

Inhoudsopgave



Waarvoor hebben we robots nodig?	Blz. 86
Robots van fischertechnik	Blz. 88
Actuatoren	Blz. 88
Sensoren	Blz. 88
ROBO Interface	Blz. 89
Software ROBO Pro	Blz. 89
Voeding	Blz. 89
Procedures voor het experimenteren	Blz. 90
De eerste stappen	Blz. 90
De eerste eenvoudige robot	Blz. 92
Intelligente rijdende robot	Blz. 94
Basismodel	Blz. 94
De Lichtzoeker	Blz. 96
De Spoorzoeker	Blz. 98
Robot met hindernisherkenning	Blz. 99
Lichtzoeker met hindernisherkenning	Blz. 102
Robot met randherkenning	Blz. 104
De lopende robot	Blz. 107
Uitbreidingsmogelijkheden	Blz. 109
Infrarood handzender	Blz. 109
ROBO RF Data Link	Blz. 109
ROBO I/O-Extension	Blz. 110
Problemen verhelpen	Blz. 111

Waarvoor hebben we robots nodig?

■ Het begrip 'robot' werd voor het eerst in 1923 gebruikt in de roman "Golem" van Carel Capek. Deze kunstmatig gecreëerde figuur moest de juiste vaardigheden hebben om werkzaamheden over te nemen van de mens.

In de jaren 30 en 40 van de 20e eeuw werd de robot veel meer als een automaat gezien. Diverse pogingen om hem van menselijke eigenschappen te voorzien, zoals bijv. een hoofd met knipperlichten als ogen, werken tegenwoordig alleen nog maar op de lachspieren. Van mobiliteit of zelfs intelligentie is bij deze machines weinig te merken. Omdat het besturingsprincipe van grote invloed is op de robottechniek werd de bouw van robots een stuk realistischer toen de eerste elektronische schakelingen beschikbaar werden. Het streven robots van "intelligentie" te voorzien is vandaag de dag nog altijd een onderwerp van onderzoek bij tal van bedrijven, instituten en universiteiten.

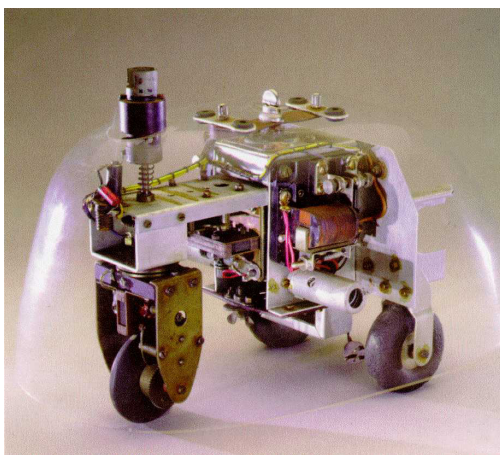


■ De eerste oplossingen dacht men te vinden met de opkomst van de zogenaamde cybernetica. De term "cybernetica" is afgeleid van het Griekse woord 'kybernetes'. De kybernetes was de navigator op een Griekse roeiboot. Hij moest de locatie van het schip bepalen en de noodzakelijke koers tot het doel berekenen.

Cybernetica moest de robot "intelligent" maken. Hoe moet je je een dergelijk intelligent gedrag eigenlijk voorstellen?

We zullen proberen om dit met een vergelijking te verduidelijken. Iedereen heeft wel eens gezien hoe een motvlinder zich in het schijnsel van een lamp gedraagt. Hij herkent de lichtbron en vliegt er naartoe om dan vlak voordat hij tegen de lamp zou vliegen uit te wijken. Het is duidelijk dat de mot voor dit gedrag de lichtbron moet herkennen, een weg daar naartoe moet bepalen en er naartoe moet vliegen. Deze vaardigheden zijn gebaseerd op instinctieve, intelligente gedragspatronen van het insect.

Laten we nu eens proberen deze vaardigheden over te brengen op een technisch systeem. We moeten de lichtbron herkennen (optische sensoren), een beweging uitvoeren (motoren aansturen) en we moeten een zinnvolle samenhang realiseren tussen herkenning en beweging (het programma).



■ Het hierboven beschreven principe heeft de Engelsman Walter Grey in de jaren 50 van de twintigste eeuw in praktijk gebracht. Met behulp van eenvoudige sensoren, motoren en elektronische schakelingen creëerde hij diverse "cybernetische" dieren, die elk voor zich een zeer specifiek gedrag, zoals bijv. dat van een mot, vertoonden. Op de foto is een kopie te zien van de "cybernetische" schildpad die te zien is in het Smithsonian Museum in Washington.

Op basis van het bovenstaande gaan wij voor onze robots ook de gewenste gedragspatronen opstellen en proberen we deze in de vorm van programma's aan de robots duidelijk te maken.

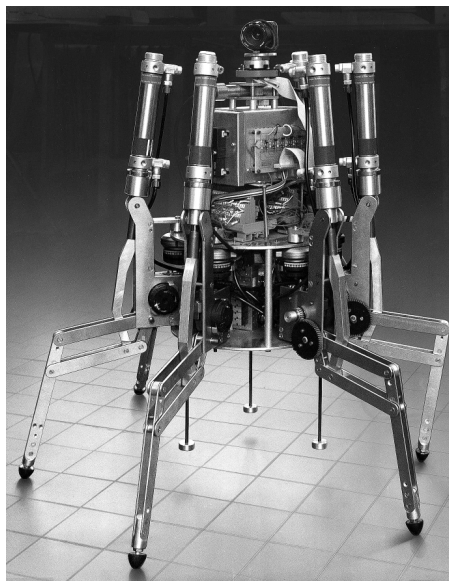
■ Maar waarvoor hebben we eigenlijk mobiele robots nodig? Laten we het gedrag van onze denkbeeldige mot eens op technische apparaten toepassen. Een eenvoudig voorbeeld daarvan is licht zoeken. We modificeren de lichtbron door een lichte streep, de richtlijn, op de vloer aan te brengen en de sensoren naar beneden te laten wijzen. Met behulp van dergelijke richtlijnen kan een mobiele robot zich bijvoorbeeld in een magazijn oriënteren. Via aanvullende informatie, bijv. in de vorm van een barcode op bepaalde posities op de lijn, krijgt de robot opdrachten om andere acties uit te voeren, zoals bijv. een pallet opnemen of afzetten.



Dergelijke robotsystemen bestaan al in de praktijk. In grote ziekenhuizen doen zich vaak lange transportwegen voor verbruiksmaterialen zoals beddengoed voor. Het transport van dit materiaal door het verplegend personeel is erg kostbaar en gaat deels gepaard met zware lichamelijke arbeid. Bovendien gaan dergelijke activiteiten ten koste van de tijd die beschikbaar is voor het verzorgen van de patiënten.

■ Sinds een aantal jaren houden wetenschappers zich bezig met een andere, in de natuur wijd verbreide bewegingsvorm, namelijk het lopen. Er worden robots ontwikkeld, die in staat zijn zich op poten voort te bewegen. Een voorbeeld van een lopende robot op zes poten is de elektropneumatische "Achille" die ontwikkeld is aan de Koninklijke Militaire Academie in Brussel. Deze robot is uitgerust met zes poten en een camera aan de bovenkant en moet mechanisch reageren op verhoogde of verlaagde hindernissen (objecten of gaten).

Dergelijke lopende machines kunnen overal worden ingezet waar wiel- of kettingvoertuigen vrijwel onbruikbaar zijn, dus bijv. op zeer ongelijk of drassig terrein, om over hindernissen te klimmen, trappen te beklimmen, greppels te passeren of op moeilijk toegankelijke of gevaarlijke locaties zoals kerncentrales, in de mijnbouw of bij reddingsacties.



Hieruit valt af te leiden dat mobiele robots een zeer belangrijke plaats kunnen innemen in de moderne maatschappij.



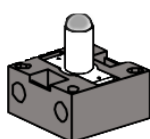
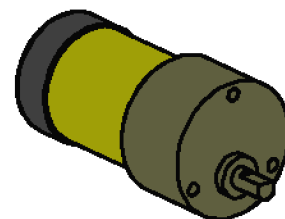
Robots van fischertechnik

■ Hoe kun je een robot maken van je fischertechnik-dozen? Voor een robot hebben we naast de sensoren (bijv. contactschakelaars) en actuatoren (bijv. motoren) veel mechanische onderdelen nodig om een model te kunnen bouwen. Hiervoor is de fischertechnik-doos ROBO Mobile Set de ideale basis. Deze doos bevat de volgende sensoren en actuatoren:

Actuatoren

Power Motor:

Twee van deze krachtige gelijkstroommotoren (9VDC/2,4W) met een ingebouwd drijfwerk en een tandwielreductie van 50:1 drijven de mobiele robotmodellen aan (d.w.z. dat de motor 50 keer ronddraait terwijl de as die uit de motor steekt in diezelfde tijd slechts eenmaal ronddraait).



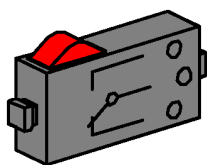
Lenslamp:

Dit gloeilampje (9VDC/150mA) dient voor het afgeven van eenvoudige lichtsignalen. In de glazen ballon van de lamp is een lens geïntegreerd, die het licht bundelt dat wordt uitgestraald. Door de lichtstraal op een helderheidssensor (zoals een fototransistor, zie hieronder) te richten, kun je een sensor bouwen die licht en donker onderscheidt. De lamp kan ook worden gebruikt voor het aangeven van bepaalde toestanden of als knipperlicht voor waarschuwingsmeldingen. In de bouwdoos wordt de lamp samen met 2 foto transistors als speciale sensor gebruikt voor lijnherkenning.

Sensoren

■ Bij sensoren maakt men onderscheid tussen digitale en analoge sensoren.

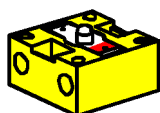
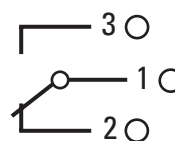
De contactschakelaar is een voorbeeld van een digitale sensor. Digitale waarden kunnen slechts 2 verschillende toestanden aannemen. Deze toestanden worden met 0 resp. 1 aangegeven. Voor de contactschakelaar betekent "0" dat er geen stroom tussen de aansluitingen loopt en betekent "1" dat er wel stroom is.



De **contactschakelaar** van fischertechnik, die in het programma ROBO PRO 'toets' wordt genoemd, is uitgevoerd als een wisselschakelaar. Daarom heeft hij 3 aansluitingen. Als de rode knop wordt ingedrukt, wordt er mechanisch een schakelaar geactiveerd, die aansluiting 1 en 3 met elkaar verbindt. Tegelijkertijd wordt het contact onderbroken tussen aansluiting 1 en 2 die in rusttoestand met elkaar verbonden waren. Op deze manier kunnen beide mogelijke uitgangsposities worden afgevraagd:

in rusttoestand gesloten (aansluiting 1 en 2 bezet)

in rusttoestand open (aansluiting 1 en 3 bezet)



De **fototransistor** kan als digitale en als analoge sensor worden gebruikt. In het eerste geval dient hij voor het herkennen van duidelijke overgangen tussen licht en donker, zoals bijv. een gemarkeerde lijn. Ook is het mogelijk verschillende lichtsterktes te onderscheiden; de fototransistor werkt dan als een analoge sensor. Analoge waarden kunnen alle mogelijke waarden tussen hun laagste en hoogste waarde aannemen. Om deze waarden door de computer te kunnen laten verwerken, moeten ze worden omgezet in desbetreffende getalswaarden.

Bij de fototransistor gaat het overigens om een zogeheten halfgeleidercomponent, met elektrische eigenschappen die afhankelijk zijn van de lichtsterkte. Iedereen kent wel zonnecellen, waarmee stroom

opgewekt wordt uit zonlicht. De fototransistor kunnen we zien als een combinatie van een minizonnecel en een transistor. Lichtimpulsen (fotonen) die op de fototransistor vallen genereren een zeer geringe stroom, die vervolgens door de transistor wordt versterkt.

