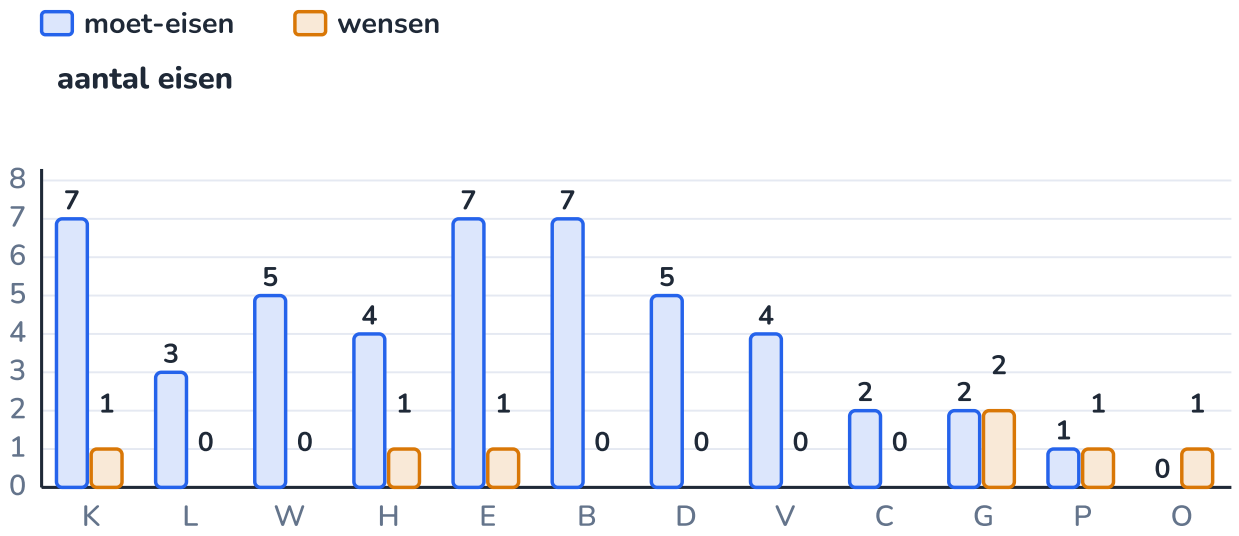
letter	groep	moet	wens	vooral uit hoofdstuk
K	klimaat	7	1	14, 31, 34
L	licht	3	0	25, 36
W	water	5	0	11, 23, 36
H	voedingsoplossing	4	1	13, 15, 36
E	energie	7	1	16, 20, 21
B	besturing	7	0	29, 31, 32
D	data	5	0	28, 30
V	veiligheid	4	0	5, 15, 26
C	constructie	2	0	8, 10, 26
G	gebruik	2	2	4, 24, 32
P	teelt	1	1	36
O	omgeving	0	1	38

Alle eisen

code	eis	soort	criterium (vooraf vast)	hoofdstuk
K1	overdag (7-22 h) minstens 17,5 °C bij de planten	moet	laagste minuut ≥ 17,5 °C	31, 34
K2	's nachts (22-6 h) minstens 15,5 °C	moet	≥ 15,5 °C	31, 34
K3	hoogstens 30 °C	moet	≤ 30 °C	31, 34
K4	ventilatoren vol boven 25 °C, dak open boven 27 °C; terug onder 24 en 25 °C	moet	drempels ± 0,3 °C	31, 32
K5	het dak haalt zijn eindpunt binnen 12 s, anders alarm	moet	alarm na 12 s	26, 32
K6	verwarming hoogstens 6 keer aan per uur	wens	≤ 6	31
K7	basisventilatie overdag minstens 11 m³/h	moet	≥ 11 m³/h	14
K8	CO_2-klep alleen in DAG met het dak dicht	moet	nooit met het dak open	14, 32
L1	lichtsom op het bed minstens 11,0 mol/(m²·dag)	moet	elke dag	25, 36
L2	lampen vullen het daglicht aan tot 200 µmol/(m²·s), hoogstens 1 stap per 200 ms	moet	± 10 µmol zolang SoC ≥ 40 %	25

code	eis	soort	criterium (vooraf vast)	hoofdstuk
L3	volle lampen: elk van 9 meetpunten ≥ 180 µmol/(m²·s)	moet	≥ 180	25, 36
W1	1,2 L/min per goot (± 20 %)	moet	0,96-1,44 L/min	11, 36
W2	pomp in minuut 0-9 van elk uur, extra beurt als slappe sla verwacht wordt	moet	juist	30, 39
W3	de pomp draait nooit droog	moet	pomp uit na 10 s droog	11, 23, 32
W4	minstens 5 dagen zonder bijvullen bij 6 L per dag	moet	≥ 5 dagen	11, 13
W5	reservoir lichtdicht: algen onder $5 \cdot 10^5$ per mL	moet	$< 5 \cdot 10^5$ per mL	5
H1	EC tussen 1,2 en 1,8 mS/cm (bij 25 °C)	moet	altijd	13
H2	pH tussen 5,5 en 6,5	moet	altijd	15
H3	voedingsoplossing nooit boven 24 °C	moet	≤ 24 °C	34, 36
H4	voedingsoplossing tussen 18 en 22 °C	wens	18-22 °C	34, 36
H5	pH-sensor om de twee weken in twee punten geijkt, EC met ijkvloeistof	moet	helling −0,18 V/pH ± 10 %; EC ± 0,05	13, 15
E1	van maart tot oktober genoeg zonnestroom: saldo per maand ≥ 0	moet	≥ 0	20
E2	de batterijen overbruggen de nacht in de slechtste maand	moet	uit de batterij ≤ 2,46 kWh	16, 20
E3	laadtoestand nooit onder 20 %	moet	≥ 20 %	20
E4	energiebeheer: onder 40 % lampen op zonnestroom, onder 20 % uit, onder 10 % alarm	moet	lampen volgen de grens	20, 29
E5	laadtoestand elk uur in het EEPROM, na een stroomuitval hersteld	moet	verschil ≤ 2 %-punt	16, 29
E6	5 V-rail aan de sensorkast minstens 4,75 V	moet	≥ 4,75 V	21
E7	batterijen niet laden onder 0 °C	moet	module ≥ 0 °C	16
E8	jaarverbruik hoogstens 700 kWh	wens	≤ 700 kWh	20, 25
B1	na een stroomuitval: alles uit, OPSTART 5 s, dan de gewone modus	moet	juist	29, 32
B2	alarm bij: geen temperatuur 30 s, vlotter droog 10 s, reservoir onder 10 cm, dak-time-out, batterij onder 10 %	moet	alarm, statusled snel	32, 40
B3	zonder temperatuur verwarmt de module met de gemiddelde inschakelduur	moet	juist	31
B4	onderhoud: lange druk van 3 s; pomp uit, dak geblokkeerd, werklicht; na 60 min terug	moet	juist	32

code	eis	soort	criterium (vooraf vast)	hoofdstuk
B5	waakhond: een hangende taak herstart de Mega binnen 2 s	moet	herstart	32
B6	langste ronde van loop() hoogstens 100 ms	moet	≤ 100 ms	29
B7	één taak per uitgang	moet	geen tweede eigenaar	40
D1	elke minuut een CSV-regel, één bestand per dag	moet	1440 regels per dag	30
D2	een ontbrekende meting geeft een lege cel	moet	lege cel	30
D3	minstens een jaar op de kaart	moet	≥ 1 jaar	30
D4	klok op UTC+1, hoogstens 1 min afwijking per maand	moet	≤ 1 min	30
D5	dag- en weekrapport in Python, gekoppeld aan de eisen	moet	draait	28, 30
V1	in de kweekruimte alleen 12,8 V en 5 V	moet	in orde	5, 20
V2	zekering per tak en bij elke batterijpool	moet	in orde	20, 23, 26
V3	niemand onder een bewegend dak	moet	dak staat stil	26, 32
V4	pH-min (salpeterzuur 0,50 mol/L) gelabeld, met bril en handschoenen	moet	in orde	15
C1	reservoir (65 kg vol) met zijn midden tussen 0,65 en 1,31 m van ligger A	moet	in de zone	10
C2	dakmotor trekt 120 N zonder te blokkeren	moet	≥ 120 N	8, 26
G1	statusled: rustig normaal, snel bij alarm, vast bij onderhoud	moet	juist	32
G2	kasdak met de hand: korte druk op S1 naast de deur	moet	juist	24, 32
G3	wekelijks onderhoud (bijvullen, EC en pH) hoogstens 20 min	wens	≤ 20 min	40
G4	hydroponie hoogstens € 800	wens	≤ € 800	4
P1	48 plantplaatsen, om de 8,75 dagen een goot oogsten	moet	in orde	36
P2	minstens 100 kg sla per jaar	wens	≥ 100 kg	36
O1	hoogstens 2 kg CO_2-eq per kg sla	wens	≤ 2	38



K klimaat, L licht, W water, H voedingsoplossing, E energie, B besturing, D data, V veiligheid, C constructie, G gebruik, P teelt, O omgeving.

UITLEG · 40.6

Interfaces: waar de deelsystemen elkaar raken

6/28

Een interface is een afspraak tussen twee deelsystemen. Het **N²-diagram** toont ze in één oogopslag: de deelsystemen staan op de diagonaal, en het vakje in rij A en kolom B zegt wat A aan B levert. Een lege rij of kolom is verdacht: een deelsysteem dat niets krijgt of niets geeft, hoort er waarschijnlijk niet bij. De **interfacetabel** schrijft elke afspraak uit: de waarde met eenheid, de pin of het adres, de eigenaar en de test. Twee interfaces komen van buiten de module: het station (warmte, deur) en de bewoner (de knop).

code	van → naar	afspraak	eigenaar	test
I1	energie → water, klimaat, licht	12,8 V-rail, elke tak een eigen zekering (hoofdstuk 20, 23, 26)	energie	spanning aan de verbruiker bij volle last
I2	energie → besturing	5 V-rail (omvormer 1 A): minstens 4,75 V aan de sensorkast (hoofdstuk 21)	energie	meten terwijl de pomp aanslaat
I3	besturing → licht	D5, PWM 490 Hz: 0-255 = 0-190 W, hoogstens 1 stap per 200 ms (hoofdstuk 25)	besturing	lampstand in de logger
I4	besturing → water	D9 (IRLZ44N): minuut 0-9 en de extra beurt; één eigenaar (hoofdstuk 23, 39)	besturing	naspelen
I5	water → besturing	vlotter D2 (LOW = water), druk A5: niveau alleen bij stilstaande pomp (hoofdstuk 11)	water	vlotter optillen
I6	voeding → besturing	pH A2 (2,50 V bij 7,00; −0,18 V per pH), EC A6 (0,50 V per mS/cm), NTC A0 (hoofdstuk 13, 15, 22)	voeding	ijken
I7	klimaat → besturing	SHT31 op I²C 0x44, sensorkast op 3 m, pull-ups samen 1,8 kΩ (hoofdstuk 30)	klimaat	I²C-scanner
I8	energie → besturing	INA226 0x40 (zon) en 0x41 (verbruik), shunt 100 A / 75 mV (hoofdstuk 30)	energie	stroomtang ernaast

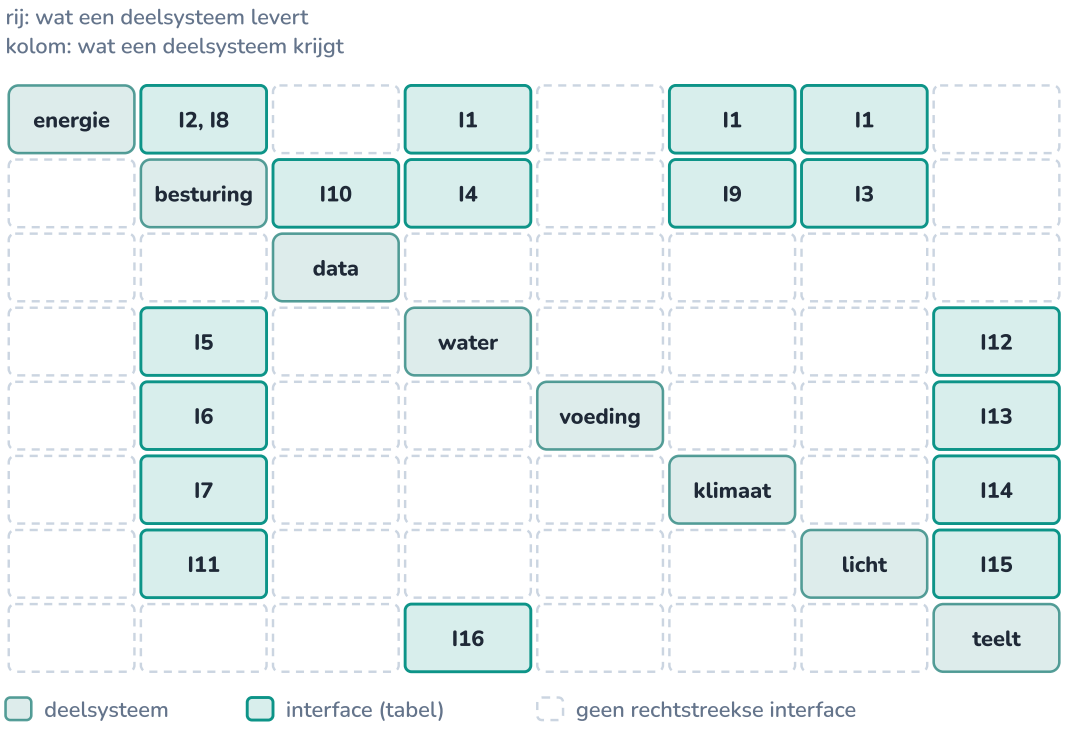
code	van → naar	afspraak	eigenaar	test
I9	besturing → klimaat	D10 CO_2-klep, D11 ventilatoren (PWM), D12 convector, D6-D8 dak, D23/D24 eindeloop (hoofdstuk 26, 31)	besturing	naspelen
I10	besturing → data	één CSV-regel per minuut, 17 kolommen, UTC+1, lege cel = geen meting (hoofdstuk 30)	data	logbestand lezen
I11	licht → besturing	LDR A1 onder een kapje: daglicht in lux, gedeeld door 54 = µmol/(m²·s) (hoofdstuk 22, 25)	licht	luxmeter ernaast
I12	water → teelt	1,2 L/min per goot, een film van enkele mm (hoofdstuk 36)	water	maatbeker
I13	voeding → teelt	EC 1,4 mS/cm, pH 5,8, 18-22 °C (hoofdstuk 13, 15, 36)	voeding	logger
I14	klimaat → teelt	18-27 °C overdag, CO_2 ≥ 350 ppm (hoofdstuk 14, 31, 34)	klimaat	logger
I15	licht → teelt	200 µmol/(m²·s) van 6 tot 22 uur: 11,5 mol per dag (hoofdstuk 25, 36)	licht	PAR-meter
I16	teelt → water	transpiratie 4 à 6 L per dag: bijvullen met regenwater en stamoplossing (hoofdstuk 13, 36)	water	niveau in de logger
I17	station → klimaat	warmwaterleiding naar de convector (±800 W), de deur, één gedeelde wand (hoofdstuk 19, 31)	klimaat	convector aan, temperatuur
I18	bewoner → besturing	S1 naast de deur op D22 (RC + 74HC14): kort = dak, lang (3 s) = onderhoud (hoofdstuk 24, 32)	besturing	knoptest

Integratieproblemen opsporen. De meeste problemen vind je niet door één deel te testen, maar door twee documenten naast elkaar te leggen: een budget naast de code, een schema naast een ander hoofdstuk, de code van twee versies naast elkaar. Het voorbeeldteam vond er zes, allemaal vóór de testweek:

gevonden bij	probleem	oplossing in versie 5
interface I8 naast hoofdstuk 20	de ACS712 op A4 meet tot 20 A, maar de panelen laden met tot ±74 A: de coulombteller telt fout	de laadtoestand telt met de twee INA226 (zon min verbruik); A4 blijft vrij
code naast code	taakExtraPomp() (hoofdstuk 39) zet pin 9 aan, acties() (versie 4) zet ze om de 2 s weer uit: de pomp flikkert	één taak beslist over de pomp: minuut 0-9 of de extra beurt
takenlijst van versie 2 en 4	modi.ino (hoofdstuk 32) liet het dimmen, het energiebeheer en de logger weg	alles terug in één programma, met de modi als kader
budget naast code	hoofdstuk 20 rekende 's nachts 2 W voor de ventilatoren; versie 4 liet ze op PWM 90 (5 W) draaien	's nachts PWM 36 (2 W): het budget en de code zeggen hetzelfde
integratietest, reservoir bijna leeg	met de pomp aan staan er ±3,9 L in slang en goten: het niveau zakt 1,6 cm; bij 11 cm gaf dat elk uur een vals alarm	het niveau telt pas 2 min nadat de pomp gestopt is
budget naast logger	hoofdstuk 20 rekende de lampen van 4 tot 20 uur UTC; de klok van de module loopt op UTC+1	één afspraak: alles in UTC+1, lampen van 6 tot 22 uur

Als elk deel werkt, werkt het geheel

Elke rij in de tabel hierboven gaat over delen die apart perfect werkten. De ACS712 meet juist; de extra pompbeurt van hoofdstuk 39 werkt; het budget van hoofdstuk 20 klopt met zijn eigen aannames. Het probleem zit telkens in de afspraak ertussen.



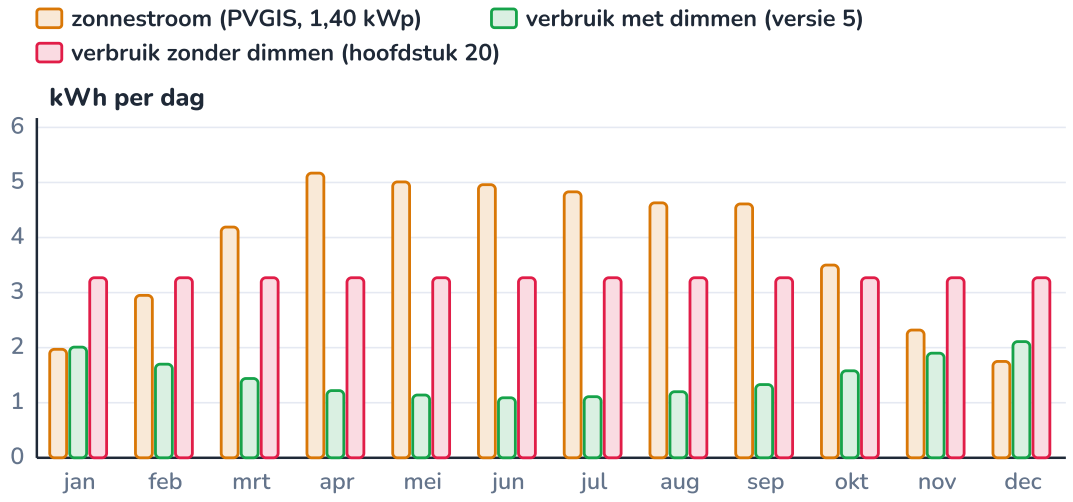
De besturing ontvangt van bijna iedereen en stuurt bijna iedereen: daar lopen de meeste interfaces samen.

UITGEWERKT VOORBEELD · 40.7

Fase 2 · energie: het budget na veertig hoofdstukken

7/28

Hoofdstuk 20 stelde het energiebudget op: 3,27 kWh per dag, waarvan 93 % voor de lampen, en besloot tot een tweede batterij. Sindsdien kwamen er dingen bij: het dimmen met het daglicht (hoofdstuk 25), de convector (hoofdstuk 31), de modi en de logger. Reken het budget na zoals de module nu is. Volstaan de panelen en de batterijen, per maand? En wat betekent dat voor een heel jaar?



Dimmen halveert het verbruik ruim; alleen in december en januari blijft de zon (geel) onder het gedimde verbruik (groen).

