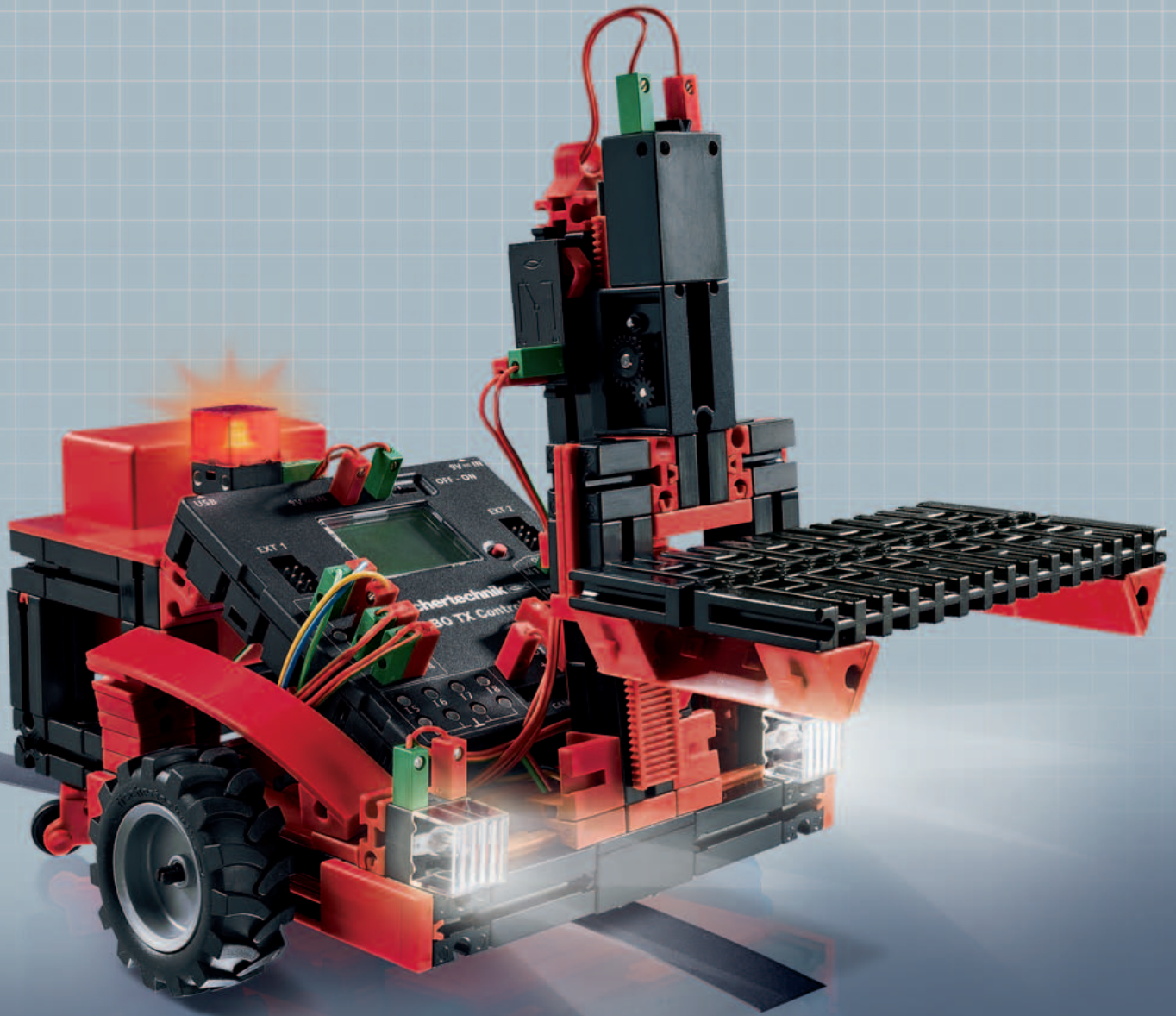


**fischertechnik** 

**COMPUTING**

Begeleidend boekje



**ROBO TX Training Lab**

**11** MODELS  
MODEL 2

<b>Welkom bij ROBO TX Training Lab</b>	<b>3</b>
<b>Een beetje algemene informatie</b>	<b>3</b>
Elektriciteit	3
Omtrent dit begeleidend boekje	3
Robots, de kunstmatige mensen?	4
Computing, (vrijwel) alles automatisch	4
<b>Uitleg van de onderdelen</b>	<b>4</b>
Encoder-motoren	5
XS-motor	5
Gloeilamp	5
Lenslamp	5
Fototransistor	5
Sporensensor	6
Toets	6
Warmtesensor (NTC)	6
<b>Een paar tips</b>	<b>7</b>
<b>Eerste stappen</b>	<b>7</b>
<b>Startersmodellen</b>	<b>8</b>
De handendroger	8
Het stoplicht	9
De lift	10
De afwasmachine	10
Temperatuurregeling	11
<b>Robots – de volgende uitdaging</b>	<b>12</b>
Basismodel	12
De sporenzoeker	13
De grasmaaier	14
De voetbalrobot	15
De meetrobot	16
De vorkheftruck	17
<b>Wanneer iets niet gelijk werkt ...</b>	<b>19</b>
<b>En hoe gaat het nu verder?</b>	<b>20</b>

## Welkom bij ROBO TX Training Lab

Hallo!

Wij zijn blij, dat jij hebt gekozen voor de bouwdoos „ROBO TX Training Lab” van fischertechnik. En wij beloven je, dat je belangstelling zal worden beloond. Met deze bouwdoos kun je namelijk een heleboel interessante experimenten uitvoeren en spannende taken oplossen.

Tijdens het doorlezen van dit beeldschermboekje en het proberen om experimenten en taken uit te voeren, zul je stap voor stap leren hoe je met behulp van de ROBO TX Controller van fischertechnik eenvoudige en ook gecompliceerde machines en robots kunt bouwen en programmeren.

Zoals het met leren nu eenmaal gaat, kun je niet onmiddellijk met de moeilijkste dingen beginnen, hoewel die natuurlijk vaak een beetje interessanter zijn dan de wat eenvoudiger. Daarom hebben wij de experimenten en taken in dit boekje zodanig opgebouwd, dat je met iedere nieuwe taak iets anders bijleert, dat je dan weer bij de volgende taak kunt gebruiken.

Dus wees maar niet bang, wij beginnen klein en werken dan samen toe naar de grote robots.

Nu willen wij je veel plezier en succes bij het experimenteren met het ROBO TX Training Lab toewensen.

Je team van

*fischertechnik*



## Een beetje algemene informatie

Voordat je echt met de bouwdoos aan de slag kunt gaan, moet je nog een paar dingetjes weten. De onderdelen waarmee wij aan het werk gaan, zijn weliswaar sterk, maar wanneer deze niet goed worden behandeld, kunnen zij eventueel worden beschadigd.

### Elektriciteit

Zoals je waarschijnlijk wel weet, werken heel veel onderdelen van het ROBO TX Training Lab op elektrische stroom. En waar stroom in het spel is, moet je er bijzonder goed op letten, dat je geen fouten maakt. Kijk daarom altijd goed in de bouwhandleiding wanneer het gaat om het bedraden van elektrische onderdelen.

In geen geval mag je de plus- en min-pool rechtstreeks met elkaar verbinden, dus kortsluiten. Daardoor kan de ROBO TX Controller, maar ook de accu, beschadigd worden.

Elektriciteit en elektronica is een net zo interessant onderwerp als robotica (dus waarom het in deze bouwdoos gaat), en fischertechnik heeft een bouwdoos, die zich speciaal met deze onderwerpen bezig houdt. Wanneer je daarvoor belangstelling hebt, zul je met de „PROFI E-Tech”-bouwdoos net zoveel plezier hebben als met het ROBO TX Training Lab.



### Omtrent dit begeleidend boekje

Dit PDF-boekje heeft een paar mogelijkheden die met een gedrukt exemplaar niet mogelijk zijn en die je misschien wel van internet kent.

#### Koppelingen binnen het boekje

Wanneer in de tekst over een onderwerp wordt gesproken, wat in een ander deel van dit boekje gedetailleerder wordt uitgelegd (bijv. over de onderdelen), is deze tekst donkerblauw en onderstreept. Je kunt de tekst aanklikken en daarmee ga je automatisch naar die pagina, waar de uitleg is vermeld. Zoiets noem je een „kruisverwijzing”.

#### Achtergrondinformatie

Soms worden in dit boekje begrippen of moeilijke woorden gebruikt, die je graag uitgelegd zou hebben. Deze begrippen zijn groen en onderstreept. Wanneer je deze tekst met de muiscursor aanraakt, verschijnt een veld met de uitleg.