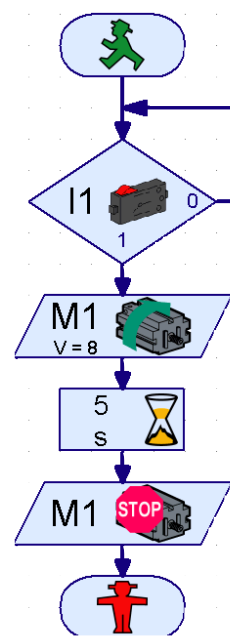
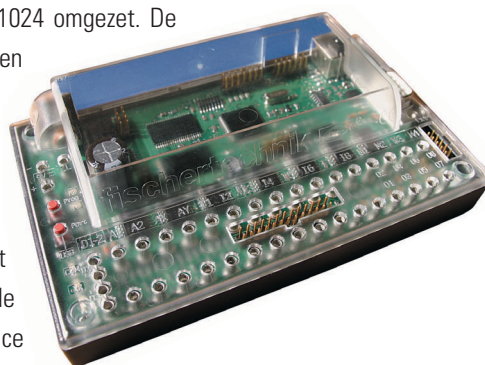
Let op:

Let bij het aansluiten van de fototransistor op de juiste aansluiting van de polen: rode markering = plus. Toelaatbare spanning: 30V max.

■ Verschillende sensoren en actuatoren kunnen worden aangesloten op de ROBO Interface en ermee worden geanalyseerd. Naast 8 digitale ingangen biedt de ROBO Interface de beschikking over een aantal analoge ingangen. Zo wordt bijvoorbeeld een op de ingangen AX en AY uitgeoefende weerstandswaarde tussen 0 en 5,5 k Ω in een getalswaarde van tussen 0 en 1024 omgezet. De meetwaarden van een helderheidssensor, zoals bijv. een fototransistor, worden daarmee geregistreerd en zijn beschikbaar voor verder gebruik. Op de analoge ingangen A1 en A2 kunnen spanningen tussen 0 en 10VDC worden gemeten.

De belangrijkste functie van de interface is de logische koppeling van de ingangswaarden. Daarvoor heeft de interface een programma nodig. Dit programma beslist hoe uit de ingangsgegevens (de sensorsignalen) de bijpassende uitgangsgegevens zoals motorstuursignalen etc. ontstaan. Met de ROBO Interface beschikken we over genoeg rekenvermogen om ook moeilijke programma's te ontwerpen.

ROBO Interface



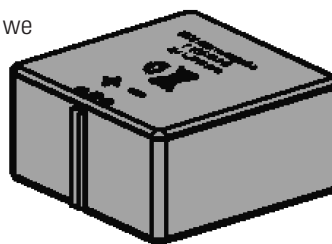
■ Om zo effectief mogelijk de voor de interface benodigde programma's te kunnen maken, is er een grafisch programmeeropervlak gemaakt. Het begrip "programmeeropervlak" staat hier voor een softwareoplossing die het ons mogelijk maakt op zeer gemakkelijke wijze onze programma's te maken. Dit gebeurt met behulp van grafische symbolen. De computer van de ROBO Interface kan eigenlijk alleen maar commando's uit zijn zogenaamde 'machinecommandopakket' uitvoeren. In principe zijn dit eenvoudige besturingsstructuren waarvan de toepassing buitengewoon moeilijk is voor beginners. De pc-software ROBO Pro stelt daarvoor de grafische elementen ter beschikking, die vervolgens worden omgezet in een taal die de interface kan uitvoeren.

■ Het enige dat we nog extra nodig hebben in aanvulling op de doos ROBO Mobile Set, is de Accu Set, art.nr. 34969. Deze bevat de accu die we gebruiken als de mobiele voeding voor onze robots en een speciale oplader voor de accu.

Het is het beste om de accu meteen met de oplader op te laden, zodat hij vol is als we met de experimenten willen beginnen.

Software ROBO Pro

Voeding



Procedures voor het experimenteren

■ We zullen stap voor stap de fascinerende wereld van mobiele robots binnengaan. We beginnen met een eenvoudige testopzet voor het testen van de basisfuncties van de interface en de sensoren. Vervolgens bouwen we eenvoudige modellen, die bepaalde opdrachten krijgen. Daarna proberen we het met steeds complexere systemen.

Als het maken van eigen programma's op een bepaald moment te gecompliceerd wordt en te lang duurt, kun je ook de meegeleverde voorbeeldprogramma's op de interface laden en de robots daarmee aansturen. Aan het einde van dit boekje vind je een hoofdstuk over het oplossen van fouten om je te helpen bij eventuele storingen.

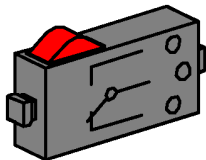
Een zeer belangrijk punt is de zorgvuldigheid bij het opbouwen en in gebruik nemen van de robots. Bij het aansluiten van de elektrische componenten moet je de instructies nauw opvolgen en liefst twee of zelfs drie keer controleren of alles klopt. Bij de mechanische constructies, ook bij eigen creaties, zijn onbelemmerde beweging en weinig speling op de tandwielen en bevestigingen van belang. Het wordt aan je creativiteit overgelaten om eigen programma's te schrijven en daarmee een nieuwe "gedraging" te definiëren. Dit wordt slechts beperkt door de geheugen- en rekencapaciteit van de hardware. De volgende voorbeelden dienen daarbij als interessante vingeroefeningen.

De eerste stappen

■ Na de theoretische bespiegelingen gaan we nu onze eigen experimenten uitvoeren. Waarschijnlijk sta je te popelen om direct te starten, en dan nog het liefst meteen met de grote lopende robot. Dat is natuurlijk mogelijk en als de bouw instructies zorgvuldig worden opgevolgd zal het ook wel lukken om het model op te bouwen.

Maar wat te doen als het niet werkt? In dergelijke gevallen moet je systematisch de oorzaak van de fout zoeken. Maar voordat we ons daarmee gaan bezighouden, controleren we eerst nog een keer de samenwerking tussen de computer en de interface.

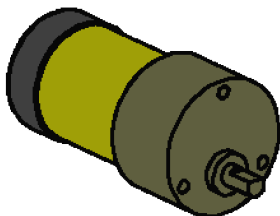
In hoofdstuk 1 en 2 van het handboek van de software ROBO Pro wordt het installeren van de besturingssoftware op de pc en het aansluiten van de interface beschreven. Met behulp van de Interfacetest kun je de verschillende sensoren en/of actuatoren testen



Contactschakelaar

Contactschakelaar

Je kunt nu bijv. een contactschakelaar aansluiten op de digitale ingang I1 en kijken hoe de toestand van de ingang verandert als de contactschakelaar geactiveerd wordt.



Powermotor

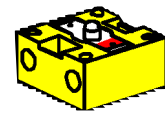
Power Motor

Je kunt de uitgangen controleren door een motor te verbinden met een motoruitgang, bijv. M1. Met de linker muisknop kun je de motor de andere kant op laten draaien en met de schuifregelaar kun je de snelheid aanpassen.

Fototransistor

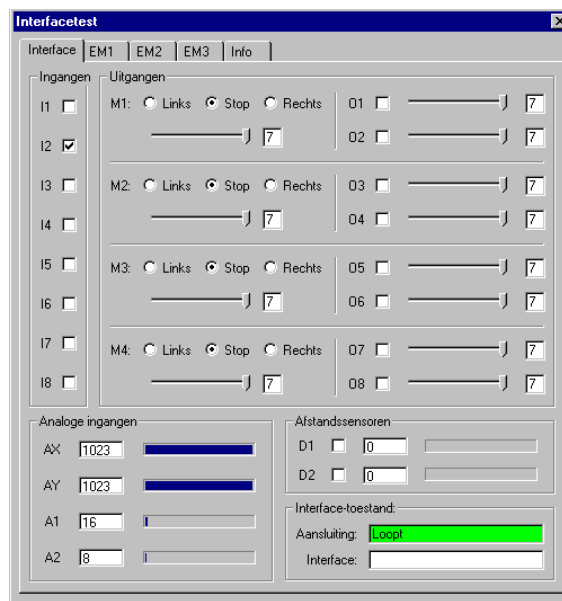
Als je ook de analoge ingang AX wil testen, kun je een fototransistor als analoge sensor toepassen. Waar bij de motor resp. de contactschakelaar de polariteit van de aansluitingen geen rol speelt (de motor draait in het ongunstigste geval verkeerd om), is een correcte aansluiting van de polen bij de fototransistor vereist om hem correct te laten werken.

Verbind het met rood gemarkeerde contact van de transistor met een rode aansluitstekker en verbind het andere contact met een groene stekker. De tweede groene stekker komt in de bus van ingang AX die zich dichtbij de rand van de interface bevindt; de tweede rode stekker past in de verder naar binnen gelegen bus van AX. (Let op: als de fototransistor wordt aangesloten op een digitale ingang I1-I8, moet de rode stekker in de bus het dichtst bij de rand van de behuizing worden gestoken).



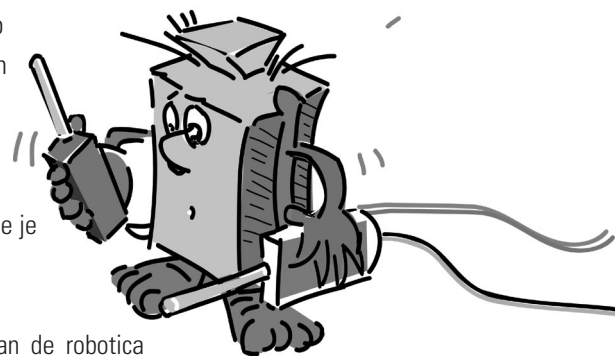
Fototransistor

Nu kun je met behulp van een zaklamp de belichtingssterkte van de fototransistor variëren en daarmee de uitslag van de blauwe balk van AX veranderen. Als de wijzer niet tot voorbij het maximum uitslaat, dan moet je de aansluitingen van de fototransistor nog eens goed bekijken. Als de wijzer daartegen ook bij een uitgeschakelde zaklamp op nul staat, dan kan het zijn dat de belichting in het vertrek, het omgevingslicht, te sterk is. De uitslag verandert dan als je de fototransistor afdekt.

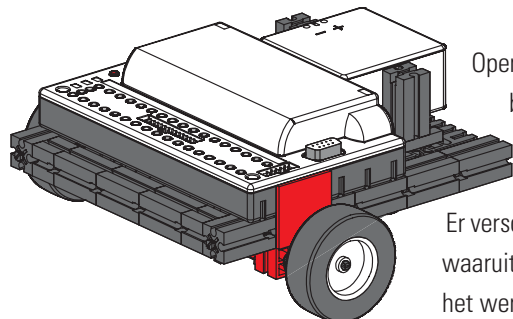


Om nog even kort terug te komen op de kleurcodering van de stekker: Let er goed op dat je tijdens het monteren altijd een rode stekker op de rode geleider aansluit en een groene stekker op de groene geleider. Als de juiste polariteit van belang is voor de opbouw van een circuit, neem je altijd een rode geleider voor de pluspool en een groene geleider voor de minpool. Dit lijkt misschien een beetje hypernauwkeurig, maar het systematisch fouten kunnen zoeken wordt aanzienlijk vergemakkelijkt als je je strikt aan de juiste kleurcoderingen houdt.

Met een eenvoudig programma gaan we onze eerste stappen op het gebied van de robotica afsluiten. In hoofdstuk 3 van het ROBO Pro handboek wordt het programma "Garagedeurbesturing" uitgelegd. Dit heeft weliswaar niets met mobiele robots te maken, maar het is prima geschikt om vertrouwd te raken met de ROBO Pro software. Om dit programma te kunnen uitvoeren is het voldoende de motor en drie contactschakelaars uit de bouwdoos ROBO Mobile Set op de interface aan te sluiten. De rest wordt uitvoerig beschreven in het softwarehandboek.