GEGEVEN

- PVGIS Brussel, 60° zuid: de gemiddelde dag van elke maand, per uur (hoofdstuk 20); 1,40 kWp
- lampen van 6 tot 22 uur (UTC+1), $P = 190\,\mathrm{W} \cdot \min\left(1,\ \frac{200-\Phi_{\mathrm{dag}}}{200}\right)$ met Φ_{dag} het daglicht in µmol/(m²·s)
- daglicht op het bed: ±50 lx per W/m² op het paneel (rekenwaarde van dit project), 54 lx per µmol/(m²·s) (hoofdstuk 25)
- batterijen: 2 × 1,23 = 2,46 kWh bruikbaar; laadrendement 95 %

GEVRAAGD

- het budget van een gemiddelde maartdag
- saldo en nachtbuffer per maand
- het jaar

OPLOSSING

1 **Het budget.** De lampen verbruiken niet meer 3040 Wh per dag, maar alleen wat het daglicht niet levert. De rest verandert weinig; twee posten komen erbij en één wordt zuiniger.

VERBRUIKER	HOOFDSTUK 20 (WH/DAG)	VERSIE 5, GEMIDDELDE MAARTDAG (WH/DAG)	WAAROM ANDERS
groeilampen 2 × 95 W	3040	1206	gedimd met het daglicht (hoofdstuk 25): de lampen vullen aan tot 200 µmol
ventilatoren	96	95	overdag PWM 90, 's nachts PWM 36 (nieuw in versie 5)
microcontroller, sensoren, logger	72	72	budget 3,0 W; hoofdstuk 21 rekende 0,96 W, de logger 0,07 W
pomp (10 min per uur)	48	48	en ±1 Wh per extra beurt van 5 min (hoofdstuk 39)
laadregelaar en BMS	12	12	
CO_2-klep	1,5	1,6	32 keer 30 s in DAG, alleen met het dak dicht
dakmotor	0,2	1,0	zonnige dag: ±24 bewegingen van ±7 s (testweek)
convector: klep en ventilator	0,0	30	nieuw: 12 W zolang er verwarmd wordt
totaal	**3270**	**1465**	met gedimde lampen: 55 % minder

2 **Per maand.** Zelfde methode als hoofdstuk 20, uur per uur, maar met gedimde lampen:

MAAND	JAN	FEB	MRT	APR	MEI	JUN	JUL	AUG	SEP	OKT	NOV	DEC
zon (kWh/dag)	1,97	2,95	4,19	5,17	5,01	4,96	4,83	4,63	4,61	3,50	2,32	1,75
verbruik, gedimd	2,01	1,70	1,44	1,22	1,14	1,09	1,11	1,20	1,33	1,58	1,90	2,11
uit de batterij	1,73	1,44	1,16	0,92	0,74	0,66	0,71	0,86	1,07	1,32	1,64	1,84
saldo	**−0,12**	1,12	2,55	3,70	3,64	3,64	3,50	3,22	3,06	1,75	0,32	**−0,44**

Alleen december (−0,44 kWh) en januari (−0,12 kWh) komen tekort; november en februari sluiten nu, zonder dimmen kwamen ze 1,0 en 0,4 kWh per dag tekort.

3 **De nacht.** Uit de batterijen komt hoogstens $1{,}84\,\mathrm{kWh}$ (december): $1{,}84/2{,}46 = 75\,\%$ van wat bruikbaar is. Eén batterij (1,23 kWh) volstaat van oktober tot februari niet: de tweede blijft nodig.

4 **Het jaar.** De panelen kunnen $1397\,\mathrm{kWh}$ leveren; de module gebruikt er $\pm 543\,\mathrm{kWh}$ van, waarvan $459\,\mathrm{kWh}$ voor de lampen. Zonder dimmen was dat $1193\,\mathrm{kWh}$. Blijft de oogst 100 kg sla, dan kost een kilo nog $4{,}6\,\mathrm{kWh}$ lampenergie in plaats van 11,1 (hoofdstuk 36).

ANTWOORD

Met dimmen daalt het verbruik van een gemiddelde maartdag van 3,27 naar 1,47 kWh. De panelen volstaan van februari tot november; alleen in december en januari is er gemiddeld te weinig zon. De twee batterijen blijven nodig voor de nacht (tot 1,84 kWh).

Klopt dit? Een gemiddelde dag is een optimistische dag. Het dimmen hangt niet lineair af van het licht: op een grijze dag moeten de lampen veel meer doen dan de gemiddelde dag voorspelt, en op een zonnige dag kunnen ze niet minder dan uit. De testweek verderop toont het: op twee grijze maartdagen zakte de laadtoestand onder 40 % en kregen de lampen alleen nog zonne-overschot. Het omgekeerde gebeurt ook: op zonnige dagen zijn de batterijen om 11 uur vol en knijpt de regelaar de zonnestroom af. De jaarbalans klopt dus ruim, de weekbalans niet altijd.

UITLEG · 40.8

Het 12,8 V-net: van paneel tot pin

8/28

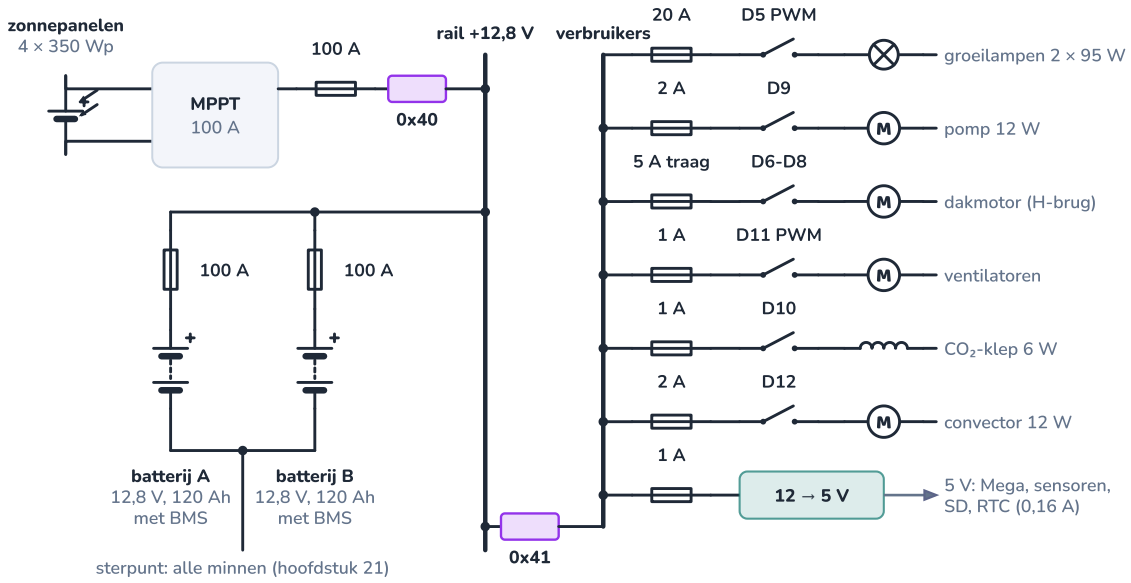
Het schema toont de pluskant van het net als één lijn per tak. Het brengt samen wat hoofdstuk 16 (batterij), 20 (panelen en regelaar), 21 (5 V-rail en sterpunt), 23 (MOSFETs), 25 (PWM), 26 (dakmotor) en 30 (INA226) apart vastlegden.

- **Bron.** De MPPT-regelaar zet de 30 à 40 V van de panelen om naar de laadspanning van de batterijen (hoofdstuk 20; hoe hij dat doet, zie je in E24). Op een zonnige middag loopt er tot ±74 A naar de rail.
- **Twee meetshunts.** De INA226 op 0x40 meet de zonnestroom naar de rail, die op 0x41 de stroom naar de verbruikers. Het verschil is de batterijstroom: daarmee telt versie 5 de laadtoestand. De ACS712 op A4 (hoofdstuk 16) meet maar tot 20 A en zou op een zonnige middag verzadigen.
- **Zekeringen.** Een zekering bij elke batterijpool (een batterij levert bij kortsluiting honderden ampère) en één per tak, net boven de stroom die de tak normaal vraagt.
- **Het sterpunt.** Elke tak heeft een eigen retourdraad naar de minpool. Een gedeelde massadraad liet de pH 0,48 verspringen als de pomp aansloeg (hoofdstuk 21).

tak	stroom (rekenwaarde)	zekering	schakelaar en pin	hoofdstuk
groeilampen 2 × 95 W	14,8 A	20 A	IRLZ44N op koelplaat, D5 (PWM 490 Hz)	23, 25
pomp 12 W	1,0 A (aanloop meer)	2 A	IRLZ44N, D9, vrijloopdiode 1N5819	9, 23

tak	stroom (rekenwaarde)	zekering	schakelaar en pin	hoofdstuk
dakmotor	1,84 A bij 60 N, 10 A blokkeren	5 A traag	H-brug 2 × BTS7960, D6-D8, eindeloop D23/D24	26
ventilatoren	0,39 A bij PWM 90	1 A	IRLZ44N, D11 (PWM)	23, 31
CO_2-klep	0,47 A	1 A	IRLZ44N, D10, vrijloopdiode	14, 23
convector: klep en ventilator	±0,9 A (rekenwaarde 12 W)	2 A	IRLZ44N, D12	31, 40
omvormer 12 → 5 V	0,08 A gemiddeld (0,16 A aan de 5 V-kant)	1 A	altijd aan	21

De grootste stroom tegelijk: lampen, pomp, een bewegende dakmotor en de ventilatoren vol, plus de omvormer: $14{,}8 + 1{,}0 + 1{,}8 + 1{,}1 + 0{,}1 \approx 19\ \text{A}$. Verwarmen en koelen gaan nooit samen, en de CO_2-klep gaat niet open met een bewegend dak. De kabels van de rail naar de lampen rekent E5 na met matrices: hoogstens 3 % spanningsval tot elke verbruiker.



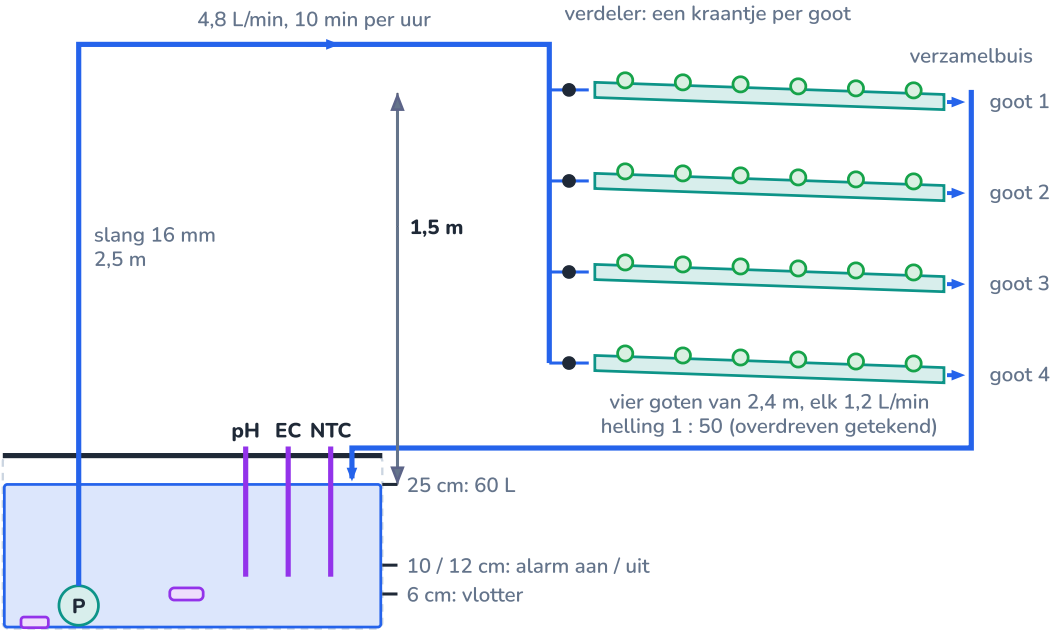
Links de bron en de batterijen, in het midden de rail met de twee meetshunts, rechts de zeven takken met hun zekering, schakelaar en pin.

UITGEWERKT VOORBEELD · 40.9

Fase 2 · water: pomp, goten en reservoir

9/28

Reken de waterkringloop van de module na. Haalt de pomp het debiet dat de goten nodig hebben? Hoeveel water staat er onderweg als de pomp draait, en wat ziet de druksensor dan? Hoe lang houdt het reservoir het vol zonder bijvullen, en hoeveel water gebruikt de module per jaar?



Het water gaat 1,5 m omhoog naar de verdeler, loopt in vier goten langs de wortels en valt via de verzamelbuis terug in het reservoir.

GEGEVEN

- pomp B: werkpunt 4,8 L/min bij 1,56 m met een slang van 16 mm en 2,5 m (hoofdstuk 11), 12 W elektrisch
- 4 goten van 2,4 m, 10 cm breed, film van ±3 mm, 1,2 L/min per goot (hoofdstuk 36)
- reservoir binnen 60 × 40 cm, vol bij 25 cm; alarm onder 10 cm (hoofdstuk 11)
- verbruik tot 6 L op een warme dag (hoofdstuk 13); 16 L water per kg sla, 100 kg per jaar (hoofdstuk 36)
- regen in Ukkel 837 mm per jaar (F9); kasdak in bovenaanzicht 2,40 × 2,00 m

GEVRAAGD

- debiet en rendement van de pomp
- water onderweg
- dagen zonder bijvullen
- de waterbalans van een jaar

OPLOSSING

1. **Debiet.** Vier goten van 1,2 L/min vragen $4{,}8\,\mathrm{L/min}$: precies het werkpunt. Het hydraulische vermogen is $P = \rho g h Q = 1000 \cdot 9{,}81 \cdot 1{,}56 \cdot \frac{4{,}8}{60\,000} = 1{,}22\,\mathrm{W}$, van de 12 W: een rendement van ±10 % (hoofdstuk 9). In 10 minuten gaat er 48 L rond.

2. **Water onderweg.** Slang: $\frac{\pi}{4} \cdot 1{,}6^2 \cdot 250 = 503\,\mathrm{cm^3}$; film in vier goten: $4 \cdot 240 \cdot 10 \cdot 0{,}3 = 2880\,\mathrm{cm^3}$; verdeler en verzamelbuis ±0,5 L. Samen $3{,}9\,\mathrm{L}$, dus zakt het reservoir $3{,}9 / 2{,}4 = 1{,}6\,\mathrm{cm}$ zolang de pomp draait.

3. **Dagen zonder bijvullen.** Tussen 25 en 10 cm zit $15 \cdot 2{,}4 = 36\,\mathrm{L}$: $36 / 6 = 6{,}0$ dagen op warme dagen. In de testweek verdampten de planten gemiddeld $4{,}3\,\mathrm{L}$ per dag.

4. **Het jaar.** Transpiratie $16 \cdot 100 = 1600\,\mathrm{L}$, het reservoir om de 5 weken verversen: $60 \cdot 365 / 35 \approx 626\,\mathrm{L}$. Samen $\approx 2{,}2\,\mathrm{m^3}$. Op het kasdak valt $4{,}80 \cdot 0{,}837 = 4{,}0\,\mathrm{m^3}$ regen.

ANTWOORD

De pomp levert precies de 4,8 L/min van vier goten. Terwijl ze draait, staan er ±3,9 L onderweg en meet de druksensor 1,6 cm te laag: versie 5 leest het niveau pas 2 minuten na de pomp. Het reservoir houdt 6 warme dagen vol; de module gebruikt ±2,2 m³ water per jaar, en het kasdak vangt er bijna het dubbele.

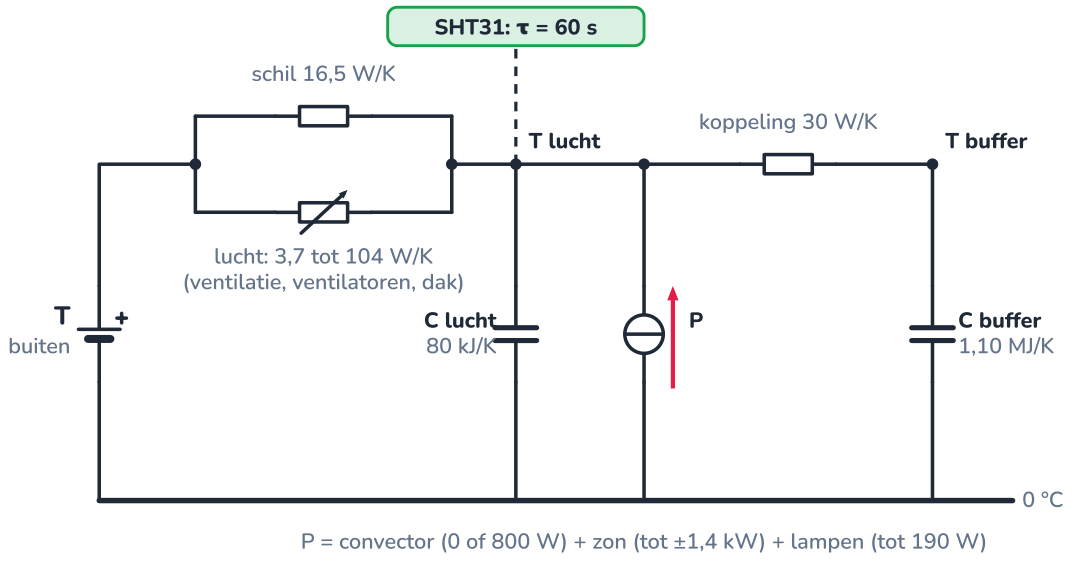
Klopt dit? De alarmgrens van 10 cm heeft een hysterese tot 12 cm (hoofdstuk 32). Die 2 cm is maar iets groter dan de 1,6 cm die de pomp wegneemt: zonder de regel 'niveau bij stilstaande pomp' gaf een reservoir op 11 cm elk uur in minuut 1 tot 9 een alarm. Nog een interface: de goten liggen in trappen, dus de onderste krijgt meer druk dan de bovenste. Het voorbeeldteam mat 1,08 tot 1,31 L/min en zette de kraantjes bij tot 1,19 à 1,21 L/min (eis W1).

UITGEWERKT VOORBEELD · 40.10

Fase 2 · warmte: verlies, buffer en verwarming

10/28

Reken de warmtebalans van de module na, voor een koude winternacht en voor een zonnige lentemiddag. Volstaat de convector van 800 W? Wat doet de buffer? En volstaan de ventilatoren en het dak om de module onder 27 °C te houden?



Temperaturen als spanningen, warmtestromen als stromen: de lucht warmt snel op en af, de buffer traag.

GEGEVEN

- H = 20,2 W/K (hoofdstuk 19); ontwerptemperatuur buiten −8 °C (Ukkel)
- overdag verwarmen onder 18 °C, 's nachts onder 16 °C; convector 800 W (hoofdstuk 31)
- buffer: 200 L vat + 60 L reservoir; lucht met alles wat snel meewarmt 80 kJ/K (hoofdstuk 31)
- zon door het kasdak op een heldere lentemiddag tot ±1,4 kW; buiten 14 °C
- ventilatoren vol +40 W/K, dak open +60 W/K (hoofdstuk 31)

GEVRAAGD

- vermogen bij −8 °C
- de buffer
- koelen op een zonnige dag

OPLOSSING

1. **Winter.** Overdag (18 °C) bij −8 °C: $P = H \cdot \Delta T = 20{,}2 \cdot 26 = 525\,\mathrm{W}$; hoofdstuk 19 rekende 567 W voor 20 °C. De convector heeft ±275 W over om op te warmen. Schakelen de lampen aan, dan komt er tot 190 W bij.

2. **De buffer.** $C = 260\,\mathrm{kg} \cdot 4186\,\mathrm{J/(kg \cdot K)} = 1{,}09\,\mathrm{MJ/K}$: veertien keer de snelle knoop. Koelt de buffer 's nachts 2 °C af, dan geeft hij 2,2 MJ = 0,6 kWh warmte af. Hij remt het afkoelen en vlakt de zaagtand van de verwarming af (hoofdstuk 18 en 31).

3. **Zonnige lentemiddag.** Om 1,4 kW zonnewarmte bij 14 °C buiten kwijt te raken zonder boven 27 °C te gaan, moet $H \geq 1400/(27-14) = 108\,\mathrm{W/K}$. De schil alleen haalt 20 W/K; met de ventilatoren vol en het dak open is het $20 + 40 + 60 = 120\,\mathrm{W/K}$: genoeg, net.

ANTWOORD

De convector van 800 W volstaat bij −8 °C met ±275 W reserve; de buffer (1,09 MJ/K) overbrugt de schommelingen. Op een zonnige lentedag moeten de ventilatoren en het dak samen werken: 120 W/K tegenover 108 W/K nodig. In de testweek kwam de module op de twee zonnige dagen tot 27,1 °C (eis K3).

