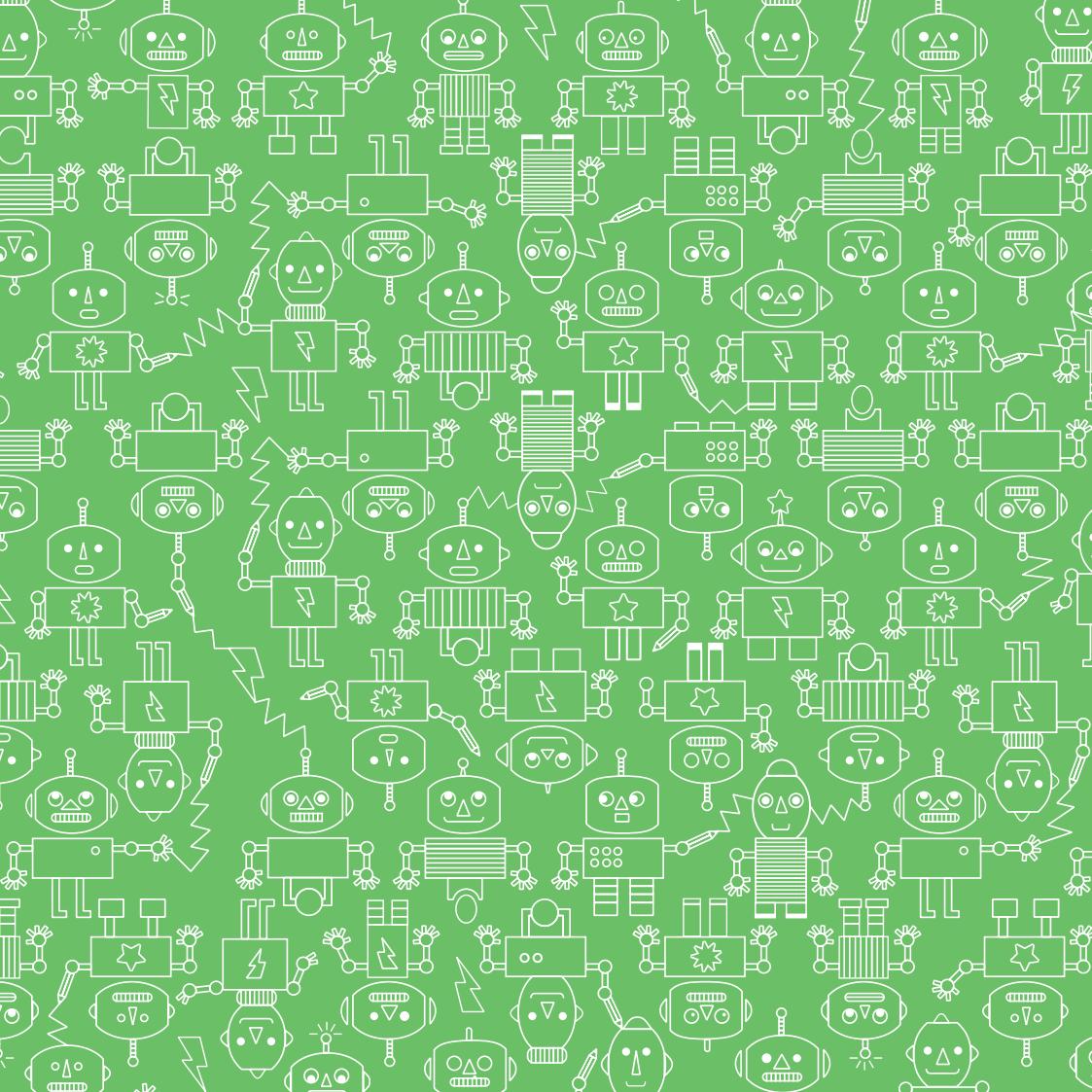


Bouw jouw eigen robot

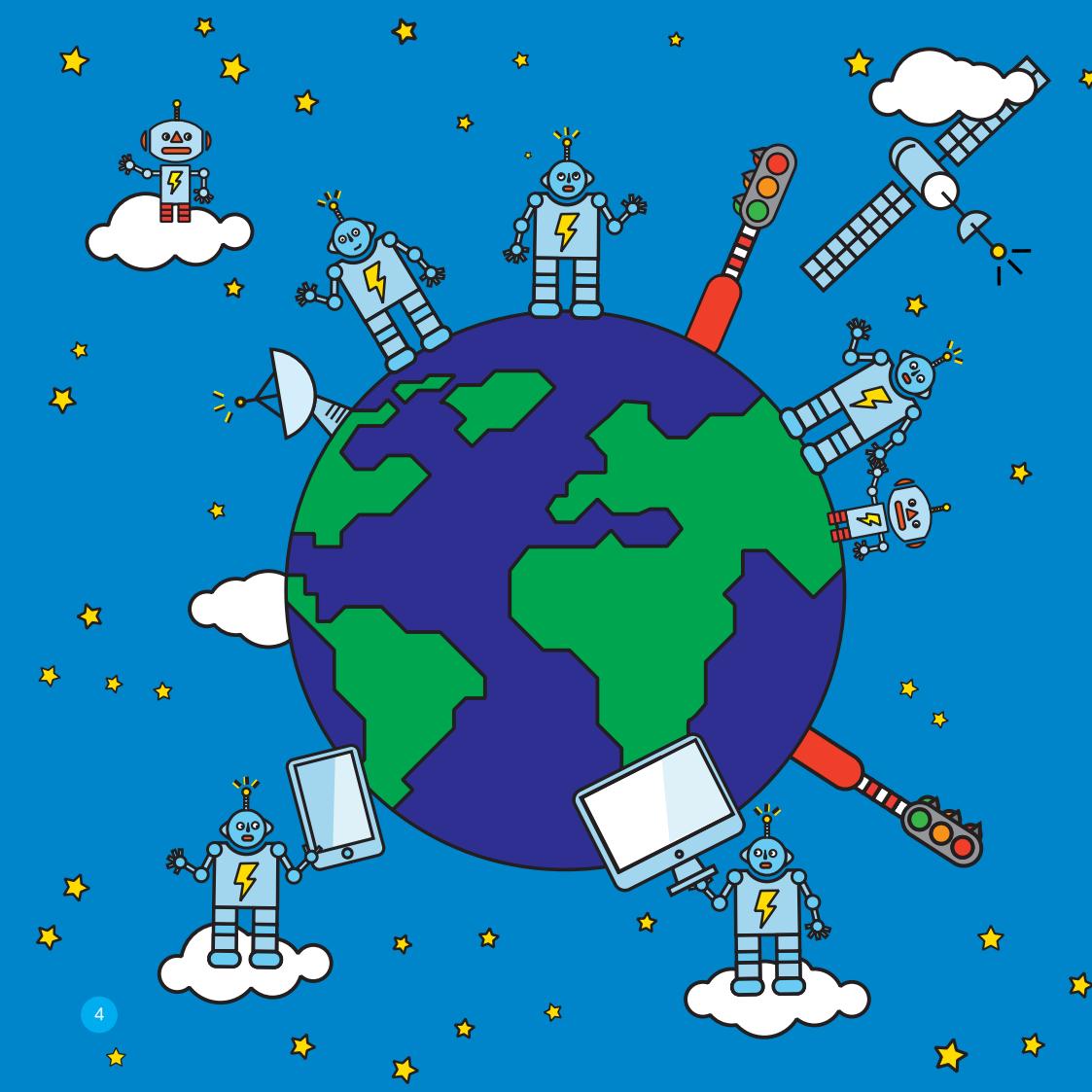
Francis wyffels, Natacha Gesquière, Tom Neutens,
Bjarne Van de Velde en Silke Van der Schueren

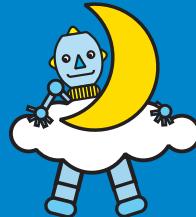
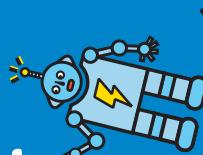


dwengo



1	Van da Vinci tot Sony: een stukje geschiedenis • 7	
2	Kunstmatige intelligentie • 8	
3	De robot • 10	
	De bouwblokken van een robot • 10	
	Een robot als (huis)dier? • 12	
	Overal microcontrollers! • 13	
4	De Dwenguino • 16	
	Veelzijdig microcontroller-platform • 16	
	De microcontroller ontleed • 18	
	Van algoritme tot programma • 20	
	De Dwenguino programmeren • 21	
5	Programmeren van jouw robot • 24	
	De robot zegt 'hallo' • 24	
	Spannend, zo'n robot bouwen • 26	
	Je robot op pad sturen • 34	
	Tijd voor robot-petanque • 36	
6	Sensoren • 38	
	Analoge en digitale sensoren • 38	
	Autonome robot met sonar • 40	
	De sonar-sensor aansluiten • 41	
	De meetwaarde van de sonar uitelezen en opslaan • 43	
	Autonome robot met grondsensor • 46	
	De grondsensor aansluiten • 47	
	De meetwaarde van de grondsensor uitelezen en opslaan • 49	
	Automaten: een manier om met sensoren om te gaan • 50	
7	Nodig eens een robot uit in de klas! • 54	
8	Spiekbriefje • 58	





Overal robots

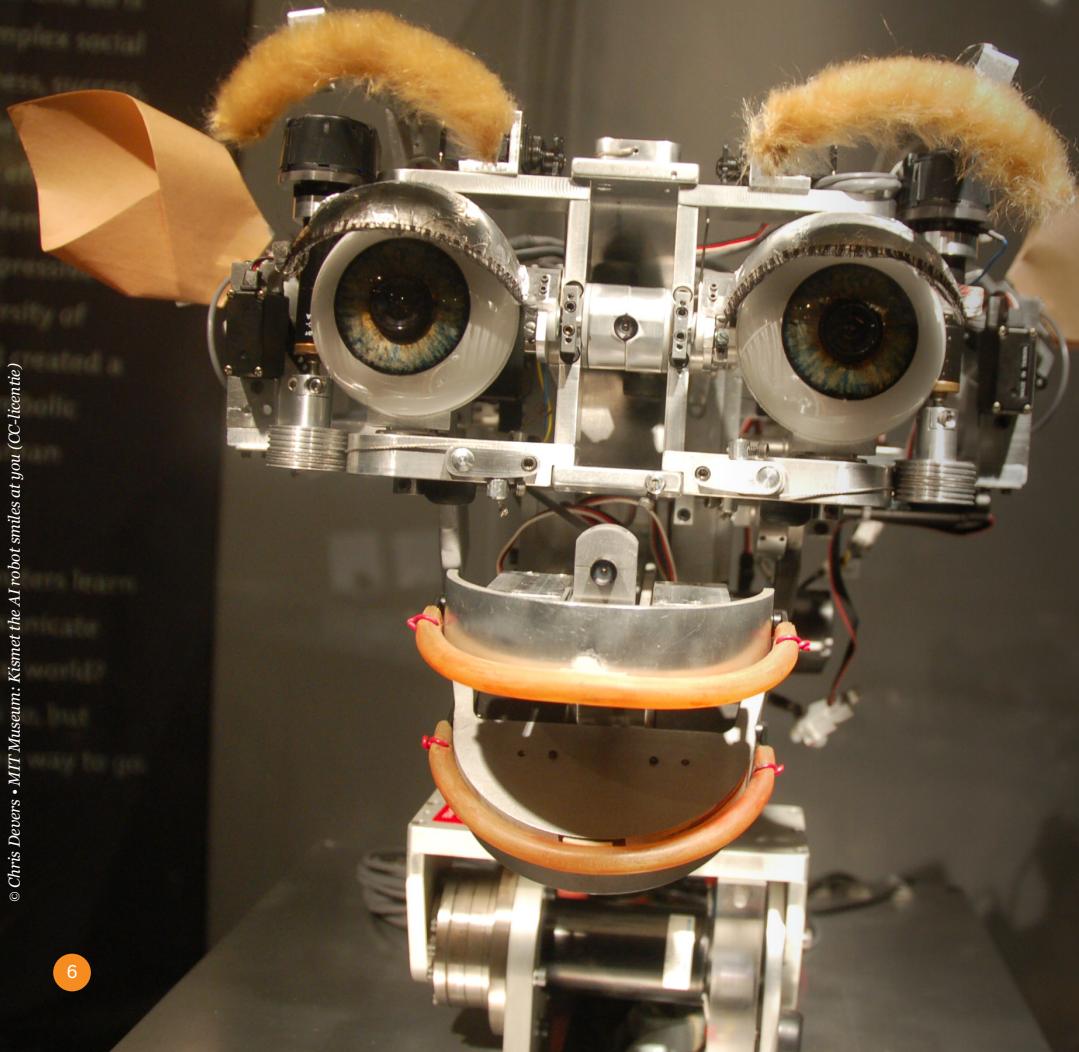
Technologie speelt een belangrijke rol in onze maatschappij.