De eerste eenvoudige robot



■ Na de Interfacetest en de Garagedeurbesturing gaan we nu eindelijk de eerste robot in gebruik nemen. We bouwen het model "Eenvoudige robot" met de twee aandrijfmotoren op zoals beschreven in de bouw instructies. Dat gaat erg eenvoudig en snel, omdat dit model met opzet alleen voorzien is van de basiselementen die een robot nodig heeft om te kunnen rijden. De motoren sluiten we aan op uitgang M1 en M2.

Open de software ROBO Pro en maak een nieuw programma aan (BESTAND – NIEUW). ROBO Pro biedt verschillende moeilijkheidsniveaus om in te werken. Deze kunnen in ROBO Pro worden ingesteld in het menu onderdeel NIVEAU. Voorlopig is Niveau 1 goed.

Er verschijnt een leeg werkblad en aan de linkerrand van het beeldscherm verschijnt het elementvenster waaruit je de verschillende programma-elementen kunt selecteren met de linker muisknop om deze op het werkblad te plaatsen. Je kunt de eigenschappen wijzigen met de rechter muisknop.

Opdracht 1 (Niveau 1):

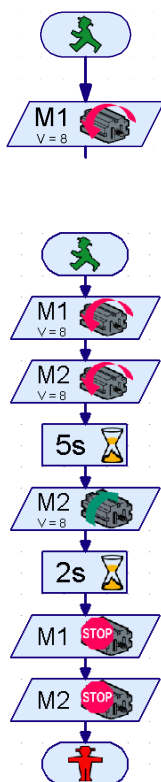
De "Eenvoudige robot" moet 5 seconden lang recht vooruit rijden, vervolgens 2 seconden lang een rondje draaien en daarna blijven stilstaan.



Tips:

De eerste robot programmeren we stap voor stap samen:

- We beginnen met het groene mannetje. Dit mannetje symboliseert de start van het programma.
- Vervolgens halen we het motorsymbool uit het elementvenster. Plaats dit symbool onder het startelement, zodat de verbindinglijn automatisch getekend wordt. In het venster met eigenschappen stellen we de motoruitgang "M1" en de draairichting "links" in en we bevestigen dit met OK.
- Onder dit symbool plaatsen we op dezelfde manier nog een motorsymbool. Daarmee schakelen we motor 2 in.
- Om gedurende een bepaalde tijd te wachten, maken we gebruik van het element Wachtijd, dat we onder het tweede motorsymbool plaatsen. We stellen de tijd hierbij in op 5 seconden.
- Vervolgens laten we de motor M2 in de andere richting draaien (rechtsom), waarna we 2 seconden wachten om tenslotte beide motoren uit te schakelen. Ons programma eindigt met het eindsymbool, het rode mannetje. Op de afbeelding is de complete programmastructuur te zien.



Als je niet zeker weet of alles klopt, kun je het programma vergelijken met het meegeleverde voorbeeldprogramma. Sla het eigen programma dan eerst op en laad het bestand Eenvoudige robot 1.rpp uit de voorbeeldmap van ROBO Pro (Standaardinstelling C:\Programmas Files\ROBO Pro\ Voorbeeldprogrammas\ROBO Mobile Set).

Als alles in orde is, dan kan het programma via de downloadfunctie in de interface worden geladen. Na het indrukken van de knop Download verschijnt er een dialoogvenster. Daar stellen we in dat het programma in FLASH-geheugen 1 moet worden geladen, waarna het direct na het downloaden moet worden gestart.

Direct na het downloaden rijdt ons model weg, hij draait even rond en blijft staan. Om het programma opnieuw te starten, kun je op de interface de Prog-knop even indrukken. De LED Prog1 knippert dan

weer, zolang het programma loopt. Daarna gaat hij constant branden. Het programma in het FLASH-geheugen blijft trouwens ook bewaard als de stroomvoorziening naar de interface onderbroken wordt. Probeer dit uit door de stekker van de accu uit te trekken. Steek de stekker weer in en selecteer het opgeslagen programma door zolang op de Prog-knop te drukken tot de LED Prog1 gaat branden. Druk nogmaals op de knop om het programma te starten.

Onze robot kan nog niet echt veel, maar daar gaan we wat aan doen.



Opdracht 2 (Niveau 1):

Om te zorgen dat onze robot niet al na 7 seconden blijft stilstaan, gaan we hem nu dansen leren.

- **Laat hem over verschillende afstanden recht vooruit, linksom, rechtsom en achteruit rijden en op verschillende snelheden.**
- **Deze procedure moet door blijven gaan totdat het programma beëindigd wordt met de Prog-knop op de interface.**

Tips:

- Pool eenvoudigweg de motoren steeds weer om om de robot in de gewenste richtingen te laten rijden.
- De snelheid van de motoren kun je in het venster met eigenschappen voor elk motorsymbool instellen tussen 1 en 8. Als M1 en M2 met verschillende snelheid in dezelfde richting draaien, dan maakt de robot een bocht.
- Om te zorgen dat het programma voortdurend herhaald wordt, moet je een verbindingslijn trekken van de uitgang van het laatste programma-element naar de lijn die in het eerste element naar binnen gaat.
- Een kant-en-klaar voorbeeld vind je onder [Eenvoudige robot 2.rpp](#).

Gefeliciteerd, je hebt nu je eerste eigen robot gebouwd en hem zelf geprogrammeerd. Hij is echter nog niet bijzonder intelligent, want hij herkent nog geen hindernissen en valt nog van de tafel als je niet oppast. Maar in de loop van de verdere experimenten verandert dat nog.



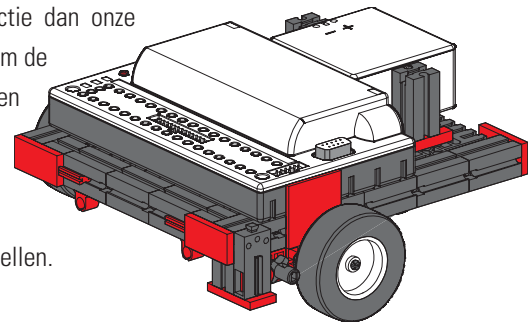
Intelligente rijdende robot

■ Om robots hun omgeving te laten herkennen, hebben ze sensoren nodig. De volgende modelsuggesties zijn voorbeelden van varianten van mobiele robots waarmee het gebruik van verschillende sensoren kan worden uitgetoond. Daarbij gaat het erom zowel interne toestanden van de robot, zoals de trajectmeting door impulswielen, als externe signalen, zoals bij het spoor- of licht zoeken, aan elkaar te koppelen. Bij elk model worden daarvoor bepaalde opdrachten gegeven. Dit zijn interessante oefeningen waardoor je vertrouwd raakt met de materie. De programma's voor de verschillende opdrachten bevinden zich in de directory van ROBO Pro onder Voorbeeldprogrammas\ROBO Mobile Set\l. Maar je mag natuurlijk ook je eigen opdrachten bedenken voor de modellen. Als je de volgende voorbeelden uitgetoond hebt, heb je vast en zeker nog een heleboel andere ideeën.

Basismodel

■ Het basismodel is stabiel en robuuster van constructie dan onze eerste "Eenvoudige robot". Hij heeft bovendien 2 sensoren om de weg te bepalen. Deze bestaan uit een contactschakelaar en een impulswiel. Het impulswiel is met de motoras verbonden en activeert bij elke draaicyclus van de motor vier maal een contactschakelaar.

Dit model dient als basis voor de verdere mobiele robotmodellen.



Bouw het basismodel zoals in de bouw instructies is beschreven. Ga daarbij zeer zorgvuldig te werk. Als mechanisch alles klaar is, controleer je of de motoren soepel draaien door elke motor even zonder de interface rechtstreeks op de accu aan te sluiten.

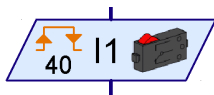


Opdracht 1 (Niveau 1):

- **Programmeer de interface zo dat het model 40 pulsen recht vooruit rijdt. Gebruik voor het meten van de impulsen de telcontactschakelaar op ingang I1.**
- **Meet de afstand die het model aflegt en bereken welke afstand er per impuls wordt afgelegd.**
- **Herhaal deze proef 3 maal en leg in de tabel vast, hoe sterk de waardes van elkaar afwijken.**

Tips:

- Schakel eerst beide motoren in (draairichting linksom).
- Om de impulsen op I1 te tellen maak je gebruik van het programma-element **Impulsteller**.
- Tel beide impulsflanken (0-1 bij het indrukken, 1-0 bij het loslaten van de contactschakelaar). Dit kun je in het venster met eigenschappen instellen onder **Impulstype**. Daarmee verhoog je de nauwkeurigheid van de wegbepaling.
- Schakel de motoren daarna weer uit en beëindig het programma.
- Je vindt het kant-en-klare programma onder [Basismodel 1.rpp](#).



**Resultaat:**

	Aantal pulsen	Afgelegde afstand	Afstand/impuls
Poging 1	40		
Poging 2	40		
Poging 3	40		

Grofweg kun je stellen dat het model per impuls ongeveer een afstand van een centimeter aflegt.

Intussen weet je ook welke draairichting je voor de afzonderlijke motoren moet instellen om het model in een bepaalde richting te laten rijden. Leg deze resultaten vast in de volgende tabel, zodat je er niet meer bij elke richtingsverandering over hoeft na te denken. Als je de bekabeling op het model exact uitvoert zoals beschreven in de bouw instructie, betekent de draairichting linksom bij elke motor dat het wiel vooruit draait. Zo zijn de motoren in alle voorbeeldprogramma's geprogrammeerd.

**Vul de tabel in:**

Rijrichting model	Draairichting M1	Draairichting M2
Vooruit	Linksom	Linksom
Achteruit		
Links		
Rechts		
Stop		

Om niet bij elke richtingsverandering twee motorsymbolen op het beeldscherm te hoeven plaatsen, kun je voor elke rijrichting een subprogramma maken dat deze taak op zich neemt. Dit vereenvoudigt het programmeren enorm. Het aanmaken van subprogramma's wordt beschreven in hoofdstuk 4 van het handboek bij de software ROBO Pro. Zodra je dit hoofdstuk hebt gelezen, kun je je aan de volgende opdracht wagen. Schakel nu in ROBO Pro over op **Niveau 2**.

**Opdracht 2 (Niveau 2):**

- **Maak een subprogramma aan voor elke rijrichting.**
- **Programmeer het model zo, dat het een vierkant met een meter lange zijden rijdt.**
- **Hoe groot is de herhalingsnauwkeurigheid?**



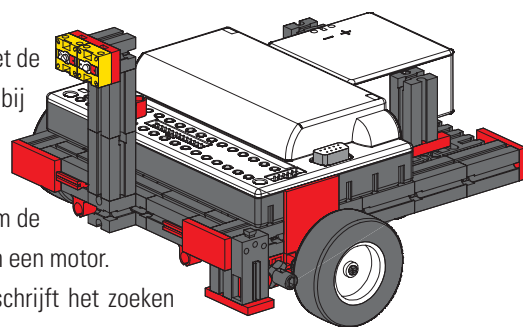
Tips:

- Maak eerst een subprogramma **Vooruit**. De andere subprogramma's kun je dan genereren door dit subprogramma te kopiëren. Daarin moet je dan nog wel de draairichtingen van de motoren aanpassen.
- Gebruik een lagere snelheid om naar links en rechts te draaien. Dat verhoogt de nauwkeurigheid.
- Om de impulsen te tellen, maak je weer gebruik van het element **Impulsteller** en de contactschakelaar bij ingang I1.
- Laad het programma eerst in het RAM om het uit te proberen en laat het daarin staan tot je hebt uitgezocht hoeveel impulsen je nodig hebt om een draaiing van 90° uit te voeren. Ten eerste gaat het laden in het RAM sneller dan het laden in het FLASH-geheugen en ten tweede heeft het FLASH-geheugen slechts een "beperkte" levensduur van ca. 100.000 downloads.
- Het kant-en-klare programma heet Basismodel 2.rpp.