Klopt dit? De testweek vroeg 14,0 kWh warmte van het station in zeven maartdagen, van 0,8 tot 4,3 uur verwarming per dag. Die warmte komt niet uit de batterijen: alleen de klep en de ventilator van de convector (12 W) staan in het energiebudget. In de zomer is het omgekeerde probleem groter: dan helpt een schaduwdoek meer dan meer ventilatie (versie 2).

UITLEG · 40.11

Fase 2 · licht: lichtsom, lampen en dimmen

Planten tellen fotonen, geen watt (hoofdstuk 36). Het ontwerp vraagt 200 µmol/(m²·s) op het bed, 16 uur per dag. Dat is de **lichtsom** (daglichtsom, DLI):

LICHTSOM VAN DE MODULE

$$\mathrm{DLI} = 200\,\mu\mathrm{mol}/(\mathrm{m}^2\cdot\mathrm{s})\cdot 16\cdot 3600\,\mathrm{s} = 11{,}5\,\mathrm{mol}/(\mathrm{m}^2\cdot\mathrm{dag})$$

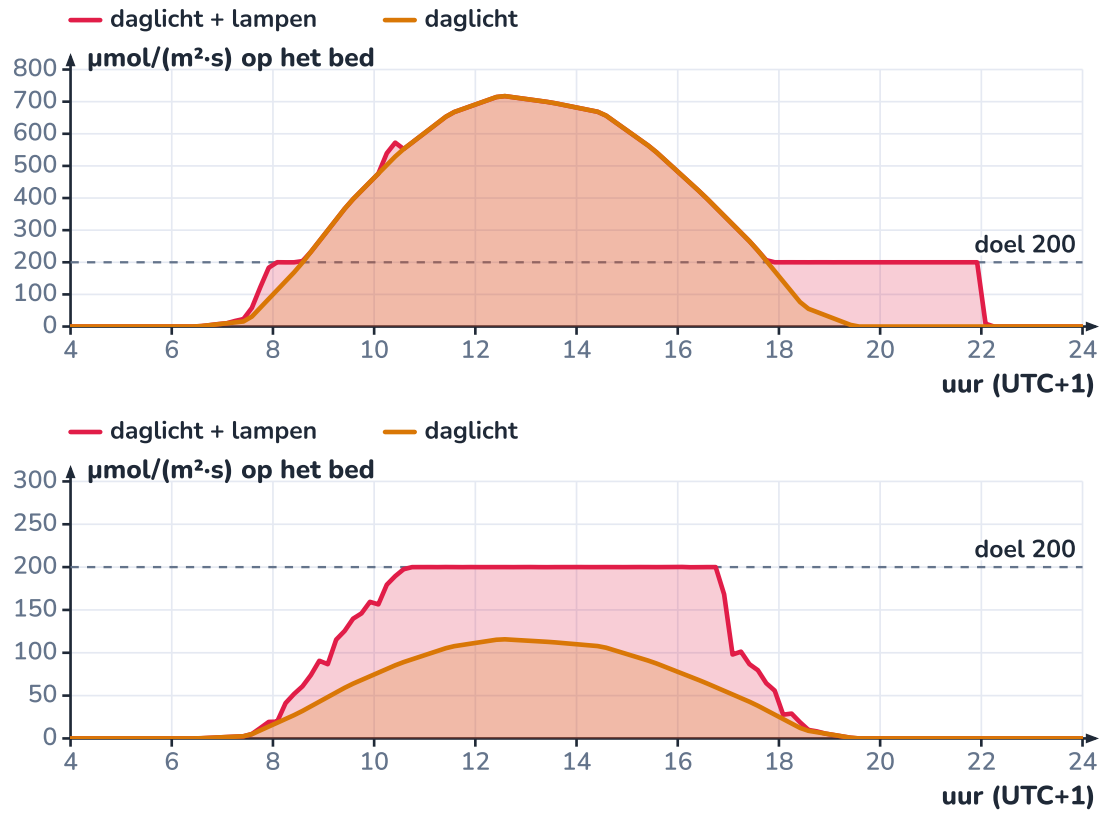
Twee lampen van 95 W moeten dus $200\cdot 2{,}88 = 576\,\mu\mathrm{mol}/\mathrm{s}$ op het bed krijgen: 3,0 µmol per joule (hoofdstuk 36 rekende 3,0, een zuinige lamp met reflecterende wanden).

Dimmen. De LDR meet het daglicht in lux; gedeeld door 54 wordt dat µmol/(m²·s) (hoofdstuk 25). De lampen vullen het verschil aan tot 200: bij 100 µmol daglicht branden ze op de helft, vanaf 200 zijn ze uit. De grafiek toont twee dagen uit de testweek. Op de zonnige woensdag is het daglicht van ongeveer 9 tot half zes ruim genoeg; de lampen helpen alleen 's morgens en 's avonds. De lichtsom wordt 21,4 mol: veel meer dan nodig. Op de grijze dinsdag komt het daglicht nauwelijks boven 100.

Energiebeheer. Dinsdag lag de laadtoestand onder 40 %. Dan krijgen de lampen alleen het zonne-overschot: de grens stijgt als de batterijen laden en daalt als ze leveren (hoofdstuk 20 en 29). 's Morgens tot 8 uur bleven de lampen uit, 's middags brandden ze op wat de panelen over hadden, en rond 18 uur gingen ze uit. De lichtsom zakte naar 6,0 mol. Dat is een keuze van het ontwerp: bij een lege batterij gaan eerst de lampen uit, niet de pomp of de besturing. De prijs is eis L1 (verderop).

Meer licht is altijd beter

Boven ±250 à 300 µmol/(m²·s) groeit sla bij deze temperatuur en CO_2 nauwelijks sneller, maar ze verdampt meer (hoofdstuk 36). De module dimt daarom naar 200 en laat de middagzon ongebruikt over het bed vallen. Extra licht is pas nuttig als de lichtsom onder het doel blijft.



Geel het daglicht, rood wat de lampen erbij doen. Dinsdag blijft de som onder 200: het energiebeheer houdt de lampen laag.

PROEF · 40.12

Fase 2 · voedingsoplossing: ijken, bewaken en bijsturen

12/28

ONDERZOEKSVRAAG

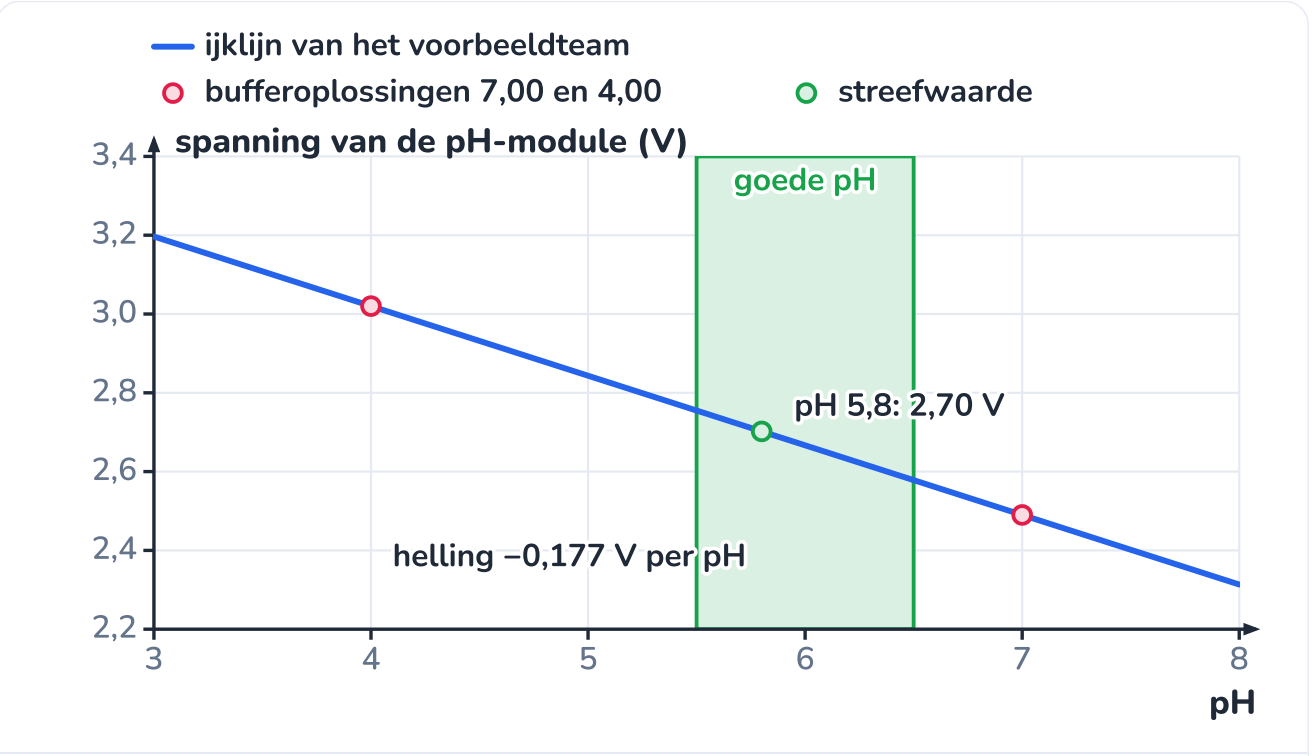
Meten de pH- en de EC-sonde in het reservoir juist, en hoe stuur je de oplossing bij als de logger een waarschuwing geeft? (Samenstelling: hoofdstuk 12; EC: hoofdstuk 13; pH: hoofdstuk 15.)

MATERIAAL

- pH-module en sonde, EC-module en sonde, de Mega met een testprogramma dat de spanning toont
- bufferoplossingen pH 7,00 en 4,00, EC-ijkvloeistof 1,413 mS/cm, gedemineraliseerd water, tissues
- maatbeker 1 L, spuitjes van 5 en 20 mL, stamoplossingen A en B, pH-min (salpeterzuur 0,50 mol/L)
- veiligheidsbril, nitrilhandschoenen, het logboek

Veiligheid

Salpeterzuur van 0,50 mol/L is bijtend: bril en handschoenen, altijd zuur in water gieten, nooit omgekeerd, en de fles gelabeld (GHS05) buiten het bereik van de planten bewaren (hoofdstuk 15). Stamoplossing A en B nooit onverdund samen gieten: calcium en fosfaat of sulfaat slaan dan neer.



Twee bufferoplossingen bepalen de ijklijn; de band van 5,5 tot 6,5 is de goede pH (eis H2).

WERKWIJZE

1. Spoel de pH-sonde, zet ze in buffer 7,00 en wacht tot de spanning 30 s stabiel is. Noteer de spanning. Spoel, en herhaal in buffer 4,00.
2. Bereken de helling in V per pH en de spanning bij pH 7. Zet beide in de code (hoofdstuk 15 rekende 2,50 V en −0,18 V per pH).
3. Zet de EC-sonde in de ijkvloeistof van 1,413 mS/cm (25 °C) en vergelijk met wat de module toont na de omrekening naar 25 °C.
4. Bijvullen: meet het niveau, vul met regenwater bij tot 25 cm, en voeg per liter bijgevuld water 10 mL A en 10 mL B toe, telkens apart en goed gemengd.
5. pH te hoog: neem 1 L uit het reservoir en titreer met pH-min tot 5,8 (proeftitratie). Vermenigvuldig met het volume in het reservoir en geef de dosis in twee helften, met 15 minuten rondpompen ertussen.
6. Noteer alles in het logboek: tijd, niveau, EC en pH voor en na, dosis.

WAARNEMING

meting	voorbeeldteam
buffer pH 7,00	2,49 V
buffer pH 4,00	3,02 V
helling	−0,177 V per pH (−0,18 ± 10 %: in orde)
EC-ijkvloeistof 1,413 mS/cm	module toont 1,42 mS/cm
woensdag: bijvullen	13 L regenwater + 130 mL A + 130 mL B; niveau 18,9 → 24,3 cm; EC 1,39 → 1,38
maandag na de testweek	pH 6,6 in ±35 L; proeftitratie: 1,6 mL pH-min per L; dosis 56 mL in twee keer 28 mL

De spanningen wijken 0,01 V af van hoofdstuk 15: elke sonde is een beetje anders, en ze verandert met de tijd. Daarom ijkt het team om de twee weken (eis H5).

VERKLARING

De pH-module geeft $U = 2{,}50 - 0{,}18 \cdot (\mathrm{pH} - 7)$. Bij pH 5,8 is dat $2{,}716\ \mathrm{V}$, of ADC-waarde 556 op de 10-bits ADC van de Mega. Eén stap van de ADC (4,9 mV) is $0{,}027$ pH: ruim fijn genoeg voor een band van 5,5 tot 6,5. De EC-module geeft 0,50 V per mS/cm (rekenwaarde van dit project): één ADC-stap is $0{,}010\ \mathrm{mS/cm}$. Versie 5 rekent de EC om naar 25 °C met de temperatuur van de NTC (2 % per graad, hoofdstuk 13).

Bijvullen met regenwater alleen zou de EC doen zakken: 13 L water op 45 L oplossing verdunt met ±23 %. Met 10 mL A en 10 mL B per liter krijgt het nieuwe water dezelfde samenstelling. De pH stijgt ondertussen elke dag met ±0,1 à 0,15 (hoofdstuk 15), omdat de planten meer nitraat opnemen dan ze afgeven: in de testweek van 5,86 naar 6,59.

NASLAG · 40.13

De module in kerngetallen

13/28

Na fase 2 vat het voorbeeldteam de module samen op één blad: de technische fiche. Ze komt vooraan in het rapport en naast de module aan de muur. Elk getal heeft een herkomst: het hoofdstuk waar het berekend of beslist werd.

deelsysteem	kerngetallen	hoofdstuk
kweekruimte	binnen 2,40 × 2,00 m, wanden 1,93 en 2,87 m hoog (11,5 m³); vloer 0,40 m boven het terrein; één wand grenst aan het station	6, 14, 19
kasdak	polycarbonaat 16 mm (U = 2,0), 5,30 m² onder 25°; schuifpaneel 10 kg op rails, slag 0,60 m, opening aan de nok	7, 8, 19
schil	wanden U = 0,22, vloer U = 0,21; H = 20,2 W/K (16,3 met nachtscherm): 567 W bij −8 °C buiten en 20 °C binnen	19
kweekbed	4 NFT-goten van 2,4 m, 48 plaatsen, 2,88 m², helling 1 : 50; om de 8,75 dagen een goot oogsten	4, 36
water	reservoir 60 L (25 cm, 2,4 L per cm), alarm onder 10 cm; pomp 12 W, 4,8 L/min bij 1,56 m, 10 min per uur; 1,2 L/min per goot	9, 10, 11, 36
voedingsoplossing	1,30 g/L zouten; EC 1,4 (1,2-1,8 mS/cm), pH 5,8 (5,5-6,5), 18-22 °C; stamoplossing A en B: 10 mL per L	12, 13, 15, 36
warmte	buffer 200 L + 60 L (1,09 MJ/K), lucht 80 kJ/K; convector 800 W; 18/19 °C overdag, 16/17 °C 's nachts	18, 19, 31
ventilatie en dak	11 m³/h (PWM 90), 's nachts PWM 36; vol boven 25 °C (+40 W/K); dak open boven 27 °C (+60 W/K)	14, 26, 31
licht	2 × 95 W, 200 µmol/(m²·s) van 6 tot 22 uur (UTC+1): 11,5 mol per dag; 3,0 µmol/J op het bed; gedimd met het daglicht	23, 25, 36
zon	4 × 350 Wp = 1,40 kWp, 60° zuid, op het dak van de woonmodule; MPPT 100 A; ±1397 kWh per jaar mogelijk	6, 20
batterijen	2 × LiFePO4 12,8 V 120 Ah: 2,46 kWh bruikbaar, BMS, in de verwarmde module; laadtoestand met twee INA226	16, 20, 30

deelsysteem	kerngetallen	hoofdstuk
energiebeheer	vanaf 40 % volgens schema; 20-40 % lampen alleen op zonne-overschot; onder 20 % lampen uit; onder 10 % alarm	20, 29
verbruik	zonder dimmen 3,27 kWh per dag; met dimmen 1,1 tot 2,1 kWh per dag; ±540 kWh per jaar	20, 25, 40
besturing	Arduino Mega 2560, klimaat.ino versie 5 (510 regels) met pinnen.h en dak.h/.cpp; regelkast RK-1; sensorkast SK-1 op 3 m	29-32, 37, 40
data	CSV per minuut, 17 kolommen, ±121 kB per dag, klok op UTC+1	30, 40
teelt	48 kroppen van ±200 g in 35 dagen; ±100 kg sla per jaar	34, 36
voetafdruk	±164 kg CO_2-eq per jaar, ±1,64 kg per kg sla	38

UITLEG · 40.14

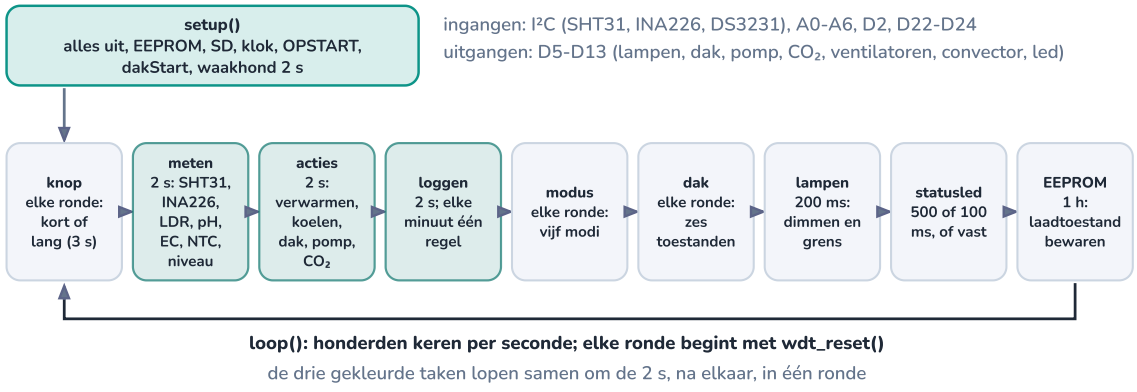
Fase 3 · de besturing: vijf versies worden één programma

De code van de module groeide over vier hoofdstukken, telkens met iets erbij en soms met iets eruit om het nieuwe duidelijk te tonen. Versie 4 (hoofdstuk 32) liet het dimmen, het energiebeheer en de logger weg. In fase 3 worden ze samengevoegd tot **versie 5**.

versie	hoofdstuk	wat erbij kwam	wat er nog ontbrak
1	29	pinnen.h, zeven taken met millis(), dimmen, pomp 10 min per uur, dak met de knop, laadtoestand in het EEPROM, energiebeheer	echte klok, logger, regeling
2	30	klok DS3231 (UTC+1), logger op SD en USB, pomp in minuut 0-9, meldingen met #, INA226	verwarming, ventilatie, modi
3	31	verwarming en ventilatoren met hysterese, noodprogramma bij een sensorfout	modi, dak als toestandsmachine
4	32	modi OPSTART, DAG, NACHT, ALARM, ONDERHOUD; dak.h en dak.cpp; alarmbits; CO_2-klep	dimmen, energiebeheer en logger (weggelaten)
taak	39	slapVerwacht(): een extra pompbeurt van 5 min om het halfuur	een eigen taak die ook pin 9 stuurt
5	40	alles in één programma; laadtoestand met twee INA226; pH, EC en watertemperatuur; één eigenaar per uitgang; waakhond	zie 'wat de module nog niet kan'

Samenvoegen is meer dan knippen en plakken. Vier regels hielden het programma overzichtelijk:

1. **Eén klok.** millis() voor de intervallen (om de 2 s meten, om de 200 ms de lampen), de DS3231 voor het uur van de dag, en alles in UTC+1 (hoofdstuk 30).
2. **Nergens delay().** Elke taak kijkt of het haar beurt is en keert meteen terug (hoofdstuk 29). Dan blijft elke ronde kort, en kan de waakhond na 2 s ingrijpen als er toch iets blijft hangen (hoofdstuk 32).
3. **Eén eigenaar per uitgang.** Alleen acties() beslist over de pomp; de extra beurt van hoofdstuk 39 is een voorwaarde in die beslissing, geen tweede taak die op pin 9 schrijft. Zo ook: alleen taakLampen() schrijft naar D5, alleen dak.cpp naar de H-brug.
4. **De modi als kader.** Wat een taak doet, hangt af van de modus (volgende blad). Een nieuwe functie krijgt dus altijd de vraag: wat doet ze in OPSTART, in ALARM, in ONDERHOUD?