### Koppelingen buiten dit boekje

Voor een paar koppelingen heb je een internetverbinding nodig (bijv. met de internetpagina van fischertechnik), c.q. een geïnstalleerde ROBO Pro (voor het openen van de online help van ROBO Pro). Deze koppelingen zijn lichtblauw en onderstreept.

### **Foto's**

Een foto zegt meer dan 1.000 woorden. Dit heb je misschien wel vaker gehoord. En omdat er zoveel waarheid in zit, kun je door het met de muiscursor aanraken van de bruine en onderstreepte woorden een foto openen, waarop je kunt herkennen wat in de tekst is bedoeld.

## **Robots, de kunstmatige mensen?**

Aan wat denk je eerst, als je het woord „robot” hoort? Heb je weleens een robot gezien? In de bioscoop of op de televisie? Of zelfs al een keertje in het echt?

Er zijn ontelbaar verschillende soorten robots. Sommige zien er een beetje uit als een mens, andere zijn vaak niet meer dan één of meerdere armen. Wat maakt een robot nou precies tot een robot?

In het woordenboek staat: „Robots zijn stationaire of mobiele machines, die volgens een bepaald programma vastgelegde taken uitvoeren.”



## **Computing, (vrijwel) alles automatisch**

Robots zijn dus machines, die door middel van een programma worden aangestuurd. En deze besturing van machines (of in ons geval modellen) noemen wij „computing”.

Met het „ROBO TX Training Lab” kun je fantastisch kennis maken met dit onderwerp. De bouwdoos bevat namelijk alles, wat je voor het bouwen en aansturen van vele verschillende machines nodig hebt.

De programma's voor het besturen van de modellen kun je op de PC met behulp van de software ROBO Pro maken en aansluitend via een USB- of Bluetooth-verbinding naar de ROBO TX Controller versturen. De Controller „controleert” en stuurt daarna het model overeenkomstig het programma, dat jij hebt geprogrammeerd.

## **Uitleg van de onderdelen**

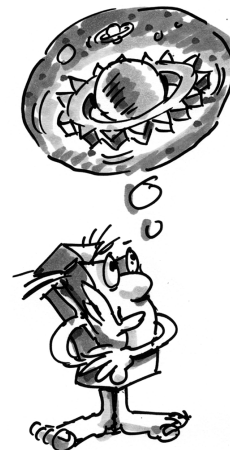
### **Dit zit allemaal in de bouwdoos**

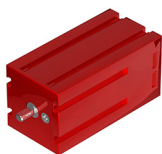
Allereerst zitten hier talrijke fischertechnik-modules in, maar daarnaast ook motoren, lampen en sensoren, alsmede een kleurige bouwhandleiding voor het bouwen van de verschillende modellen.

Wanneer je de modules allemaal hebt uitgepakt, moet je enkele componenten eerst monteren, voordat je kunt beginnen (bijv. kabels en stekkers). Welke dat precies zijn, staat in de bouwhandleiding onder „Montagetips” beschreven. Het beste is om daarmee te beginnen.

## **Schakelaars**

Als schakelaars worden alle onderdelen bedoeld, die een actie uit kunnen voeren. Dat betekent dat, wanneer zij op een elektrische stroom worden aangesloten, zij op enigerlei wijze „actief” worden. Meestal kun je dat direct zien. Een motor gaat draaien, een lamp brandt, enz.





### Encoder-motoren

Als aandrijving voor onze robots maken wij gebruik van de twee encoder-motoren uit de bouwdoos. Op het eerste gezicht zijn dit normale elektromotoren, die bedoeld zijn voor een spanning van 9 volt en een stroomverbruik van maximaal 0,5 ampère.

De encoder-motoren kunnen echter nog meer: Naast de aansluiting voor de voedingsspanning van de motor hebben ze nog een bus voor een 3-polige aansluitkabel, waarmee m.b.v. de zogenaamde encoder de draaibeweging van de motor kan worden bepaald.

De encoder werkt eigenlijk net zo'n beetje als de kilometer teller op een fiets. Een magneet (bij de fiets zit deze meestal op één van de spaken) komt bij iedere omwenteling langs een sensor (zit bij de fiets meestal aan de vork), waardoor de sensor een puls opwekt. Deze pulsen kunnen geteld worden en bij het voorbeeld van de kilometer teller met de omtrek van de band worden vermenigvuldigd. Hierdoor weet je hoever je hebt gefietst.

De encoders op encoder-motoren van fischertechnik wekken per omwenteling van de motoras 3 pulsen op. En omdat de encoder-motoren bovendien nog een overbrenging met een [overbrengingsverhouding van 25:1](#) (spreek uit als „25 op 1“) hebben, komt één omwenteling van de as, die uit de overbrenging komt, overeen met 75 pulsen van de encoder.

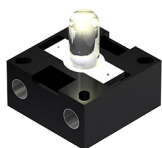


### XS-motor

De XS-motor is een elektromotor, die net zo lang en hoog is als een fischertechnik-module. Bovendien is de motor erg licht. Daardoor kun je hem op plaatsen monteren, waar geen ruimte is voor grote motoren.

De beide overbrengingen, die eveneens in de bouwdoos zitten, passen precies op de XS-motor.

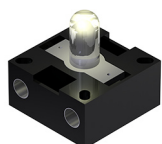
De XS-motor is bedoeld voor een voedingsspanning van 9 volt en een stroomverbruik van maximaal 0,3 ampère.



### Gloeilamp

In de bouwdoos zitten twee gloeilampen. Deze kunnen op veel manieren worden gebruikt, bijv. als signaallampen bij een stoplicht, maar ook als knipperlicht op een robot.

De gloeilampen zijn bedoeld voor een spanning van 9 volt en verbruiken ca. 0,1 ampère stroom.



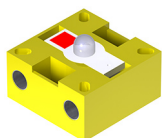
### Lenslamp

In deze lamp is een lens verwerkt, die het licht bundelt. Deze lijkt erg op een bolle lamp. Je moet dus oppassen, dat je deze niet verwisselt. Om het verschil beter te kunnen zien, is de insteekfitting van deze lamp grijs, terwijl de bolle lamp een witte fitting heeft. De lenslamp heb je nodig voor het bouwen van een [fotocel](#).

De lenslamp is net als de gloeilampen bedoeld voor een spanning van 9 volt en verbruikt ca. 0,15 ampère stroom.

## Sensoren

Sensoren zijn in zekere zin de contrastukken van de schakelaars. Deze voeren geen acties uit, maar reageren op bepaalde situaties en gebeurtenissen. Een toets reageert bijv. op een „druk op de knop“, doordat deze een elektrische stroom doorlaat of onderbreekt. Een warmtesensor reageert op de temperatuur in zijn omgeving.



### Fototransistor

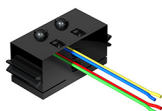
De fototransistor wordt ook wel omschreven als „helderheidssensor“. Het is een „voeler“, die op helderheid reageert.

Deze vormt bij een [fotocel](#) de reflector voor de lenslamp. Bij een grotere helderheid, dus wanneer de transistor door de lenslamp wordt belicht, geleidt de transistor stroom. Wanneer de lichtstraal wordt onderbroken, geleidt de transistor geen stroom.

**Let op:**

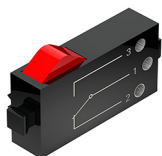
**Wanneer de fototransistor op de voedingsspanning wordt aangesloten, moet je er op letten, dat de polen correct zijn aangesloten. De plus-pool moet op de rode markering op de fototransistor worden aangesloten.**





### Sporensensor

De **infrarood** sporensensor is een digitale sensor voor de herkenning van een zwart spoor op een witte ondergrond op een afstand van 5 - 30 mm. Hij bestaat uit twee zend- en twee ontvangelementen. Als aansluiting heb je twee digitale ingangen en de 9 volt voedingsspanning (plus- en min-pool) nodig op de [ROBO TX Controller](#).



### Toets

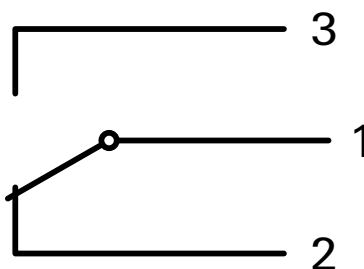
De toets wordt ook wel aanraaksensor genoemd. Wanneer de rode knop wordt ingedrukt, wordt mechanisch een schakelaar geschakeld en vloeit er een stroom tussen de contacten 1 (middencontact) en 3. Gelijktijdig wordt het contact tussen de aansluitingen 1 en 2 onderbroken. Hierdoor kun je de toets op twee manieren gebruiken:

#### Als „Normally open“ (in rusttoestand geopend):

De contacten 1 en 3 worden aangesloten.