**De Lichtzoeker**

■ Je hebt het basismodel nu uitgebreid onderzocht. Nu moet de robot leren om op omgevingsignalen te reageren. Evenals bij de mot in onze vergelijking in het eerste hoofdstuk moet hij een lichtbron herkennen en volgen. De bouwdoos bevat 2 fototransistors, die we als lichtdetector gaan toepassen. Om de lichtbron te kunnen volgen wordt elke sensor gekoppeld aan een motor.

Het programma bestaat uit twee delen. Het ene deel beschrijft het zoeken naar een lichtbron en in het andere deel wordt het volgen resp. het aansturen van de lichtbron gerealiseerd. Daarvoor worden weer subprogramma's gebruikt. Na het inschakelen wordt het subprogramma 'Licht zoeken' geactiveerd. Dit subprogramma wordt pas afgesloten nadat er een lichtbron is gevonden. Het hoofdprogramma probeert de robot op basis van de lichtbron te sturen. Telkens als de richting van de robot sterk van de ideale lijn afwijkt, wordt een van de twee sensoren niet meer beschoren door de lichtbron. Als gevolg daarvan corrigeert de robot zijn rijrichting, zodat beide sensoren de lichtbron weer kunnen herkennen.



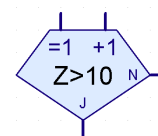
Bouw eerst het model Lichtzoeker zoals in de bouw instructies is beschreven.

**Opdracht 1 (Niveau 2):**

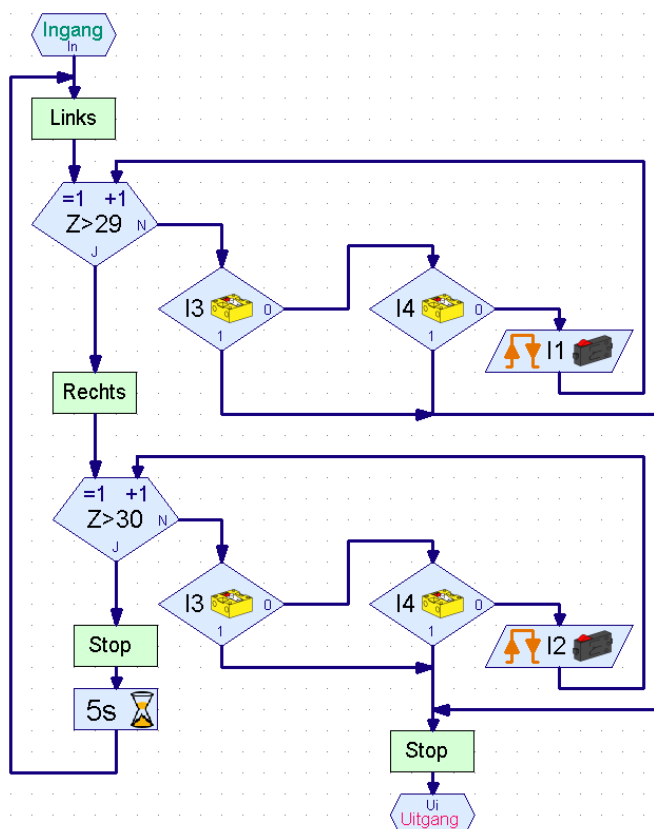
- **Programmeer eerst de functie "Licht zoeken". De robot moet zich daarbij langzaam ten minste 360° ronddraaien om te zoeken. Als de robot tijdens het zoeken een lichtbron vindt, stopt de robot. Vindt hij niets, dan draait hij nog eens 360° in de andere richting. Als hij dan nog altijd geen lichtbron vindt, moet hij 5 seconden wachten en daarna opnieuw gaan zoeken.**
- **Als het licht gevonden is, moet het model door de lichtbron worden aangestuurd. Als de lichtbron naar links of rechts beweegt, moet de robot de bewegingen van het licht volgen. Als hij het contact verliest, moet het programma opnieuw beginnen met het licht zoeken. Probeer de robot met een zaklamp te lokken en door een hindernisparcours te leiden.**

Tips:

- Pas voor de verschillende rijrichtingen de subprogramma's toe die je al voor het basismodel hebt geprogrammeerd. Zodra het programma Basismodel 2.rpp geopend is, vind je **in het venster met elementgroepen** van ROBO Pro onder **Geladen programma** het programma Basismodel 2 en daaronder de subprogramma's die zich bevinden in het programma Basismodel 2. Deze subprogramma's kun je dan eenvoudig invoegen in je nieuwe programma.
- Voor het subprogramma "Licht zoeken" gebruik je het element **Getallenlus**. (Zie voor de beschrijving van dit element het handboek van ROBO Pro).
- In de lus tussen aansluiting "N" en aansluiting "+1" vraag je de fototransistors af en tel je een impuls op de impulscontactschakelaar I1. De lus wordt zo vaak doorlopen totdat de robot licht heeft gevonden of zich 360° heeft gedraaid. Je moet gewoon even uitproberen hoe vaak hij de lus voor een volledige draai moet doorlopen. Stel de waarde "Z" in het element Getallenlus op basis daarvan in.
- Vervolgens programmeer je precies op dezelfde manier een tweede lus voor het zoeken met omgekeerde draairichting.
- Als de robot licht vindt, stopt hij en verlaat hij het subprogramma.
- Hieronder volgt het complete subprogramma voor het licht zoeken:



Getallenlus

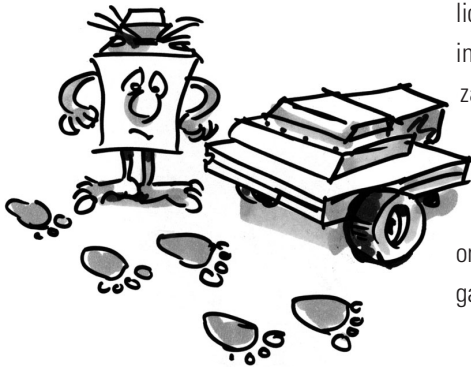


- In het hoofdprogramma vraag je weer de fototransistors af en stuur je de motoren aan op basis van welke fototransistor licht ziet:

Licht bij I3 en I4	Vooruit
Licht alleen bij I3	Bocht naar rechts
Licht alleen bij I4	Bocht naar links
Geen licht gevonden	Stop en terug naar het subprogramma Licht zoeken

- De bocht naar rechts of naar links genereer je door M1 en M2 op verschillende snelheden te laten lopen bij een gelijke draairichting. Dit leidt tot een zeer harmonische rijstijl.

- Het hoofdprogramma ziet er dan als volgt uit:
- Je vindt het kant-en-klare programma onder [Lichtzoeker.rpp](#).
- Gebruik een zaklamp als lichtbron. Probeer daarbij de lichtbundel niet te klein te maken, anders worden niet allebei de fotosensoren door de lichtbron beschenen. Besef wel dat in zeer fel verlichte ruimtes je zaklamp kan worden overstraald door andere lichtbronnen, zoals zonlicht door een groot raam. De robot rijdt dan onder bepaalde omstandigheden de lamp voorbij en gaat naar het felle licht toe.

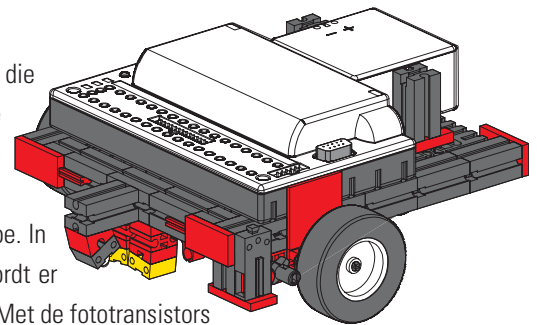
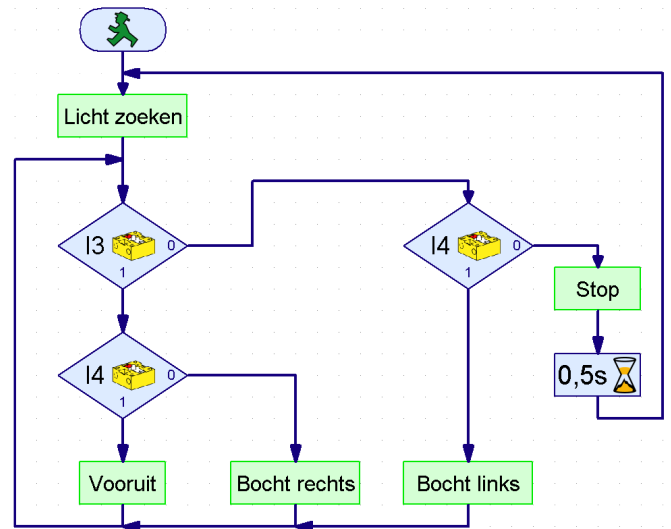


De Spoorzoeker

■ Zoeken en volgen zijn wezenlijke eigenschappen, die intelligente wezens bezitten. Met de Lichtzoeker heb je een robot gebouwd en geprogrammeerd die op directe signalen van zijn doel reageert.

Met de Spoorzoeker passen we een ander zoekprincipe toe. In plaats van doelgericht naar de lichtbron toe te rijden, wordt er nu een zwarte lijn aangebracht die de robot moet volgen. Met de fototransistors is deze opdracht betrekkelijk gemakkelijk op te lossen. Het door de zwarte lijn gereflecteerde licht wordt gemeten en de motoren worden aan de hand daarvan gecorrigeerd. Om te zorgen dat dit nauwkeurig werkt, wordt de lijn door de lamp verlicht. Let op dat de fotosensoren niet als gevolg van een ongunstige configuratie worden verblind door het verstrooide licht van de lamp. In deze samenhang heeft bundeling van het licht van de optische lens van de gloeilamp een zeer goed effect.

Bouw nu het model Spoorzoeker zoals in de bouw instructies is beschreven.



Opdracht 1:

- **Schrijf eerst een subprogramma om het spoor te zoeken. De robot moet daarbij eenmaal in de rondte draaien.**
- **Vindt hij daarbij geen spoor, dan moet hij een stukje recht vooruit rijden en daar opnieuw gaan zoeken. Om het spoor te herkennen worden de fototransistors afgevraagd.**
- **Als de robot het spoor ontdekt heeft, moet hij dit volgen.**
- **Als het spoor ten einde is of als de robot het spoor kwijtraakt, bijv. omdat de richting van het spoor te sterk verandert, moet het zoeken opnieuw beginnen.**

Tips:

- Na het inschakelen van de lamp moet er even gewacht worden (ca. een seconde) voordat de fototransistors worden afgevraagd. Anders herkent de fototransistor "donker", dat wil zeggen een spoor, waar er geen is, omdat het afvragen al begint voordat de lamp echt fel brandt.
- Als spoor maak je gebruik van ca. 20 mm brede zwarte isoleertape of je tekent met viltstift een zwart spoor van deze breedte op een wit vel papier. De bochten mogen niet te krap zijn, anders verliest de robot het spoor te vaak.
- Controleer eerst met de interfacetest of de fototransistors het spoor ook goed herkennen. Vergeet daarbij niet de lamp in te schakelen.
- Regel de lamp zo af dat op een lichte ondergrond beide fototransistors de waarde 1 opleveren, ook als de motoren M1 en M2 worden ingeschakeld. Als de accu wat zwakker is, gaat de lamp iets minder fel branden als de motoren worden ingeschakeld. Als de lamp niet goed is afgeregeld, kan het zijn dat een fototransistor "donker" aangeeft hoewel hij helemaal geen spoor heeft gevonden.
- Het spoorzoeken functioneert bijna hetzelfde als het licht zoeken. Je moet alleen het zoeken zo aanpassen, dat het model bij een niet-geslaagde zoekpoging na een volle draai in de rondte een stuk recht vooruit rijdt, voordat hij verder zoekt.
- Let erop dat het model bij het volgen van het spoor recht vooruit moet rijden, als beide fototransistors de waarde "donker" (=0) opleveren.
- Je vindt het kant-en-klare programma onder [Spoorzoeker.rpp](#).