Denk maar aan de smartphone die je dagelijks gebruikt of de webshops die je bezoekt. Maar technologie betekent ook: drones, zelfrijdende auto's, het internet der dingen (IoT, Internet of Things) en... robots, die steeds vaker ingezet worden in de industrie. Gelukkig maar! Ze nemen saaie of gevaarlijke taken van mensen over, en ze helpen een handje in onder meer de chirurgie en de ouderenzorg.

Hierdoor zullen **jobs sneuvelen en nieuwe, andere banen ontstaan**. Robots creëren met andere woorden **maatschappelijke uitdagingen**, maar ze dragen tegelijk **oplossingen** aan. In deze context worden creativiteit, kennis van technologie, en digitale en probleemoplossende vaardigheden almaar belangrijker. Daar komt dit boek bij van pas! **We helpen je een robot ontwerpen, bouwen en programmeren, en bereiden je zo voor op een boeiende toekomst.**

Door zelf een robot te bouwen en te programmeren begrijp je wat er aan de basis ligt van die alomtegenwoordige technologie: mechanica, sensoren, microcontrollers, programmeertaal, ... **De robotica doet je stilstaan bij tal van biologische processen, leert je probleemoplossend denken én je creativiteit inzetten.**

Bouw je met ons mee?



1

Van da Vinci tot Sony: een stukje geschiedenis

Robots bestaan veel langer dan je misschien zou denken. Zo ontwierp **Leonardo da Vinci al in 1495 een ‘mechanische mens’**. Het was echter pas in 1961 dat de industrie de eerste robot mocht verwelkomen: de **Unimate**. Deze robotarm kon je programmeren om taken zelfstandig uit te voeren. De Unimate wordt vandaag nog altijd gebruikt: geavanceerde varianten ervan doen bijvoorbeeld dienst **in de microchirurgie**.

Parallel met de ontwikkeling van zo’n industriële robots startten onderzoekers in 1950 met het systematisch bouwen van robots met poten. De meest bekende afstammelingen hiervan zijn ongetwijfeld **Honda’s ASIMO** (die een menselijke gedaante heeft) en **Sony’s Aibo** (een robothond).

Onderzoekers zijn tot slot niet alleen bezig met de mechanische bouw; ze focussen ook meer en meer op de sociale vaardigheden van robots. KISMET van MIT uit de jaren negentig (zie foto) lacht bijvoorbeeld naar je. Robots met gelaatsexpressie worden onder andere ingezet als baliemedewerkers in hotels of ziekenhuizen.

2

Kunstmatige intelligentie

Intelligente machines: mensen dromen er al eeuwenlang van. Zo bouwde de Hongaarse klokkenmaker von Kempelen in 1769 een 'schaakmachine'. Hoewel in 1820 aan het licht kwam dat een mens deze machine stiekem bediende en ze dus nep was, zette ze de wereld wel aan het denken over kunstmatige of artificiële intelligentie (AI).

Pas met de **opkomst van de digitale computer in 1950** konden onderzoekers machines écht programmeren om te schaken. **Sindsdien ging de kunstmatige intelligentie sterk vooruit**, zeker de voorbije tien jaar. Zo versloeg IBM's schaakcomputer Deep Blue in 1997 de toenmalige wereldkampioen Kasparov. In 2016 versloeg Google's programma Alpha Go dan weer de wereldkampioen Lee Sedol in het spel Go – een nieuwe mijlpaal, want hoewel dit spel maar twee (heel eenvoudige) regels heeft, is het veel complexer dan schaken.

Vandaag voeren slimme, lerende algoritmes heel wat taken uit. Ze herkennen spraak, vertalen teksten en analyseren medische gegevens. Er bestaan zelfs programma's die volledig zelfstandig muziek creëren, animaties maken of filmscripts schrijven.



Landschapsfoto na bewerking met het Deep Dream-algoritme. Dit algoritme is getraind op het herkennen van honden en gebouwen en transformeert op die manier gewone foto's. Francis Wyffels, 2017

3

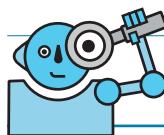
De robot

De bouwblokken van een robot

Het 'lichaam' van een robot bevat mechanische onderdelen die uit veel materialen kunnen bestaan: wielen, een arm, een hoofd, ... De robot heeft sensoren (ingangen) om te 'voelen' en actuatoren (uitgangen) om te 'handelen'.

Je kan je robot voorzien van afstandssensoren, grondsensoren, tastsensores, lichtsensoren of geluidssensoren. Voorbeelden van actuatoren zijn een lcd-scherm, een zoemer en een servomotor.

Cruciaal is de rekeneenheid waarmee je robot 'beslissingen' neemt. Hiervoor gebruik je bijvoorbeeld een **processor** (zoals in een computer) of een **microcontroller** (zoals op de Dwenguino). De bedrading zorgt ervoor dat de rekeneenheid, de sensoren en de actuatoren met elkaar in verbinding staan. Hoe de robot zijn actuatoren aanstuurt, hangt af van de informatie die hij verzamelt via zijn sensoren en van de manier waarop de rekeneenheid geprogrammeerd is. De batterij in de robot voorziet de motoren van de nodige energie.



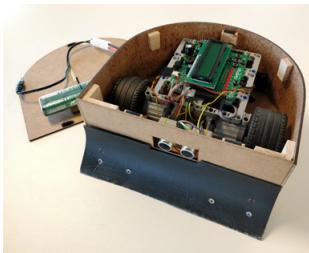
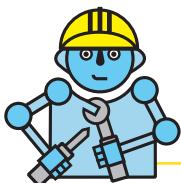
Jouw computer heeft ook **in- en uitgangen**. Het toetsenbord en de muis zijn voorbeelden van ingangen; het scherm is een uitgang.

Een robot kan alleen een bepaalde taak uitvoeren als zijn rekeneenheid daarvoor geprogrammeerd is. Die taak ligt niet per se vast: je kan de rekeneenheid telkens opnieuw zo programmeren dat de robot een andere taak vervult. Die taak kan bovendien afhangen van de informatie die de microcontroller via de ingangen krijgt, maar als het programma op de microcontroller er niets mee doet, heeft deze informatie geen effect.

Benieuwd wat je robot zoal kan? Je zou hem zo kunnen programmeren dat hij telkens opnieuw bekijkt of de ondergrond waarop hij rijdt donker- of lichtgekleurd is, wat zijn afstand tot de muur is, of waar een andere robot zich bevindt. Je kan hem lawaai laten maken, snel of traag laten rijden, of meetwaarden op het lcd-scherm laten verschijnen. Met een servomotor waaraan een potlood bevestigd is, kan je de robot zelfs laten tekenen.

DOE-HET-ZELF!

Silke bouwde op school een sumorobot met haar zus. Herken jij de verschillende onderdelen (wielen, motoren, bedrading, sensoren, batterij, rekeneenheid,...)? Wat beschouw je als een ingang? Wat als een uitgang?



3

De robot Een robot als (huis)dier?

Een robot wordt weleens met een dier vergeleken.

Die vergelijking leidt tot interessante vragen. Heeft een dier sensoren? Met welk deel van een dier kan je de motoren van een robot vergelijken? En heeft de bedrading een dierlijk equivalent?

Het is voer voor discussie, maar de volgende interpretatie houdt wel steek: de **rekeneenheid is het brein van de robot en komt overeen met de hersenen van een dier**. Een robot kan namelijk alleen een taak uitvoeren als zijn rekeneenheid daarvoor geprogrammeerd is. De motoren doen de robot bewegen, dus je zou ze kunnen vergelijken met de spieren van een dier.

Het **programma** vertelt de robot waar hij naartoe moet, hoe snel hij moet bewegen en wanneer hij via zijn sensoren informatie uit de omgeving moet ophalen. Een deel van de bedrading vormt het zenuwstelsel van de robot. Via deze draden worden signalen gestuurd: enerzijds van de rekeneenheid naar de motoren en de sensoren, en anderzijds van de sensoren naar de rekeneenheid. Een ander deel van de bedrading zorgt voor het transport van de nodige energie. Dieren halen energie uit hun voeding, bijvoorbeeld de energie die ze nodig hebben om hun spieren te bewegen. Robots krijgen hun energie van hun batterijen. Ze gebruiken de elektriciteit die de batterijen leveren om hun motoren te bewegen.

Waarin verschilt een dier dan van een robot? Het verschil zit hem vooral in de manier waarop een dier beslissingen neemt. Een dier houdt, al dan niet bewust, rekening met de omstandigheden. Een robot beweegt alleen als zijn rekeneenheid hem daartoe beveelt. Als er iets mis is met het programma, als de betreffende sensor stuk is of als de robot over te weinig energie beschikt, dan doet de robot niet wat hij zou moeten doen. Voor een dier geldt dat niet, aangezien het dierlijk brein in staat is om oplossingen te bedenken voor nieuwe problemen.

Overal microcontrollers!

De rekeneenheid, bijvoorbeeld een microcontroller, is het belangrijkste onderdeel van een robot.