Na setup() draait loop() rond; de drie gekleurde taken lopen om de 2 s, de andere bij elke ronde of op hun eigen ritme.

UITLEG · 40.15

De modi en het dak als toestandsmachines

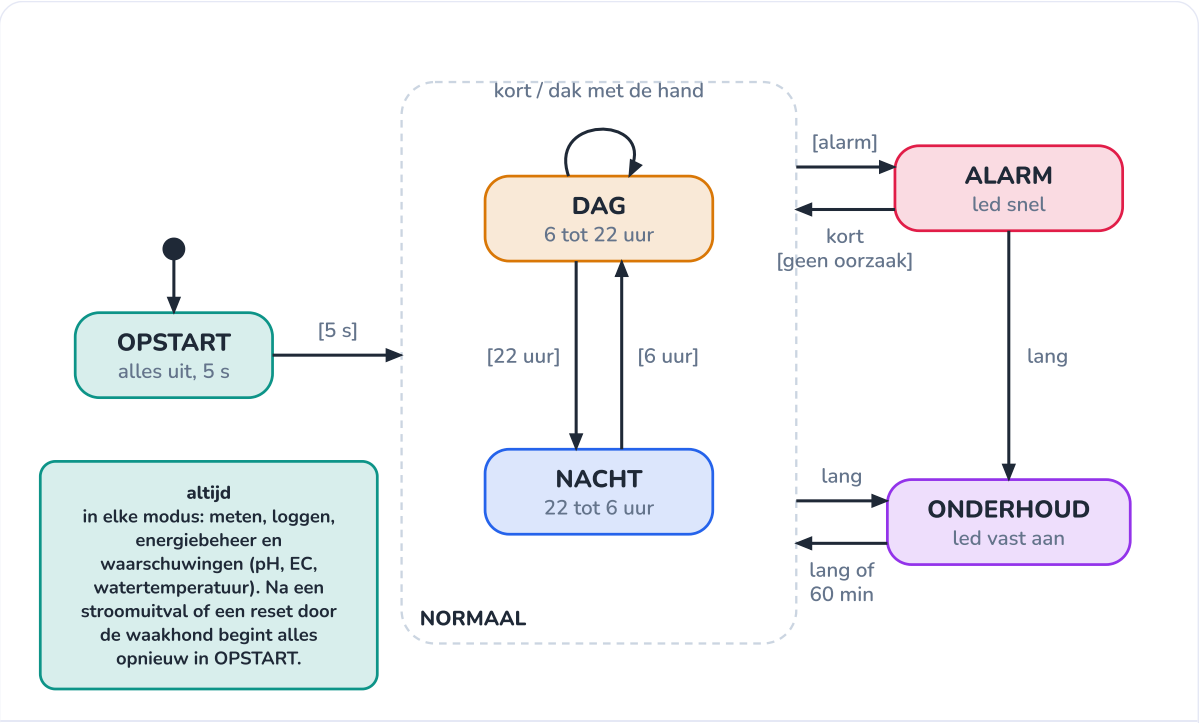
15/28

De modi komen ongewijzigd uit hoofdstuk 32: OPSTART na elke herstart, DAG en NACHT volgens de klok, ALARM bij een oorzaak die meteen ingrijpen vraagt, ONDERHOUD met een lange druk op S1. Versie 5 voegt er één alarm aan toe (batterij onder 10 %) en drie **waarschuwingen** die geen modus zijn: pH buiten 5,5-6,5, EC buiten 1,2-1,8 en een voedingsoplossing boven 24 °C. Een waarschuwing vraagt actie vandaag, geen actie nu; ze komt als melding en in de logger. Waarom niet gewoon alarm? Een pH van 6,6 schaadt de sla niet in een uur, en een module die om de haverklap in ALARM staat, leert zijn bewoners het alarm te negeren.

Wat elke taak doet, hangt af van de modus:

modus	verwarming	ventilatoren	lampen	pomp	dak	CO_2
OPSTART	uit	laag	uit	uit	dakStart: eindeloop lezen	dicht
DAG	18/19 °C	90; vol boven 25 °C	dimmen, grens	min. 0-9 + extra	volgt 27/25 °C	30 s per halfuur, dak dicht
NACHT	16/17 °C	36; vol boven 25 °C	uit	min. 0-9	dicht	dicht
ALARM	zoals DAG of NACHT	zoals DAG of NACHT	zoals DAG	niet bij vlotter of niveau	overdag automatisch	dicht
ONDERHOUD	zoals DAG of NACHT	vol (verse lucht)	werklicht 60	uit	geblokkeerd	dicht

Het **dak** blijft een eigen toestandsmachine in dak.h en dak.cpp (hoofdstuk 32): DICHT, OPENEN, OPEN, SLUITEN, GESTOPT en DAK_FOUT. De klimaatregeling vraagt alleen 'open' of 'dicht'; dak.cpp beslist wanneer de motor draait, stopt op de eindeloop en geeft DAK_FOUT als het eindpunt na 12 s niet bereikt is. Bij het binnenkomen van DAG volgt het dak meteen de temperatuur: na een herstart overdag sluit het dus eerst, en gaat het opnieuw open zodra het warmer is dan 27 °C.



NORMAAL omvat DAG en NACHT: een overgang vanuit het kader geldt voor beide. Waarschuwingen zijn geen modus.

CODE · 40.16

klimaat.ino versie 5

16/28

Het volledige programma, 510 regels. pinnen.h en dak.h/dak.cpp zijn dezelfde bestanden als in hoofdstuk 32 (hieronder te downloaden). Lees het van boven naar onder: de instellingen verwijzen naar de eisen (K1, L1, ...), elk blok begint met een kopregel, en de volgorde in loop() is de volgorde van het blokschema.

C++ (Arduino) **klimaat.ino** 510 regels

```
// klimaat.ino - de sturing van de klimaatmodule, versie 5 (Arduino Mega 2560, hoofdstuk 40)
// Samengevoegd: klok, logger, dimmen en energiebeheer (versie 2, hoofdstuk 30), de klimaatregeling (hoofdstuk 31),
// de modi en het kasdak als toestandsmachines (versie 4, hoofdstuk 32) en de extra pompbeurt (hoofdstuk 39).
// Nieuw: de laadtoestand telt met de twee INA226, pH, EC en de temperatuur van de voedingsoplossing worden bewaakt,
// één taak beslist over de pomp, 's nachts draaien de ventilatoren trager, het niveau telt alleen als de pomp
// stilstaat, en een waakhond van 2 s. I2C: INA226 0x40 en 0x41, SHT31 0x44, DS3231 0x68 (wintertijd).
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <EEPROM.h>
#include <RTClib.h>
#include <Adafruit_SHT31.h>
#include <avr/wdt.h>
#include "pinnen.h"
#include "dak.h"

// ---------------------------------------------------------------- instellingen (programma van eisen, hoofdstuk 40)
const unsigned long MEET_MS = 2000;                 // meten en beslissen
const unsigned long LAMP_MS = 200;                  // lampen: hoogstens één stap per 200 ms (L4)
const unsigned long LANG_MS = 3000;                 // lange druk op S1: onderhoud aan of uit (B4)
const unsigned long ONDERHOUD_MAX_MS = 3600000UL;
const unsigned long BEWAAR_MS = 3600000UL;          // laadtoestand elk uur in het EEPROM (B1)
const unsigned long POMP_RUST_MS = 120000UL;        // niveau pas 2 min na het stoppen van de pomp (W3)
const float DAG_LAAG = 18.0, NACHT_LAAG = 16.0;     // verwarming aan onder LAAG, uit boven LAAG + 1 (K1, K2)
const float KOEL_AAN = 25.0, KOEL_UIT = 24.0;       // ventilatoren vol (K4)
const float DAK_OPEN = 27.0, DAK_DICHT = 25.0;      // kasdak (K4)
const int VENT_DAG = 90, VENT_NACHT = 36, VENT_VOL = 255, WERKLICHT = 60;
const float DOEL = 200.0, LAMPEN_VOL = 200.0, LUX_PER_UMOL = 54.0;  // umol/(m2 s) op het kweekbed (L1, L2)
const float SOC_VOL = 40.0, SOC_ZON = 20.0, SOC_ALARM = 10.0;       // energiebeheer (L3, E5)
```

```ino
const float CAPACITEIT_AH = 240.0;                  // twee LiFePO4 van 120 Ah (hoofdstuk 20)
const float PH_LAAG = 5.5, PH_HOOG = 6.5, EC_LAAG = 1.2, EC_HOOG = 1.8, WATER_WARM = 24.0;   // H1-H3
const float SLAP_LICHT = 25.0, SLAP_VOCHT = 60.5;    // extra pompbeurt: klx en % (hoofdstuk 39)
const int SD_CS = 53;
const byte INA_ZON = 0x40, INA_VERBRUIK = 0x41;
const float R_SHUNT = 0.00075;                       // shunt 100 A / 75 mV
const int ADRES_SOC = 0;
const char KOP[] = "tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V,soc_pct,lampen_pwm,"
                   "verw_pct,dak,water_C,pH,EC_mScm,niveau_cm,modus,alarm";

enum Modus { OPSTART, DAG, NACHT, ALARM, ONDERHOUD };
const char* NAAM[] = {"OPSTART", "DAG", "NACHT", "ALARM", "ONDERHOUD"};
const byte A_SENSOR = 1, A_WATER = 2, A_NIVEAU = 4, A_DAK = 8, A_BATTERIJ = 16;   // alarm: meteen ingrijpen
const byte W_PH = 1, W_EC = 2, W_WARM = 4, W_SD = 8;                              // waarschuwing: vandaag bijsturen

// ---------------------------------------------------------------- toestand
Adafruit_SHT31 sht;
RTC_DS3231 rtc;
DateTime klok, minuut;
Modus modus = OPSTART;
byte alarmen = 0, waarschuwingen = 0;
unsigned long modusSinds = 0, vorigeMeting = 0, vorigeLamp = 0, vorigeBewaring = 0, ledSinds = 0;
unsigned long knopSinds = 0, droogSinds = 0, pompSinds = 0;
float luchtT = NAN, luchtRV = NAN, lux = 0, uBat = 0, uBus = NAN, iZon = NAN, iVerbruik = NAN;
float soc = 50.0, waterT = NAN, ph = NAN, ec = NAN, niveau = 0, gemAan = 0.5;
int geenMeting = 0, grens = 255, stand = 0;
bool verwarmen = false, koelen = false, warm = false, laden = false, pompAan = false;
bool extraBeurt = false, extraBeslist = false, niveauLaag = false, batLeeg = false;
bool knopWas = false, langGemeld = false, ledAan = false, kaartOk = false, bezig = false;
float somT, somRV, somLux, somZon, somVerbruik, somU, somWT, somPH, somEC, somNiveau, somStand;   // logger
int nT, nMeting, nVermogen, nWT, nPH, nVerw;
byte eindModus = OPSTART, eindAlarm = 0, eindDak = 0;
float eindSoc = 50.0;
char opdracht[32];
int lengte = 0;

// ---------------------------------------------------------------- meldingen
void printTwee(int x) {
  if (x < 10) Serial.print('0');
  Serial.print(x);
}

void melding(const char* tekst) {                    // "# 14:02:31 tekst" (ook dak.cpp gebruikt ze)
  Serial.print("# ");
  printTwee(klok.hour());
  Serial.print(':');
  printTwee(klok.minute());
  Serial.print(':');
  printTwee(klok.second());
  Serial.print(' ');
  Serial.println(tekst);
}

// ---------------------------------------------------------------- sensoren (hoofdstuk 11, 13, 15, 22, 30)
int gemiddeld(int pin) {                             // 16 metingen middelen tegen ruis
  long som = 0;
  for (int i = 0; i < 16; i++) som += analogRead(pin);
  return som / 16;
}

float daglichtLux() {                                // LDR onder een kapje, 1,0 kOhm naar GND
  int n = gemiddeld(LDR);
  if (n < 1) n = 1;
  float r = 1000.0 * (1024.0 - n) / n;
  return 10.0 * pow(r / 15000.0, -1.0 / 0.7) / 0.10;
}

float batterijspanning() {                           // deler 100 kOhm / 33 kOhm
```

```
  return gemiddeld(BATTERIJ) * 5.0 / 1024.0 * (133.0 / 33.0);
}

float waterTemperatuur() {                          // NTC in het reservoir, B = 3913 K
  int n = gemiddeld(NTC);
  if (n < 5 || n > 1018) return NAN;                // draad los of kortsluiting
  float r = 10000.0 * (1024.0 - n) / n;
  return 1.0 / (1.0 / 298.15 + log(r / 10000.0) / 3913.0) - 273.15;
}

float phWaarde() {                                  // 2,50 V bij pH 7,00, -0,18 V per pH (hoofdstuk 15)
  float u = gemiddeld(PH) * 5.0 / 1024.0;
  if (u < 0.1 || u > 4.9) return NAN;
  return 7.0 + (2.50 - u) / 0.18;
}

float ecWaarde(float t) {                           // 0,50 V per mS/cm, omgerekend naar 25 graden (hoofdstuk 13)
  float e = gemiddeld(EC) * 5.0 / 1024.0 / 0.50;
  if (isnan(t)) return e;
  return e / (1.0 + 0.02 * (t - 25.0));
}

float niveauCm() {                                  // MPX5010 onderaan het reservoir (hoofdstuk 11)
  float u = gemiddeld(DRUK) * 5.0 / 1024.0;
  float p = (u / 5.0 - 0.04) / 0.09 * 1000.0;       // Pa
  return p / (1000.0 * 9.81) * 100.0;
}

long inaLees(byte adres, byte reg) {                // INA226: 16 bits uit een register, of -1
  Wire.beginTransmission(adres);
  Wire.write(reg);
  if (Wire.endTransmission() != 0) return -1;
  if (Wire.requestFrom(adres, (byte)2) != 2) return -1;
  byte hoog = Wire.read();
  byte laag = Wire.read();
  return ((long)hoog << 8) | laag;
}

float inaStroom(byte adres) {                       // A: shuntspanning (2,5 uV per stap) gedeeld door 0,75 mOhm
  long s = inaLees(adres, 0x01);
  if (s < 0) return NAN;
  return (int16_t)s * 0.0000025 / R_SHUNT;
}

float inaSpanning(byte adres) {                     // V: busspanning, 1,25 mV per stap
  long b = inaLees(adres, 0x02);
  if (b < 0) return NAN;
  return b * 0.00125;
}

byte band(byte w, byte bit, float x, float laag, float hoog, float marge) {   // waarschuwing met hysterese
  if (isnan(x)) return w;
  if (x < laag || x > hoog) return w | bit;
  if (x > laag + marge && x < hoog - marge) return w & ~bit;
  return w;
}

// ------------------------------------------------------------------ meten, laadtoestand, alarmen (om de 2 s)
void meten(unsigned long nu) {
  float t, rv;
  if (sht.readBoth(&t, &rv)) { luchtT = t; luchtRV = rv; geenMeting = 0; }
  else { luchtT = luchtRV = NAN; geenMeting++; }
  lux = daglichtLux();
  uBat = batterijspanning();
  iZon = inaStroom(INA_ZON);
  iVerbruik = inaStroom(INA_VERBRUIK);
  uBus = inaSpanning(INA_VERBRUIK);
  if (!isnan(iZon) && !isnan(iVerbruik)) {          // coulombteller (hoofdstuk 16) met de twee INA226
```

```
    float iLaad = iZon - iVerbruik;                // A, positief = de batterijen laden
    soc += iLaad * (MEET_MS / 3600000.0) / CAPACITEIT_AH * 100.0;
    if (uBat >= 14.2 && iLaad > 0.0 && iLaad < 2.0) soc = 100.0;    // vol: opnieuw ijken
    soc = constrain(soc, 0.0, 100.0);
    laden = iLaad > 0.0;
  }
  if (soc >= SOC_VOL) grens = 255;                 // energiebeheer (hoofdstuk 20)
  else if (soc < SOC_ZON) grens = 0;
  else if (laden) grens = min(grens + 5, 255);     // er is zon over: meer licht mag
  else grens = max(grens - 5, 0);                  // de batterij levert: minder licht
  waterT = waterTemperatuur();
  ph = phWaarde();
  ec = ecWaarde(waterT);
  if (digitalRead(VLOTTER) == LOW) droogSinds = nu;  // LOW = genoeg water; een losse draad = droog
  if (!pompAan && nu - pompSinds >= POMP_RUST_MS) niveau = niveauCm();   // water in de goten telt niet mee
  if (!niveauLaag && niveau < 10.0) niveauLaag = true;                // 10 en 12 cm
  else if (niveauLaag && niveau > 12.0) niveauLaag = false;
  if (!batLeeg && soc < SOC_ALARM) batLeeg = true;                    // 10 en 15 %
  else if (batLeeg && soc > SOC_ALARM + 5.0) batLeeg = false;
  byte nieuw = 0;
  if (geenMeting >= 15) nieuw |= A_SENSOR;         // 30 s geen luchttemperatuur
  if (nu - droogSinds >= 10000) nieuw |= A_WATER;  // 10 s droog
  if (niveauLaag) nieuw |= A_NIVEAU;
  if (dak == DAK_FOUT) nieuw |= A_DAK;
  if (batLeeg) nieuw |= A_BATTERIJ;
  if (nieuw & ~alarmen & A_SENSOR) melding("oorzaak: geen temperatuur");
  if (nieuw & ~alarmen & A_WATER) melding("oorzaak: vlotter droog");
  if (nieuw & ~alarmen & A_NIVEAU) melding("oorzaak: reservoir onder 10 cm");
  if (nieuw & ~alarmen & A_DAK) melding("oorzaak: dak haalt zijn eindpunt niet");
  if (nieuw & ~alarmen & A_BATTERIJ) melding("oorzaak: batterij onder 10 %");
  alarmen = nieuw;
  byte w = band(waarschuwingen, W_PH, ph, PH_LAAG, PH_HOOG, 0.05);
  w = band(w, W_EC, ec, EC_LAAG, EC_HOOG, 0.05);
  w = band(w, W_WARM, waterT, 0.0, WATER_WARM, 0.5);
  if (w & ~waarschuwingen & W_PH) melding("waarschuwing: pH buiten de band, bijsturen");
  if (w & ~waarschuwingen & W_EC) melding("waarschuwing: EC buiten de band, bijsturen");
  if (w & ~waarschuwingen & W_WARM) melding("waarschuwing: voedingsoplossing te warm");
  waarschuwingen = w | (kaartOk ? 0 : W_SD);
}

// ---------------------------------------------------------------- de knop: kort of lang (hoofdstuk 24, 32)
void leesKnop(unsigned long nu, bool& kort, bool& lang) {
  bool in = digitalRead(KNOP) == HIGH;             // na de 74HC14: hoog = ingedrukt
  kort = lang = false;
  if (in && !knopWas) { knopSinds = nu; langGemeld = false; }
  if (in && !langGemeld && nu - knopSinds >= LANG_MS) { lang = true; langGemeld = true; }
  if (!in && knopWas && !langGemeld) kort = true;
  knopWas = in;
}

// ---------------------------------------------------------------- de modi (hoofdstuk 32)
bool isDag() { return klok.hour() >= 6 && klok.hour() < 22; }
Modus gewoon() { return isDag() ? DAG : NACHT; }

void naar(Modus nieuw, unsigned long nu) {
  Modus oud = modus;
  modus = nieuw;
  modusSinds = nu;
  char tekst[32] = "modus ";
  strcat(tekst, NAAM[nieuw]);
  melding(tekst);
  if (nieuw == ONDERHOUD) dakBlokkeer(true, nu);     // niemand onder een bewegend dak
  if (oud == ONDERHOUD) dakBlokkeer(false, nu);
  if (nieuw == NACHT) dakVraag(false);
  if (nieuw == DAG) dakVraag(warm);
}

void taakModus(unsigned long nu, bool kort, bool lang) {
```

```
  switch (modus) {
    case OPSTART:                                  // 5 s: sensoren en klok lezen
      if (nu - modusSinds >= 5000) naar(gewoon(), nu);
      break;
    case DAG:
    case NACHT:
      if (alarmen) naar(ALARM, nu);
      else if (lang) naar(ONDERHOUD, nu);
      else if (gewoon() != modus) naar(gewoon(), nu);
      else if (kort) dakKnop(nu);                  // korte druk: het dak met de hand (G2)
      break;
    case ALARM:
      if (kort) {                                  // bevestigen
        if (alarmen & A_DAK) {
          dakHerstel(nu);
          alarmen &= ~A_DAK;
        }
        if (alarmen == 0) naar(gewoon(), nu);
        else melding("alarm blijft: oorzaak nog aanwezig");
      } else if (lang) naar(ONDERHOUD, nu);
      break;
    case ONDERHOUD:
      if (lang || nu - modusSinds >= ONDERHOUD_MAX_MS) naar(gewoon(), nu);
      break;
  }
}

// ------------------------------------------------------------------ klimaat, pomp en CO2 (om de 2 s)
void acties(unsigned long nu) {
  bool dag = isDag();
  float laag = dag ? DAG_LAAG : NACHT_LAAG, hoog = laag + 1.0;
  if (isnan(luchtT)) {                             // noodprogramma: blind met de gemiddelde inschakelduur
    verwarmen = (nu / 1000UL) % 600UL < (unsigned long)(gemAan * 600);
    koelen = false;
  } else {
    if (!verwarmen && luchtT < laag) verwarmen = true;
    else if (verwarmen && luchtT > hoog) verwarmen = false;
    if (!koelen && luchtT > KOEL_AAN) koelen = true;
    else if (koelen && luchtT < KOEL_UIT) koelen = false;
    if (verwarmen) koelen = false;
    gemAan += ((verwarmen ? 1.0 : 0.0) - gemAan) * (MEET_MS / 3600000.0);
    bool automatisch = dag && (modus == DAG || (modus == ALARM && !(alarmen & A_DAK)));
    if (!warm && luchtT > DAK_OPEN) { warm = true; if (automatisch) dakVraag(true); }
    else if (warm && luchtT < DAK_DICHT) { warm = false; if (automatisch) dakVraag(false); }
  }
  if (modus == ONDERHOUD) koelen = true;           // verse lucht voor wie binnen werkt
  bool verw = verwarmen && modus != OPSTART;
  digitalWrite(VERWARMING, verw ? HIGH : LOW);
  if (verw) nVerw++;
  analogWrite(VENTILATOREN, koelen ? VENT_VOL : (dag ? VENT_DAG : VENT_NACHT));
  // de pomp: één taak beslist over pin 9 (minuut 0-9 van elk uur, extra beurt om het halfuur)
  int m = klok.minute();
  if (m == 30 && !extraBeslist) {
    extraBeslist = true;
    extraBeurt = dag && lux / 1000.0 > SLAP_LICHT && luchtRV <= SLAP_VOCHT;
    if (extraBeurt) melding("extra pompbeurt: slappe sla verwacht");
  }
  if (m < 30 || m >= 35) extraBeurt = extraBeslist = false;
  bool pomp = (modus == DAG || modus == NACHT || modus == ALARM) && (m < 10 || extraBeurt)
              && !(alarmen & (A_WATER | A_NIVEAU));
  if (pomp != pompAan) { pompAan = pomp; pompSinds = nu; }
  digitalWrite(POMP, pomp ? HIGH : LOW);
  // CO2: alleen in DAG, met het dak dicht en zonder koelen: om het halfuur 30 s (K8)
  bool co2 = modus == DAG && dak == DICHT && !koelen && m % 30 == 0 && klok.second() < 30;
  digitalWrite(CO2_KLEP, co2 ? HIGH : LOW);
}

// ------------------------------------------------------------------ groeilampen: dimmen en energiebeheer
```

```
void taakLampen(unsigned long nu) {
  if (nu - vorigeLamp < LAMP_MS) return;
  vorigeLamp += LAMP_MS;
  int doel = 0;
  if (modus == ONDERHOUD) doel = WERKLICHT;
  else if ((modus == DAG || modus == ALARM) && isDag()) {
    float f = constrain((DOEL - lux / LUX_PER_UMOL) / LAMPEN_VOL, 0.0, 1.0);
    doel = min((int)(f * 255 + 0.5), grens);
  }
  if (doel > stand) stand++;
  else if (doel < stand) stand--;
  analogWrite(LAMPEN, stand);
}

// ------------------------------------------------------------- logger (hoofdstuk 30): elke minuut één regel
void wisSommen() {
  somT = somRV = somLux = somZon = somVerbruik = somU = somWT = somPH = somEC = somNiveau = somStand = 0;
  nT = nMeting = nVermogen = nWT = nPH = nVerw = 0;
}

size_t schrijfRegel(Print& uit) {
  size_t n = 0;
  char tijd[17];
  snprintf(tijd, sizeof(tijd), "%04d-%02d-%02d %02d:%02d", minuut.year(), minuut.month(), minuut.day(),
           minuut.hour(), minuut.minute());
  n += uit.print(tijd);
  n += uit.print(',');
  if (nT > 0) n += uit.print(somT / nT, 1);
  n += uit.print(',');
  if (nT > 0) n += uit.print(somRV / nT, 1);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somVerbruik / nVermogen / 60.0, 2);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somZon / nVermogen / 60.0, 2);
  n += uit.print(',');
  if (nMeting > 0) n += uit.print(somLux / nMeting, 0);
  n += uit.print(',');
  if (nVermogen > 0) n += uit.print(somU / nVermogen, 2);
  n += uit.print(',');
  n += uit.print(eindSoc, 1);
  n += uit.print(',');
  if (nMeting > 0) n += uit.print(somStand / nMeting, 0);
  n += uit.print(',');
  if (nMeting > 0) n += uit.print(100.0 * nVerw / nMeting, 0);
  n += uit.print(',');
  n += uit.print(eindDak);
  n += uit.print(',');
  if (nWT > 0) n += uit.print(somWT / nWT, 1);
  n += uit.print(',');
  if (nPH > 0) n += uit.print(somPH / nPH, 2);
  n += uit.print(',');
  if (nPH > 0) n += uit.print(somEC / nPH, 2);
  n += uit.print(',');
  if (nMeting > 0) n += uit.print(somNiveau / nMeting, 1);
  n += uit.print(',');
  n += uit.print(eindModus);
  n += uit.print(',');
  n += uit.print(eindAlarm);
  n += uit.println();
  return n;
}

void opslaan() {
  size_t lang = schrijfRegel(Serial);
  if (!kaartOk) return;
  char naam[20];
  snprintf(naam, sizeof(naam), "LOG/%04d%02d%02d.CSV", minuut.year(), minuut.month(), minuut.day());
  bool nieuw = !SD.exists(naam);
```

```
  File f = SD.open(naam, FILE_WRITE);
  if (!f) {
    melding("waarschuwing: SD, kan het bestand niet openen");
    kaartOk = false;
    return;
  }
  if (nieuw) f.println(KOP);
  if (schrijfRegel(f) < lang) {
    melding("waarschuwing: SD-kaart vol");
    kaartOk = false;
  }
  f.close();
}

void loggen() {                                  // om de 2 s, na meten() en acties()
  DateTime begin(klok.year(), klok.month(), klok.day(), klok.hour(), klok.minute(), 0);
  if (!bezig) {
    minuut = begin;
    bezig = true;
  } else if (begin.unixtime() != minuut.unixtime()) {
    opslaan();
    wisSommen();
    minuut = begin;
  }
  if (!isnan(luchtT)) { somT += luchtT; somRV += luchtRV; nT++; }
  if (!isnan(iZon) && !isnan(iVerbruik) && !isnan(uBus)) {
    somZon += iZon * uBus;
    somVerbruik += iVerbruik * uBus;
    somU += uBus;
    nVermogen++;
  }
  if (!isnan(waterT)) { somWT += waterT; nWT++; }
  if (!isnan(ph) && !isnan(ec)) { somPH += ph; somEC += ec; nPH++; }
  somLux += lux;
  somStand += stand;
  somNiveau += niveau;
  nMeting++;
  eindModus = modus;
  eindAlarm = alarmen;
  eindDak = dak;
  eindSoc = soc;
}

void leesOpdracht() {                            // T2021-03-08 09:00:00 + Enter: de klok zetten
  while (Serial.available() > 0) {
    char c = Serial.read();
    if (c == '\n' || c == '\r') {
      opdracht[lengte] = '\0';
      int j, mo, d, u, mi, s;
      if (lengte > 0 && sscanf(opdracht, "T%d-%d-%d %d:%d:%d", &j, &mo, &d, &u, &mi, &s) == 6) {
        rtc.adjust(DateTime(j, mo, d, u, mi, s));
        klok = rtc.now();
        melding("klok gezet");
        wisSommen();
        bezig = false;
      }
      lengte = 0;
    } else if (lengte < (int)sizeof(opdracht) - 1) {
      opdracht[lengte] = c;
      lengte++;
    }
  }
}

// ---------------------------------------------------------------- statusled en EEPROM
void taakStatusled(unsigned long nu) {           // normaal 500 ms, alarm 100 ms, onderhoud vast aan (G1)
  if (modus == ONDERHOUD) { digitalWrite(STATUSLED, HIGH); return; }
  unsigned long periode = (modus == ALARM) ? 100 : 500;
```

```
  if (nu - ledSinds < periode) return;
  ledSinds = nu;
  ledAan = !ledAan;
  digitalWrite(STATUSLED, (ledAan && modus != OPSTART) ? HIGH : LOW);
}

void taakBewaren(unsigned long nu) {                // de laadtoestand overleeft een stroomonderbreking
  if (nu - vorigeBewaring < BEWAAR_MS) return;
  vorigeBewaring += BEWAAR_MS;
  EEPROM.put(ADRES_SOC, soc);                       // schrijft alleen de bytes die veranderen
}

// ------------------------------------------------------------- setup en loop
void setup() {
  wdt_disable();                                    // misschien zijn we net door de waakhond herstart
  const int UIT[] = {LAMPEN, PWM_DICHT, PWM_OPEN, DAK_ENABLE, POMP, CO2_KLEP, VENTILATOREN, VERWARMING, STATUSLED};
  for (int p : UIT) { pinMode(p, OUTPUT); digitalWrite(p, LOW); }   // veilig starten: alles uit
  digitalWrite(DAK_ENABLE, HIGH);
  pinMode(VLOTTER, INPUT_PULLUP);
  pinMode(KNOP, INPUT);
  pinMode(EIND_OPEN, INPUT_PULLUP);
  pinMode(EIND_DICHT, INPUT_PULLUP);
  Serial.begin(115200);
  Wire.begin();
  rtc.begin();
  sht.begin(0x44);                                  // antwoordt ze niet, dan volgt het alarm na 30 s
  klok = rtc.now();
  byte ina[] = {INA_ZON, INA_VERBRUIK};
  for (int i = 0; i < 2; i++) {                     // INA226: 128 metingen middelen (282 ms)
    Wire.beginTransmission(ina[i]);
    Wire.write(0x00);
    Wire.write(0x49);
    Wire.write(0x27);
    Wire.endTransmission();
  }
  kaartOk = SD.begin(SD_CS);
  if (kaartOk && !SD.exists("LOG")) SD.mkdir("LOG");
  EEPROM.get(ADRES_SOC, soc);
  if (isnan(soc) || soc < 0.0 || soc > 100.0) soc = 50.0;   // leeg EEPROM: een ruwe schatting
  eindSoc = soc;
  niveau = niveauCm();
  unsigned long nu = millis();
  droogSinds = nu;
  melding("klimaatmodule v5 (hoofdstuk 40)");
  Serial.print("# laadtoestand uit het EEPROM: ");
  Serial.print(soc, 1);
  Serial.println(" %");
  if (!kaartOk) melding("waarschuwing: geen SD-kaart");
  Serial.println(KOP);
  naar(OPSTART, nu);
  dakStart(nu);
  wdt_enable(WDTO_2S);                              // komt loop() 2 s niet langs wdt_reset(): herstart (B5)
}

void loop() {
  wdt_reset();
  unsigned long nu = millis();
  leesOpdracht();
  bool kort, lang;
  leesKnop(nu, kort, lang);
  if (nu - vorigeMeting >= MEET_MS) {
    vorigeMeting += MEET_MS;
    klok = rtc.now();
    meten(nu);
    acties(nu);
    loggen();
  }
  taakModus(nu, kort, lang);
```

```
  taakDak(nu);
  taakLampen(nu);
  taakStatusled(nu);
  taakBewaren(nu);
}
```