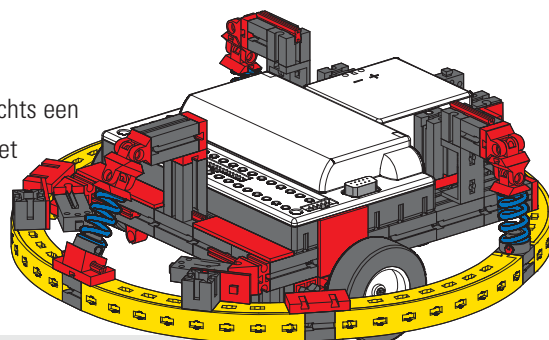
**Opdracht 2:**

- **Maak een spoor met verschillende krappe bochten. Welke radius kan het model nog net halen?**
- **Experimenteer bij het corrigeren van het spoor met verschillende snelheden voor M1 en M2. Welke combinatie levert het beste resultaat op?**
- **Zet nu een rond traject uit als spoor. Probeer de snelheden zo te optimaliseren dat de robot een zo kort mogelijke rondetijd haalt. Deze opdracht is prima geschikt voor een wedstrijd tussen meer robots.**

■ Alle tot dusverre gebouwde robots kunnen een bepaald traject afleggen en een lichtbron of een spoor volgen. Maar wat gebeurt er als ze een hindernis tegenkomen? Waarschijnlijk zullen ze de hindernis opzij duwen of blijft de robot er tegenaan stoten totdat de accu leeg is. Het zou natuurlijk veel intelligenter zijn als de robot de hindernis herkende en op de juiste wijze zou uitwijken. Daarvoor wordt de robot voorzien van een bewegende stootstang rondom met drie contactschakelaars. Met deze stootstang kan hij bepalen of er zich links of rechts van hem of achter hem een hindernis bevindt. Hoe hij daarop moet reageren, is dan alleen nog een kwestie van programmeren.

Bouw nu eerst het model "Robot met hindernisherkenning". Voor wegbepaling is slechts een contactschakelaar (I1) nodig. Daarom verwijderen we contactschakelaar I2 van het basismodel en gaan we die gebruiken voor de hindernisherkenning.

Robot met hindernisherkenning





Opdracht 1 (Niveau 2):

- De robot moet eerst recht vooruit rijden. Als hij links tegen een hindernis (E4) stoot, moet hij een stukje terug en daarna naar rechts uitwijken.
- Als hij rechts tegen een hindernis (E3) stoot, moet hij een stukje terug en daarna naar links uitwijken.

Tips:

- De hindernisherkenning bij het achteruit rijden wordt momenteel nog niet toegepast.
- In het hoofdprogramma worden de contactschakelaars afgevraagd. Afhankelijk van de contactschakelaar die geactiveerd wordt, wijkt het model naar links of naar rechts uit. Dit gebeurt telkens in een subprogramma.
- Het impulsgetal voor een draai naar rechts moet afwijken van het impulsgetal voor een draai naar links (bijv. 3 impulsen naar rechts, 5 impulsen naar links). Anders kan het zijn dat het model in een hoek terecht komt en daar niet meer uitkomt, omdat hij altijd evenveel naar links als naar rechts draait.
- Het kant-en-klare programma heet Hindernis1.rpp.

Twee dingen kan de hindernisherkenner nog niet. Tijdens het achteruit rijden herkent hij nog geen hindernissen. Hij merkt ook nog niet of er zich een hindernis recht voor hem bevindt. Hij zou ze echter wel allebei kunnen herkennen. Als tijdens het achteruit rijden I5 wordt ingedrukt, is er een hindernis achter het model. Als tijdens het vooruit rijden I3 en I4 tegelijk worden ingedrukt, bevindt er zich een hindernis recht voor het model. In dit geval zou de robot zich direct 90° kunnen draaien. In totaal hebben we nu dus de volgende mogelijkheden waarop de robot moet reageren:

Hindernis	Contact schakelaar	Reactie
Rechts	Alleen I3	Naar links uitwijken (ca. 30° draaien)
Links	Alleen I4	Naar rechts uitwijken (ca. 45° draaien)
Voor	I3 en I4	Naar links uitwijken (ca. 90° draaien)
Achter	I5	Wordt alleen afgevraagd tijdens het achteruit rijden. Stoppen en daarna zoals gepland verder uitwijken

Om deze opdracht elegant op te lossen, kunnen enkele nieuwe programma-elementen zoals bijv. bedieners (ook wel operatoren genoemd, bijv. EN, OF) uit niveau 3 van ROBO Pro zeer goed van pas komen. In niveau 3 bestaat tevens de mogelijkheid via oranje pijlen gegevens uit te wisselen tussen verschillende elementen. Schakel daarom in de software over op dit niveau. Daarna kun je het beste eerst even hoofdstuk 5 van het ROBO Pro handboek grondig doorlezen. Je bent dan klaar voor de volgende opdracht.



Opdracht 2 (Niveau 3):

- Stel het hindernisprogramma zo op dat het model reageert zoals in bovenstaande tabel is aangegeven.
- Maak daarbij gebruik van de mogelijkheden uit ROBO Pro Niveau 3.

Tips:

- Met behulp van bedieners (of operatoren) worden via een subprogramma "Afvragen Hindernis" de verschillende mogelijke combinaties van contactschakelaars afgevraagd. Voor elke mogelijkheid heeft het subprogramma een eigen uitgang.

Gegevensinvoer CR = contactschakelaar rechts

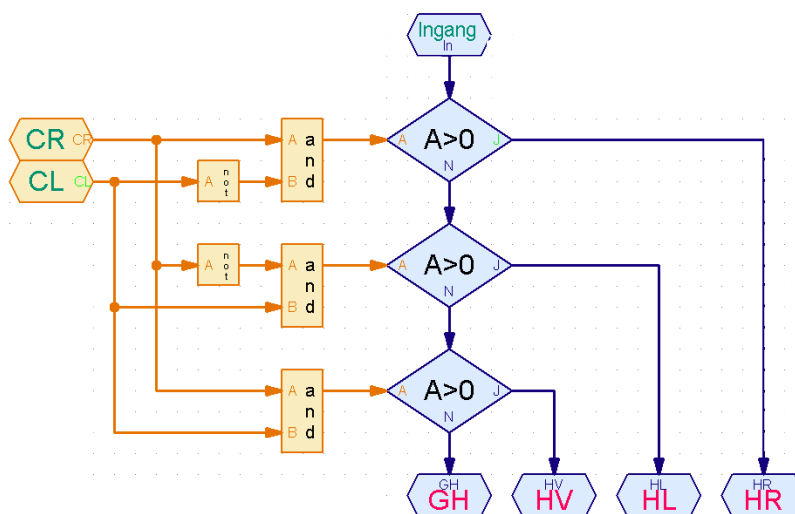
Gegevensinvoer CL = contactschakelaar links

Uitgang GH = geen hindernis

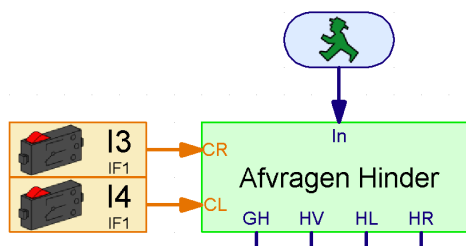
Uitgang HV = hindernis voor

Uitgang HL = hindernis links

Uitgang HR = hindernis rechts

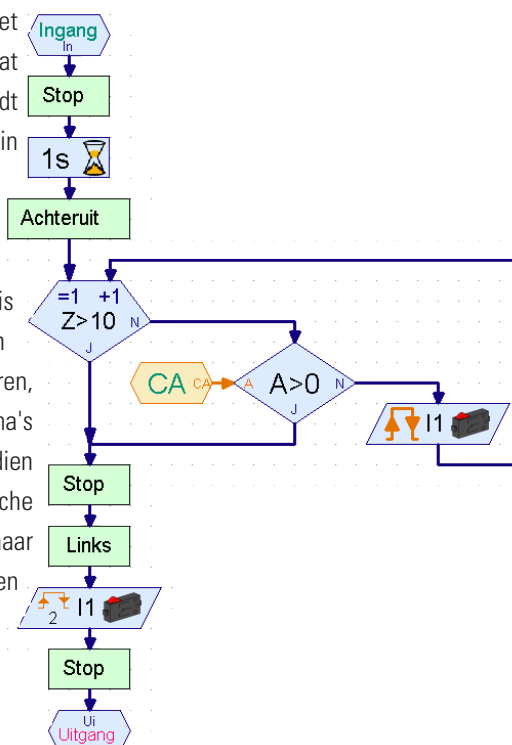


- Om te zorgen dat het onmiddellijk duidelijk is welke contactschakelaar wordt afgevraagd, plaats je de oranje contactschakelaarelementen in het hoofdprogramma en verbind je ze via gegevensinvoer met het subprogramma.



- In de verschillende subprogramma's voor het uitwijken wordt tijdens het achteruit rijden I5 afgevraagd. Het model rijdt dan zolang achteruit totdat het ingestelde impulsgetal wordt bereikt of I5 wordt ingedrukt. I5 wordt weer in het hoofdprogramma gezet, zodat onmiddellijk zichtbaar is in welk subprogramma hij wordt afgevraagd.
- Je vindt het complete programma onder [Hindernis2.rpp](#).

Een voordeel van de in deze opdracht toegepaste programmeertechniek is dat je direct in het hoofdprogramma kunt zien welke contactschakelaar in welk subprogramma wordt afgevraagd. Als je de ingang wilt veranderen, hoef je dat slechts op één plek te doen en hoef je niet in alle subprogramma's te zoeken waar de contactschakelaar overal verstopt kan zitten. Bovendien kun je met de operatoren op zeer aanschouwelijke wijze logische koppelingen maken. In principe kan dat ook met Vertakkingselementen, maar dit wordt al snel zeer onoverzichtelijk als er meer situaties moeten worden afgevraagd.



Lichtzoeker met hindernisherkenning



■ We hebben nog lang niet alle mogelijkheden uitgeprobeerd die de ROBO Mobile Set biedt. Daarom gaan we nu de twee functies licht zoeken en hindernis herkennen met elkaar combineren. Uit wetenschappelijk opzicht is de robot dan van twee gedragingen voorzien. Omdat echter niet allebei de gedragspatronen tegelijk actief kunnen zijn, krijgen ze verschillende prioriteiten. De robot is normaal gesproken op zoek naar licht. Herkent hij een hindernis, dus een gevaar voor zichzelf, dan wordt de gedraging "hindernis vermijden" actief. Als alles in het groene bereik (veilig) blijft, kan de robot verder zoeken naar de lichtbron.

Als professionele softwareontwikkelaars een dermate gecompliceerde opdracht aanpakken, programmeren ze er niet zomaar op los, maar maken ze gebruik van een bepaalde strategie voor het ontwikkelen van het programma. Een van deze methodes noemt men "Top-Down ontwerpen". Bij deze procedure wordt het totale systeem van bovenaf gedefinieerd, zonder al direct in te gaan op alle details. Van deze methode maken wij voor dit probleem ook gebruik.

Opdracht 1 (Niveau 3):

Leer de robot de volgende gedragingen aan:

- Zoek een lichtbron.
- Volg deze zodra je hem gevonden hebt.
- Als er onderweg een hindernis opduikt, moet je ervoor uitwijken.
- Zoek daarna opnieuw een lichtbron.

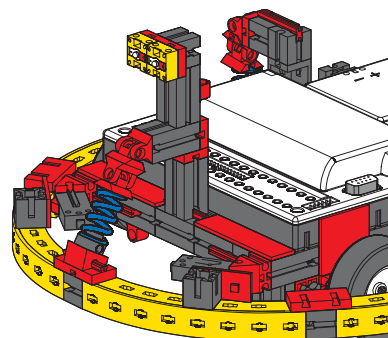


Maak voor de oplossing gebruik van de programma-elementen uit ROBO Pro Niveau 3. Los de opdracht "van bovenaf" op volgens de Top-Down-aanpak.