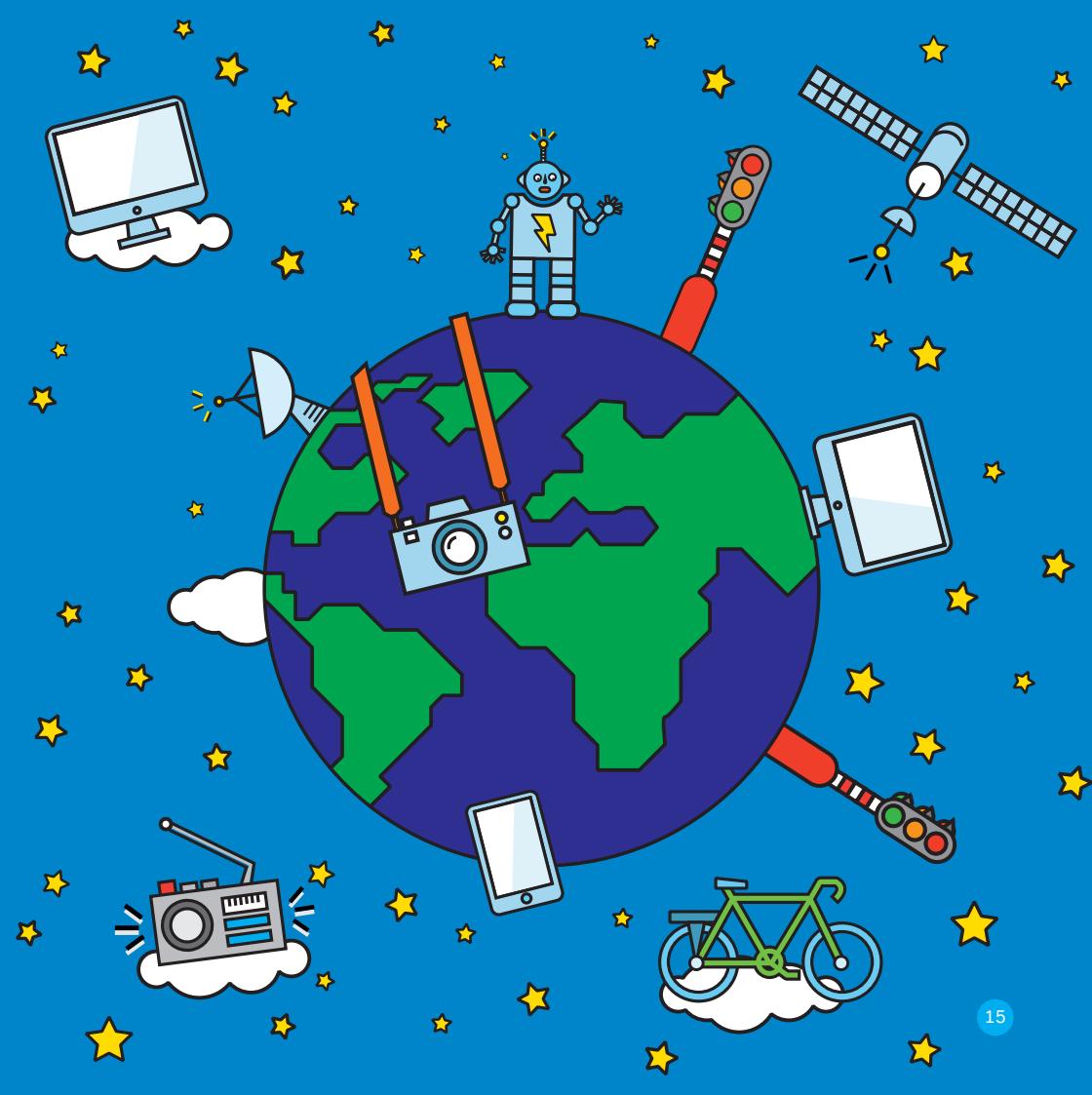
Microcontrollers **bevatten alles wat nodig is** om sensorgegevens te verwerken, berekeningen uit te voeren en de gepaste acties te ondernemen. Je bent er vast al mee in aanraking gekomen, want ze zitten **in tal van digitale apparaten**. En die apparaten zijn sinds een **paar decennia natuurlijk niet meer weg te denken uit het huis, de tuin of op de werkvloer**. Het begon met kleine toestellen, zoals wekkerradio's en elektrische tandenborstels, maar ondertussen is zo goed als elk apparaat intelligent. Misschien heb je thuis wel een stofzuigrobot? Ook zelfrijdende auto's zijn allang geen sciencefiction meer.

3

De robot

Die apparaten staan steeds vaker in verbinding met het internet en vormen zo samen het **Internet of Things**. Experts voorspellen dat tegen 2020 meer dan 50 miljard apparaten verbonden zullen zijn met het internet. Ze doen daarmee niet alleen op smartphones en tablets, maar bijvoorbeeld ook op slimme rookmelders, intelligente koelkasten en robots.

De **digitale rekensystemen** die instaan voor de noodzakelijke intelligentie, zoals dus de microcontroller, vormen de **basis voor al deze apparaten**.



4

De Dwenguino

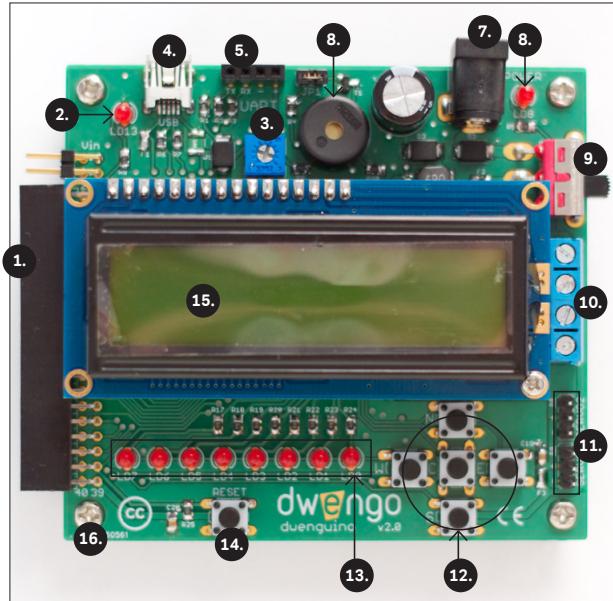
Een veelzijdig microcontroller-platform

Als microcontroller-platform kiezen we voor de Dwenguino, die heel wat extra voorzieningen heeft zodat je snel je eigen robot kan bouwen. De mogelijkheden van de Dwenguino zijn eindeloos: handige drukknoppen, analoge ingangen, een lcd-scherm, een zoemer, alles om motoren aan te sturen en nog veel meer!

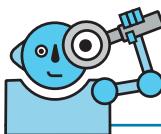
Met de uitbreidingsconnector kan je elke pin benaderen en apparaten zoals een led aansturen, en dankzij het handige uitbreidingsbord breid je de Dwenguino uit met een elektronische schakeling of een van de vele Arduino shields. De Dwenguino is namelijk compatibel met de Arduino en kan je dus programmeren in Arduino IDE, ArduBlock, DwenguinoBlockly en Scratch4Arduino.

Zo leer je microcontrollers programmeren zonder je zorgen te maken over het vinden van de juiste elektronica. **Kortom: de Dwenguino is de perfecte tool voor startende hobbyisten, leraren en scholen.**

1. uitbreidingsconnector
2. led op pin 13
3. contrast van led-scherm aanpassen
4. USB-connector
5. UART-connector
6. zoemer
7. batterij-connector
8. power led
9. aan- en uitschakelaar
10. motor-connector
11. servo-connector
12. vijf drukknoppen
13. acht leds
14. resetknop
15. led-scherm
16. bevestigingsgaten



De Dwengo is volledig open source: alle schema's, alle ontwerp-bestanden en de softwarebibliotheek staan online. Zo kan je de Dwengo-hardware tot in het kleinste detail bestuderen, jouw eigen uitbreidingsbord in elkaar steken of zelf een Dwengo-hardware variant maken.



4

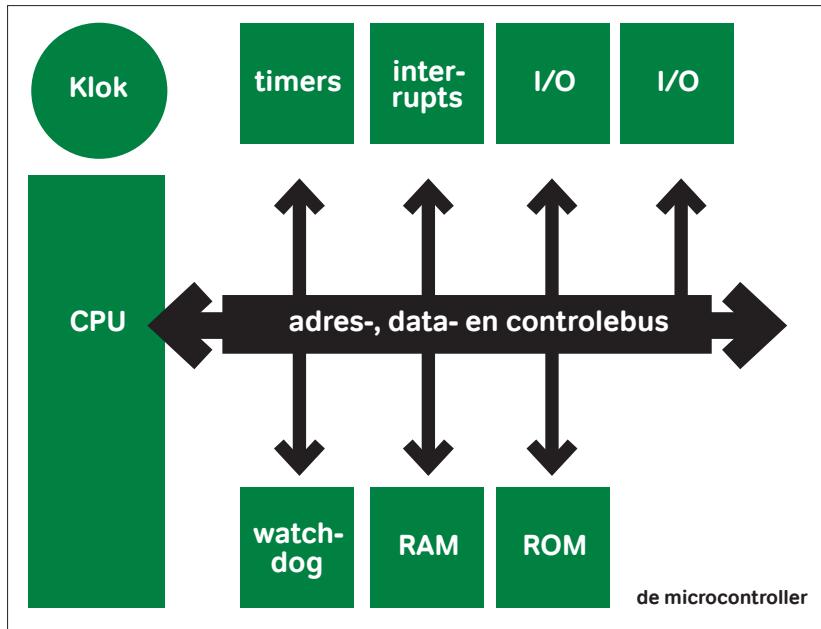
De Dwenguino

De microcontroller ontleed

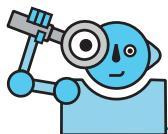
De microcontroller, het kloppend hart van elk digitaal apparaat, verschilt van een klassieke processor zoals die in je pc. Zo'n processor doet bijvoorbeeld een beroep op een extern geheugen (RAM en ROM) en externe ingangs- en uitgangspinnen (I/O) om te interageren met andere componenten.

De microcontroller daarentegen bundelt al wat hij nodig heeft om te functioneren op **één chip**. Het geheugen zit op de microcontroller zelf en de ingangs- en uitgangspinnen zijn rechtstreeks verbonden met de microcontroller. Terwijl de klassieke processor een gespecialiseerde component is met veel meer rekenkracht dan de microcontroller, is de microcontroller eerder een manusje-van-alles.

Om de microcontroller te laten functioneren, moet je hem **programmeren**. Jouw programma kan bijvoorbeeld vertellen dat de microcontroller twee motoren moet aansturen wanneer de knopjes ingedrukt zijn. Je kan hiervoor een grafische programmeertaal zoals **Google Blockly** of **Ardublock** gebruiken. Maar je kan evengoed tekstueel programmeren in C++.



Jouw programma wordt omgezet in een **opeenvolging van instructies** die de **microcontroller begrijpt**. Deze instructies worden opgeslagen in het zogenoemde **hex-bestand**: elke instructie wordt er voorgesteld door een **uniek getal**. Je programma komt dus eigenlijk neer op een opeenvolging van nullen en enen. Bij elke tik van zijn interne klok overloopt de microcontroller deze nullen en enen, zet hij ze om naar een gekende instructie en voert hij die uit.



4

De Dwenguino

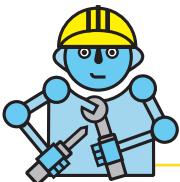
Van algoritme tot programma

Nadat je de software hebt geïnstalleerd die geschikt is voor de gekozen microcontroller, kan je een eigen programma beginnen te schrijven!

Hiervoor vertrek je vanuit een bepaald probleem dat je wilt oplossen. Vervolgens bedenk je **een algoritme voor dat probleem**: een reeks van stappen die in de juiste volgorde uitgevoerd moeten worden om vanuit de begintoestand tot het gewenste eindresultaat te komen.