OOK TE DOWNLOADEN

Wat er nieuw is tegenover versie 2 en 4:

- **De laadtoestand** telt nu met de twee INA226: `iLaad = iZon - iVerbruik`, om de 2 s opgeteld (een coulombteller, hoofdstuk 16). Bij 14,2 V met een kleine laadstroom zijn de batterijen vol en wordt de teller opnieuw op 100 % gezet.
- **Het energiebeheer** zit in meten(): boven 40 % mogen de lampen alles, onder 20 % niets, daartussen schuift de grens 5 stappen op of af naargelang de batterijen laden of leveren (hoofdstuk 29).
- **band()** geeft een waarschuwing met hysterese: aan buiten de band, pas weer uit als de waarde 0,05 (of 0,5 °C) binnen de band zit. Zonder hysterese zou een pH die rond 6,50 schommelt tientallen meldingen geven (hoofdstuk 31).
- **De pomp** wordt in acties() beslist: minuut 0-9, of minuut 30-34 als de beslisboom om 30 minuten slappe sla verwachtte, nooit bij een water- of niveau-alarm en niet in ONDERHOUD. Het niveau telt pas 2 minuten na de pomp.
- **De logger** schrijft 17 kolommen: de 7 van hoofdstuk 28 en 30, en 10 nieuwe achteraan, zodat oude Python-scripts blijven werken (hoofdstuk 30).

Getest. Het programma compileert zonder waarschuwingen met g++ tegen een nagebootste Arduino, ook met int = 16 bit zoals op de Mega. In een gesloten lus met een model van de module draaide het een week: 59 miljoen rondes van loop(), de langste 48,8 ms (eis B6: hoogstens 100 ms; de SHT31 alleen wacht al 20 ms), 432 schrijfbeurten in het EEPROM (alleen bytes die veranderen) en acht dagbestanden op de kaart. De testweek van dit hoofdstuk komt uit dezelfde simulatie, in Python nagebouwd: per dag verschillen verbruik, zon en warmte minder dan 0,5 %.

NASLAG · 40.17

Het logformaat van versie 5

17/28

Eén regel per minuut, op de SD-kaart in LOG/JJJJMMDD.CSV en tegelijk over USB (hoofdstuk 30). Een lege cel betekent: geen geldige meting in die minuut. Nieuwe kolommen komen achteraan, zodat een script dat de eerste zeven kolommen leest, blijft werken.

kolom	eenheid	wat erin staat	sinds
tijdstip	jjjj-mm-dd uu:mm	begin van de minuut, UTC+1	28
temperatuur_C	°C	gemiddelde van de SHT31 (leeg zonder meting)	28
vocht_pct	%	relatieve vochtigheid van de lucht	28
verbruik_Wh	Wh	energie naar de verbruikers in die minuut (INA226 0x41)	28
zon_Wh	Wh	geleverde zonnestroom in die minuut (INA226 0x40)	28
licht_lx	lx	daglicht op de LDR	30
batterij_V	V	busspanning van de rail	30
soc_pct	%	laadtoestand aan het einde van de minuut	40
lampen_pwm	0-255	gemiddelde lampstand	40
verw_pct	%	deel van de minuut met de verwarming aan	40
dak	0-5	DICHT, OPENEN, OPEN, SLUITEN, GESTOPT, DAK_FOUT	40
water_C	°C	voedingsoplossing (NTC)	40
pH		gemiddelde van de minuut	40
EC_mScm	mS/cm	omgerekend naar 25 °C	40

kolom	eenheid	wat erin staat	sinds
niveau_cm	cm	reservoir, gemeten bij stilstaande pomp	40
modus	0-4	OPSTART, DAG, NACHT, ALARM, ONDERHOUD	40
alarm	bits	1 temperatuur, 2 vlotter, 4 niveau, 8 dak, 16 batterij	40

Vijf regels uit de testweek (donderdag, de sensorkast is van 14:02 tot 14:07 losgekoppeld)