Tips:

Verdeel de opdracht eerst in drie onderdelen:

- Afvragen of de robot een lichtbron ziet (subprogramma "Licht")
- Afvragen of hij tegen een hindernis botst (subprogramma "Hindernis")
- Afhankelijk van deze resultaten laat je de robot weten, wat hij moet doen (subprogramma "Rijden")



Voor de subprogramma's "Licht" en "Hindernis" bedenk je nu welke verschillende situaties de robot kan waarnemen. Je wijst een getalswaarde toe aan elke situatie en deze sla je met behulp van een Bevelsegment op in een variabele. Uit elke situatie volgt dan een reactie die wordt uitgevoerd in het subprogramma "Rijden".

Subprogramma Licht:

Nr.	Situatie	Toestand van de sensoren	Reactie
0	Geen lichtbron beschikbaar	I6=0; I7=0	Licht zoeken
1	Lichtbron recht voor robot	I6=1; I7=1	Recht vooruit rijden
2	Lichtbron links van de robot	I7=1	Bocht naar links maken
3	Lichtbron rechts van de robot	I6=1	Bocht naar rechts maken

Subprogramma Hindernis:

Nr.	Situatie	Toestand van de sensoren	Reactie
4	Hindernis direct voor de robot	I3=1; I4=1	90° uitwijken
5	Hindernis rechts van de robot	I3=1	Naar links uitwijken
6	Hindernis links van de robot	I4=1	Naar rechts uitwijken

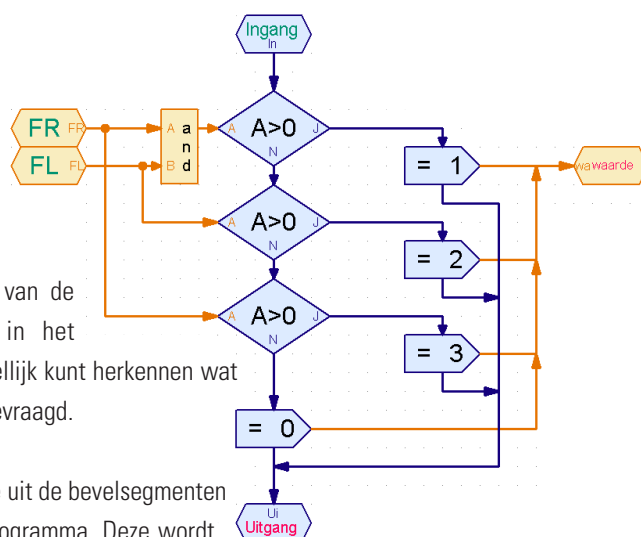
Nu moet je deze inzichten nog met programma-elementen reproduceren in ROBO Pro.

Subprogramma Licht:

FR=Fototransistor rechts
FL=Fototransistor links

De elementen voor het afvragen van de fototransistors plaats je weer in het hoofdprogramma, zodat je onmiddellijk kunt herkennen wat er in het subprogramma wordt afgevraagd.

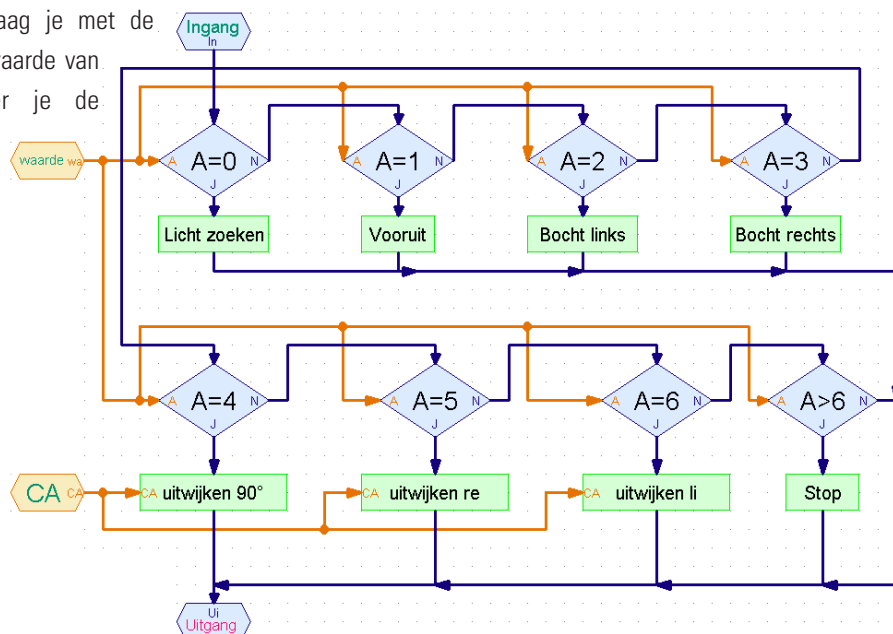
Ook de variabele waarin de waarde uit de bevelsegmenten is opgeslagen komt in het hoofdprogramma. Deze wordt in meer subprogramma's toegepast. Je verbindt deze via een gegevensuitgang met het subprogramma.



Het subprogramma "Hindernis" maak je volgens hetzelfde principe als het subprogramma "Licht".

In het subprogramma "Rijden" vraag je met de Vertakkings-elementen de actuele waarde van de variabelen af en programmeer je de bijbehorende reactie:

CA=contactschakelaar achter



Als laatste detail moet je nu nog de subprogramma's maken die in dit subprogramma worden toegepast. Maar wacht eens even! Die hebben we bijna allemaal al. Zo kun je het subprogramma "Licht zoeken" bijvoorbeeld al kopiëren uit het programma voor het model Lichtzoeker. Als je niet meer weet hoe dit moet, moet je hoofdstuk 4 van het ROBO Pro handboek even doorlezen.

Let op:

Bij het model Lichtzoeker waren de fototransistors aangesloten op ingang I3 en I4. Nu zijn ze echter aangesloten op I6 en I7. Bovendien werd daar voor het tellen van de impulsen bij het draaien naar links contactschakelaar I1 afgevraagd en bij het draaien naar rechts I2. Nu is alleen nog maar I1 beschikbaar voor het tellen van de impulsen, maar dat werkt even goed. Je moet het subprogramma "Licht zoeken" na het kopiëren dus aanpassen. Omdat het afvragen van de contactschakelaar in het subprogramma verstopt zit, kun je er gemakkelijk overheen kijken. Dit gebeurt niet meer als je de ingangen in het hoofdprogramma plaatst en deze via gegevensinvoeren met het subprogramma verbindt. Bij de Lichtzoeker kende je deze mogelijkheid echter nog niet.

Ook de subprogramma's voor het uitwijken zijn al beschikbaar, namelijk bij het model voor hindernisherkenning. Hier is de contactschakelaar I5, die extra wordt afgevraagd bij het achteruit rijden, al naar buiten gebracht.

Je vindt het kant-en-klare programma onder [Hindernis-Licht.rpp](#).

Het hoofdprogramma ziet er op het eerste gezicht zeer overzichtelijk en eenvoudig uit. Toch zitten er heel wat hoofdbreken in de subprogramma's. Maar met behulp van de stapsgewijze procedure volgens de Top-Down-methode kun je dus ook een dergelijk complex probleem oplossen.

Mocht je trouwens iemand kennen die ook een bouwdoos voor de ROBO Mobile Set heeft, dan kunnen jullie nog verder experimenteren met deze robots. Monteer dan een lichtbron aan beide robots. Ze gaan elkaar dan zoeken.



Robot met randherkenning

■ Nu je in het vorige voorbeeld hebt gezien hoe je te werk moet gaan om een complexer probleem te programmeren, kun je de mobiele robot een zeer belangrijke gedraging bijbrengen. Hij moet namelijk leren niet van de tafel te vallen. Als de robot tegen een hindernis rijdt, dan maakt dat voor hem in de meeste gevallen niets uit. Als hij echter van een tafel valt en één meter lager op de grond valt, dan kan hij behoorlijk beschadigd raken, ondanks het feit dat de componenten van fischertechnik zeer robuust zijn. Daarom krijgt de robot sensoren zodat hij randen kan herkennen. Deze randdetectoren bestaan telkens uit een contactschakelaar, die geactiveerd wordt door een draaibaar gelagerd wiel. Dit wiel kan ook omhoog en omlaag bewegen. Zodra het wiel voorbij de rand van de tafel komt, valt het naar beneden. De contactschakelaar wordt nu niet meer geactiveerd en het programma weet dat het model zich bij een afgrond bevindt en reageert daar op gepaste wijze op. De robot heeft in totaal 4 randdetectoren, zodat hij zowel tijdens het vooruit- als het achteruitrijden aan beide kanten naar afgronden kan zoeken. Als gevolg daarvan heeft dit model geen impulscontactschakelaar voor trajectmeting. Het afgelegde traject wordt aangestuurd door de inschakelduur van de motoren. Bouw eerst het model zoals in de bouw instructies is beschreven.

Controleer zorgvuldig of de randdetectoren goed reageren:

- als het model bij de tafelrand komt en de contactschakelaar weer precies wordt ingedrukt;
- op het moment dat het wiel weer op de tafel staat.

Misschien moet je de ene of andere contactschakelaar nog iets omlaag of omhoog verstellen.



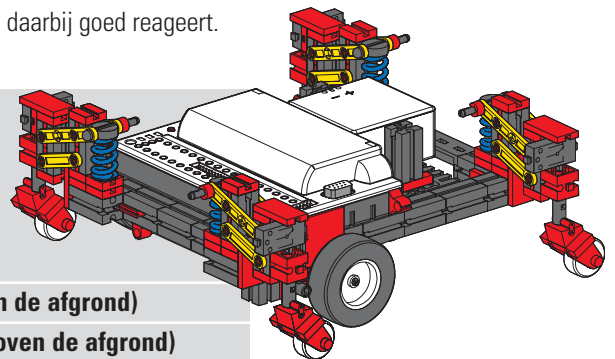
Opdracht 1 (Niveau 3):

- **Bedenk eerst hoe de robot moet reageren als hij bij een afgrond komt.**
- **Als je hier iets dieper over nadenkt, zal het je opvallen dat er erg veel combinaties mogelijk zijn van sensoren die zich boven de afgrond kunnen bevinden. Er kan slechts een van de 4 detectoren worden geactiveerd, 2 of 3 verschillende tegelijk of alle 4 de sensoren.**
- **Hoe moet de robot op elke verschillende situatie reageren?**

Tips:

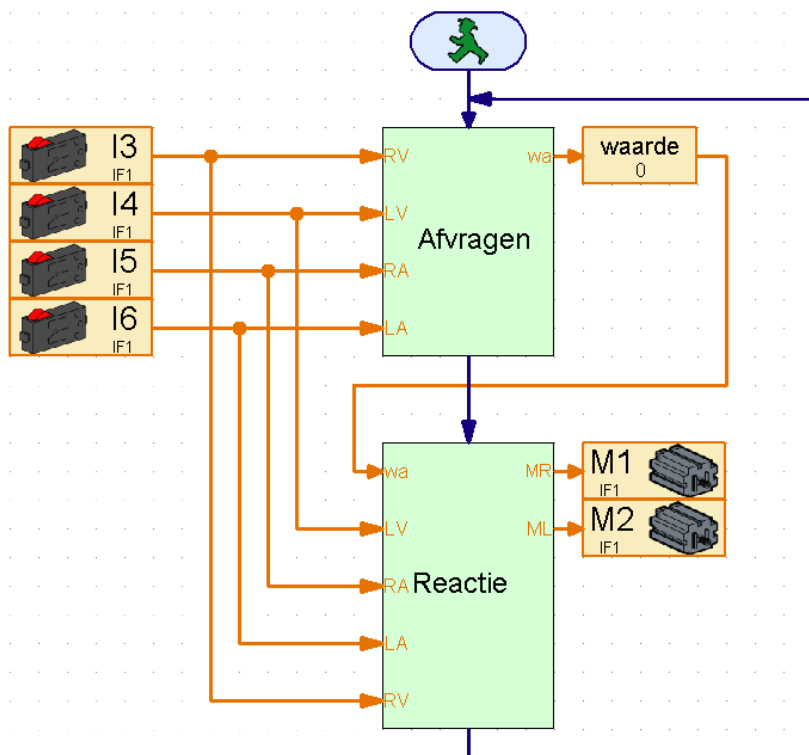
De oplossing vind je in de onderstaande tabel. De sensoren die zich boven de afgrond bevinden (contactschakelaar=0) zijn aangekruist. Elke combinatie krijgt een nummer. In het programma dat je gaat maken, krijgt elke mogelijkheid het bijbehorende getal. Aan de hand van dit getal reageert de robot op de actuele situatie. Maar hierover later meer. Je hoeft je in eerste instantie alleen maar te bedenken hoe de robot moet staan om de desbetreffende combinatie te laten optreden en of hij daarbij goed reageert.