Tijdens het programmeren zet je je algoritme om in (grafische of tekstuele) **code**: dit zijn opdrachten die eenduidig beschrijven wat je programma moet doen. Als je een grafische taal gebruikt en het programma uploadt naar de microcontroller worden de blokken van je programma omgezet naar C++. Je kan deze C++-code ook bekijken. Tijdens het compilatieproces wordt deze C++-code vertaald naar machinetaal: een **reeks instructies die verstaanbaar zijn voor de microcontroller**.

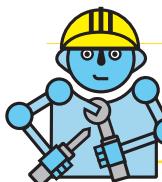
DOE-HET-ZELF!



Een goed algoritme bedenken is niet altijd even evident. Twijfel je hieraan? Kijk dan eens naar het beroemde filmpje 'How to program your teacher to make a jam sandwich'. Je vindt het door via Google te zoeken naar 'Sandwich Bot'.

De Dwenguino programmeren

Zoals we eerder al vertelden, is de Dwenguino compatibel met de Arduino. Hierdoor kan je de vele mogelijkheden van je microcontroller-platform programmeren met behulp van Arduino IDE en een zelfgekozen grafische of tekstuele programmeertaal. Wij gebruiken de grafische programmeertaal DwenguinoBlockly.



DOE-HET-ZELF!

Installeer de **plug-in DwenguinoBlockly** voor de Arduino IDE.
Instructies vind je op <http://www.dwengo.org/blockly>.

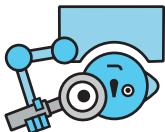
Nadat je Arduino IDE en DwenguinoBlockly hebt geïnstalleerd, kan je een eigen programma schrijven met DwenguinoBlockly. Je kan hiervoor heel wat instructies gebruiken die de Dwenguino al begrijpt. Er zijn bijvoorbeeld blokken voorzien die het gemakkelijk maken om de motoren of het lcd-scherm aan te sturen of om een led aan te zetten.

4

De Dwenguino

Verbind nu eerst de Dwenguino met je computer en start Arduino IDE en DwenguinoBlockly op.

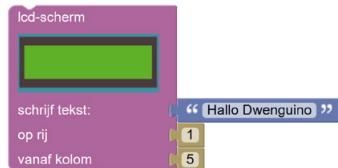
De basis van elk DwenguinoBlockly-programma is het ‘zet klaar / herhaal’-blok.



In het ‘zet klaar / herhaal’-blok plaats je al je code. Je kan zowel in het ‘zet klaar’- gedeelte als in het ‘herhaal’-gedeelte blokken toevoegen. Belangrijk om te weten: alles in het ‘zet klaar’-gedeelte wordt slechts één keer uitgevoerd, namelijk bij het aanzetten van de Dwenguino. De blokken in het ‘herhaal’-gedeelte worden steeds opnieuw uitgevoerd tot je zelf het programma opnieuw start (met de resetknop of door de stroom te onderbreken (de juiste kabel uittrekken en opnieuw insteken)).

Je kan de Dwenguino zo programmeren dat er een **boodschap** verschijnt op het lcd-scherm. Op het lcd-scherm van de Dwenguino passen maximaal 32 karakters, bijvoorbeeld letters of cijfers, verspreid over twee regels.

Voor het programma op de volgende pagina gebruik je twee blokken uit het Dwenguino-menu in de linkermenubalk: een eerste blok om de tekst te wissen die eventueel al op het lcd-scherm staat, en een tweede blok om de gewenste tekst op dat scherm te laten verschijnen. Dit tweede blok moet je twee keer gebruiken als je tekst uit twee regels bestaat. Bij 'op rij' geef je in op welke lijn de tekst moet komen (rij 0 of rij 1), 'vanaf kolom' laat je toe om de tekst naar rechts op te schuiven. Op elke lijn kan je 16 karakters zetten (van kolom 0 tot en met kolom 15).



5

Programmeren van jouw robot

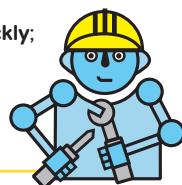
De robot zegt 'hallo'

Je programmeert de Dwenguino nu zo dat er 'Hallo mens' op de eerste regel van het lcd-scherm verschijnt, en 'Ik ben robot' op de tweede regel. Wil je iets anders op het scherm zetten? Dat kan perfect, zolang je maar niet meer dan 32 karakters gebruikt.

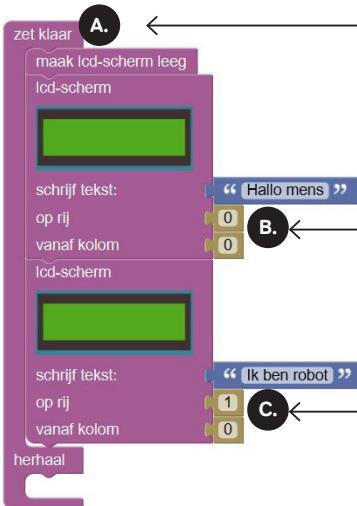
DOE-HET-ZELF!

Begin met het programma van hiernaast te **ontcijferen**. Probeer te voorspellen wat het programma zal doen, en test het:

1. Sluit de Dwenguino aan;
2. Start **Arduino** op;
3. Selecteer Dwenguino in het menu **Hulpmiddelen > Board**;
4. Selecteer de poort waarop de Dwenguino is aangesloten in het menu **Hulpmiddelen > Poort**;
5. Open via het menu Hulpmiddelen de plug-in **DwenguinoBlockly**;
6. Sleep de juiste blokken, zodat je hetzelfde programma krijgt als in het voorbeeld hiernaast;
7. Klik op het upload-icoontje ► en kijk wat er gebeurt!



Pas het programma aan. Probeer eens om de tekst op de eerste lijn op te schuiven naar rechts. Kan je op de tweede lijn van het lcd-scherm de tekst veranderen in 'Daar gaan we!?'?



A.

Het 'zet klaar'-gedeelte:
dit wordt slechts één keer uitgevoerd
bij het opstarten van de Dwenguino.

B.

De eerste plaats:
programmeurs beginnen steeds bij nul
te tellen. Daarom staat er hier bij rij en
kolom telkens een 0 .

C.

De tweede plaats:
rij 1 is de tweede rij op het lcd-scherm.

5

Programmeren van jouw robot

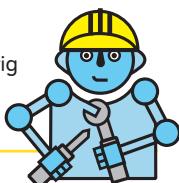
Spannend, zo'n robot bouwen

Als je een robot wilt bouwen, moet je eerst nadenken over **welk soort robot** dat precies gaat zijn. Sumorobots moeten bijvoorbeeld aan heel andere vereisten voldoen dan racerobots.

Een sumorobot heeft onder meer een sterker lichaam nodig, en een sterke 'kast' om de mechanische onderdelen te beschermen. Legoblokken zijn prima geschikt om het lichaam van de robot te bouwen (de Dwenguino kan je makkelijk bevestigen op een lichaam uit lego), maar je kan evengoed recyclagemateriaal gebruiken, zoals een lege ijsdoos. Wil je een persoonlijker ontwerp? Je kan robotonderdelen 3D-printen en de kast van een sumorobot lasercutten. Je zal moeten afwegen welke sensoren en motoren geschikt zijn. Op de foto zie je bijvoorbeeld een tekenende robot. Zijn armen werden gelasercut en bevestigd op gelijkstrommotoren van Lego.

DOE-HET-ZELF!

Wil je de kast van je robot lasercutten, maak dan eerst een nauwkeurig ontwerp in karton. Meet de afmetingen zorgvuldig op en geef ze in in een gratis softwareprogramma, zoals Inkscape.



In het project WeGoSTEM bouwden 5000 kinderen een tekenende robot. Foto door Francis wyffels, 2017.



5

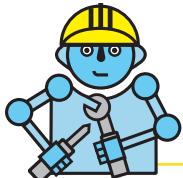
Programmeren van jouw robot

Zodra je het type robot hebt gekozen, moet je nog heel wat andere knopen doorhakken. Moet je robot bepaalde afmetingen hebben? Dan moet je daar rekening mee houden bij het plaatsen van de motoren en het gebruiken van het uitbreidingsbord. Wordt je robot niet te breed? Is hij stevig genoeg? Hangen de motoren niet door? Zullen de wielen er niet af vallen tijdens het rijden? Kunnen de wielen vrij draaien of worden ze belemmerd?

Denk verder goed na over de **manier waarop je de sensoren bevestigt**. Je kan ze bijvoorbeeld vastkleven of inbouwen in een geraamte. Als je grondsensoren gebruikt, speelt de afstand tot de grond een grote rol. Heb je plaats voorzien voor de batterij? Kan je makkelijk bij de aansluiting om de batterij op te laden of eventueel te vervangen? Is je Dwenguino bereikbaar en kan hij oververhit geraken? Kan je de aan-uitknop vlot bedienen? Kan je het lcd-scherm gebruiken, of hoeft dat niet?

Voor de motoren kan je legomotoren kiezen. Je sluit ze aan op de Dwenguino, zodat de microcontroller ze kan aansturen. Je begint met het stekkertje te verwijderen. Van de vier oorspronkelijke draden verwijder je de buitenste draden links en rechts. De middelste twee draden vertin je met behulp van een soldeerbout, daarna verbind je ze met de Dwenguino.

DOE-HET-ZELF!

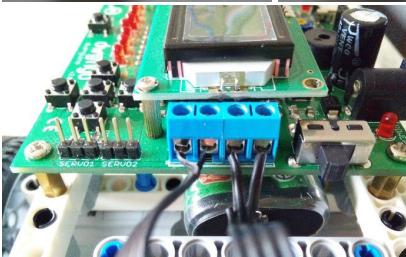


Ontwerp en bouw je eigen robot. Een rijdende robot is ideaal om mee te beginnen. Je vertrekt hiervoor van een paar legoblokken, twee legomotoren en een Dwenguino. Van de legomotoren gebruik je telkens de twee middelste draden, die je met de blauwe motor-connector bevestigt. Onderstaande figuren laten zien hoe dat moet.