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V,soc_pct,lampen_pwm,verw_pct,dak,water_C,
pH,EC_mScm,niveau_cm,modus,alarm
2021-03-11 14:01,25.4,49.6,0.49,0.58,39142,14.30,100.0,0,0,2,21.5,6.21,1.36,21.5,1,0
2021-03-11 14:02,,,0.35,0.43,39112,14.30,100.0,0,0,2,21.5,6.21,1.36,21.5,3,1
2021-03-11 14:03,,,0.34,0.42,39082,14.30,100.0,0,0,2,21.5,6.21,1.36,21.5,3,1
2021-03-11 14:07,26.2,47.6,0.49,0.57,38961,14.30,100.0,0,0,2,21.6,6.21,1.36,21.5,3,0
2021-03-11 14:08,26.2,47.8,0.49,0.58,38930,14.30,100.0,0,0,2,21.6,6.21,1.36,21.5,3,0
```

Om 14:02 zijn temperatuur en vocht leeg, springt de modus naar 3 (ALARM) en staat alarmbit 1 (geen temperatuur) aan. Om 14:07 is de meting terug en het alarmbit uit, maar de modus blijft 3 tot iemand kort op S1 drukt (14:09). Het verbruik zakt in die minuten van 0,49 naar 0,35 Wh per minuut: zonder meting koelt het noodprogramma niet, en de ventilatoren vallen terug.

grootte	waarde
regel	±84 tekens
dagbestand	±121 kB (hoofdstuk 30, met 7 kolommen: 68,5 kB)
kaart van 1 GB	±23 jaar

UITLEG · 40.18

Fase 4 · testen: van deeltest tot testweek

18/28

Testen volgt de rechterkant van het **V-model**: van de delen naar het geheel, en elke test toetst de ontwerpstap die ertegenover staat.

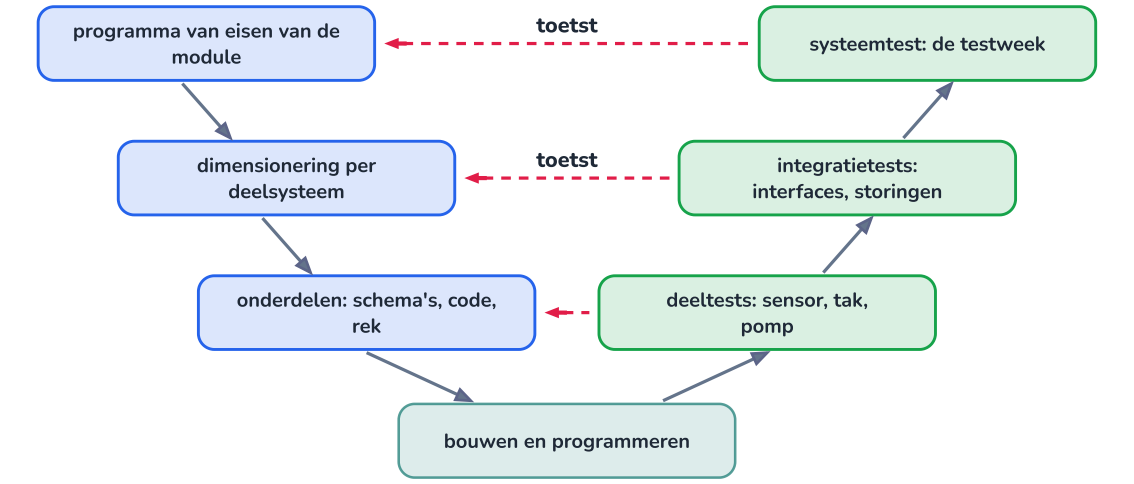
1. **Deeltests**, zodra een onderdeel klaar is: elke sensor naast een referentie, elke tak met de stroomtang, de pomp met een maatbeker per goot, het dak met een blokje in de rail.
2. **Integratietests**: werkt elke interface zoals de tabel zegt? De I²C-scanner vindt vier adressen, de logger schrijft 17 kolommen, de pomp slaat aan in minuut 0 en niet in minuut 11.
3. **Storingstests**: je veroorzaakt een fout en kijkt of het systeem reageert zoals ontworpen. Wat er moet gebeuren, schrijf je vooraf op.
4. **De testweek**: de module draait zeven dagen alleen. Elke dag kijkt één teamlid in de logbestanden, zonder in te grijpen tenzij de planten of de veiligheid het vragen.

storingstest	wanneer	wat moet er gebeuren (vooraf)	wat de logger toonde	eis
stekker van de sensorkast los (5 min)	do 14:02	na 30 s alarm, noodprogramma voor de verwarming, lege cellen	alarm om 14:02:28, 5 regels zonder temperatuur	B2, B3, D2
vlotter met de hand omhoog (1 min)	do 14:15	na 10 s alarm, pomp uit	alarm om 14:15:08	W3, B2
stekker van de regelkast uit (2 min)	do 15:56	alles uit; daarna OPSTART, laadtoestand uit het EEPROM	OPSTART 15:58:00, DAG 15:58:06; 3 regels ontbreken	B1, E5, D1

storingstest	wanneer	wat moet er gebeuren (vooraf)	wat de logger toonde	eis
lange druk op S1: onderhoud	wo 10:12	pomp uit, dak geblokkeerd, werklicht, ventilatoren vol	modus 4 van 10:12 tot 10:29	B4
blokje in de rail van het dak	deeltest, vr	na 12 s DAK_FOUT en alarm; kort drukken: opnieuw sluiten	zie K5 (voor de testweek)	K5
een taak die blijft hangen	deeltest (hoofdstuk 32)	de waakhond herstart de Mega na 2 s	herstart	B5

De simulatie als generale repetitie. Voor de module een week alleen blijft, draait het voorbeeldteam versie 5 in een simulatie: het programma krijgt nagebootste sensoren, en een model van de module (warmte zoals hoofdstuk 31, zon zoals hoofdstuk 20) reageert op wat het programma schakelt. Zo vind je fouten in de code voor ze in de echte module zitten. De testweek van dit hoofdstuk is zo'n simulatie; wie de module echt heeft (spoor A), vervangt ze door de eigen logbestanden.

> ⚠ **Een model is geen meting**
>
> Een simulatie toont wat het programma doet met de module zoals je ze gemodelleerd hebt. Klopt het model niet (een dak dat minder koelt, een sla die meer verdampt), dan klopt de uitkomst ook niet. Schrijf daarom bij elk resultaat of het uit een meting, een berekening of een simulatie komt, en vergelijk het model met echte metingen zodra je ze hebt (E17, E36).



Links ontwerp je van het geheel naar de delen, rechts test je van de delen naar het geheel; elke stippellijn is een toets.

UITLEG · 40.19

De testweek: 8 tot 14 maart 2021

19/28

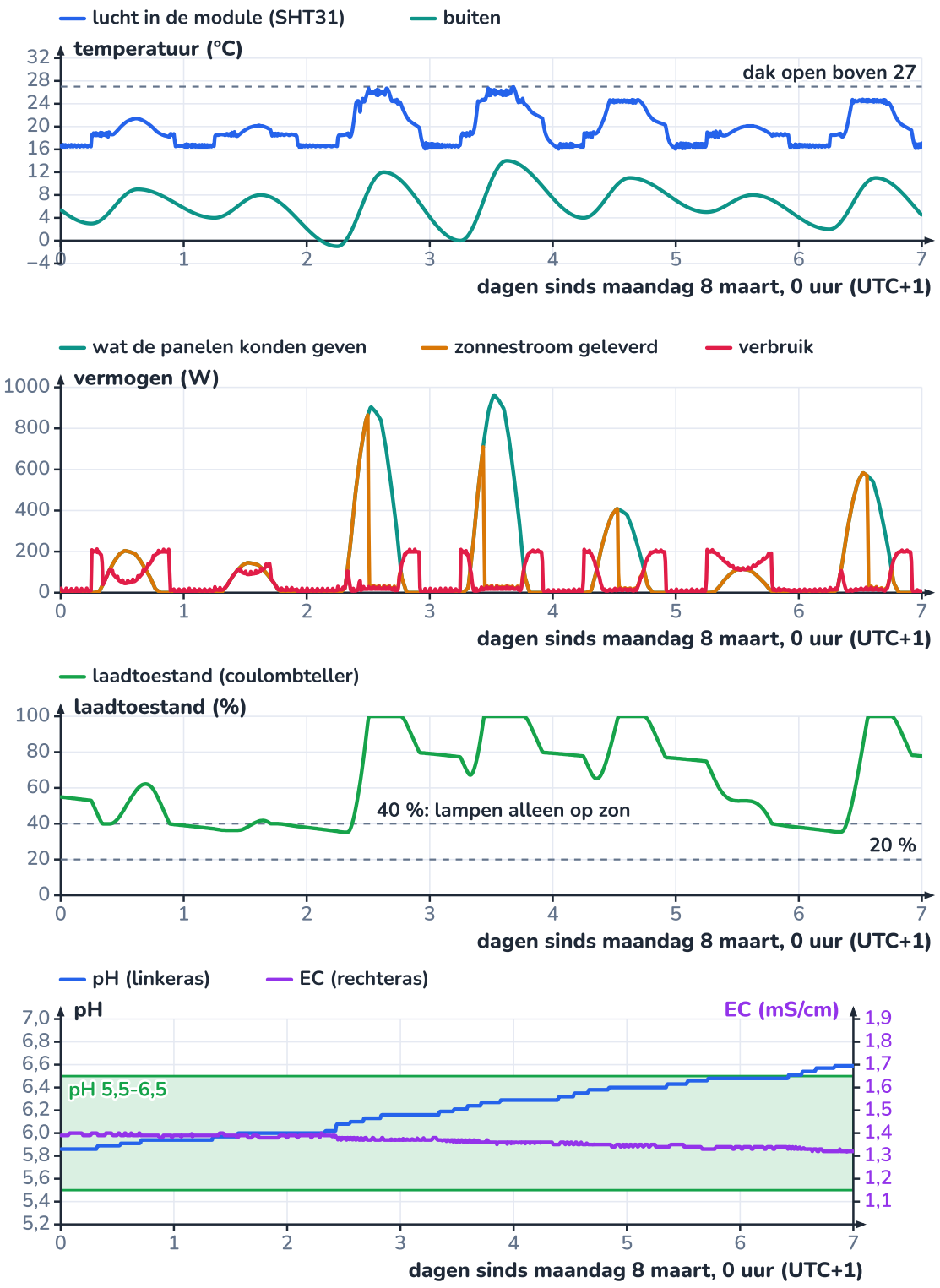
De testweek begon op maandag met een grijze start: laadtoestand 55 %, reservoir op 22 cm. Het logboek van het team:

wanneer	wat	gevolg in de logbestanden
ma 8 maart 0:00	start van de testweek; laadtoestand 55 %, reservoir 22 cm, EC 1,40, pH 5,86	—

wanneer	wat	gevolg in de logbestanden
ma en di	grijs (zonfactor 0,35 en 0,25), 3 tot 9 °C	laadtoestand onder 40 %: de lampen krijgen alleen zonne-overschot
wo 10:12-10:29	onderhoud: 13 L regenwater en 130 mL A + 130 mL B bijgevuld	modus 4; niveau van 18,9 naar 24,3 cm
wo en do	zonnig, tot 14 °C buiten	dak 10 en 14 keer open; batterijen vol, zonnestroom afgeknepen
do 14:00-16:00	storingstests: sensorkast, vlotter, stroom	alarmen, lege cellen, 3 ontbrekende regels
za	heel grijs (zonfactor 0,20)	laadtoestand van 77 naar 38 %; lichtsom 9,2 mol
zo 10:12	pH boven 6,5	waarschuwing in de meldingen; niemand in de module
ma 15 maart	pH bijgestuurd met pH-min, reservoir bijgevuld	(na de testweek)

Wat de vier grafieken tonen:

- **Temperatuur.** 's Nachts de zaagtand van de verwarming rond 16-17 °C, overdag 18-19 °C op grijze dagen. Woensdag en donderdag stijgt de lucht tot 27,1 °C: dan gaat het dak 10 en 14 keer open. De verwarming vroeg 14,0 kWh warmte van het station, van 0,8 tot 4,3 uur per dag.
- **Energie.** De panelen hadden 23,9 kWh kunnen leveren, maar de regelaar nam er maar 11,4 af: op de zonnige dagen waren de batterijen voor de middag vol, en 12,4 kWh bleef ongebruikt. Het verbruik was 10,2 kWh, waarvan 8,2 kWh voor de lampen.
- **Laadtoestand.** Tussen 35 en 100 %. Onder 40 % van maandagavond tot dinsdagmiddag, van dinsdagavond tot woensdagochtend en van zaterdagavond tot zondagochtend: dan kregen de lampen alleen zonne-overschot. Nooit onder 20 %.
- **Voedingsoplossing.** De pH stijgt gestaag, van 5,86 naar 6,59; zondag om 10:12 passeert ze 6,5. De EC zakt langzaam, van 1,40 naar 1,31: de sla neemt verhoudingsgewijs meer zouten op dan water. Het niveau zakt van 21,9 naar 14,6 cm, met het bijvullen van woensdag ertussen.



Gemiddelden per 10 minuten uit de logbestanden (simulatie). De grijze stippellijn in de tweede grafiek is wat de panelen hadden kunnen geven.

CODE · 40.20

De testweek analyseren in Python

20/28

Zeven dagbestanden van 1440 regels lees je niet met het oog. Het script leest ze met het csv-moduul (hoofdstuk 28 en 30), maakt per dag een rapport en toetst een deel van de eisen. Twee keuzes: een lege cel wordt `None` en telt niet mee in een minimum; en 'overdag' begint voor K1 pas om 7 uur, omdat de module om 6 uur van de nacht- naar de daginstelling springt en even moet opwarmen (dat staat zo in het programma van eisen, niet achteraf bedacht).

Python **testweek.py** 64 regels

```python
# testweek.py - de logbestanden van de klimaatmodule toetsen aan het programma van eisen
import csv

DAGEN = ["20210308", "20210309", "20210310", "20210311", "20210312", "20210313", "20210314"]
NAAM = ["ma", "di", "wo", "do", "vr", "za", "zo"]


def getal(tekst):
    """Tekst naar kommagetal; een lege cel (geen meting) wordt None."""
    return float(tekst) if tekst != "" else None


def lees(dag):
    with open("LOG/" + dag + ".CSV", newline="") as f:
        return [{k: (v if k == "tijdstip" else getal(v)) for k, v in r.items()} for r in csv.DictReader(f)]


def uur(r):
    return int(r["tijdstip"][11:13])


def dagrapport(rijen):
    dag = [r for r in rijen if 7 <= uur(r) < 22 and r["temperatuur_C"] is not None]
    nacht = [r for r in rijen if not 6 <= uur(r) < 22 and r["temperatuur_C"] is not None]
    par = [r["licht_lx"] / 54 + 200 * r["lampen_pwm"] / 255 for r in rijen if 6 <= uur(r) < 22]
    paren = list(zip(rijen, rijen[1:]))                    # (deze minuut, volgende minuut)
    return {
        "regels": len(rijen),
        "T_dag": (min(r["temperatuur_C"] for r in dag), max(r["temperatuur_C"] for r in dag)),
        "T_nacht": min(r["temperatuur_C"] for r in nacht),
        "zon": sum(r["zon_Wh"] for r in rijen) / 1000,                    # kWh
        "verbruik": sum(r["verbruik_Wh"] for r in rijen) / 1000,
        "soc": min(r["soc_pct"] for r in rijen),
        "lichtsom": sum(p * 60 for p in par) / 1e6,                       # mol/(m² dag)
        "verwarming": sum(r["verw_pct"] for r in rijen) / 100 / 60,       # uur
        "starts": sum(1 for a, b in paren if a["verw_pct"] == 0 and b["verw_pct"] > 0),
        "dak": sum(1 for a, b in paren if a["dak"] == 0 and b["dak"] > 0),
        "pH": max(r["pH"] for r in rijen),
        "EC": min(r["EC_mScm"] for r in rijen),
        "niveau": min(r["niveau_cm"] for r in rijen),
    }


week = [dagrapport(lees(dag)) for dag in DAGEN]
print("dag regels  T overdag  nacht   zon verbr  SoC licht verw starts dak  pH   EC niveau")
for naam, d in zip(NAAM, week):
    print(f"{naam}  {d['regels']:5d}  {d['T_dag'][0]:4.1f}-{d['T_dag'][1]:4.1f}  {d['T_nacht']:4.1f}  "
          f"{d['zon']:4.2f}  {d['verbruik']:4.2f}  {d['soc']:3.0f}  {d['lichtsom']:4.1f}  {d['verwarming']:3.1f}"
          f"  {d['starts']:3d}  {d['dak']:3d} {d['pH']:4.2f} {d['EC']:4.2f}  {d['niveau']:4.1f}")


def toets(code, tekst, fout):
    print(f"{code}  {tekst:38s} {'gehaald' if not fout else 'NIET: ' + ', '.join(fout)}")


print()
toets("K1", "overdag (7-22 h) nooit onder 17,5 C", [n for n, d in zip(NAAM, week) if d["T_dag"][0] < 17.5])
toets("K2", "'s nachts nooit onder 15,5 C", [n for n, d in zip(NAAM, week) if d["T_nacht"] < 15.5])
toets("K3", "nooit boven 30 C", [n for n, d in zip(NAAM, week) if d["T_dag"][1] > 30])
toets("L1", "lichtsom minstens 11,0 mol per dag", [n for n, d in zip(NAAM, week) if d["lichtsom"] < 11.0])
toets("E3", "laadtoestand nooit onder 20 %", [n for n, d in zip(NAAM, week) if d["soc"] < 20])
toets("H1", "EC nooit onder 1,2 mS/cm", [n for n, d in zip(NAAM, week) if d["EC"] < 1.2])
toets("H2", "pH nooit boven 6,5", [n for n, d in zip(NAAM, week) if d["pH"] > 6.5])
toets("D1", "1440 regels per dag", [n for n, d in zip(NAAM, week) if d["regels"] != 1440])
```

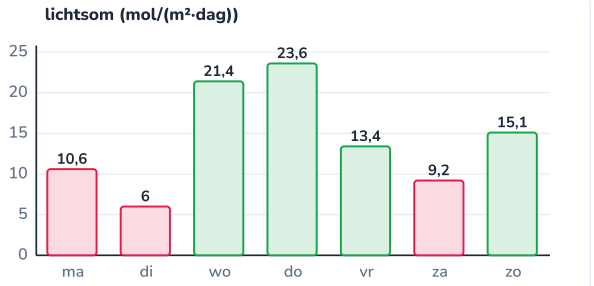
OOK TE DOWNLOADEN

De uitvoer:

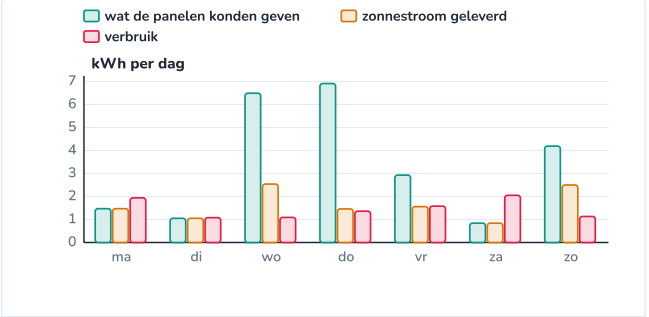
```
dag regels  T overdag  nacht   zon verbr  SoC licht verw starts dak  pH   EC niveau
ma   1440  18.0-21.4  16.0  1.47  1.94   39  10.6  2.8   48    0 5.94 1.38  20.0
di   1440  18.0-20.2  16.0  1.05  1.08   36   6.0  4.3   70    0 6.00 1.38  19.0
wo   1440  17.9-27.0  16.0  2.54  1.09   35  21.4  3.0   43   10 6.16 1.37  18.6
do   1437  18.0-27.1  16.0  1.46  1.36   67  23.6  1.7   30   14 6.29 1.35  20.1
vr   1440  18.0-25.0  16.0  1.56  1.58   65  13.4  0.8   16    0 6.40 1.34  18.0
za   1440  18.0-20.1  16.0  0.84  2.05   38   9.2  2.3   41    0 6.48 1.33  16.6
zo   1440  18.0-25.2  16.0  2.50  1.13   35  15.1  2.6   40    0 6.59 1.31  14.6