Nr.	Rechts-voor (I3)	Links-voor (I4)	Rechts-achter (I5)	Links-achter (I6)	Reactie
0					Vooruit (geen sensor boven de afgrond)
1	●	●	●	●	Stop (alle 4 de sensoren boven de afgrond)
2	●	●	●		Een stukje naar rechts draaien
3	●	●		●	Een stukje naar links draaien
4	●		●	●	Een stukje naar links draaien
5		●	●	●	Een stukje naar rechts draaien
6	●	●			Eerst terug en daarna naar rechts draaien
7	●		●		Een stukje naar links draaien
8	●			●	Een stukje naar links draaien
9		●	●		Een stukje naar rechts draaien
10		●		●	Een stukje naar rechts draaien
11			●	●	Een stukje vooruit rijden
12	●				Eerst terug en daarna naar links draaien
13		●			Eerst terug en daarna naar rechts draaien
14			●		Een stukje vooruit rijden
15				●	Een stukje vooruit rijden



Dat is best wel pittig, hè? Maar wees niet bang, voor dit model is er een kant-en-klaar programma, dat gebruikmaakt van alle voorkeuren van ROBO Pro. Dit heet Randen.rpp.

De belangrijkste elementen bevinden zich in het hoofdprogramma, zodat je de totale procedure kunt begrijpen. Het complexe afvragen van de contactschakelaars en de aansturing van de motoren is in subprogramma's verborgen. Hier eerst het hoofdprogramma:



Het programma begint met het afvragen van de 4 contactschakelaars. Helemaal links zie je welke contactschakelaars worden afgevraagd. Deze zijn via gegevensinvoer met het subprogramma verbonden. Het subprogramma "Afvragen" bepaalt welke contactschakelaars zijn ingedrukt en levert dan de in de tabel beschreven waarde op. Deze waarde wordt toegewezen aan de gelijknamige variabele, die je weer kunt herkennen in het hoofdprogramma. De waarde van de variabelen wordt doorgegeven aan het subprogramma "Reactie", dat vervolgens onafhankelijk van deze waarde de beide motoren aanstuurt. In het subprogramma "Reactie" worden ook de randsensoren nog ingelezen, omdat de randsensoren ook afgevraagd worden, terwijl het model uitwijkt.

Je zou de contactschakelaartoewijzing op de interface evenals de motoruitgangen nu kunnen veranderen zonder dat je alle subprogramma's hoeft door te nemen om te

controleren waar er nog een invoerlement of een motorsymbool verstopt zit. Elke ingang en elke uitgang komt slechts eenmaal voor.

Deze programmeertechniek kun je vooral toepassen in situaties waar een subprogramma voor veel verschillende modellen moet worden toegepast en je vooraf nog niet exact weet, welke in- en uitgangen op de interface daarvoor moeten worden gebruikt.

Als je nu nieuwsgierig bent geworden, moet je maar eens in de subprogramma's kijken om te zien of je ze begrijpt. Het programmeerprincipe is hetzelfde als bij het model "Lichtzoeker met hindernisherkenning".



Opdracht 2 (Niveau 3):

Laad het programma in de interface en laat het model op een tafel rijden.

- Reageert het model altijd correct?
- Moet het zich bij bepaalde combinaties van contactschakelaars anders gaan gedragen?
- Optimaliseer het programma eventueel als dat nodig is.

■ Nadat we ons uitvoerig hebben bezig gehouden met de rijdende robot, stappen we nu over op een andere manier van voortbewegen die we kunnen gebruiken voor mobiele robots: lopen.

De manier van lopen van insecten is prima geschikt als voorbeeld voor de aandrijving van "machinale zespotigen". Bij wat wij zullen aanduiden als het "lopen op drie voeten" worden steeds drie van de zes poten tegelijk opgetild van de grond: de voorste en achterste poot aan de ene kant, samen met de middelste poot aan de andere kant:

Lopen op drie voeten

De poten die op de grond blijven staan (met zwart weergegeven) vormen een stabiele driepoot, zodat het model altijd stevig staat en niet omvalt tijdens het lopen.



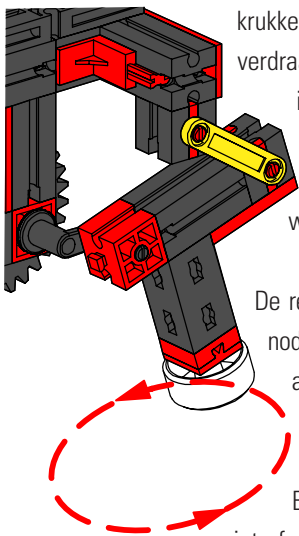
De poten van de lopende robot van fischertechnik zijn zo geconstrueerd, dat zij een zogenaamde vierstangsaandrijving vormen. De constructie van de hier toegepaste aandrijving is een soort van schaarblokbeweging. Aangedreven door een kruk voeren de bewegend gelagerde elementen van het drijfwerk schommelbewegingen uit. De afstanden tussen de afzonderlijke scharnierpunten en de positie van het voetpunt (het onderste uiteinde van de poot) zijn zo geselecteerd dat het voetpunt een elliptische beweging beschrijft, als de aandrijfkruk draait. Daardoor ontstaat een beweging, die lijkt op een het uitvoeren van een stap tijdens het lopen.

De 6 krukken die de poten aandrijven, moeten precies worden afgesteld zoals aangegeven in de bouw instructies. De drie poten die tegelijkertijd op de vloer staan, hebben dezelfde krukstand. De krukken van de 3 poten die op dat moment in de lucht steken, zijn ten opzichte daarvan 180° verdraaid. De juiste stand van de krukken ten opzichte van elkaar garandeert dat het model in de juiste volgorde van stappen op drie voeten kan lopen.

De naafmoeren, waarmee de tandwielen op de assen worden vastgezet, moeten goed worden vastgedraaid, zodat de afstelling van de krukken tijdens het lopen niet verandert.

De rechter- en linkerzijde van het model worden elk door een eigen motor aangedreven (is nodig om bochten te kunnen nemen). Daarom moet je ervoor zorgen dat de middelste poot aan de ene zijde altijd in dezelfde stand staat als de twee buitenste poten aan de andere zijde. Deze synchronisatie gebeurt door de software via contactschakelaar I1 en I2.

Bouw eerst het model zoals in de bouw instructies is beschreven. Controleer met de interfacetest of alle contactschakelaars en motoren correct zijn aangesloten. Draairichting van de motoren: draairichting linksom = vooruit.



Opdracht 1 (Niveau 1):

Leer de robot lopen.

- Programmeer het model zo dat het op drie voeten recht vooruit loopt.
- Maak gebruik van de contactschakelaars I1 en I2 om de linker- en rechterpoot met elkaar te synchroniseren.
- Let er daarbij op dat de twee buitenste poten aan de ene zijde en de middelste poot aan de andere zijde altijd in dezelfde stand moeten staan.

Tips:

- Breng eerst de poten aan de linker- en rechterzijde in hun uitgangspositie. Schakel daarvoor beide motoren in (draairichting linksom).
- De procedure mag pas verdergaan als contactschakelaar I1 en I2 allebei niet zijn ingedrukt (dit moet worden afgevraagd zodra het model de tweede stap moet nemen).
- Laat de motoren lopen totdat de desbetreffende contactschakelaar (I1 voor M1, I2 voor M2) weer is ingedrukt. Daarbij is het belangrijk dat het model pas begint met de volgende stap als beide contactschakelaars zijn ingedrukt. Dan staan de poten namelijk in de juiste stand ten opzichte van elkaar. Een voorwaarde daarvoor is echter ook dat de krukken die de poten aandrijven correct zijn afgesteld, zoals in de bouw instructies is beschreven.
- Nu kan de cyclus weer vooraan beginnen en voert de robot zijn tweede stap uit. Het model blijft nu vooruit lopen totdat jij het programma stopt.
- Je vindt het kant-en-klare programma onder Lopende robot1.rpp.

Op dezelfde manier als bij het rijdende basismodel kun je het model nu naar links, rechts of achteruit laten lopen door de motordraairichtingen te veranderen. Voor het tellen van de stappen kun je I1 of I2 gebruiken.



Opdracht 2 (Niveau 2):

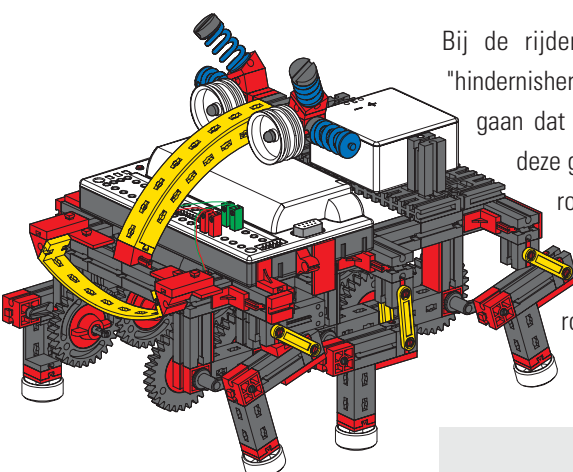
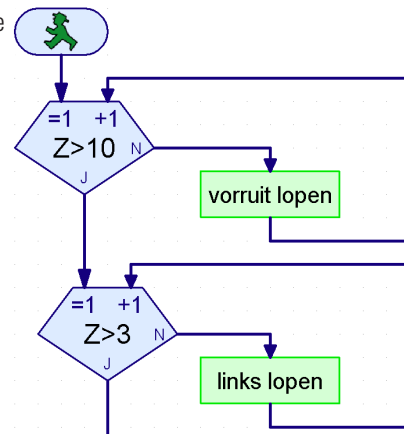
- Programmeer je model zo, dat het 10 stappen vooruit, 3 stappen naar links, 3 stappen naar rechts en weer 10 stappen achteruit loopt.
- Maak voor elke richting een afzonderlijk subprogramma aan.
- Maak voor het tellen van de stappen gebruik van het element Getallenlus.

Tips:

- Kopieer gewoon het programma Laufroboter1.rpp naar een subprogramma.
- Kopieer dit subprogramma zo vaak als nodig is voor de verschillende looprichtingen. Verander de draairichting van de motoren in elk subprogramma zodat het model zich in de gewenste richting verplaatst.
- Gebruik het element Getallenlus om het aantal stappen voor elke draairichting te tellen. Het model voert een stap uit telkens als er een subprogramma wordt doorlopen. Als het programma de lus met het subprogramma 10 maal doorloopt, dan voert het model 10 stappen uit.

Op deze manier kun je je lopende robot een willekeurige reeks stappen bijbrengen (Lopende robot 2.rpp).

Bij de rijdende robots hebben we het thema "hindernisherkenning" al uitvoerig behandeld. We gaan dat hier niet nog eens herhalen. Probeer deze gedraging echter eens door de lopende robot te laten uitvoeren. De sensoren daarvoor vind je in de bouwdoos. Bij het programmeren kun je de rijdende robot als voorbeeld nemen. Veel succes!



■ De ROBO Interface biedt nog veel meer functionaliteit dan we tot nu toe hebben laten zien met de mobiele robots. Daarvoor heb je echter extra componenten nodig die niet zijn meegeleverd bij deze bouwdoos. Omdat deze echter zeer interessant zijn voor het maken van robots, willen we een aantal van die componenten hier kort presenteren.