Toets ingedrukt: er vloeit stroom.

Toets niet ingedrukt: er vloeit geen stroom.

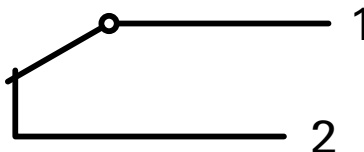


#### Als „Normally closed“ (in rusttoestand gesloten):

De contacten 1 en 2 worden aangesloten.

Toets ingedrukt: er vloeit geen stroom.

Toets niet ingedrukt: er vloeit stroom.



### Warmtesensor (NTC)

Bij dit onderdeel gaat het om een warmtesensor, waarmee temperaturen gemeten kunnen worden. Bij 20°C draagt de **elektrische weerstand** 1,5 kΩ (kilo-ohm). NTC betekent Negatieve Temperatuur Coëfficiënt. Simpel gezegd betekent dit, dat de weerstand daalt wanneer de temperatuur stijgt.

De informatie die door de sensoren wordt geleverd (bijv. licht-donker, ingedrukt, niet ingedrukt, temperatuurwaarde) kun je, zoals wij later nog zullen zien, via de [ROBO TX Controller](#) doorsturen naar de pc en daar met behulp van de software een motor zodanig programmeren, dat een ventilator gaat draaien, zodra de fotocel wordt onderbroken.



## Software ROBO Pro 2.x

ROBO Pro is een grafisch programmeeroppervlak waarmee je de programma's voor de [ROBO TX Controller](#) kunt programmeren.

„Grafisch programmeeroppervlak“ betekent, dat je de programma's niet regel voor regel met de hand moet „schrijven“, maar met behulp van grafische symbolen eenvoudig figuurlijk kunt samenstellen. Een voorbeeld voor een dergelijk programma staat op de afbeelding rechts.

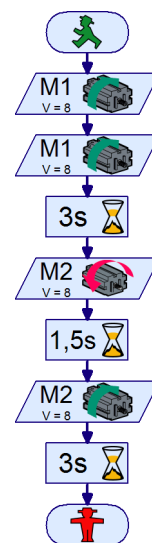
Hoe een programma precies kan worden geprogrammeerd, is uitvoerig beschreven in de hoofdstukken 3 en 4 van de [online help van ROBO Pro](#).

De installatie van ROBO Pro en de driver voor de ROBO TX Controller is beschreven in de installatiehandleiding, die in de bouwdoos zit.

De software staat op dezelfde CD als dit begeleidend boekje.

Wanneer je klaar bent met de installatie van ROBO Pro, kun je ROBO Pro gelijk starten, omdat wij vervolgens de online help van de software nodig hebben.

Het beste kun je nu de beide eerste hoofdstukken van de [online help van ROBO Pro](#) doorlezen. Daarbij leer je de software al een beetje kennen, zodat wij aansluitend direct met het experimenteren kunnen beginnen.





## ROBO TX Controller

De ROBO TX Controller is het hart van deze computing-bouwdoos. Deze stuurt namelijk de [schakelaars](#) aan en analyseert de informatie die van de [sensoren](#) wordt ontvangen.

Voor deze taak is de ROBO TX Controller voorzien van talrijke aansluitingen, waar je de onderdelen op aan kunt sluiten. Welke onderdelen op welke aansluitingen kunnen worden aangesloten en wat de functies van de aansluitingen zijn, is beschreven in de gebruiksaanwijzing van ROBO TX Controller.

De geïntegreerde Bluetooth-interface is een bijzondere highlight. Hiermee kun je zonder kabels je PC met de ROBO TX Controller verbinden. Of meerdere controllers met de PC en met elkaar.

Hoe de controller omgaat met de afzonderlijke onderdelen en wat deze elk moeten doen, leg je vast in het programma, dat jij in de [software ROBO Pro](#) schrijft.



## Stroomvoorziening (niet meegeleverd)

Omdat je waarschijnlijk wel weet, dat heel veel onderdelen van het ROBO TX Training Lab op elektrische stroom werken, heb je ook een stroomvoorziening nodig.

Hier is de accuset van fischertechnik het beste voor geschikt. Deze wordt niet met het ROBO TX Training Lab meegeleverd.

## Een paar tips

Experimenteren geeft het meeste plezier, wanneer de experimenten ook werken. Daarom moet je tijdens het bouwen van de modellen in paar basisregels in acht nemen:

- **Zorgvuldig werken**

Neem je tijd en kijk goed naar de bouwhandleiding voor het model. Wanneer je achteraf naar een fout moet zoeken, duurt het nog veel langer.

- **De beweeglijkheid van alle onderdelen controleren**

Controleer tijdens het bouwen steeds weer of onderdelen die moeten kunnen bewegen, ook daadwerkelijk gemakkelijk kunnen bewegen.

- **De interfacetest gebruiken**

Voordat je begint met het schrijven van een programma voor een model, moet je alle op de ROBO TX Controller aangesloten onderdelen met behulp van de interfacetest van ROBO Pro testen. Hoe dat precies werkt wordt uitgelegd in de [online help van ROBO Pro](#) in hoofdstuk 2.4.



## Eerste stappen

Zo. Na alle voorbereidingen en informatie kun je nu eindelijk zelf beginnen.

Omdat je voor het experimenteren, behalve met fischertechnik-onderdelen, vooral met de [Software ROBO Pro](#) zult werken, zou je jezelf eigenlijk als eerste vertrouwd moeten maken met hoe je een programma schrijft en dit leren. En omdat dit in de hoofdstukken 3 en 4 van de [online help van ROBO Pro](#) werkelijk uitstekend wordt uitgelegd, kun je nu het beste deze hoofdstukken zorgvuldig doorwerken.

**Ook hier geldt de tip:** Neem je tijd en besteed hier aandacht aan, dan zul je aansluitend des te meer plezier aan je modellen beleven.



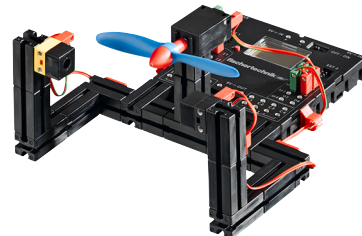
## Startersmodellen

Nadat je de hoofdstukken 3 en 4 van de [online help van ROBO Pro](#) hebt doorgelezen, kun je nu al enkele modellen van de bouwdoos programmeren. Daarom kunnen wij nu ook onmiddellijk beginnen. Altijd wanneer je een model hebt afgebouwd en bedraad, moet je met behulp van de [interfacetest](#) controleren, of alle in- en uitgangen op de [ROBO TX Controller](#) correct zijn aangesloten en de [sensoren](#), [motoren](#) en [lampen](#) goed werken.

### De handendroger

Op je school zijn op het toilet naast de wasbakken nieuwe handendrogers geïnstalleerd. Deze zijn voorzien van een [fotocel](#), waarmee de ventilator in- en uitgeschakeld kan worden.

- Bouw allereerst het model zoals dat in de bouwhandleiding is beschreven.



#### Taak 1:

- De handendroger moet nu zodanig worden geprogrammeerd, dat, zodra de [fotocel](#) wordt onderbroken, de ventilator ingeschakeld en na 5 seconden weer uitgeschakeld wordt.

#### Programmeertips:

- Schakel in het programmaverloop eerst de lamp voor de [fotocel](#) op [uitgang M2](#) in.
- Daarna wacht je een seconde, zodat de [fototransistor](#) voldoende tijd heeft om op het licht te reageren. Pas dan werkt de [fotocel](#) goed.
- Dan vraag je op [ingang I1](#) om de waarde van de [fototransistor](#). Is de waarde „1“ ([fotocel](#) niet onderbroken), moet de waarde op de ingang in een lus permanent worden opgevraagd.
- Zodra de waarde „0“ wordt ([fotocel](#) onderbroken), schakel je de motor M1 in en na 5 seconden weer uit.
- Daarna moet de waarde van [fototransistor](#) opnieuw worden opgevraagd enz.

Start je programma met behulp van de [start-button](#) en controleer, of het werkt zoals het moet. Wanneer dat het geval is ben je al een heel eind op weg om een professionele ROBO Pro-programmeerder te worden.