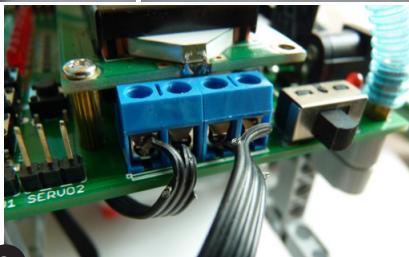
A. motor



B. motor gestript

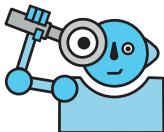


C. in de connector

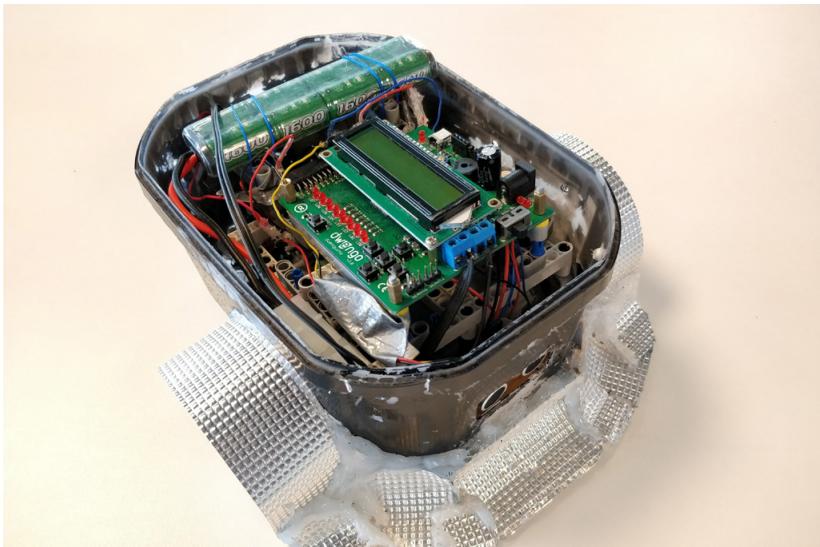




Ook over deze bal-robot is goed nagedacht. De robot kan rollen door het intrekken van zijn poten. De batterij, de microcontroller en de sensoren werden geïntegreerd in de binnenste (zwarte) kern. Natan Doms en Francis Wyffels, 2017, foto door Guilherme Gerais



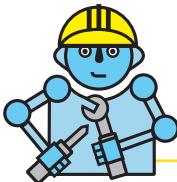
Deze sumorobot van Mathieu en zijn vrienden van het eerste jaar weegt net geen 1,5 kilogram en is kleiner dan 24 bij 24 centimeter. De robot voldoet aan de spelregels van de robotcompetitie van Dwengo en de UGent. (<http://ugent.be/robotcompetitie>)



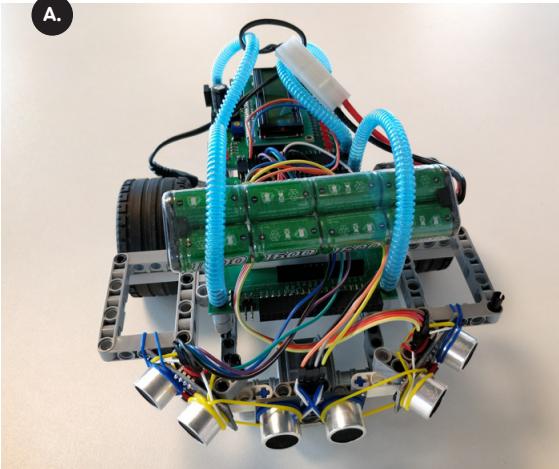
5

Programmeren van jouw robot

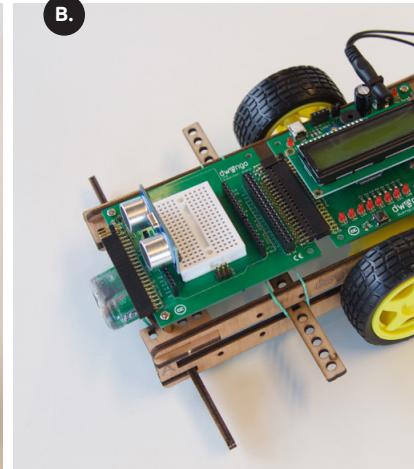
Met Lego bouw je heel wat interessante robotchassis. Op onze website <http://www.dwengo.org/nl/bouwjerobot> vind je instructies voor het bouwen van drie eenvoudige rijdende legorobots waarop je de Dwenguino kan monteren. Daarnaast bevat het Dwenguino-'Startpakket Plus' van Dwengo voorgemaakte onderdelen en motoren voor een eenvoudige rijdende robot.



A.



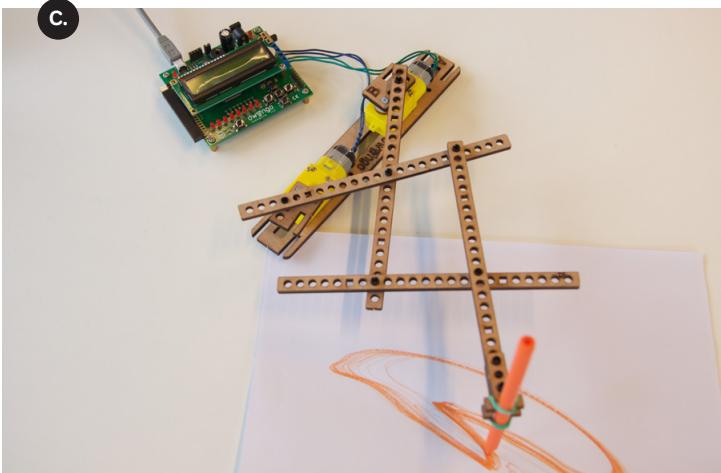
B.



A. Legorobot

B. Rijdende robot via onderdelen Startpakket Plus

C. Tekenende robot via onderdelen Startpakket Plus



5

Programmeren van jouw robot

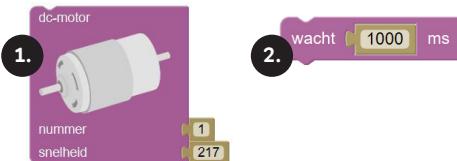
Je robot op pad sturen

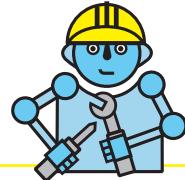
Het is de bedoeling je robot een traject te laten afleggen.

Daarvoor moet je de motoren leren **programmeren**.

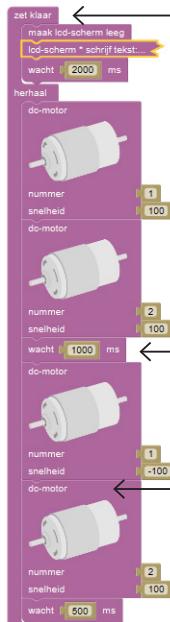
Probeer je robot eerst eens **in een vierkant te laten rijden** met behulp van deze twee nieuwe blokken:

1. **Met het motor-blok regel je de snelheid** van één van de motoren die aangesloten zijn op de motor-connector. De snelheid wordt bepaald door een getal tussen -255 en 255. Het teken bepaalt de draairichting van de motor.
2. **Het wacht-blok laat je toe om je programma te pauzeren** gedurende een bepaalde tijd in milliseconden. Hiermee controleer je met andere woorden het tijdsgedrag van je programma.





Combineer meerdere motor- en wacht-blokken in het 'herhaal'-gedeelte om je robot in een vierkant te laten rijden. Programmeer en test je robot. Inspiratie nodig? Bekijk dan aandachtig het programma op de bladzijde hieronder.



A.

Het 'herhaal'-gedeelte:

hiermee heb je de robot geprogrammeerd zodat er tekst op het scherm verschijnt (zie de vorige opdracht). Om het overzichtelijk te houden werd de code ingeklappt.

Het 'herhaal'-gedeelte:

dit gedeelte vormt een oneindige lus waarmee je de robot zo programmeert dat hij telkens opnieuw een stukje vooruit rijdt en 90° naar rechts draait. Het motorblok en de wachtfunctie spelen hierbij een belangrijke rol.

B.

Een wacht-blok:

met dit blok kan je jouw programma pauzeren. Geef hiervoor een tijd mee in milliseconden. De motoren blijven wel draaien zolang je ze niet stopt of opnieuw instelt. Met het wacht-blok bepaal je dus het tijds gedrag van je programma.

C.

Een motor-blok:

met dit blok stuurt je een motor aan. Je kan twee DC-motoren aansluiten op het Dwenguiño-bord: nummer 1 en nummer 2. Het getal dat de snelheid bepaalt, kan variëren tussen -255 en 255.

5

Programmeren van jouw robot

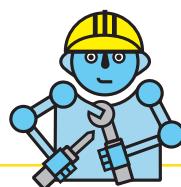
Tijd voor een rondje Robot-petanque

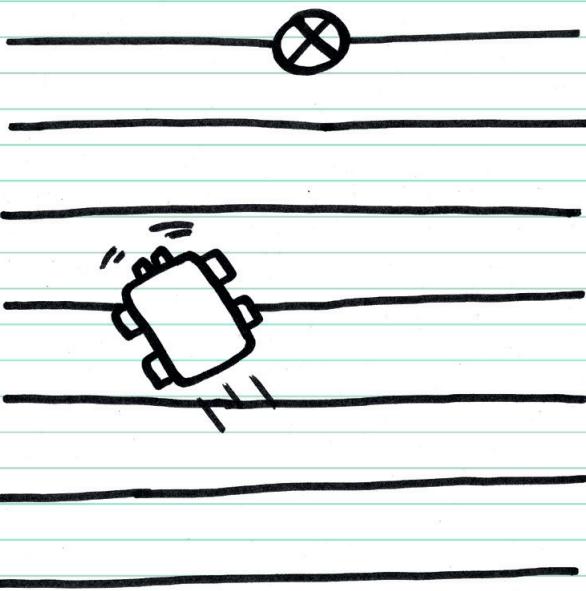
Voor de competitiebeesten: probeer eens te petanqueren met enkele robots! Leg een voorwerp in het midden van een cirkel en rijd er zo dicht mogelijk bij met je robot. Je zal de snelheden van de robots moeten modelleren om te kunnen voorspellen wat ze gaan doen.

DOE-HET-ZELF!

Meet de snelheid van je robot tien keer. Neem het gemiddelde van deze waarden. Gebruik het resultaat als de snelheid van je robot voor de volgende opdracht.

- Plaats een poppetje op een terrein.
- Maak met plakband een cirkel rond het poppetje.
- Zet de robots aan de buitenkant van de cirkel met de voorkant gericht naar het poppetje.
- Laat de robots rijden. De robot die het dichtst bij het poppetje stopt, wint.
- Opgelet: de robots mogen het poppetje niet raken of omverrijden!





6

Sensoren

Analoge en digitale sensoren

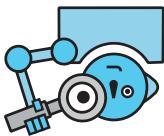
Sensoren zijn simpelweg onmisbaar in moderne elektronica. Neem nu een smartphone: zo'n toestel heeft een lichtsensor die het omgevingslicht meet en de helderheid van het scherm daaraan aanpast; een afstandssensor die zorgt dat je scherm uitgaat wanneer je het tegen je oor houdt en nog heel wat andere sensoren, zoals een versnellingsmeter en een gyroscoop.

Robots gebruiken sensoren om **informatie te verzamelen over hun omgeving**. Mensen doen hetzelfde met hun zintuigen (tastzin, gehoor, smaakzin, ...). Sensoren vormen dus eigenlijk de zintuigen van de robot. Twee soorten sensoren zijn belangrijk wanneer je met de Dwenguino aan de slag gaat: analoge sensoren en digitale sensoren.

Om een **analoge sensor te lezen**, sluit je hem aan op een van de analoge pinnen van de Dwenguino (A0-A7). Een digitale sensor kan je rechtstreeks aansluiten op een digitale pin van de Dwenguino (0-23).

De Dwenguino is voorzien van de meest eenvoudige soort sensoren: knoppen. Een knop is een digitale sensor. Hij kan namelijk maar twee toestanden aannemen: ingedrukt of niet ingedrukt.

Wanneer een knop niet ingedrukt is, wordt een '1' gelezen. Wanneer de knop ingedrukt is, wordt een '0' gelezen. Had je het tegenovergestelde verwacht? Lees dan zeker het kadertje!