K1  overdag (7-22 h) nooit onder 17,5 C    gehaald
K2  's nachts nooit onder 15,5 C           gehaald
K3  nooit boven 30 C                       gehaald
L1  lichtsom minstens 11,0 mol per dag     NIET: ma, di, za
E3  laadtoestand nooit onder 20 %          gehaald
H1  EC nooit onder 1,2 mS/cm               gehaald
H2  pH nooit boven 6,5                     NIET: zo
D1  1440 regels per dag                    NIET: do
```

Lees ze eerst per kolom. **regels**: donderdag ontbreken er drie (de stroomtest). **licht**: maandag, dinsdag en zaterdag onder 11,0 mol. **starts**: dinsdag 70 keer verwarmen op één dag, maar in één uur nooit meer dan 6 keer (eis K6, een aparte telling per uur). **dak**: alleen woensdag en donderdag. **pH**: zondag boven 6,5.

Twee eisen zijn niet gehaald (L1 en H2), en D1 geeft 'NIET' voor donderdag: dat is geen fout van de module maar van de toets, want de stroomtest stond in het testplan. Het testrapport noteert daarom 'ja' met een verklaring. Een toets in code is streng en eerlijk; het oordeel blijft werk voor mensen.



Rood: onder 11,0 mol/(m²·dag). De zonnige dagen geven twee keer meer licht dan nodig.



Woensdag en donderdag konden de panelen 6,5 en 6,9 kWh geven; de module nam er 2,5 en 1,5 kWh van af.

NASLAG · 40.21

Het testrapport van het voorbeeldteam

21/28

Per eis de toets, het resultaat en het besluit. Van de 47 moet-eisen zijn er 43 gehaald, één voorlopig (K3: de zomer komt nog), één kan pas na vijf weken getoetst worden (W5) en twee zijn niet gehaald. Van de 7 wensen zijn er 5 gehaald, één niet (H4) en één pas na een jaar te toetsen (P2).

L1, de lichtsom. Op drie grijze dagen bleef de lichtsom onder 11,0 mol. De oorzaak is geen fout maar een botsing tussen twee eisen: E3 (nooit onder 20 %) en E4 (eerst de lampen afknijpen) winnen van L1. Opvallend: dezelfde week werd 12,4 kWh zonnestroom afgeknepen. Een grotere buffer of stroom van het station zou beide eisen laten samengaan (werkboek).

H2, de pH. De waarschuwing kwam op zondag om 10:12, maar niemand zag ze tot maandag. De module deed wat het ontwerp vroeg; het ontwerp vroeg te weinig. Twee oplossingen: vrijdag de pH preventief naar 5,8 brengen (bij +0,1 à 0,15 per dag blijft ze dan tot maandag onder 6,5), of de melding naar een telefoon sturen (E37).

code	eis	toets	resultaat (voorbeeldteam)	besluit
K1	overdag (7-22 h) minstens 17,5 °C bij de planten	logger, testweek	laagste 17,9 °C	ja
K2	's nachts (22-6 h) minstens 15,5 °C	logger, testweek	laagste 16,0 °C	ja
K3	hoogstens 30 °C	logger, zonnige dagen	hoogste 27,1 °C (do)	voorlopig
K4	ventilatoren vol boven 25 °C, dak open boven 27 °C; terug onder 24 en 25 °C	naspelen, logger	24 keer open (wo, do), drempels juist	ja
K5	het dak haalt zijn eindpunt binnen 12 s, anders alarm	deeltest: blokje in de rail	sluit in 6,8 s; met blokje alarm na 12 s	ja
K6	verwarming hoogstens 6 keer aan per uur	logger: starts per uur	hoogstens 6 (wo 5-6 h)	ja, net
K7	basisventilatie overdag minstens 11 m³/h	anemometer aan het rooster	11,9 m³/h	ja
K8	CO_2-klep alleen in DAG met het dak dicht	naspelen, code	in orde	ja
L1	lichtsom op het bed minstens 11,0 mol/(m²·dag)	logger: daglicht + lampen	ma 10,6, di 6,0, za 9,2	nee
L2	lampen vullen het daglicht aan tot 200 µmol/(m²·s), hoogstens 1 stap per 200 ms	logger, code	in orde	ja
L3	volle lampen: elk van 9 meetpunten ≥ 180 µmol/(m²·s)	PAR-meter, 's nachts	186 tot 214	ja
W1	1,2 L/min per goot (± 20 %)	maatbeker, 1 min per goot	1,08 tot 1,31; na bijstellen 1,19 tot 1,21	ja
W2	pomp in minuut 0-9 van elk uur, extra beurt als slappe sla verwacht wordt	logger, meldingen	7 extra beurten	ja
W3	de pomp draait nooit droog	vlotter met de hand omhoog	alarm na 8 s, pomp uit	ja
W4	minstens 5 dagen zonder bijvullen bij 6 L per dag	berekening, logger	36 L / 6 L = 6,0 dagen	ja
W5	reservoir lichtdicht: algen onder $5 \cdot 10^5$ per mL	telling na 5 weken	pas na 5 weken	later
H1	EC tussen 1,2 en 1,8 mS/cm (bij 25 °C)	logger	1,31 tot 1,40	ja
H2	pH tussen 5,5 en 6,5	logger	5,86 tot 6,59; zo 10:12 boven 6,5	nee
H3	voedingsoplossing nooit boven 24 °C	logger	hoogste 23,0 °C	ja
H4	voedingsoplossing tussen 18 en 22 °C	logger	17,1 tot 23,0 °C	nee
H5	pH-sensor om de twee weken in twee punten geijkt, EC met ijkvloeistof	ijkverslag	−0,177 V/pH; 1,42 in 1,413	ja
E1	van maart tot oktober genoeg zonnestroom: saldo per maand ≥ 0	berekening PVGIS, testweek	maart +2,6, oktober +1,8 kWh/dag	ja
E2	de batterijen overbruggen de nacht in de slechtste maand	berekening	1,84 kWh (december)	ja
E3	laadtoestand nooit onder 20 %	logger	laagste 35 %	ja
E4	energiebeheer: onder 40 % lampen op zonnestroom, onder 20 % uit, onder 10 % alarm	logger (dinsdag)	dinsdag: lampen begrensd	ja
E5	laadtoestand elk uur in het EEPROM, na een stroomuitval hersteld	stekker van de regelkast uit	100,0 → 100,0 %	ja
E6	5 V-rail aan de sensorkast minstens 4,75 V	meten: pomp aan, SD schrijft	4,91 V	ja

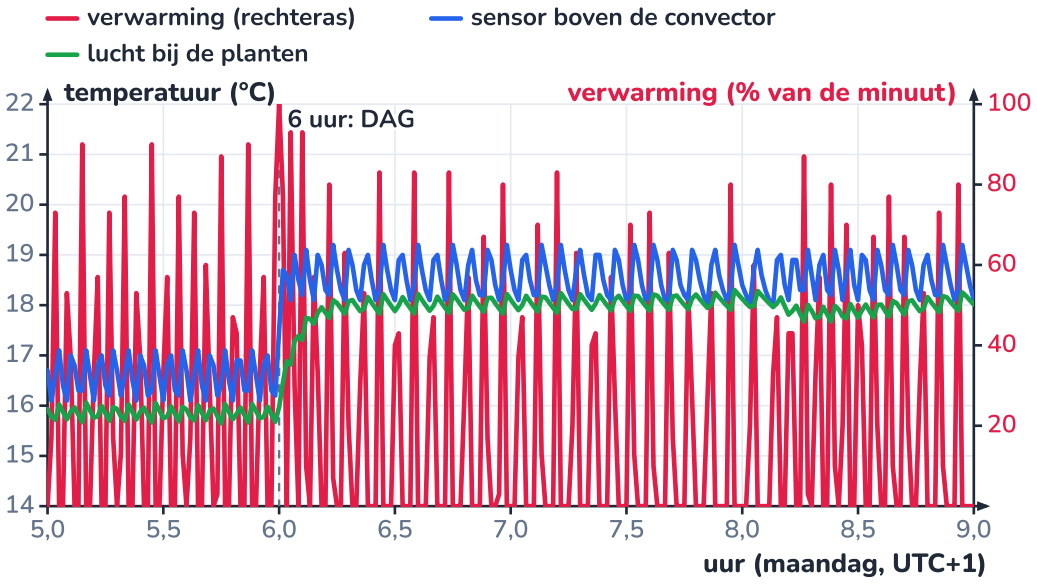
code	eis	toets	resultaat (voorbeeldteam)	besluit
E7	batterijen niet laden onder 0 °C	BMS-instelling, logger	laagste 16,0 °C	ja
E8	jaarverbruik hoogstens 700 kWh	berekening	±540 kWh	ja
B1	na een stroomuitval: alles uit, OPSTART 5 s, dan de gewone modus	stekker van de regelkast uit	15:58:00 OPSTART, 15:58:06 DAG	ja
B2	alarm bij: geen temperatuur 30 s, vlotter droog 10 s, reservoir onder 10 cm, dak-time-out, batterij onder 10 %	storingstests	sensor na 28 s, vlotter na 8 s	ja
B3	zonder temperatuur verwarmt de module met de gemiddelde inschakelduur	storingstest sensor	in orde	ja
B4	onderhoud: lange druk van 3 s; pomp uit, dak geblokkeerd, werklicht; na 60 min terug	woensdag	wo 10:12 tot 10:29	ja
B5	waakhond: een hangende taak herstart de Mega binnen 2 s	deeltest (hoofdstuk 32)	in orde	ja
B6	langste ronde van loop() hoogstens 100 ms	simulatie, micros()	48,8 ms	ja
B7	één taak per uitgang	code nalezen	in orde (pomp samengevoegd)	ja
D1	elke minuut een CSV-regel, één bestand per dag	logbestanden	1440; do 1437 (stroomtest)	ja
D2	een ontbrekende meting geeft een lege cel	storingstest sensor	5 regels	ja
D3	minstens een jaar op de kaart	berekening	±121 kB per dag: 1 GB ≈ 23 jaar	ja
D4	klok op UTC+1, hoogstens 1 min afwijking per maand	referentieklok	±5 s per maand (±2 ppm)	ja
D5	dag- en weekrapport in Python, gekoppeld aan de eisen	testweek.py	in orde	ja
V1	in de kweekruimte alleen 12,8 V en 5 V	checklist	in orde	ja
V2	zekering per tak en bij elke batterijpool	schema, keuring	in orde	ja
V3	niemand onder een bewegend dak	test in onderhoud	in orde	ja
V4	pH-min (salpeterzuur 0,50 mol/L) gelabeld, met bril en handschoenen	checklist	in orde	ja
C1	reservoir (65 kg vol) met zijn midden tussen 0,65 en 1,31 m van ligger A	rolmeter	0,98 m	ja
C2	dakmotor trekt 120 N zonder te blokkeren	trekproef met veerunster	120 N bij 3,4 A	ja
G1	statusled: rustig normaal, snel bij alarm, vast bij onderhoud	test	in orde	ja
G2	kasdak met de hand: korte druk op S1 naast de deur	test	in orde	ja
G3	wekelijks onderhoud (bijvullen, EC en pH) hoogstens 20 min	tijd opnemen	17 min	ja
G4	hydroponie hoogstens € 800	kostenraming	€ 788	ja
P1	48 plantplaatsen, om de 8,75 dagen een goot oogsten	teeltschema	in orde	ja
P2	minstens 100 kg sla per jaar	wegen, een jaar lang	model: 100 kg	later
O1	hoogstens 2 kg CO_2-eq per kg sla	levenscyclusanalyse	1,64	ja

Foutzoeken met de logger: de klep klikt voortdurend

Een ander team hing de sensorkast SK-1 op 40 cm boven de convector: daar kwam de kabel van 3 m net toe. De eerste ochtend hoort een bewoner de klep van de convector aan één stuk klikken. De sla staat er goed bij, de statusled knippert rustig. Het logbestand van maandag, 7:00 tot 7:08:

```
tijdstip,temperatuur_C,vocht_pct,verbruik_Wh,zon_Wh,licht_lx,batterij_V,soc_pct,lampe
n_pwm,verw_pct,dak,water_C,pH,EC_mScm,niveau_cm,modus,alarm
2021-03-08 07:00,18.7,83.5,3.51,0.04,107,12.95,46.5,252,0,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:01,18.3,83.7,3.47,0.04,110,12.95,46.4,252,0,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:02,18.1,83.9,3.53,0.04,114,12.94,46.3,252,37,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:03,18.9,83.2,3.58,0.05,117,12.94,46.2,252,47,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:04,19.0,83.3,3.47,0.05,121,12.95,46.0,252,0,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:05,18.5,83.5,3.47,0.05,124,12.95,45.9,252,0,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:06,18.2,83.8,3.47,0.05,128,12.95,45.8,252,0,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:07,18.2,83.7,3.60,0.05,131,12.94,45.7,252,70,0,16.9,5.86,1.40,21.8,1,0
2021-03-08 07:08,19.2,83.1,3.51,0.05,135,12.94,45.6,252,13,0,16.9,5.86,1.40,21.8,1,0
```

Waar zit de fout, hoe bevestig je ze, en hoe los je ze op?



De sensor (blauw) volgt de pluim van de convector, de lucht bij de planten (groen) blijft rond 18 °C. De verwarming (rood) klapt aan en uit.

GEGEVEN

- het logbestand hierboven (kolommen: blad 'Het logformaat van versie 5')
- verwarming aan onder 18 °C, uit boven 19 °C (hoofdstuk 31); eis K6: hoogstens 6 keer aan per uur
- een referentiethermometer bij de planten, op 1,2 m hoogte

GEVRAAGD

- wat het logbestand zegt
- een hypothese en een test die één ding verandert
- de oplossing en de hertest

OPLOSSING

1. **Lees wat de module zelf vertelt.** Geen alarm (kolom alarm 0, modus 1): de module denkt dat alles in orde is. Maar verw_pct springt van minuut tot minuut: 0, 37, 47, 0, 70, 13 %. De verwarming gaat binnen één minuut aan en weer uit. Geteld over een uur: 14 keer aan tussen 7 en 8 uur, tegenover ±4 op dezelfde ochtend in de testweek. Eis K6 is ruim overschreden.

2 **Vergelijk met een referentie.** Tussen 7 en 9 uur meet de SHT31 gemiddeld 18,6 °C, de referentie bij de planten 18,0 °C. De sensor meet warmer dan de lucht waarin de planten staan, en de temperatuur in het logbestand springt tot 1 °C per minuut: veel sneller dan een ruimte met 80 kJ/K kan opwarmen (hoofdstuk 31).

3 **Hypothese.** De sensor zit in de warme luchtstroom van de convector. Zodra die blaast, meet hij binnen een minuut 19 °C en schakelt de verwarming uit; de luchtstroom stopt, de sensor koelt af tot onder 18 °C, en de klep gaat weer open. De regeling regelt de lucht rond de sensor, niet de module.

4 **Test die één ding verandert.** Hang de referentiethermometer naast de SHT31, boven de convector. Zijn ze het daar eens, dan meet de sensor juist en is de *plaats* fout. Hang daarna alleen de sensorkast om, midden in de ruimte op planthoogte, en laat de rest zoals het was.

5 **Oplossen en hertesten.** Na het verplaatsen: een nacht loggen en tellen. In het model valt de verwarming terug naar ±4 starts per uur en verdwijnt het verschil met de referentie. Noteer de verplaatsing in de interfacetabel (I7: de plaats van de sensor hoort bij de afspraak).

ANTWOORD

De sensorkast hangt in de warme lucht van de convector: de verwarming schakelt 14 keer per uur en de sensor meet ±0,5 °C te warm. De sensorkast moet midden in de ruimte hangen, niet boven de warmtebron (hoofdstuk 31).

Klopt dit? Hetzelfde symptoom (veel korte beurten) kan ook een te smalle hysterese of een te grote verwarming zijn. Daarom verander je telkens maar één ding en kijk je opnieuw in het logbestand. Een minimum aantijd van 2 minuten in de code (werkboek) beschermt de klep ook tegen zo'n fout, maar lost de oorzaak niet op: de sensor zou nog altijd de verkeerde lucht meten. Het model rekent met een pluim van +3 °C boven de convector (rekenwaarde); de grafiek is dus een simulatie, zoals de testweek.

NASLAG · 40.23

De foutzoektabel van de module

23/28

Per symptoom de mogelijke oorzaken, de waarschijnlijkste eerst, met een test die telkens één ding controleert. Begin altijd bij wat de module zelf vertelt: de statusled, de meldingen en de logbestanden. Voeg de fouten toe die jullie zelf vinden: zo groeit de tabel mee met de module.

symptoom	mogelijke oorzaak (waarschijnlijkste eerst)	test (telkens één ding)	oplossing
alles donker, statusled uit	zekering van de omvormer (1 A); BMS schakelde af; stekker van de regelkast	12,8 V op de rail? 5 V na de omvormer?	oorzaak van de zekering zoeken, dan vervangen; batterijen laden
statusled knippert snel	een alarm: geen temperatuur, vlotter, niveau, dak of batterij	de meldingen (#) en de kolom alarm lezen	de oorzaak oplossen, dan kort drukken op S1
lampen zwak op een grijze dag	laadtoestand onder 40 %: het energiebeheer; LDR ziet een lamp; lamp of MOSFET defect	logger: soc_pct, lampen_pwm, licht_lx naast elkaar	bij een lage laadtoestand is het een ontwerpkeuze (eis L1, werkboek)
pomp slaat om de paar seconden aan en uit	twee taken schrijven naar pin 9	elke plaats in de code die de pomp schakelt opzoeken	één eigenaar (versie 5)
een goot krijgt weinig water	kraantje dicht of verstopt; wortels in de verzamelbuis; pomp zuigt lucht	maatbeker aan het einde van elke goot, 1 min	kraantje bijstellen, goot reinigen bij het oogsten
elk uur een niveau-alarm in minuut 1 tot 9	niveau gemeten terwijl er water in de goten staat	niveau_cm per minuut rond minuut 0	niveau pas 2 min na de pomp (versie 5); reservoir bijvullen
pH springt als de pomp aanslaat	gedeelde massadraad (hoofdstuk 21)	pH-kolom rond minuut 0 en 10	eigen retourdraad per tak naar het sterpunt

symptoom	mogelijke oorzaak (waarschijnlijkste eerst)	test (telkens één ding)	oplossing
pH loopt op tot boven 6,5	normale stijging (0,1 à 0,15 per dag); niemand stuurde bij	pH per dag in de logger	pH-min na een proeftitratie; vrijdag preventief
EC zakt na het bijvullen	alleen regenwater bijgevuld	EC voor en na het bijvullen	10 mL A en 10 mL B per liter bijgevuld water
temperatuur leeg in de logger	SHT31 los, kabel van 3 m, pull-ups	I²C-scanner: staat 0x44 erbij?	stekker, kabel, pull-ups nakijken (hoofdstuk 30)
klep van de convector klikt tot 14 keer per uur	sensorkast in de warme lucht van de convector; hysterese te smal	starts per uur; referentiethermometer bij de planten	sensorkast midden in de ruimte (hoofdstuk 31); eventueel een minimum aantijd
dak blijft half open, alarm	blokje of ijs in de rail; eindeloop defect; zekering 5 A	melding 'dak haalt zijn eindpunt niet'; eindeloop met de hand	vrijmaken, dan kort drukken: het dak sluit opnieuw
laadtoestand 100 % terwijl de spanning zakt	de teller is lang niet opnieuw geijkt	rustspanning 's morgens tegenover soc_pct	de batterijen één keer laten vollopen tot 14,2 V
logbestand ontbreekt of is leeg	kaart vol of weg; naam niet 8.3; stroom viel weg voor close()	melding 'SD'; kaart in de laptop	kaart vervangen; bestandsnaam nakijken (hoofdstuk 30)
module te warm met het dak open	zon op een zomerdag groter dan dak en ventilatoren aankunnen	temperatuur tegenover zonnestroom	schaduwdoek of nachtscherm als zonnescherm (versie 2)

UITLEG · 40.24

Fase 5 · het technisch rapport

24/28

Een technisch rapport is geschreven voor iemand die het project niet volgde, maar er verder mee moet: een volgend team, een ingenieur van het station, de jury. Die lezer leest eerst de samenvatting, dan de figuren, en pas daarna de tekst. De opbouw volgt de fasen:

onderdeel	wat erin staat	bij het voorbeeldteam
titelblad	titel als boodschap, team, datum, versie	Eigen stroom, eigen sla: de klimaatmodule van Habitat in een testweek (versie 1.0)
samenvatting	een halve bladzijde: doel, aanpak, belangrijkste getallen, besluit	zie hieronder
inleiding	probleem, doel, afbakening, spoor	alleen de kweekruimte, niet de woonmodule; spoor B
programma van eisen	de eisen per groep, met herkomst; de volledige tabel in een bijlage	54 eisen, 47 moet
ontwerp per deelsysteem	berekening, keuze en schema, per eigenaar	energiebudget, waterkringloop, warmtemodel, licht, voeding
integratie	N²-diagram, interfacetabel, de problemen die je vond	18 interfaces, 6 problemen
besturing	versies, blokschema, toestandsdiagram; de code in een bijlage	klimaat.ino versie 5
testen	testplan, storingstests, testweek, testrapport	43 van 47 moet-eisen gehaald

onderdeel	wat erin staat	bij het voorbeeldteam
besluit en aanbevelingen	antwoord op het doel; wat beter moet, gekoppeld aan eisen	L1 en H2: twee voorstellen
bronnen	alles wat je gebruikte, zo dat een ander het terugvindt	PVGIS, datasheets, hoofdstukken
bijlagen	code, logbestanden, tekeningen, ijkverslagen	klimaat.ino, zeven logbestanden

De samenvatting van het voorbeeldteam (één alinea, alle getallen erin):

> De klimaat- en energiemodule van Habitat moet 48 kroppen sla telen op eigen zonnestroom, met een geregeld klimaat en een logger. We stelden 54 eisen op, waarvan 47 moet-eisen, rekenden de energie-, water- en warmtebalans na, voegden vijf versies van de besturing samen tot klimaat.ino versie 5 en lieten de module een week alleen draaien in een simulatie met hetzelfde programma (8-14 maart 2021). 43 moet-eisen zijn gehaald, één voorlopig (de zomer komt nog) en één kan pas na vijf weken getoetst worden. Twee zijn niet gehaald: op drie grijze dagen bleef de lichtsom onder 11,0 mol/(m²·dag), omdat het energiebeheer de lampen afknijpt onder 40 % laadtoestand, terwijl op de zonnige dagen 12,4 kWh zonnestroom ongebruikt bleef; en zondag steeg de pH boven 6,5 zonder dat iemand de waarschuwing zag. We stellen een grotere buffer of stroom van het station voor, en meldingen die een telefoon bereiken.

Zinnen die iets bewijzen. Een rapport overtuigt met getallen, eenheden en criteria:

zwak	sterk	waarom
De batterij is groot genoeg.	In december leveren de batterijen 's nachts 1,84 kWh van de 2,46 kWh die bruikbaar is (75 %).	getal, eenheid en de grens ernaast
De test was geslaagd.	Met de vlotter omhoog volgde na 8 s een alarm en stopte de pomp (eis W3: binnen 10 s).	wat, wanneer en tegen welk criterium
We denken dat de sensor slecht hing.	Boven de convector las de SHT31 0,5 °C hoger dan de referentie bij de planten; op de nieuwe plaats verdween het verschil.	een meting in plaats van een mening
Zie grafiek.	Figuur 7 toont dat de laadtoestand dinsdag onder 40 % bleef: de lampen kregen alleen zonne-overschot.	de tekst zegt wat de figuur bewijst
Alles werkte goed.	43 van de 47 moet-eisen zijn gehaald; L1 en H2 niet (zie 6.3).	eerlijk, ook over wat niet lukte

Bronnen. Vermeld alles wat je gebruikte, zo dat een ander het terugvindt: wie, wat, welke versie, wanneer geraadpleegd (hoofdstuk 2). Bijvoorbeeld: *Europese Commissie, Joint Research Centre. PVGIS, versie 5.3. Brussel, 60° zuid, 1,40 kWp. Documentatie gecontroleerd op 30 september 2026; de getallen zijn rekenwaarden uit het cursusmodel.* Voor een datasheet: de fabrikant, het onderdeel en de revisie. Voor dit boek: *StemExpert Intermediate, hoofdstuk 20: Zonne-energie dimensioneren.*

Een rapport vertelt hoe hard we gewerkt hebben

Een rapport vertelt wat er is, waarom, en hoe goed het werkt. Het logboek bewaart het verhaal van het werk; het rapport bewaart het resultaat. Wat niet lukte, hoort er wel in: met de oorzaak en een voorstel.

UITLEG · 40.25

Fase 5 · de verdediging voor een jury

De verdediging duurt twintig minuten: tien minuten presenteren, tien minuten vragen. De jury (de leraar, een collega, iemand van buiten de school) kent het programma van eisen en heeft het rapport gelezen. Ze wil drie dingen weten: begrijpen jullie je eigen ontwerp, zijn jullie keuzes onderbouwd, en zijn jullie eerlijk over de grenzen?

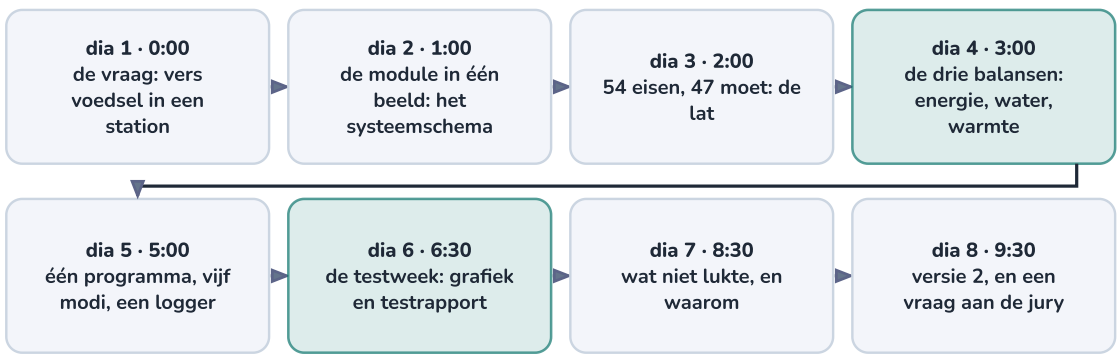
- **Het storyboard** legt de dia's vast voor je ze maakt: één boodschap per dia, als titel geschreven ('De batterijen dekken de nacht, niet de grijze week'), met één figuur die ze bewijst.
- **Iedereen spreekt**, elke eigenaar over zijn deelsysteem. De integrator opent en sluit.
- **De demo** toont de module aan het werk: een hand op de SHT31 en de ventilatoren slaan aan, de vlotter omhoog en de statusled knippert snel, de logger die live een regel schrijft. Houd een plan B klaar (een filmpje, een logbestand).

Vragen die een jury stelt, en wat een sterk antwoord bevat:

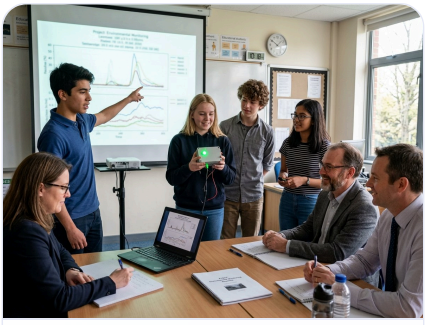
vraag van de jury	waar het echt over gaat	wat een sterk antwoord bevat
Waarom twee batterijen en geen drie?	dimensioneren en afwegen	de nacht in december (1,84 kWh van 2,46); een derde zou de afgeknepen zonnestroom van een zonnige dag bewaren voor een grijze, maar kost geld en CO_2 (hoofdstuk 38)
Hoe weet je dat je simulatie klopt?	model tegenover werkelijkheid	de waarden komen uit hoofdstuk 19, 20 en 31; C++ en Python geven dezelfde week; valideren doe je met echte logbestanden (E36)
Wat gebeurt er als de SHT31 uitvalt?	fail-safe ontwerp	alarm na 30 s, noodprogramma met de gemiddelde inschakelduur (eis B2, B3), getest op donderdag
Waarom aan/uit en geen PID?	regeltechniek	de convector kan alleen aan of uit; met 1 °C hysterese hoogstens 6 starts per uur; PID past bij een traploze verwarming (E39)
Kan de module zonder het station?	systeemgrens	energie: van februari tot november wel, in december en januari niet; warmte: nee; water: het kasdak vangt 4,0 m³ tegenover 2,2 m³ nodig
Wat hebben jullie niet getest?	eerlijkheid	de zomer (K3), de algen (W5), de opbrengst van een jaar (P2); en de testweek was een simulatie
Wie had zondag de pH moeten zien?	eigenaarschap	de eigenaar van water en voeding; het ontwerp stuurde de melding naar niemand: voorstel in de aanbevelingen

Een vraag die je niet kan beantwoorden

Zeg wat je wel weet, wat je niet weet, en hoe je het zou uitzoeken. 'Dat hebben we niet getest; de logger zou het tonen als we de sensorkast een week in de zon hangen' is een sterk antwoord. Raden is dat niet.



Acht dia's, tien minuten: de balansen en de testweek krijgen de meeste tijd.



De verdediging: één figuur per boodschap, en een jury die vragen stelt over de getallen.

NASLAG · 40.26

Beoordeling: eindrubric, peer- en zelfevaluatie

26/28

De jury en de leraar beoordelen het project met de eindrubric; het team gebruikt dezelfde rubric voor de zelfevaluatie (werkboek). Per criterium kies je het niveau dat het best past, met bewijs: een berekening, een logbestand, een regel in het logboek. De eindscore is de som van niveau × gewicht, gedeeld door 4. Het voorbeeldteam haalde niveau 3 op alles en niveau 4 voor testen en data-analyse: 80 op 100.

Daarnaast beoordeelt elk teamlid zichzelf en de anderen op vijf punten (peer-evaluatie), van 1 (zelden) tot 4 (altijd). Grote verschillen tussen zelf- en peer-evaluatie bespreek je in een kort teamgesprek.

criterium	gewicht	1 · begint	2 · op weg	3 · goed	4 · uitstekend
programma van eisen en interfaces	10 %	eisen vaag, zonder toets	meetbaar, maar herkomst of criterium ontbreekt vaak	alle eisen meetbaar, met criterium, toets en hoofdstuk; interfacetabel volledig	ook: botsende eisen opgemerkt en beslist
dimensionering	15 %	berekeningen ontbreken	berekeningen met fouten in eenheden of aannames	energie, water, warmte, licht en voeding juist nagerekend, met herkomst	ook: gevoeligheid getoond (wat als het grijs is, wat als het vriest)
bouw en veiligheid	10 %	niet volgens het schema	werkt, maar pas na ernstige herstellingen gekeurd	volgens het schema, zekering per tak, sterpunt, in één keer gekeurd	ook: nette bekabeling, alles gemeten en genoteerd
besturing en code	15 %	versie 5 draait niet	draait, maar niet alle situaties getest	draait, alle naspeeltests en storingstests in orde	ook: een eigen uitbreiding, getest en uitgelegd
testen en data-analyse	20 %	geen testplan	testplan, maar criteria achteraf of analyse met de hand	testplan vooraf, storingstests, testweek met een script getoetst	ook: niet-gehaalde eisen tot op de oorzaak ontleed

criterium	gewicht	1 · begint	2 · op weg	3 · goed	4 · uitstekend
technisch rapport	15 %	onvolledig	alle delen, maar zonder getallen of bronnen	volledig, met getallen, figuren die in de tekst uitgelegd worden, bronnen	ook: kort, helder, klaar voor een volgend team
verdediging	10 %	geen presentatie of demo	presentatie, maar vragen ontweken	storyboard gevolgd, iedereen spreekt, vragen met getallen beantwoord	ook: eerlijk over grenzen, sterke demo met plan B
samenwerking en logboek	5 %	logboek leeg	logboek onvolledig	logboek elke les, eigenaars kennen elkaars interfaces	ook: conflicten opgelost en genoteerd

Peer- en zelfevaluatie

criterium	1 · zelden	4 · altijd
komt afspraken en deadlines na	levert te laat of niet	op tijd, en meldt vooraf als het niet lukt
kwaliteit in het eigen deelsysteem	berekeningen of bouw met fouten	juist, gemeten, gedocumenteerd
bewaakt de eigen interfaces	verandert een pin of waarde zonder te melden	overlegt met de andere eigenaar, past de tabel aan
helpt anderen	werkt alleen	legt uit, test mee, deelt wat hij kan
houdt het logboek mee bij	nooit	elke les: wie, wat, probleem, volgende stap

UITLEG · 40.27

Wat de module nog niet kan, en wat Expert toevoegt

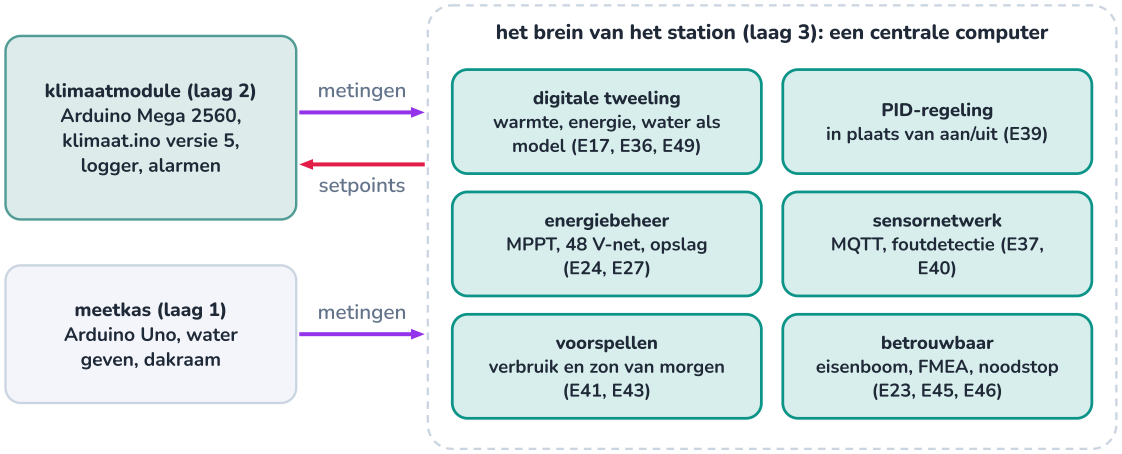
27/28

De module werkt: ze teelt, regelt, beheert haar energie en houdt alles bij. Maar de testweek toonde ook haar grenzen. Ze reageert op wat er nu is, ze voorspelt niet; ze meldt, maar alleen aan wie in de module staat; ze vertrouwt op één sensor per grootheid; en haar gedrag is getest in een model dat nog niet op echte metingen geijkt is.

wat de module nog niet kan	gevolg (testweek of berekening)	wat Expert toevoegt
een melding bereikt niemand buiten de module	de pH-waarschuwing van zondag werd pas maandag gezien	sensornetwerk en MQTT naar een centrale computer en een telefoon (E37)
aan/uit-regeling	zaagtand van 1 °C, tot 6 starts per uur	PID-regeling met een traploze verwarming (E39)
het energiebeheer reageert, maar voorspelt niet	dinsdag lampen afgeknepen, woensdag ±4 kWh zonnestroom ongebruikt	verbruik en zon voorspellen (E41, E43), opslag en het 48 V-net van het station (E19-E24, E27)
één temperatuursensor	sensor los: blind verwarmen	sensorfusie en foutdetectie (E40), redundantie en FMEA (E46)
het model is niet geijkt op metingen	de testweek is een simulatie	parameters schatten uit logbestanden (E17, E36), de digitale tweeling (E49)
geen noodstop in hardware	hangt de Mega, dan beslist alleen de waakhond	noodstoplogica met poorten (E23)
CO_2 niet gemeten	de klep gaat open op tijd, niet op concentratie	een CO_2-sensor en minuutgemiddelden (E6, E14)

wat de module nog niet kan	gevolg (testweek of berekening)	wat Expert toevoegt
de MPPT is een zwarte doos	je weet niet waarom hij afknijpt	buck- en boostomzetters en MPPT zelf (E24)

In Expert bouw je het **brein van het station**: een centrale computer die de meetkas (laag 1) en de klimaatmodule (laag 2) verbindt met de woonmodule, metingen verzamelt, een **digitale tweeling** meedraait (een simulatie zoals de testweek van dit hoofdstuk, maar geijkt op de echte module) en setpoints terugstuurt. De Mega blijft de module regelen, met versie 5 als basis; het brein zegt haar wat het beste setpoint is. Het ontwerpproces wordt het V-model in het groot, met een eisenboom en een FMEA (E45, E46). Het echte station EDEN ISS in Antarctica werkte zo: een serre die zichzelf regelde, bewaakt door een controlecentrum op afstand (hoofdstuk 36).



De Mega's blijven regelen; het brein kijkt mee, rekent vooruit en stuurt setpoints terug.

HET GEHEEL · 40.28

De tweede laag is af

28/28

De klimaat- en energiemodule is de tweede laag van Habitat. Elk deel van Intermediate leverde er iets aan, en in dit hoofdstuk kwam alles samen: meten met een onzekerheid (I1), modellen met hun geldigheidsgebied (I2), krachten op dak en vloer (I3), de chemie van de voedingsoplossing en de batterij (I4), de drie balansen (I5), de elektronica van sensor tot MOSFET (I6), Python en C++ (I7), regelen en toestanden (I8), de plant zelf (I9), en ontwerpen met data (I10).

Waar het vandaan komt. De meetkas van Fundamental (F33) was hetzelfde project in het klein: 36 eisen, één programma met tellers, een testweek met een laptop. Nu: 54 eisen, vijf versies samengevoegd, een logger en een analyse in Python, en een verdediging voor een jury.

Waar het naartoe gaat. Expert neemt de module over als bouwsteen van het brein van het station. Het warmtemodel met twee knopen wordt de kern van de digitale tweeling (E17), de aan/uit-regeling een PID-regeling (E39), het energiebeheer een voorspellend beheer met het 48 V-net van het station (E24, E41), de meldingen een sensornetwerk (E37), en het programma van eisen een eisenboom in het V-model (E45). Het integratieproject van Expert (E49) volgt dezelfde fasen, voor het hele station.

Tien delen, negenendertig hoofdstukken, één module.

F33 Integratieproject: de meetkas
Het eerste integratieproject: dezelfde fasen, in het klein.

I2 Experimenten ontwerpen en rapporteren
Het verslag en de bronvermelding van het technisch rapport.

I11 Druk en hydraulica
Het werkpunt van de pomp en de druksensor in het reservoir.

I13 Chemische bindingen en stofeigenschappen
De EC van de voedingsoplossing en de temperatuurcompensatie.

I15 Zuren, basen en pH
De pH, de tweepuntsijking en het bijsturen met pH-min.

I16 Redox en batterijen
De LiFePO4-batterijen en de coulombteller.

I19 Warmtetransport en isolatie
Het warmteverlies van 20,2 W/K.

I20 Zonne-energie dimensioneren
Panelen, batterijen en energiebeheer: het budget dat hier herrekend is.

I25 Van analoog naar digitaal
Dimmen met PWM en het daglicht in µmol.

I29 Microcontrollers programmeren in C++
Versie 1, pinnen.h en taken met millis().

I30 Datalogging en communicatie
De logger, I²C en de klok van versie 2.

I31 Regelsystemen
Hysterese, het warmtemodel met twee knopen en het noodprogramma.

I32 Toestandsmachines
De modi en het dak als toestandsmachines: het kader van versie 5.

I36 Hydroponie en voedselproductie
Het NFT-rek, de lichtsom en de opbrengst.

I38 Duurzaam ontwerpen
De voetafdruk van de module en de keuze van de batterijen.

I39 Data visualiseren en eerste stappen in AI
De extra pompbeurt uit de beslisboom, en grafieken die niet misleiden.

E17 Warmteoverdracht kwantitatief
Het warmtemodel van het station: de kern van de digitale tweeling.

E24 Vermogenselektronica en elektrische machines
De MPPT-regelaar van binnen.

E37 Embedded systemen en IoT

Meldingen die iemand bereiken: een sensornetwerk met MQTT.

E39 PID-regeling

Van aan/uit naar PID.

E41 Van data naar model: regressie

Het verbruik en de zon van morgen voorspellen.

E45 Systems engineering en het V-model

Het V-model en de eisenboom voor het hele station.

E49 Integratieproject: het brein van het station

Het integratieproject van Expert: het brein van het station.

Habitat

De tweede laag van Habitat is af. Nieuw in het stationsdossier: het programma van eisen van de module (54 eisen, 47 moet), 18 interfaces, klimaat.ino versie 5 met een logformaat van 17 kolommen, het testrapport van een testweek, en de eindbalans: met dimmen ±540 kWh per jaar (tekort alleen in december en januari), ±2,2 m³ water per jaar (het kasdak vangt er 4,0), en warmte van het station (±2 kWh per maartdag).

SAMENVATTING

In het kort

- Integreren is nagaan of delen die elk werken, ook samen werken. De plaats waar ze elkaar raken, heet een interface: een waarde, een pin, een adres, een eenheid, een tijdzone.
- Een project met vier eigenaars en een integrator loopt in vijf fasen: eisen, dimensioneren, bouwen en programmeren, integreren en testen, rapporteren en verdedigen, met mijlpalen en een logboek.
- Het programma van eisen van de module telt 54 eisen (47 moet), elk meetbaar, met een vooraf vastgelegd criterium, een toets en het hoofdstuk waar de waarde vandaan komt.
- Het N²-diagram en de interfacetabel leggen 18 interfaces vast; door budget, schema en code naast elkaar te leggen vond het voorbeeldteam zes integratieproblemen vóór de testweek.
- Met dimmen daalt het verbruik van 3,27 naar ±1,5 kWh op een gemiddelde maartdag en naar ±540 kWh per jaar; alleen december en januari komen tekort, en de twee batterijen blijven nodig voor de nacht.
- Water: 4,8 L/min over vier goten, ±3,9 L onderweg als de pomp draait, 6 warme dagen zonder bijvullen, ±2,2 m³ per jaar. Warmte: 20,2 W/K, convector 800 W, buffer 1,09 MJ/K; op zonnige dagen dak en ventilatoren samen.
- Klimaat.ino versie 5 voegt vijf versies samen met vier regels: één klok, nergens delay(), één eigenaar per uitgang, de modi als kader; ze schrijft elke minuut 17 kolommen.
- Testen gaat van deeltest over integratie- en storingstest naar een testweek; de logbestanden worden in Python getoetst aan het programma van eisen.
- In de testweek werden 43 van de 47 moet-eisen gehaald; L1 (lichtsom op grijze dagen) en H2 (pH op zondag) niet: de oorzaken zijn een botsing tussen eisen en een melding die niemand bereikte.
- Het technisch rapport begint met een samenvatting vol getallen; de verdediging volgt een storyboard; de eindrubric en de peer-evaluatie beoordelen resultaat en samenwerking.

BEGRIPPEN

Wat je nu kent

integrator
Het teamlid dat bewaakt dat de deelsystemen op elkaar passen: programma van eisen, interfaces, planning en testrapport.

interfacetabel
Lijst van alle plaatsen waar twee deelsystemen elkaar raken, met per interface de afspraak (waarde, eenheid, pin, adres), de eigenaar en de test.

N²-diagram
Vierkante tabel met de deelsystemen op de diagonaal; het vakje in rij A en kolom B zegt wat A aan B levert.

deeltest
Test van één onderdeel of deelsysteem apart, zodra het klaar is (een sensor, een tak, de pomp).

integratietest
Test van de samenwerking tussen deelsystemen: werkt de interface zoals afgesproken?

storingstest
Test waarbij je bewust een fout veroorzaakt (sensor los, vlotter omhoog, stroom eraf) om na te gaan of het systeem zoals ontworpen reageert.

testweek
Systeemtest waarin het systeem een week alleen draait, terwijl de logger alles bijhoudt.

lichtsom
Het aantal mol lichtdeeltjes (PAR) dat op één dag op een vierkante meter valt, daglicht en lampen samen: DLI, in mol/(m²·dag).

afknijpen
Wat een laadregelaar doet als de batterij vol is: hij neemt minder zonnestroom af dan de panelen kunnen leveren.

één eigenaar per uitgang
Ontwerpregel: één enkele taak in het programma beslist over een pin; andere taken vragen, ze schakelen niet zelf.

technisch rapport
Verslag van een ontwerp: van eisen over berekeningen en keuzes tot tests en besluit, zo dat een ander het kan nagaan en verderzetten.

verdediging
Presentatie van een ontwerp voor een jury, gevolgd door vragen, waarin je je keuzes met getallen onderbouwt.