Wanneer het nog niet werkt, moet je proberen uit te vinden waar dat aan ligt:

- Met behulp van de [interfacetest](#) kun je controleren, of alle in- en uitgangen werken en correct zijn aangesloten.
- Terwijl het programma draait, kun je aan de hand van de rood gemarkeerde onderdelen het programmaverloop volgen. Daarmee kun je snel herkennen, waar er een foutje is ingeslopen.
- Tenslotte kun je jou programma vergelijken met het kant-en-klare voorbeeldprogramma, dat je met het symbool rechts kunt openen:

Nadat deze hindernis is genomen, willen wij de taakstelling nog graag een beetje veranderen:



#### Taak 2:

- De rector, die altijd probeert om energie te sparen, bevalt het niet dat de handendroger altijd nog een bepaalde tijd doorloopt, hoewel de handen reeds droog zijn. Hij geeft jou opdracht om het programma zodanig te programmeren, dat de ventilator wordt uitgeschakeld op het moment dat de handen teruggetrokken worden. Geen probleem voor jou, of wel?

#### Programmeertips:

- Net als in het eerste programma vraag je met behulp van een vertakking de waarde van de [fototransistor](#) I1 op. Wanneer de waarde „0“ is, moet je de motor M1 inschakelen, wanneer de waarde „1“ is moet je de motor M1 uitschakelen, enz.
- Ook voor deze taak is er in geval van nood een kant-en-klaar programma:



## Het stoplicht

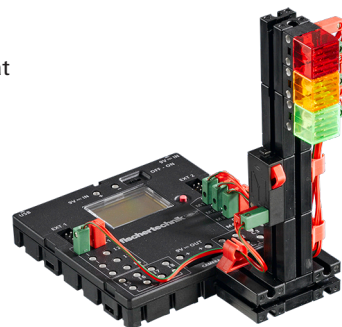
Voor jullie huis wordt een stoplicht neergezet. Omdat de monteur van het bedrijf dat het stoplicht plaatst weinig tijd heeft, biedt jij hem aan, om de programmering voor de besturing van het stoplicht voor je rekening te nemen.

De man legt uit hoe de besturing moet werken. Bouw echter eerst het model.



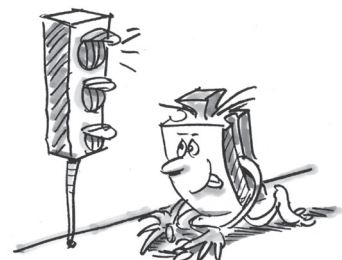
### Taak 1:

- In eerste instantie moet het stoplicht op groen staan. Wanneer een voetganger op de [toets](#) I1 drukt, moet het stoplicht 3 seconden later op oranje en na nog eens 4 seconden op rood springen. De rode fase moet 10 seconden duren, de daarop volgende rood-oranjefase 3 seconden. Daarna moet het verkeerslicht weer groen worden.



### Programmeertips:

- De verschillende lampen horen bij de onderstaande interface-uitgangen:
  - rood – M1
  - geel – M2
  - groen – M3
- Schakel de lampen achtereenvolgens zodanig aan en uit, dat de gewenste volgorde tot stand komt.
- Je kunt het kant-en-klare programma weer laden door de afbeelding rechts aan te klikken:



### Taak 2:

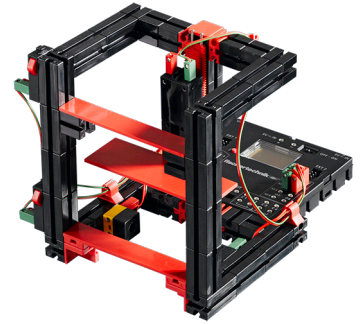
- De volgende dag belt de monteur van het stoplichtenbedrijf jou. Hij heeft vergeten om te vertellen, dat in de schakelkast op het trottoir een schakelaar I2 zit, die het stoplicht oranje moet laten knipperen, op het moment dat hier op wordt gedrukt. Je belooft de monteur om deze functie snel in je programma te integreren.

### Programmeertips:

- Bevestig een tweede [toets](#) op je stoplichtmodel en sluit deze op [ingang I2](#) aan.
- Vraag met behulp van een tweede vertakking om de waarde van [ingang I2](#). Wanneer op de [toets](#) I2 wordt gedrukt, vertakt het verloop naar het knipperlicht. Voor de rest verloopt de stoplichtbesturing gelijk als in taak 1.
- Je kunt de knipperlichtfunctie inschakelen door middel van het met tussenpozen van 0,5 seconde in- en uitschakelen van de [lamp](#) M2. Gebruik daarvoor een subprogramma. Hoe je een subprogramma moet maken kun je in hoofdstuk 4 van de [onlinehelp van ROBO Pro](#) nalezen.
- Je opent het voorbeeldprogramma door zoals je gewend bent op het symbool te klikken. Probeer echter, voordat je dit naleest, eerst zelf om een oplossing te vinden. Veel succes!

## De lift

Je buurman heeft in zijn werkplaats een goederenlift ingebouwd, zodat hij zijn zware reservoirs niet meer de trap op naar de eerste verdieping hoeft te sjouwen. Nu heeft hij echter nog een besturing nodig, die jij natuurlijk graag voor hem wilt programmeren.



### Taak 1:

- Programmeer de lift zodanig, dat hij om te beginnen naar beneden naar zijn uitgangspositie beweegt. In deze positie is de [fotocel](#) op I3 onderbroken. De lift moet, wanneer op één van de beide [toetsen](#) wordt gedrukt (I1 op de begane grond of I2 op de eerste verdieping) altijd naar de andere verdieping bewegen.

### Programmeertips:

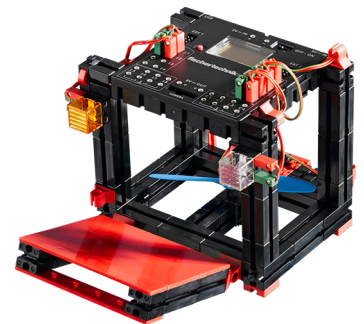
- Wanneer de [fotocel](#) is onderbroken, staat de lift op de begane grond, wanneer deze niet is onderbroken, ga je ervan uit dat de lift op de eerste verdieping staat.
- Hoe ver de lift van beneden omhoog beweegt, bestuur je met behulp van de tijd, waarin de motor is ingeschakeld.
- Ook wanneer je dit helemaal zonder hulp lukt, hebben wij voor alle gevallen weer een oplossing klaar staan:

Voordat je je aan de volgende programmeertaken waagt, moet je eigenlijk eerst weer de [online help van ROBO Pro](#) openen. Werk dan zorgvuldig hoofdstuk 5 door. Schakel in ROBO Pro over op level 3. Langzaam maar zeker worden de programmeertaken moeilijker. Wij maken gebruik van analoge ingangen, bedieningselementen, operatoren en variabelen. Maar wanneer je de online help van ROBO Pro zorgvuldig doorleest, zal het later gemakkelijker zijn om daarmee om te gaan.

## De afwasmachine

Als volgende zul jij je aan het programmeren van een afwasmachine wagen. De afwasmachine moet onderstaande functies hebben:

- [Toets](#) voor het in- en uitschakelen (I1)
- [Toets](#), die herkent of de deur gesloten is (I2)
- Afwasfunctie (propeller op M1)
- Droogfunctie (rode [lamp](#) op M2)
- Aanduiding dat de machine is ingeschakeld (oranje [lamp](#) op M3)
- Aanduiding in welke bedrijfsmodus de machine staat (transparante [lamp](#) op M4)
  - Snel knipperen: de machine wast af
  - Langzaam knipperen: de machine droogt
  - Permanent brandend: de machine is klaar



### Taak 1:

- Schrijf een afwasprogramma, dat pas begint wanneer de deur is gesloten en de startknop I1 is ingedrukt. Als eerste wordt afgewassen en daarna gedroogd. De bedrijfstoestand moet met behulp van de beide lampen op M3 en M4 worden aangegeven.

### Programmeertips:

- Hier weer ons voorstel voor de oplossing:


**Taak 2:**

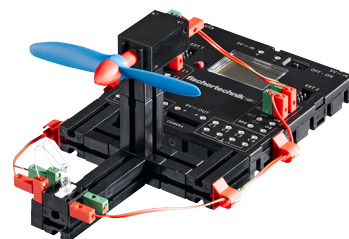
- Wanneer de deur wordt geopend moet de afwascyclus worden onderbroken. Nadat de deur is gesloten gaat het programma op die plaats verder waar het werd onderbroken.
- Geef bovendien de bedrijfstoestand van de afwasmachine op het display van de [ROBO TX Controller](#) aan.

**Programmeertips:**

- Toegegeven, deze taak is al wat moeilijker. En wanneer je deze helemaal zelf kunt oplossen: super! Wanneer je ondanks al het harde nadenken niet verder komt, is dat ook geen enkel probleem. Kijk maar eens rustig naar ons voorstel.