Je dacht waarschijnlijk dat er een '0' zou gelezen worden als de knop niet ingedrukt is. Toch is het tegenovergestelde waar. Hoe dat komt? **De 'pull-up'-weerstand** is hiervoor verantwoordelijk. In de praktijk staat een knop meestal tussen de ingang van de microcontroller en de grond (0 V). **Wanneer de knop ingedrukt is, wordt er dus een '0' ingelezen.** Daarnaast wordt de ingang van de microcontroller via een weerstand met de 5 V verbonden. Deze 'pull-up'-weerstand trekt de ingang van de microcontroller als het ware op zolang de knop niet ingedrukt is. Met andere woorden: je krijgt een '1' wanneer de knop niet ingedrukt wordt. Als er geen 'pull-up'-weerstand in het spel is, dan is enkel de situatie met ingedrukte knop gegarandeerd, in het andere geval staat de pin te zweven.

Andere voorbeelden van digitale sensoren zijn de **sonar-sensor**, de **kleursensor**, de **geluidssensor** en de **temperatuursensor**. Deze zitten allemaal in het Dwenguino-'Startpakket Plus'. Daarnaast bevat het pakket enkele **analoge sensoren**, zoals een **lichtsensor**.

6

Sensoren

Autonome robot met sonar

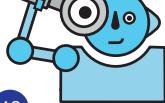
Een veelgebruikte sensor is de **sonar-sensor**. Zo'n sensor zorgt er enerzijds voor dat je robot niet tegen een muur rijdt en anderzijds dat hij een parcours langs een muur kan afleggen.

Een sonar-sensor meet de afstand tot aan een object. Hiervoor stuurt hij een ultrasoon geluidssignaal uit. Als er een voorwerp binnen bereik is, weerkaatst deze geluidsgolf erop. Door de tijd te meten tussen het verzenden en het ontvangen van het geluidssignaal, bepaalt de sensor heel nauwkeurig de afstand tot het object. De sonar-sensor gaat met andere woorden op dezelfde manier tewerk als vleermuizen of walvissen die obstakels willen detecteren.



Vooraleer je je robot programmeert, sluit je de sonar-sensor aan. Dit gaat gemakkelijk via het **uitbreidingsbord** van de Dwenguino.

De afstand Δ tussen het obstakel en de sonar-sensor bepaal je met de **formule $\Delta = 340 * (8t / 2)$** , met Δ de afstand in meter en $8t$ de gemeten tijd in seconden tussen het uitsuren en het ontvangen van het ultrasoon signaal. Hierbij veronderstel je dat de geluidssnelheid gelijk is aan 340 meter per seconde. Ingewikkeld? Gelukkig voorziet de **Dwenguino-bibliotheek** deze berekening en krijg je bij het programmeren de afstand meteen in centimeter terug.



De sonar-sensor aansluiten

Het uitbreidingsbord van de Dwenguino bevat gaatjes die kolomsgewijs met elkaar verbonden zijn.

Daarnaast vind je er **de connector van de Dwenguino** terug, met onder meer:

- Aansluitingen voor de referentiespanning of de grond (aangeduid met GND of -);
- 3,3 V (3.3) of 5 V-spanning (5 of +);
- Digitale in- en uitgangen (aangeduid met een nummer);
- Analoge ingangen (aangeduid met A<x>).

De sonar-sensor heeft vier pinnen:

- **V_{cc}:** de 5 V-voeding, soms ook aangeduid met een +;
- **Trigger:** een pin die aangeeft wanneer de ultrasone sensor moet beginnen te meten;
- **Echo:** een pin die aangeeft wanneer het weerkaatste ultrasone signaal ontvangen is;
- **Ground:** de referentiespanning of de grond, soms ook aangeduid met een -.

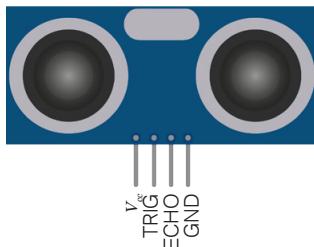
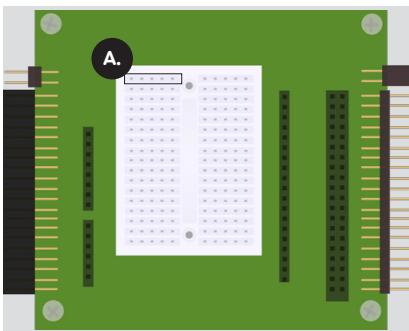
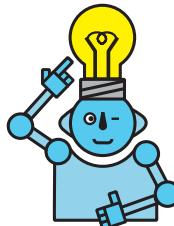
Connecteer de 5 V- en de GND-pin van de sonar-sensor met respectievelijk de + en de - op het uitbreidingsbord. Je volgt het best de **conventie**, namelijk: een rode draad gebruiken voor de 5 V en een zwarte draad voor de grond.

6

Sensoren

Tot slot verbind je de **trigger- en de echopin** met eender welke beschikbare digitale I/O-pin (aangeduid met een nummer). In dit voorbeeld sluiten we de triggerpin aan op pin 11 en de echopin op pin 12.

Om beschadigingen te vermijden, koppel je de elektronica het best los van je computer en/of de batterij wanneer je de sensoren aansluit. Controleer je schakeling grondig voordat je de Dwenguino terug aanzet (via de computer of de batterij). Zijn alle pinnen wel aangesloten zoals op het schema? Kijk bovendien heel goed na of er nergens een kortsluiting is: de 5 V (de +) mag nooit rechtstreeks verbonden zijn met de grond (de -).



A. De gaatjes binnen de kader zijn met elkaar verbonden

De meetwaarde van de sonar uitlezen en opslaan

Wil je de **afstand** van de sonar-sensor lezen? Daarvoor gebruik je het **sonar-blok**. Aan dit blok geef je mee op welke pin-nummers de sonar-sensor aangesloten is.

In ons voorbeeld zijn de trigger- en echopinnen respectievelijk aangesloten op de pinnen 11 en 12 van de Dwenguino.

Het sonar-blok geeft de afstand in centimeter tot het dichtstbijzijnde object terug.
Let wel: als het dichtstbijzijnde object op meer dan twee meter van de sensor staat, krijg je de waarde '0' terug. Hou hier rekening mee in je programma.

Om de afstand van de sonar-sensor uit te lezen op het lcd-scherm, maak je in DwenguinoBlockly een variabele met het **'stel... in op'-blok**. De waarde die je daarbij gebruikt? De meetwaarde van het sonar-blok.

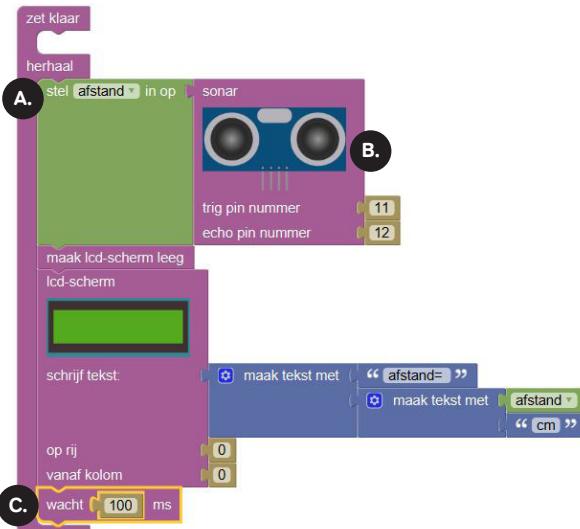
Geef deze variabele de logische naam 'afstand' en laat ze op het lcd-scherm verschijnen. Je doet dit met behulp van het lcd-blok en het 'maak tekst met'-blok (om tekst en variabele samen te gebruiken). Vergeet niet dat je de tekst **ook effectief moet kunnen aflezen** van het scherm. Als het te snel gaat, zal dat niet lukken. Je moet met andere woorden een **'wacht'-blok** invoegen.

6

Sensoren

A.

De variabele "afstand":
een plaats in het
geheugen waar je gehele
getallen kan opslaan.



B.

De afstandsmeting:
de sonar-sensor meet de
afstand tot een object.
Deze afstand wordt in
centimeter opgeslagen
in de variabele met de
naam 'afstand'.

C.

Het wacht-blok:
met dit blok leg je vast
om de hoeveel tijd de lus
doorlopen wordt. Deze
tijd wordt weergegeven
in milliseconden (ms).

We introduceerden **een variabele** in de tekst en het programma. Een variabele is een plaats in het geheugen die je een naam geeft en waaraan je een waarde toekent. Deze waarde kan je verder in het programma gebruiken door de variabele op te roepen.

In de voorbeeldcode hiernaast kennen we de meetwaarde van de sonar-sensor toe aan een variabele met de naam 'afstand'. Omdat deze variabele in het 'herhaal'-gedeelte staat, zal de sonar-sensor zijn meting telkens opnieuw uitvoeren en zal de gevonden meetwaarde telkens opnieuw worden toegekend aan de variabele 'afstand'. In je programma zorg je ervoor dat de robot een taak uitvoert die aangepast is aan die (veranderende) meetwaarde.

Merk op dat je meerdere variabelen kan aanmaken. Stel, je gebruikt twee verschillende sonar-sensoren. Dan kan je hun meetwaarden toekennen aan de variabelen 'linkerafstand' en 'rechteraafstand', zoals hieronder:



6

Sensoren

Autonome robot met grondsensor

Wil je dat je robot een zwarte lijn op een witte ondergrond waarneemt, of de witte rand van een sumoring? Dan heb je een grondsensor nodig.

Dankzij zo'n sensor legt je robot een parcours af langs een bepaalde lijn en blijft hij binnen de arena. Je kan de sensor ook gebruiken als nabijheidssensor om kleine afstanden tot objecten te meten (< 1 cm).

Hoe dit nu allemaal in zijn werk gaat? De sensor stuurt infrarood licht uit via de led. Dit licht weerkaatst op een reflecterend object (meestal een wit oppervlak) naar een fototransistor. Afhankelijk van de afstand tot de ondergrond en de kleur en ruwheid ervan, vangt de fototransistor meer of minder licht op. De hoeveelheid weerkaatsing krijg je terug als een getal tussen 0 en 1023 met behulp van het 'lees analoge waarde van'-blok.



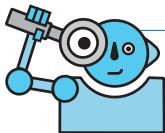
De grondsensor aansluiten

Eerst en vooral moet je de grondsensor voorzien van de nodige bedrading en weerstanden.

De stroom door de infrarood-led moet beperkt worden door middel van een voorschakelweerstand van $220\ \Omega$. De fototransistor moet dan weer voorzien worden van een weerstand van $10\ k\Omega$. Na het solderen heeft de grondsensor drie pinnen:

- V_{cc} : de 5 V-voeding, soms ook aangeduid met een +;
- V_{out} : met deze pin kan je de waarde opvragen die de sensor heeft gemeten;
- **Ground**: de referentiespanning of de grond, soms ook aangeduid met een -.