■ De ROBO Interface bevat een infrarood ontvangerdiode voor de handzender uit de IR Control Set art.-nr. 30344. In de ROBO Pro software kun je daarmee de toetsen van de handzender als digitale ingangen afvragen en zo bijv. de motoren in- en uitschakelen.

Als programmavoorbeeld hebben wij een afstandsbesturing voor de looprobot geprogrammeerd. Met de 4 ovale pijltjestoetsen op de afstandsbediening kun je het model vooruit, achteruit, naar links en naar rechts sturen. Vooraf moet je echter wel het programma Lopende robot-IR.rpp op de interface laden.

Een ander geniaal programma met betrekking tot de afstandsbediening is het programma Mobile-Teach-IR.rpp. Met dit Teach-In-programma kun je een van de rijdende robots, bijv. de Eenvoudige robot of het Basismodel, op afstand besturen. Het model registreert daarbij het afgelegde traject en kan dit daarna willekeurig vaak herhalen. Het opgeslagen traject wordt echter gewist als het programma gestopt wordt.

Hiermee wordt een programma mogelijk gemaakt via het programma-element "Lijst" in ROBO Pro. In dit element kun je een groot aantal waarden opslaan en weer uitlezen (zie ook het handboek van ROBO Pro). Het programma op zich is behoorlijk complex, maar de toepassing ervan is zeer eenvoudig:

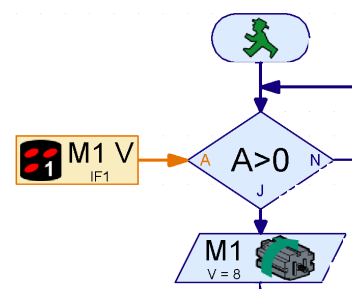
1. Laad het programma Mobile-Teach-In.rpp in het FLASH-geheugen van de ROBO Interface en start het.
2. Druk op de afstandsbediening op de toets **M1** ▶ / ▶▶. De "Leerprocedure" wordt gestart.
3. Stuur het model met de ovale pijltjestoetsen in de gewenste richting.
4. Druk op de toets **M2** ▶ / ▶▶. Het afgelegde traject wordt opgeslagen.
5. Druk op de toets **M3** ▶ / ▶▶. Het opgeslagen traject wordt afgelegd.

Met dergelijke toepassingen wordt het programmeren van robots kinderspel! Realiseer je wel dat het opgeslagen traject gewist wordt, zodra je het programma met de toets Prog. op de interface stopt.

■ De ROBO RF Data Link art.-nr. 93295 vervangt de interfacekabel tussen de pc en de interface door draadloze datatransmissie. Dat is heel prettig. Ten eerste hoeft je dan niet telkens als je een programma op de interface laadt de kabel aan te sluiten en weer los te koppelen. Ten tweede kun je programma's draadloos in de online-modus laten werken. Je kunt de fouten dan veel gemakkelijker vinden dan in de download-modus. En tenslotte kun je de mobiele robots in de online-modus via een bedieningsveld in ROBO Pro op dezelfde manier als met de IR-afstandsbediening online via het beeldscherm besturen. In tegenstelling tot de afstandsbediening kun je op het beeldscherm bovendien ook nog gegevens bekijken die de interface verstrekt, zoals waarden van variabelen of analoge ingangen, de voedingsspanning die de accu levert en de snelheid van de motoren.

Uitbreidingsmogelijkheden

Infrarood handzender



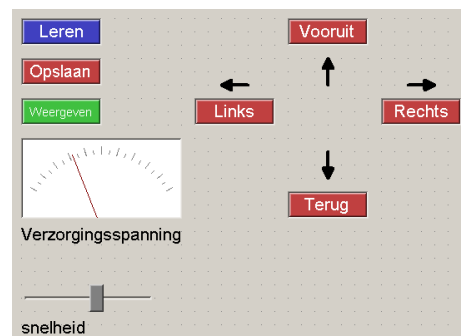
ROBO RF Data Link



Als voorbeeld hebben we het Teach-In-programma gemodificeerd en sturen we de rijdende robot via een bedieningsveld aan. Dit programma heet Mobile-Teach-RF.rpp. Je kunt het natuurlijk ook met de interfacekabel proberen. Dat is echter vrij lastig. De actieradius van het model is dan beperkt, de kabel draait zich op en de robot kan niet goed meer draaien. De kans is zeer groot dat je daarna direct op pad gaat om de RF Data Link aan te schaffen.

Laad het programma Mobile-Teach-RF.rpp.

Schakel in de Functie balk van het hoofdprogramma over naar Bedieningsveld. Daarna start je het programma in de Online-modus. Nu kun je het model met de knoppen in het bedieningsveld besturen en programmeren.



1. Druk op de knop "Leren". De "Leerprocedure" wordt gestart.
2. Stuur het model met de pijltjesknoppen in de gewenste richting.
3. Druk op de knop "Opslaan". Het afgelegde traject wordt opgeslagen.
4. Druk op de knop "Weergeven". Het opgeslagen traject wordt afgelegd.

Ook hier raakt het opgeslagen traject verloren, als het programma wordt beëindigd.

Meer details over het maken van bedieningsvelden vind je in het handboek bij ROBO Pro.

ROBO I/O-Extension

■ Als je een model met zo veel sensoren en motoren bouwt dat de in- en uitgangen van de ROBO Interface niet voldoende zijn, kun je een ROBO I/O-Extension art.-nr. 93294 op de interface aansluiten. Daarmee heb je de beschikking over nog eens 8 digitale ingangen, 4 motoruitgangen en een analoge weerstandsingang. Op deze I/O-Extension kun je een tweede module aansluiten en daarna weer een derde module, die vervolgens allemaal worden aangestuurd door een ROBO Interface. In totaal heb je dan de beschikking over 16 motoruitgangen, 32 digitale ingangen, 5 analoge weerstandsingenen, 2 analoge spanningsingenen en 2 ingangen voor afstandssensoren.

Als je dan nog niet genoeg hebt, kun je via de pc ook meer interfaces in de Online-modus besturen, bijv. een op een seriële COM-interface, een op de USB-poort of 2 interfaces op de USB-poort en dan ook nog eens elk met max. 3 ROBO I/O-Extensions! Kun jij het nog helemaal volgen? Om dit allemaal goed op een rijtje te krijgen, kun je hoofdstuk 6 van het ROBO Pro handboek doornemen.

Problemen verhelpen

■ Experimenteren kan leuk zijn. Zolang alles tenminste naar wens werkt. Meestal is dat ook het geval. Maar helaas niet altijd.

Pas als een model niet werkt, weet je of je het mechanische principe goed hebt begrepen, waarna je de fout meestal snel kunt vinden.

Bij mechanische fouten kun je in ieder geval nog iets zien (foute montage) of voelen (het draait moeilijk).

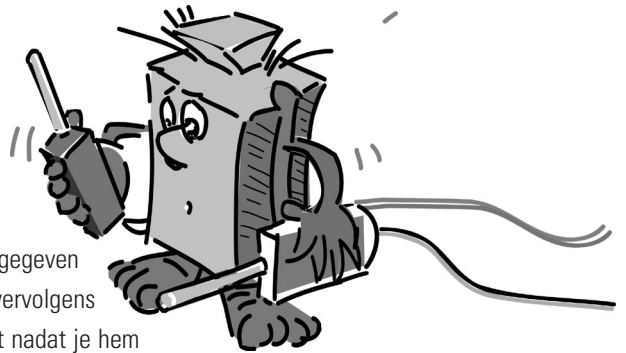
Als er elektrische problemen bijkomen dan wordt het al lastiger.

Professionals gebruiken allerlei meetinstrumenten om fouten op te zoeken, zoals een spanningsmeter of een oscillograaf. Dergelijke apparatuur hebben we niet allemaal in huis. Daarom zullen wij proberen om een fout met eenvoudige middelen te lokaliseren en te verhelpen.

Kabelmontage

Voordat we met onze experimenten beginnen, moeten we eerst een paar componenten uit de fischertechnik-bouwdoos voorbereiden. Zo moet je bijvoorbeeld de meegeleverde stekkers aan de verschillende draden bevestigen.

Daarvoor moet je de kabels eerst op maat maken. Meet daarvoor de aangegeven lengte af en maak de kabel op maat en strip het uiteinde. Elke kabel wordt vervolgens doorgemeten. Daarvoor heb je de accu en de lamp nodig. Als de lamp brandt nadat je hem aan de accu hebt aangesloten, dan is de kabel in orde. Controleer ook of de kleuren kloppen: rode stekker aan rode kabel en groene stekker aan groene kabel.



Interfacetest

Als het programma (dit geldt ook voor het meegeleverde programma) niet met ons model samenwerkt, dan starten we de interfacetest. Dit hulpprogramma stelt je in staat de in- en uitgangen apart te testen. Werken de sensoren? Draaien de motoren in de juiste richting? Bij al onze mobiele robots zijn de motoren zo aangesloten, dat bij draairichting=links het wiel of de poot vooruit beweegt. Als hier alles in orde is, dan zoeken we naar de mechanische oorzaak.

Losse contacten

Een veel voorkomende fout zijn losse contacten. Zo kunnen de aansluitstekkers los in de aansluitbussen zitten. Is dit geval dan moet je de contactveren van de stekker met een kleine schroevendraaier iets uit elkaar drukken. Wees daarbij wel voorzichtig want als je de contactveren te veel verbuigt kunnen de contacten breken of kun je de stekker moeilijk insteken.

Een andere oorzaak voor losse contacten is dat de schroefjes van de klemverbindingen in de stekker zijn losgeraakt. Draai ze voorzichtig vast! Controleer dan tegelijkertijd of geen van de dunne koperdraadjes zijn afgebroken.

Kortsluiting

Het zou kunnen gebeuren dat er kortsluiting optreedt als een kabel verkeerd wordt aangesloten. Dan werkt ook niets meer zoals het hoort. In de accu is een zekering ingebouwd, die de stroom uitschakelt als de stroom of de temperatuur te hoog wordt. Ook de uitgangen van de interface worden bij oververhitting uitgeschakeld.

Een kortsluiting kan ook optreden als het schroefje van de elektrische stekkers waarmee de kabel wordt vastgeklemd niet goed wordt vastgedraaid. Het kan dan buiten de stekker uitsteken. Als er dan twee

stekkers in twee bussen naast elkaar op de interface worden gestoken waarbij de schroefjes elkaar raken, dan ontstaat er kortsluiting. Daarom moeten de schroefjes altijd goed worden vastgedraaid en moet je de stekkers zo insteken dat de schroefjes elkaar niet kunnen raken.

Voeding

Als de apparatuur plotseling om onverklaarbare reden uitvalt, kan het zijn dat de accu bijna leeg is. De spanning daalt kort even als er een last extra wordt ingeschakeld (motor aan) waarmee een reset voor de processor wordt gegenereerd op de interface. Op de ROBO Interface gaat een rode LED branden als de spanning van de voeding te laag is. De accu moet dan worden opgeladen.

Programmeerfouten

Als er fouten optreden bij programma's die je zelf hebt geschreven en je het probleem zelf niet kunt vinden, moet je voor de zekerheid een meegeleverd programma draaien dat het doel van je eigen programma zo dicht mogelijk benadert, om elektrische of mechanische defecten te kunnen uitsluiten. In de Online-modus kun je het verloop van het programma op het beeldscherm volgen. Als het programma vanaf een bepaalde positie niet meer verdergaat, kun je daar de oorzaak zoeken, bijv. verkeerde ingang of motor geselecteerd, bij een vertakking een verkeerde waarde afgevraagd of J/N-aansluitingen verwisseld.

Als dit alles niet tot het gewenste resultaat leidt, kun je altijd nog contact opnemen met de serviceafdeling van fischertechnik (e-mail: info@fischertechnik.de).

Of bezoek ons op internet op www.fischertechnik.de. Daar vind je een forum, chatmogelijkheden, een ruilmarkt, een galerie en kun je gratis lid worden van de Fischertechnik Fanclub.

We wensen je nog heel veel uren plezier en verbazing met de ROBO-Mobile-Set.