**Temperatuurregeling**

Bij jullie thuis is een nieuwe airconditioning geïnstalleerd. Natuurlijk heb je onmiddellijk aan de installateur gevraagd hoe de temperatuurregeling werkt. Deze heeft bereidwillig uitgelegd, dat een temperatuursensor voortdurend de heersende temperatuur meet. Zodra een bovenste grenswaarde wordt overschreden, wordt de koeling ingeschakeld. Wanneer daarentegen de onderste grenswaarde wordt overschreden, wordt de koeling uitgeschakeld en de verwarming ingeschakeld. Nu wil je aan de hand van het model „Temperatuurregeling” proberen om eveneens een dergelijk [regelcircuit](#) te programmeren. Bouw eerst het model.

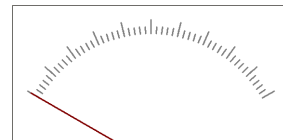

**Taak 1:**

De verwarming wordt gesimuleerd met behulp van de [lenslamp M2](#). Als „koelaggregaat” wordt de ventilator op [uitgang M1](#) gebruikt. Voor de temperatuurmeting gebruiken wij de [NTC-weerstand](#) op de [ingang I2](#).

- Programmeer het model zodanig, dat boven een bepaalde temperatuur de verwarming uitgeschakeld en de ventilator ingeschakeld wordt. Deze moet zolang koelen tot een onderste grenswaarde wordt bereikt. Daarna moet de ventilator uitgeschakeld en de verwarming ingeschakeld worden.
- De actuele waarde van de analoge ingang moet aan een meetinstrument en een tekstaanduiding in ROBO Pro worden uitgegeven, alsmede aan het display van de [ROBO TX Controller](#).

**Programmeertips:**

- **Let op:** De weerstandswaarde van de [NTC-weerstand](#) daalt wanneer de temperatuur stijgt. De bovenste temperatuurgrenswaarde is derhalve de laagste waarde van I2. Bij deze grenswaarde moet de ventilator inschakelen. De onderste temperatuurgrenswaarde is de hoogste waarde van I2. Bij deze grenswaarde moet de verwarming inschakelen.
- Met behulp van de [interfacetest](#) kun je er achter komen welke waarde I2 bij kamertemperatuur heeft. Schakel lamp M2 in en hou in de gaten hoever de waarde daalt. Daarna schakel je de ventilator in om er achter te komen hoe ver de waarde stijgt. Kies aan de hand daarvan de grenswaarden voor verwarmen en koelen.
- Geef de waarde van de analoge ingang in jouw programma met behulp van een tekst-aanduiding en/of met een meetinstrument aan (zie ook de [online help van ROBO Pro](#) hoofdstuk 8.1)



Temp. = 0

- Door op het symbool rechts te klikken kun je het kant-en-klare programma openen.

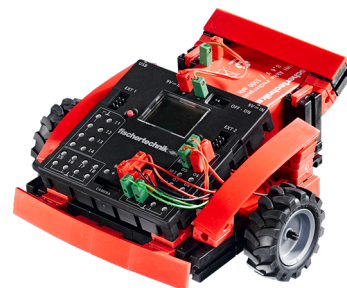


## Robots – de volgende uitdaging

### Basismodel

Met dit model willen wij er samen achter komen hoe je een rijdende robot aan kunt sturen. Hoe zorg je er voor dat de robot gaat rijden, hoe werkt de besturing en kan de nauwkeurigheid misschien nog worden verbeterd? Deze vragen zullen we met behulp van de taken in dit hoofdstuk beantwoorden.

Maar eerst moet je natuurlijk de robot bouwen. De beschrijving staat zoals altijd in de bouwhandleiding.



- Bouw de robot zoals in de bouwhandleiding is beschreven.
- Neem voor dit bouwen alle tijd. Bekijk de tekeningen in de bouwhandleiding nauwkeurig, met name die van de bedrading. Wanneer je de onderdelen anders met de [ROBO TX Controller](#) verbindt dan in de bouwhandleiding is beschreven, kan het zijn dat de robot zich anders gedraagt dan jij verwacht.
- Controleer na het bouwen alle op de [ROBO TX Controller](#) aangesloten onderdelen met behulp van de [interface-test](#) van de [ROBO Pro software](#). Wanneer jij de motoren linksom laat draaien moet de robot vooruit rijden.



#### Taak 1: Eerst maar eens gewoon rechthoek (niveau 1)

- Laat de robot 3 seconden lang rechthoek rijden (**niet op de tafel, gevaar voor vallen!**) en aansluitend weer 3 seconden achteruit.
  - *Is de robot daadwerkelijk weer op zijn uitgangspunt terecht gekomen?*
- Herhaal het programma meerdere malen en observeer of de robot werkelijk nauwkeurig steeds weer rechthoek voor- en achteruit rijdt.

#### Programmeertips:

- Ook wanneer deze taak voor jou geen enkel probleem is, komt hier ons voorstel:

#### De besturing

Ook wanneer het leuk is om te bekijken hoe de robot rechthoek rijdt, is dat misschien toch wel een beetje eentonig. Daarom moet de robot nu maar eens leren om bochten te gaan rijden. En hoe dat gaat? Heel eenvoudig:



#### Taak 2: Ook eens een bocht rijden (niveau 1)

- Laat de robot weer 3 seconden rechthoek rijden (beide motoren draaien even snel), verander daarna gedurende 1 seconde de draairichting van de rechter motor (M1) en laat de robot aansluitend weer 3 seconden rechthoek rijden (dus beide motoren weer even snel in dezelfde richting laten draaien).
- Probeer te ontdekken hoe lang je de motoren in verschillende richtingen moet laten draaien om de robot 90° te laten draaien.

#### Programmeertips:

- Verander hiervoor de wachttijd achter het besturingselement, waarbij de richting van de tweede motor wordt veranderd.
- Je kunt het kant-en-klare programma weer laden door de afbeelding rechts aan te klikken:



#### Taak 3: Een figuur rijden (niveau 2)

- Nu je weet hoe lang de draairichting van een motor moet worden omgedraaid, zodat de robot rechts of links afslaat, moet je de robot nu zodanig programmeren, dat deze een vierhoek rijdt en aansluitend weer op de uitgangspositie terecht komt.
  - *Controleer met behulp van een markering of de robot werkelijk weer precies naar zijn uitgangspositie rijdt.*

**Programmeertips:**

- Voor het „om de hoek rijden” kun je een subprogramma schrijven. Daardoor blijft je hoofdprogramma overzichtelijker.
- Uiteraard heb jij jou oplossing voor de taak al in je hoofd. Maar voor alle gevallen is hier weer ons voorstel:

**Altijd hetzelfde en nooit gelijk?**

Zoals je hebt gemerkt is de [herhalingsnauwkeurigheid](#) van de robot nog voor verbetering vatbaar. Ook wanneer hij meerdere malen precies dezelfde taak uitvoert, is het resultaat niet altijd gelijk. Dat heeft verschillende oorzaken. Een daarvan is dat beide motoren niet precies even snel draaien. Zo kan bijv. de overbrenging van de ene motor zwaarder draaien dan die van de andere. En aangezien beide motoren op dezelfde spanning (9 volt) werken, draait de ene motor dan langzamer dan de andere. Een aangezien wij onze robots door middel van wachttijden hebben aangestuurd, heeft het ene wiel zich misschien gedurende deze tijd verder gedraaid dan het andere.

De oplossing zou dus zijn om beide motoren even snel te laten draaien. En dat nu kan met behulp van de [encoder-motoren](#) heel gemakkelijk worden gedaan.


**Taak 4: De encoder-motoren gebruiken**

- Herhaal de laatste drie taken en gebruik in plaats van de gewone [motoruitgangs-](#) en [wachttijd-elementen](#) de [encodermotor-elementen](#). Hoe deze worden gebruikt wordt beschreven in de [onlinehulp van ROBO Pro](#) in hoofdstuk 11.6.

**Programmeertip:**

- Met behulp van het [encodermotor-element](#) kun je met één programma-element beide motoren gelijktijdig aansturen. Met behulp van het [afstands invoerveld](#) zorg je ervoor dat elke motor ook daadwerkelijk net zover draait, als hij moet draaien.
- De voorstellen voor de oplossing van dit probleem kun je weer door het aanklikken van de symbolen rechts openen:

Voor het tellen van de pulsen aan de snelle [telingangen C1-C4](#) heb je geen verder programma-element in [ROBO Pro](#) nodig. De motor M1 wordt intern automatisch aan de telingang C1 toegewezen, M2 hoort bij C2 enz.

**Opmerking:**

Wanneer je model ondanks het gebruik van de [encodermotor-elementen](#) niet rechtuit rijdt, kan de oorzaak in het model zelf liggen. Wanneer bijv. een naafmoer, die de kracht van de as op de wielen overdraagt, niet strak genoeg is vastgedraaid, draait de as door en rijdt het model een bocht, hoewel de motoren even snel draaien. Draai de naafmoeren daarom goed vast.