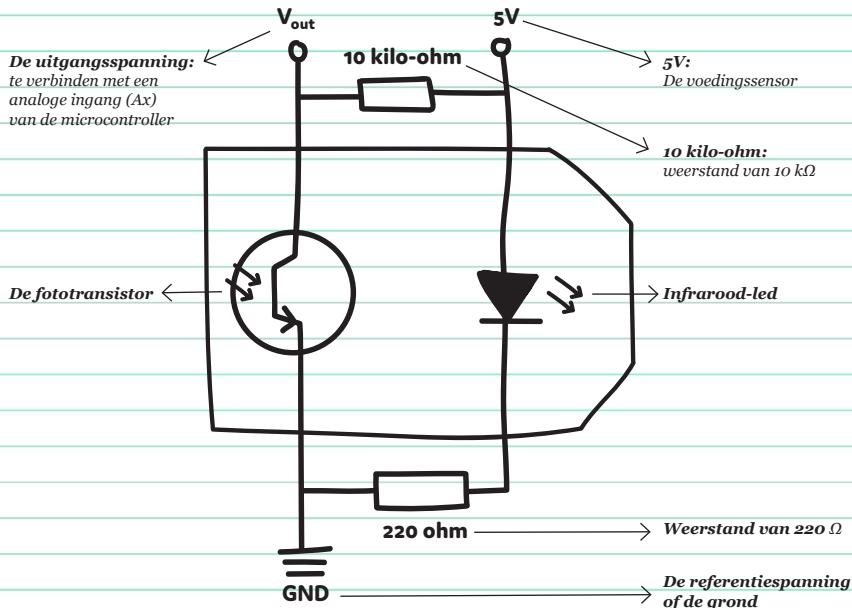
Connecteer de 5 V- en de GND-pin van de sensor met respectievelijk de + en de - op het uitbreidingsbord. Je doet er goed aan om de **kleurconventies** te volgen: een rode draad voor de 5 V en een zwarte draad voor de grond. Tot slot verbind je de V_{out} -pin met eender welke beschikbare analoge I/O-pin (aangeduid met A0 tot A7). In dit voorbeeld sluiten we de V_{out} aan op pin A1.



De grondsensor is een **analoge sensor**.
De sonar-sensor is een **digitale sensor**.

6

Sensoren



Schema van een grond-sensor: de TCRT5000

De meetwaarde van de grondsensor uitlezen en opslaan

Om de meetwaarde van de grondsensor te lezen, gebruik je het 'lees analoge waarde van'-blok.

Aan dit blok geef je mee op welk pin-nummer de V_{out} -pin van de grondsensor aangesloten is. In ons voorbeeld is de V_{out} -pin aangesloten op de analoge pin A1 van de Dwenguino.

De grondsensor geeft de hoeveelheid weerkaatsing weer als een **getal tussen 0 en 1023**. Een donker oppervlak weerkaatst weinig; een licht oppervlak veel.

stel oppervlak in op lees analoge waarde van pin A1

Om de meetwaarde van de grondsensor te kunnen uitlezen op het lcd-scherm, maak je in DwenguinoBlockly een variabele met het 'stel... in op'-blok. De variabele krijgt als waarde de meetwaarde van het 'lees analoge waarde van'-blok, en als naam 'oppervlak'. Wil je de waarde van deze variabele op het lcd-scherm laten verschijnen? Gebruik hiervoor het lcd-blok en het 'maak tekst met'-blok (om tekst en variabele samen te gebruiken). Geef jezelf opnieuw de kans om de tekst op het scherm rustig te lezen door een wacht-blok in te voegen.

6

Sensoren

Automaten: een manier om met sensoren om te gaan

Een robot bevindt zich altijd in één toestand, en die toestand hangt af van de waarden die zijn sensoren meten. Wanneer een van deze waarden verandert, kan de robot overgaan naar een andere toestand.

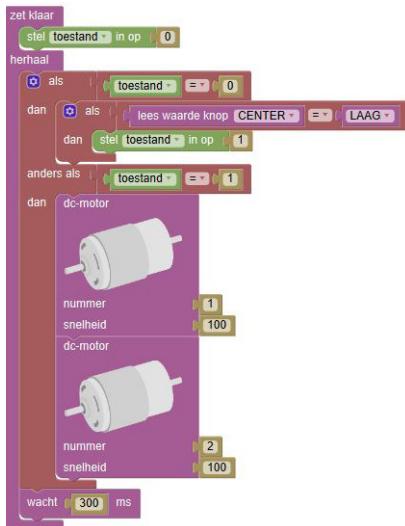
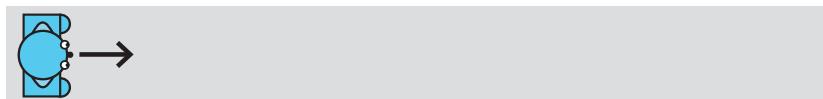
Om het gedrag van de robot voor te stellen – maar ook van stofzuigers, broodverdelers, ... – maken we gebruik van een (eindige) automaat. Zo'n automaat is een diagram dat bestaat uit cirkels (die overeenkomen met de toestanden) en pijlen (die de overgangen tussen de toestanden weergeven).

Op de volgende pagina's bouwen we samen de automaat voor een rijdende robot die na een druk op de centrale knop op de Dwenguino op zoek gaat naar een zwart vlak op een witte vloer.

De toestand waarin de robot zich kan bevinden wordt beschouwd als een variabele met de naam 'toestand'. Elke mogelijke toestand komt overeen met een andere waarde van de variabele.

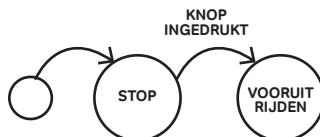
Naargelang de toestand waarin de robot verkeert, voert hij een andere taak uit. Om dit te programmeren, maak je gebruik van een **keuzestructuur**. Je gebruikt **meerdere als-blokken**.

Om te beginnen geef je je robot een **startknop**. Wanneer je deze knop op de robot indrukt, rijdt hij vooruit.



Je kan de startknop beschouwen als een eenvoudige sensor.

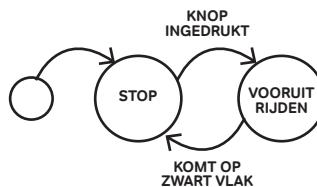
Deze sensor leest een waarde LAAG of HOOG: LAAG als de knop ingedrukt is, HOOG als hij niet ingedrukt is.



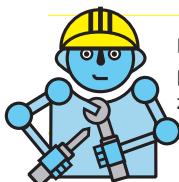
6

Sensoren

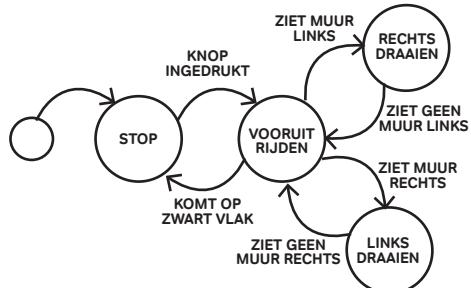
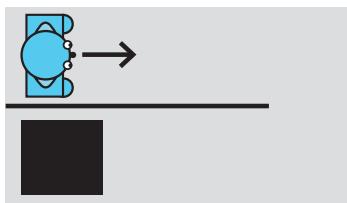
De **volgende stap**? De robot laten rijden tot aan een zwart vierkant. Hiervoor gebruik je de **analoge grondsensor**, die infrarood licht uitstuurt en meet hoeveel van dit licht weerkaatst wordt door de grond. Een donker oppervlak weerkaatst weinig; een licht oppervlak veel. De hoeveelheid weerkaatsing krijg je terug als een getal tussen 0 en 1023.



Als **laatste stap** voeg je **twee afstandssensoren** toe om te kijken of er links of rechts een muur is. De meetwaarde van deze digitale sensor kan je lezen met het sonar-blok. Gelukt? Dan heb je een **volwaardige, intelligente robot**: proficiat!



Probeer het zelf. Gebruik het **toestandsdiagram** om een robot te programmeren die op zoek gaat naar een zwart vlak. Zorg ervoor dat hij zijn afstandssensoren gebruikt om weg te draaien van de muur.



7

Nodig eens een robot uit in de klas!

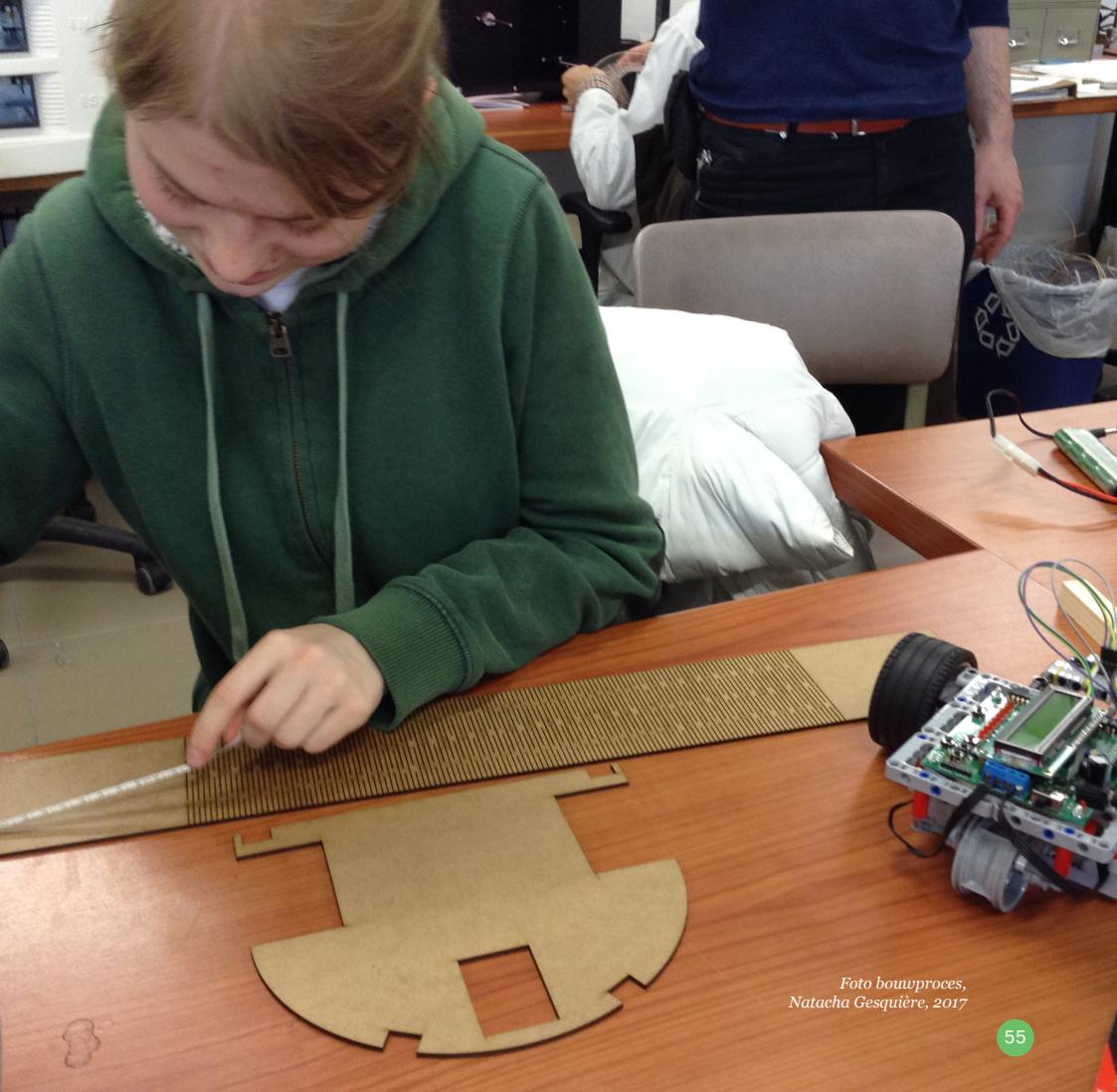
Een robotproject opzetten in de klas? Dat kan je op verschillende manieren doen. Ofwel als een integraal STE(A)M-project* ofwel als een losstaand initiatief om de beginselen van programmeren en elektronica mee uit te leggen.