**De sporenzoeker**


De robot kan nu rechtuit rijden en bochten maken. En tot dusver doet hij dat precies zo, zoals jij hem dat met behulp van het programma hebt aangegeven.

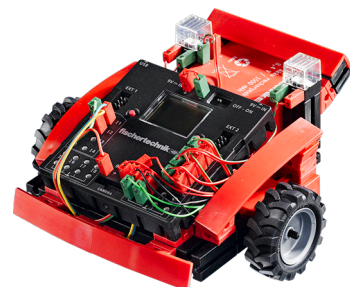
Maar een robot moet eigenlijk waar mogelijk zelfstandig kunnen reageren. Daarom willen wij hem nu iets geven waar hij op kan reageren: Een zwarte lijn als markering op de vloer.

Het doel is om de robot de zwarte lijn op te laten zoeken en hier dan langs te laten rijden.

Maar het ene na het andere. Als eerste moet je het basismodel ombouwen tot sporenzoeker. Hoe je dat moet doen, staat natuurlijk in de bouwhandleiding.

Nadat je het model helemaal hebt omgebouwd, moet je met behulp van de [interfacetest](#) controleren of alle onderdelen goed op de [ROBO TX Controller](#) zijn aangesloten en werken. Je kunt de [sporensensor](#) testen door hem boven het zwarte spoor van het parcours te houden en dan naar de zijkanten toe te verplaatsen. De signalen op de ingangen waarop de [sporensensor](#) is aangesloten moeten duidelijk veranderen.

Denk eraan om de ingangen in de [interfacetest](#) op „Digitaal 10 V (sporensensor)” in te stellen.






**Taak 1: Een spoor herkennen (niveau 2)**

- Programmeer de robot zo dat deze een recht zwart spoor volgt, waarop de robot is neergezet. Wanneer hij het spoor verliest of aan het einde hiervan is, moet hij stil blijven staan en de beide lampen moeten ieder 3 maal knipperen. Gebruik voor deze taak het parcours 1a uit de bouwdoos.

**Programmeertips:**

- Allereerst vraag je de waarden van de beide ingangen van de [sporensensor](#) op. Wanneer beide ingangen het signaal „0” ontvangen, staat de robot op het zwarte spoor. Je kunt de robot laten rijden.
- De knipperfunctie leg je in een subprogramma vast.
- Gebruik voor het rechthoekig rijden weer het [encodermotor-element](#), uiteraard zonder dat een afstand is ingevoerd.
- Het kant-en-klare programma vind je onder:

Nu kan je robot reageren. Uiteraard is deze functie nog wel erg beperkt. Het zou veel beter zijn wanneer de robot in plaats van te stoppen de richting zou kunnen corrigeren om het spoor verder te vervolgen.


**Taak 2: Het spoor volgen (niveau 2)**

- Breidt je programma uit met de functie dat de robot bij het verlaten van het spoor de richting overeenkomstig corrigeert en het spoor blijft volgen. Probeer deze taak eerst met parcours 1a, daarna met parcours 1b.

**Programmeertips:**

- Er zijn meerdere mogelijkheden, om de richting te corrigeren. Je kunt een motor laten stoppen en de andere door laten draaien, of één motor in tegengestelde richting laten draaien. Probeer gewoon welke methode het beste geschikt is.
- Hier ons voorstel voor de oplossing:

Ziezo. Nu kan de robot op een optische „rail” rijden. Het nadeel is alleen dat je de robot eerst op de rails moet zetten. Dat willen wij veranderen. De robot moet zijn spoor nu zelfstandig opzoeken.


**Taak 3: Het spoor opsporen en aansluitend volgen (niveau 2)**

- Schrijf een programma „zoeken”, waarmee de robot naar een spoor kan zoeken, wanneer hij bij het starten geen heeft gevonden. Daarvoor moet de robot eerst 1 maal een rondje draaien. Kan hij daarbij geen spoor vinden, moet hij een klein stukje rechthoekig rijden. Zodra de robot een spoor herkent, moet hij dit volgen. Anders moet hij opnieuw beginnen met zoeken. Wanneer hij 10 maal een rondje gedraaid heeft zonder een spoor te vinden, moet hij blijven staan en 3 maal knipperen.

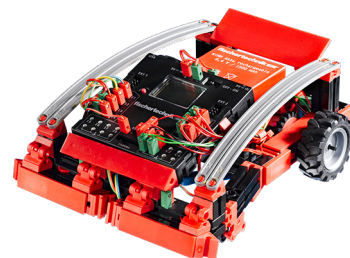
**Programmeertips:**

- Wanneer je nog niet helemaal op het goede spoor zit, is hier ons voorstel om het probleem op te lossen:

**De grasmaaier**

Kunnen robots gras maaien? Natuurlijk kunnen zij dat. Je hoeft ze alleen maar te vertellen, hoe zij met hindernissen om moeten gaan en waar het gazon ophoudt. En dan kun je het werk al aan de robot overlaten en de vrije middag in het zwembad doorbrengen.

Maar voordat je zover bent moet je eerst aan de hand van de bouwhandleiding de grasmaairobot nog bouwen. Aansluitend moet je met behulp van de [interfacetest](#) weer controleren of alles precies zo werkt als het moet werken.



Dan kunnen wij gaan programmeren. De grasmatt wordt gesymboliseerd door middel van het witte vlak van ons voetbalstadion parcours 1b, dat door een zwarte rand wordt begrensd. De grasmaaier mag deze rand niet overschrijden (omdat hij anders op de atletiekbaan terecht komt). Wanneer een hindernis op de grasmatt staat, moet de robot uitwijken wanneer hij hier tegenaan rijdt. Bovendien moet het maaien worden uitgeschakeld, wanneer een hindernis wordt herkend.



#### Taak 1: Grenzen en hindernissen herkennen en uitwijken (niveau 2)

- Programmeer de grasmaaier zodanig, dat deze vanuit zijn startpositie (binnen de begrenzing) rechthoekig vooruit rijdt, tot hij op een hindernis stoot of de begrenzing van de grasmatt (zwarte lijn) bereikt.
- Wanneer de grasmaaier op een hindernis stoot (bumpers voorkant), moet hij onmiddellijk stil blijven staan, het maaien moet uitgeschakeld worden, de grasmaaier moet een stukje achteruit rijden, naar links draaien en opnieuw vooruit rijden en het maaien opnieuw inschakelen. Deze functie verpak je in het subprogramma „uitwijken”.
- Wanneer de grasmaaier op de grasmattbegrenzing stoot, moet hij eveneens stil blijven staan en het subprogramma „uitwijken” starten.

#### Programmeertips:

- Onze oplossing:

Afhankelijk van de afmetingen van jou „grasmatt” zou het onderstaande probleem kunnen ontstaan: Afhankelijk van het feit hoe de duur van de „omkering” wordt ingeschakeld, rijdt de robot of alleen langs de rand, of blijft eventueel altijd in hetzelfde gebied van de „grasmatt”. Daarom moet de robot zich bij het uitwijken altijd een beetje anders gedragen.



#### Taak 2: Het toeval (niveau 3)

- Verander het programma van de grasmaaier zodanig, dat de grasmaaier iedere keer tijdens het uitwijken een andere hoek om te draaien neemt. De ene keer moet hij dus wat verder, de andere keer wat minder ver draaien. Bovendien moet hij bij een hindernis, die hij met behulp van de rechter bumper herkent, naar links uitwijken en omgekeerd.

#### Programmeertips:

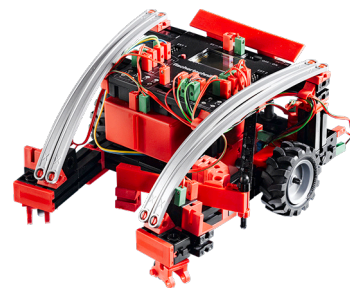
- Om gebruik te kunnen maken van variabelen in ROBO Pro, moet je de software op „Level 3” (niveau 3) instellen.
- Hiervoor heb je een [toevalsgenerator](#) nodig. Die kun je bijv. maken door de waarde van een variabele van een [tellus](#) steeds weer vanaf „0” tot een bepaalde waarde op te laten tellen. En in het subprogramma „uitwijken” stel je na het commando voor het draaien een „Wachten op...” element in, dat erop wacht dat de variabele „0” wordt. Aangezien de variabele waarschijnlijk iedere keer, wanneer deze wordt opgevraagd, een andere waarde heeft, duurt het steeds langer tot deze weer bij „0” is. En dit resulteert in verschillende tijden tijdens het draaien van de robot.
- Wij geven toe dat dit niet heel eenvoudig is. En wanneer je ondanks het vele nadenken geen goede oplossing kunt bedenken, hebben wij natuurlijk altijd weer een oplossing waarmee je aan de slag kunt:

### De voetbalrobot

Heb je ooit van de Robo-Cup gehoord? Dat is het wereldkampioenschap voetbal voor robots. Deze vinden elk jaar in een ander land plaats. Er zijn verschillende divisies, die bestemd zijn voor de verschillende soorten robots. Meer informatie vindt je bijv. op de Robo-Cup-Homepage <http://www.robocup.org>.

In de bouwhandleiding staat een voorstel voor het bouwen van een voetbalrobot. Deze is net zo beweeglijk als onze andere robots, maar heeft bovendien nog een fotocel om een bal te herkennen en een „schietmechanisme”. Bouw deze robot eerst maar eens en dan kunnen wij hem aansluitend programmeren en een paar trucjes „trainen”. En zoals altijd moet je altijd eerst de werking van het model met behulp van de [interfacetest](#) controleren, voordat je begint met het programmeren.

Als bal kun je bijv. een tennisbal gebruiken (wordt niet met de bouwdoos meegeleverd). Afhankelijk van de bal die je gebruikt, moet je het schietmechanisme misschien een beetje aanpassen.





### Taak 1: „Daar heeft hij de bal onderschept en schiet ...“ (niveau 2)

- In de eerste stap moet onze elektronische balkunstenaar leren om de bal te herkennen en daarop te reageren. Programmeer hem zodanig dat hij de bal schiet, zodra deze door de [fotocel](#) is herkend. Experimenteer maar rustig een beetje met de „schietsnelheid“. Eventueel kan ook een korte pauze tussen het „herkennen“ en „schieten“ tot een verbetering leiden.
- Dan moet hij leren om een „penalty“ te schieten. Leg een bal op de penaltystip van parcours 1a. Zet de robot aan het begin van de zwarte lijn. Nu moet hij langs de lijn bewegend een aanloop nemen en zodra de bal wordt herkend, deze op het doel schieten. Aan het einde van de lijn moet hij stoppen en omdraaien.

#### Programmeertips:

- Net als bij de [handendroger](#) moet je bij de [fotocel](#) na het inschakelen van de [lenslamp](#) een seconde wachten, voordat je de waarde van de [fototransistor](#) opvraagt.
- Je hebt het als trainer niet gemakkelijk. Wanneer jou robotspeler niet naar je luistert, kun je hem mis-schieten met behulp van ons programmavoorbeeld overtuigen:

Maar aangezien een echte balkunstenaar nog meer moet kunnen dan alleen maar penalty's nemen, willen wij de vaardigheden van onze voetbalrobot nog een beetje uitbreiden.



### Taak 2: Wie zoekt zal vinden. (niveau 2)

- De voetbalrobot moet nu binnen de grenzen van het stadion (parcours 1b) rondrijden en daarbij de begrenzingslijn niet overschrijden. Wanneer hij de bal vindt, moet hij deze wegschieten, bij voorkeur op doel.

#### Programmeertips:

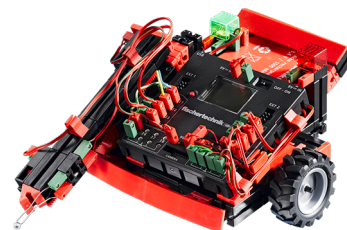
- De meeste functies voor dit programma heb je bij de [grasmaaier](#) al eens geprogrammeerd. Je kunt dus het grasmaaierprogramma gebruiken, onder een andere naam opslaan en het programma uitbreiden met de functies van de voetbalrobot.
- Waarschijnlijk zul je het niet nodig hebben, maar ook hier hebben wij weer een oplossing voor het probleem:

### De meetrobot

Is het onder je bed warmer dan onder je bureau? En hoe heet is een kaarsenvlam? En kun je met een ijsblokje je kamer koelen?

Voor zulke (en andere) vragen kun je met behulp van de meetrobot antwoorden vinden. De robot is voorzien van een [temperatuursensor \(NTC\)](#) en kan op verschillende plaatsen de temperatuur meten en weergeven. Bovendien is de meetrobot voorzien van een [sporensensor](#), waarmee je hem een zwarte lijn kunt laten volgen.

Bouw eerst de meetrobot zoals dat in de bouwhandleiding is beschreven en controleer de functies met behulp van de [interfacetest](#).



### Taak 1: Besturing en temperatuurmeting (niveau 3)

- Neem het programma van de [sporenzoeker](#) en breidt dit uit met een subprogramma voor het besturen van de meetarm. De robot moet op het spoor van parcours 1b rijden en met bepaalde tussenpozen de temperatuur meten.  
De temperatuur moet op het display van de [ROBO TX Controller](#) worden weergegeven.

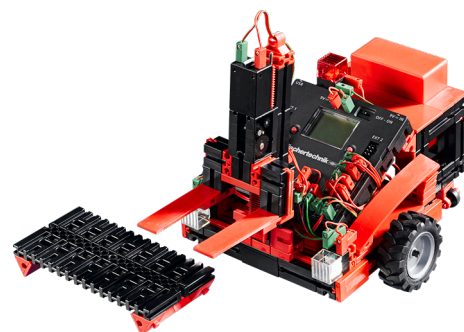
**Programmeertips:**

- Om de weerstand van de [temperatuurvoeler](#) weer te geven maak je gebruik van een [analoge ingang](#), een [bedieningsvelduitgang](#) en een [„display“-bedieningselement](#).
- Op de analoge ingang wordt de weerstandswaarde van de [temperatuurvoeler](#) weergegeven, niet de temperatuur. Om deze waarde in een temperatuuraanduiding om te kunnen rekenen, kun je een subprogramma gebruiken.
- Onze complete oplossing voor deze taak staat hier:

**De vorkheftruck**

Een gebied waarvoor in de industrie graag robots worden gebruikt, is de logistiek. Dus overal daar waar voorwerpen moeten worden verplaatst, van „A“ naar „B“, zoals dat zo mooi wordt gezegd.

Dergelijke transporttaken kunt je schitterend naspelen met behulp van het vorkheftruckmodel. Bouw het model zoals dat in de bouwhandleiding is beschreven. Controleer aansluitend met behulp van de [interfacetest](#), of alle onderdelen correct functioneren.


**Taak 1: Erop en eronder (niveau 3)**

- Schrijf voor de functies „optillen” en „laten zakken” elk een subprogramma.
- Omdat de vorken tijdens het rijden niet te ver omhoog of omlaag mogen staan, heb je nog een subprogramma „rijpositie” nodig.

**Programmeertips:**

- Tijdens het „optillen” moeten de vorken zover naar boven worden verplaatst (motor M3 draait linksom), dat de bovenste eindschakelaar I4 wordt geactiveerd.
- Tijdens het „laten zakken” moeten de vorken zover naar beneden worden verplaatst (motor M3 draait rechtsom), tot de overbrenging de onderste eindschakelaar I3 voorbij is en de schakelaar weer open staat. Hiervoor gebruik je een [„wachten op“-element](#) en stelt deze in op „1 -> 0 (vallend)”.
- Voor de „rijpositie” moeten de vorken zover naar boven toe worden verplaatst, dat de vorken net boven de onderste eindschakelaars staan.
- Zoals altijd staat hier een oplossing van ons:

Nu het mechanische gedeelte ook heel goed werkt, willen wij daar ook gebruik van maken. Tenslotte moet een vorkheftruck voorwerpen transporteren en niet alleen maar de vorken kunnen bewegen.


**Taak 2: Van „A” naar „B” (niveau 3)**

- In de bouwdoos zit een parcours 2a voor de vorkheftruck. Deze moet nu vanuit zijn startpositie weggrijpen, een pallet, die op veld „A” staat, oppakken, langs het spoor naar veld „B” transporteren en daar lossen.

**Programmeertips:**

- Programmeer de vorkheftruck zo, dat deze altijd wanneer het spoor ten einde is, de pallet oppakt of lost. Gebruik voor elk van deze beide handelingen een subprogramma.
- Laat de vorkheftruck zo langzaam rijden, dat hij in geen geval het spoor per ongeluk kwijt raakt. Anders zou hij kunnen denken, dat hij aan het einde van het spoor is aangekomen en de pallet op moet pakken of moet lossen.
- Onze oplossing:



### Taak 3: Aftakkingen herkennen

Nu wordt het spannend. De vorkheftruck moet een aftakking herkennen. Gebruik daarvoor parcours 2b.

- De vorkheftruck start op zijn startpositie. Hij moet naar veld „A” rijden, daar de pallet ophalen en naar veld „B” transporteren. Aansluitend moet hij terug naar zijn startpositie rijden. Terwijl hij het spoor volgt, moeten de schijnwerpers branden. Wanneer de vorkheftruck draait of achteruit rijdt, moet de rode lamp branden.