Misschien kan je je collega's geschiedenis, kunst of Nederlands wel warm maken om mee te werken? Als je wilt, kan je er zelfs een **wedstrijd** aan koppelen: organiseer een robot-wedstrijd op een opendeurdag of neem deel aan de **robotcompetitie van de UGent en Dwengo**.

De ervaring leert dat de meeste leerlingen graag een eigen robot ontwerpen, en voor velen is het wedstrijdelement de kers op de taart. Je kan zelf opdrachten verzinnen om de activiteit voor elke leerling boeiend te houden, maar leerlingen bedenken ook zelf spontaan opdrachten. Zin om te differentiëren? De brede waaier aan sensoren en motoren biedt veel mogelijkheden op het vlak van **soorten opdrachten en moeilijkheidsgraden**.

Hou er wel rekening mee dat de bouw van een robot wat tijd vergt. Naast **ruimtelijk inzicht** moeten de leerlingen over een zekere **handigheid en doorzettingsvermogen** beschikken.

*STE(A)M = Science, Technology, Engineering, Art en Mathematics



*Foto bouwproces,
Natacha Gesquière, 2017*

7

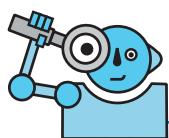
Nodig eens een robot uit in de klas!

Ga je ook solderen, lasercutten of 3D-printen? Het is een uitdaging, maar laat dat je vooral niet tegenhouden!

Een **mogelijke didactische aanpak** bestaat erin om leerlingen in groepjes van twee of drie te laten samenwerken en hen stelselmatig nieuwe uitdagingen voor te schotelen, gekoppeld aan specifieke leerdoelstellingen. Door deze opdrachten stap voor stap te doorlopen, komen de leerlingen steeds dichter bij de realisatie. Het is motiverend en ontroerend om te zien hoe ze hun successen vieren.

De grootste uitdaging voor de leerkracht? Die ligt misschien wel in het opzoeken van de grens: je leerlingen net genoeg informatie geven om hen toch zo zelfstandig mogelijk tot een oplossing te laten komen. Te veel informatie geven is verleidelijk, maar zo hou je je leerlingen niet voldoende gemotiveerd en intellectueel geprikkeld.

Dit boekje is ontwikkeld om absolute beginners te leren programmeren. Als grafisch programmeren niet uitdagend genoeg is voor jou, dan kan je een kijkje nemen in ons boek over tekstueel programmeren.



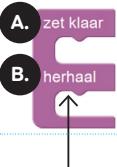
Dit boekje kwam tot stand in samenwerking met de leerlingen van de eerste en de derde graad van de Sint-Bavohumaniora in Gent. Een **filmpje van het volledige proces** vind je op <http://www.dwengo.org/nl/bouwjerobot>.



Foto Dwengo-project 'Udavi' in India,
kinderen bouwden hun eigen robot uit
recyclage-materialen, Francis wyffels, 2017

8

Spiekbriefje



Het 'zet klaar'-gedeelte:
wordt één keer uitgevoerd bij het
opstarten van de Dwenguino.

Het 'herhaal'-gedeelte:
bevat acties die herhaaldelijk uitgevoerd
moeten worden.



Pauzeren van de code:
geef een tijd mee in milliseconden
(1000 milliseconden = 1 seconde).



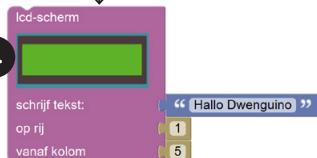
Aansturen van een motor:
verduidelijk welke motor je wilt aansturen
(1 of 2) en geef bij snelheid een getal in van
-255 tot 255.

E. maak lcd-scherm leeg

E.

Het lcd-scherm leegmaken

F.



F.

Schrijven naar het lcd-scherm:
schrijf de tekst Hallo Dwenguino op de
tweede rij en de zesde kolom van het
lcd-scherm.

G.

stel [oppervlak v] in op [Lees analoge waarde van [pin A1 v]]

G.

*Variabele een naam geven en er de
meetwaarde van een analoge sensor
aan te kennen:*
gebruik het 'stel... in op'-blok om de
meetwaarde in een variabele op te slaan.
Ken de waarde eraan toe. De gebruikte pin
moet hier ook vermeld worden.

8

Spiekbriefje

H.



H.

Afstanden meten met de sonar-sensor:

lees de waarde van de sensor met het sonar-blok, geef hierbij de juiste trigger- en echopin op. Gebruik het 'stel... in op'-blok om de waarde in een variabele op te slaan.

I.

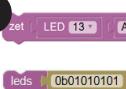


I.

Gebruik de variabele 'afstand' waar nodig:

je kan de variabele die de afstand bevat dan gebruiken waar je hem nodig hebt, bijvoorbeeld om de gemeten afstand op het lcd-scherm te laten verschijnen.

J.



J.

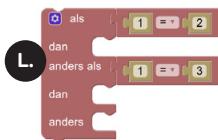
Leds aanzetten:

je kan dit doen met twee blokken. Het tweede blok laat je toe om meerdere leds tegelijk aan of uit te zetten.



K.

Keuzestructuur als-blok:
pas wanneer de voorwaarde na de
als waar is, wordt de code na de dan
uitgevoerd.



L.

Keuzestructuur als... anders als... anders:
door op het icon met het tandwiel te
klikken, kan je de keuzestructuur als
aanpassen naar een keuzestructuur als...
anders als... anders. Zo kan je meerdere
voorwaarden checken en een optie
toevoegen wanneer niet voldaan is
aan de andere voorwaarden.

Copyright © 2017 Dwengo vzw

ISBN: 9789081991759

D/2017/Dwengo/2

NUR 980, 257

We willen graag zoveel mogelijk mensen laten kennismaken met programmeren en ze warm maken voor wetenschap en techniek.
Daarom brengen we dit boek uit onder Creative Commons:



De gebruiker mag:

- het werk kopiëren, verspreiden en doorgeven
- remixen – afgeleide werken maken

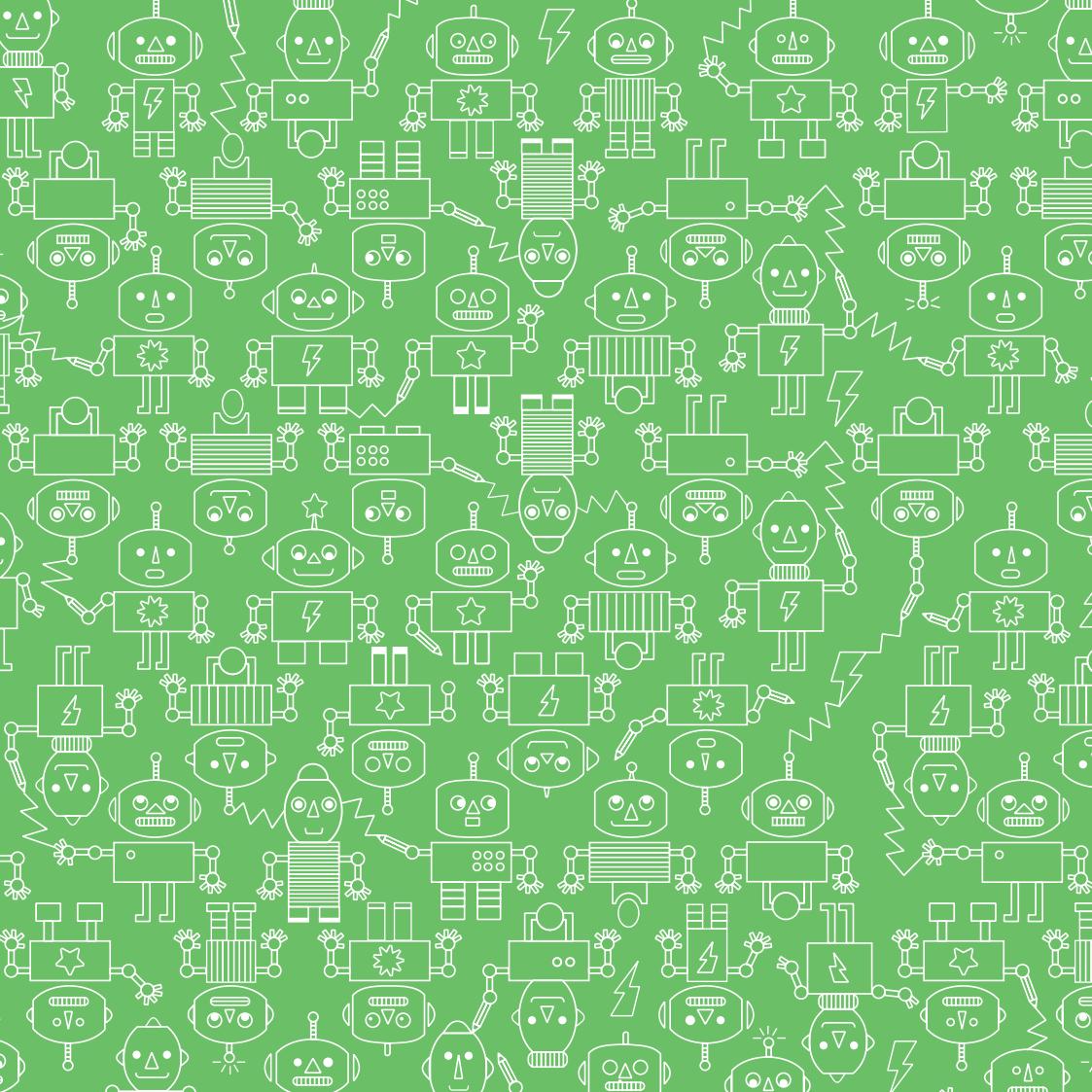
Onder de volgende voorwaarden:

- **Naamsvermelding:** de gebruiker dient bij het werk steeds Dwengo vzw te vermelden (maar zonder de indruk te wekken dat Dwengo vzw instemt met je werk of jouw gebruik van het werk).
- **Niet-commercieel:** de gebruiker mag het werk niet voor commerciële doeleinden gebruiken.
- **Gelijk delen:** indien de gebruiker het werk bewerkt, kan het daaruit ontstane werk uitsluitend krachtens dezelfde licentie als de onderhavige licentie of een gelijksoortige licentie worden verspreid.

info@dwengo.org • www.dwengo.org

Design: Studio Blomme

dwengo





De voorbereiding van de robot-wedstrijd vroeg heel veel tijd, maar die stak ik er met plezier in!

Anthony, eerste middelbaar

In dit project verleggen leerlingen hun grenzen. Het zijn mooie momenten wanneer leerlingen hun eigen oplossing beoordelen en evalueren, om dan te zoeken naar verbeteringen. Het werkt, dus blijf er nu af! moeten we in een gewone les zelden gebruiken.

Jan Colson, leerkracht wiskunde
en programmeren,
Sint-Bavohumaniora Gent

Het was een superleuke
en leerrijke ervaring.

Kamron, vijfde middelbaar



9 789081 991759

Bekijk alle boeken van Dwengo
op <http://dwengo.org/nl/books>.