#### Programmeertips:

- De onderbreking van het spoor kan de vorkheftruck als aftakking herkennen. Wanneer hij deze plaats bereikt, kun je beslissen hoe hij verder moet rijden: De vorkheftruck kan rechtuit rijden tot hij het spoor vindt, links afslaan en daarbij het spoor opzoeken of rechts afslaan tot hij het spoor terug vindt. Schrijf voor deze drie mogelijkheden elk een afzonderlijk subprogramma.
- Om bij het naar links of rechts afslaan het spoor weer terug te kunnen vinden, moet de robot eerst een klein stukje rechtuit rijden. Anders draait hij voorbij aan spoor.
- Wanneer je tijdens het programmeren het spoor bijster bent geraakt, kun je misschien wel even naar ons voorbeeld kijken:



### Taak 4: Eindeloos transporteren

- Breidt het programma uit taak 3 zodanig uit, dat de vorkheftruck na terugkomst in zijn startpositie, een korte pauze inlast en dan de pallet uit veld „B” haalt, terug brengt naar „A” en dan weer naar start rijdt. De vorkheftruck moet het geheel nu als een lopende band voortdurend herhalen.

#### Programmeertips:

- Schrijf een eindeloze lus door vanaf het laatste programma-element een lijn naar het begin van het programma terug te brengen.
- Ook wanneer deze taak voor jou geen enkel probleem is, komt hier ons voorstel:





## Wanneer iets niet gelijk werkt ...

... heeft dat in de meeste gevallen een eenvoudige oorzaak. Alleen is die oorzaak niet altijd eenvoudig op te sporen. Daarom willen wij je op deze plaats een paar aanwijzingen geven omtrent mogelijke storingsbronnen.

### Interfacetest

Ook hier nogmaals de aanwijzing: Controleer de werking van de afzonderlijke onderdelen met behulp van de [interfacetest](#) in [ROBO Pro](#).



### Kabels en bedrading

Wanneer een elektrisch onderdeel helemaal niet werkt, moet je de kabel controleren waarmee je dit onderdeel op de [ROBO TX Controller](#) hebt aangesloten. Verbind daarvoor met behulp van de kabel de [accu](#) en een [gloeilamp](#) met elkaar. Wanneer de gloeilamp gaat branden, is de kabel in orde.

Ook verkeerd gemonteerde stekkers (bijv. een groene stekker op een rode kabel) kunnen een bron van storingen zijn. Controleer tevens of „+“ en „-“ correct zijn aangesloten. Vergelijk daarvoor jou model met de afbeeldingen in de bouwhandleiding.

### Los contact

Een onderdeel, dat afwisselend wel en dan weer niet werkt, heeft vermoedelijk ergens in de bedrading een los contact.

De meest voorkomende oorzaak daarvoor is:

- **los zittende stekkers**

Wanneer de stekkers van de kabels te los in de bussen zitten, is er onvoldoende contact. In dit geval kun je met behulp van de schroevendraaier de contactveren voor op de desbetreffende stekkers **voorzichtig** uit elkaar buigen. Echter zo weinig, dat de stekkers weer vast in de bussen zitten, wanneer je ze insteekt.

- **Een slecht contact tussen kabel en stekker**

Controleer ook het contact tussen de [gestripte](#) kabeluiteinden in de stekker en de stekker zelf. Mogelijkerwijs is het voldoende om de schroeven in de stekker een beetje vaster te draaien.

### Kortsluiting

Er is sprake van kortsluiting wanneer een plus- en een min-aansluiting elkaar raken. Zowel de [accu](#), alsook de [ROBO TX Controller](#) hebben een ingebouwde zekering, waardoor deze bij kortsluiting niet beschadigd raken. De zekeringen schakelen de stroom eenvoudig gedurende enige tijd uit. Jou model werkt dan natuurlijk ook niet meer.

De oorzaak voor een kortsluiting kan hetzij een fout in de bedrading zijn, maar ook niet strak genoeg aangedraaide schroeven in de stekkers. Die kunnen elkaar raken wanneer de stekkers dienovereenkomstig zijn ingestoken en zo kortsluiting veroorzaken. Je moet de schroeven altijd helemaal indraaien en de stekkers zodanig insteken, dat de schroeven elkaar niet kunnen raken.

### Voeding

Kleine uitvalverschijnselen of langzaam draaiende motoren duiden meestal op een lege [accu](#). In dit geval moet je de accu met behulp van de meegeleverde acculader opladen. Wanneer de rode LED op de oplader stopt met knipperen en permanent gaat branden is de accu volledig opgeladen.

### Fouten in het programma

Ook wanneer niemand het graag wil toegeven: iedereen maakt fouten. En vooral bij complexere programma's kunnen er heel snel kleine foutjes insluipen.

Wanneer je dus zelf alles aan het model controleert en alle fouten verholpen hebt en je model desondanks niet dat doet wat jij wilt, moet je ook nog je programma controleren. Loop het programma stukje voor stukje door en kijk of je de fout kunt vinden.

In de online-modus, dus wanneer de [ROBO TX Controller](#) met de PC is verbonden, kun je het programma ook op het beeldscherm volgen, terwijl dit draait. Het op dat moment actieve programma-element wordt benadrukt, zodat je altijd kunt zien op welke plaats het programma op dat moment is en waar de fout optreedt.

## De laatste joker

Wanneer je ondanks alle genomen maatregelen de fout niet hebt kunnen vinden, zijn er twee mogelijkheden waarop je hulp kunt krijgen:

- **E-mail-joker**

Je kunt ons bij fischertechnik een e-mailtje sturen en ons jou probleem uitleggen.

Het e-mailadres is [info@fischertechnik.de](mailto:info@fischertechnik.de).

- **Publieksjoker**

Je kunt ons ook op internet onder <http://www.fischertechnik.de> bezoeken. Daar bestaat onder andere een forum, dat je graag zal willen helpen. Bovendien kun je gratis lid worden van de fischertechnik-fanclub.

## En hoe gaat het nu verder?

Was dat alles? Nee, natuurlijk niet. De experimenten en modellen, die je in dit boekje hebt leren kennen en uitgetest, zijn slechts een begin. Bij wijze van spreken je eerste „pogingen” op het gebied van het reusachtig veelomvattende en spannende onderwerp „computing”.

Wat wij je hier hebben laten zien is slechts een tipje van de sluier van alle mogelijkheden die je met de [ROBO TX Controller](#) en de onderdelen van fischertechnik hebt. Maar nu ben je zelf aan de beurt. Je mag je fantasie de vrije loop laten en gewoon de meest fantastische modellen bouwen.

Wanneer je nog geen idee voor een compleet eigen model hebt, kijk dan gewoon eens naar de modellen in dit boekje. Misschien bedenk je iets wat jij aan jou model anders zou doen. Of je verandert de functie van een model.

Je kunt bijv. bij de [vorkheftruck](#) in plaats van de vorken ook een pen bevestigen, die dan omhoog en omlaag kan en op een groot stuk papier tekent, terwijl de robot er over heen rijdt. Je kunt de figuren dan niet alleen maar rijden, maar ook vastleggen. En heb je van de vorkheftruck een tekenmachine gemaakt.

En wanneer je vrienden ook een [ROBO TX Controller](#) hebben, wordt het nog interessanter. Met behulp van de Bluetooth-interface kan niet alleen jouw PC met [ROBO TX Controller](#) communiceren, maar kunnen meerdere Controllers ook met elkaar communiceren. Jullie kunnen dan bijv. 2 robots programmeren die op elkaar reageren. Of met elkaar dansen. In hoofdstuk 7 van de [online help van ROBO Pro](#) staat interessante informatie met betrekking tot dit onderwerp.

Heeft jou computer een Bluetooth-interface? Wanneer dat zo is, kun je de [ROBO TX Controller](#) in plaats van via een USB-kabel ook via Bluetooth met de computer verbinden. Wanneer dat niet het geval is, kun je ook een USB-Bluetooth-stick kopen en via die mogelijkheid de [ROBO TX Controller](#) draadloos met jou PC verbinden. Hoe dat gaat, kun je nalezen in de gebruiksaanwijzing van de [ROBO TX Controller](#) en op <http://www.fischertechnik.de>.

Dus waar wacht je op? Aan de slag. Vind uit en experimenteer. En laat je niet door een beetje tegenspoed tegenhouden. Je hebt voor het experimenteren vooral geduld en doorzettingsvermogen nodig. De beloning is een werkend model.

Wij wensen je veel plezier bij het uitproberen van je eigen ideeën.

