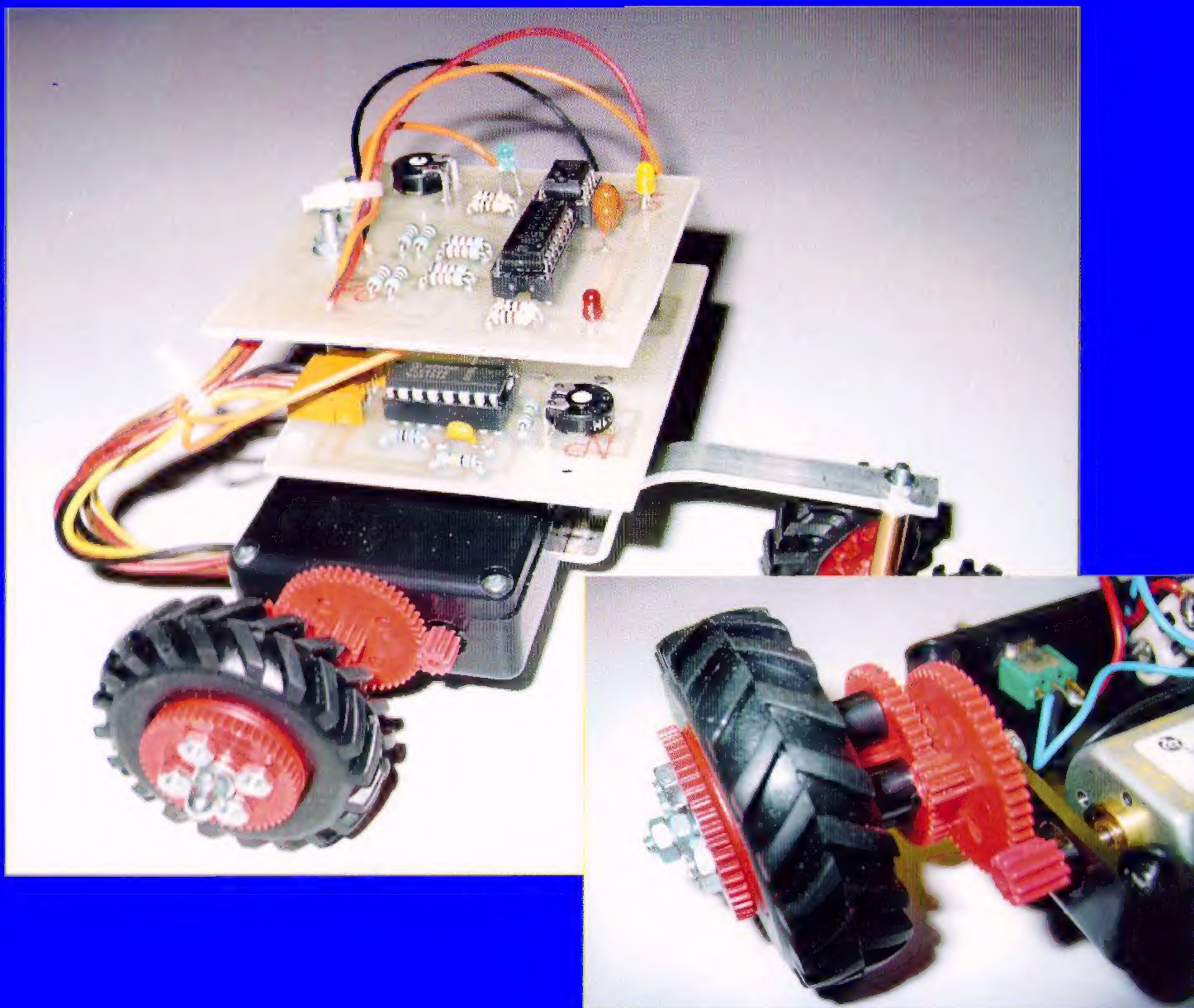


# **Mechatronica en robotica**

**Werkboek voor de robot hobbyist**



**Willem H. M. van Dreumel**

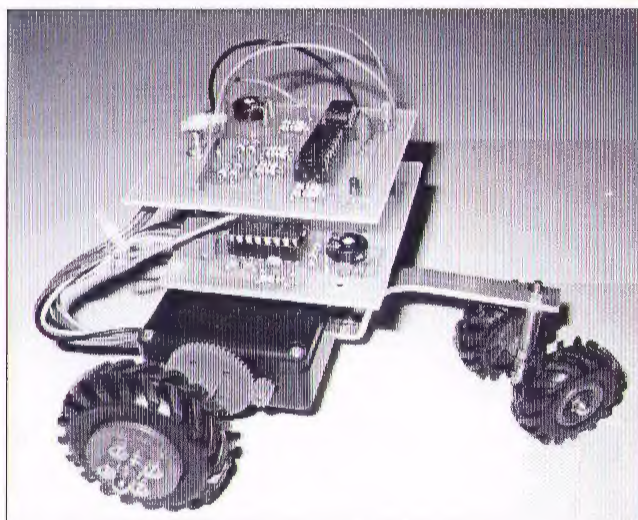
**Vego**



# Mechatronica en robotica

Werkboek voor de robot hobbyist

Willem H. M. van Dreumel



**ISBN:** 90-805610-7-X

**NUGI:** 433

**Eerste druk:** januari 2003

**Tweede druk:** maart 2004

**Derde druk:** augustus 2005

**© 2005:** Willem H. M. van Dreumel, Nijverdal, alle auteurswet rechten voorbehouden

**Vego** vof

Vego vof, Postbus 32.014, 6370 JA Landgraaf (NL), [vego\\_vof@compuserve.com](mailto:vego_vof@compuserve.com), [www.vego.nl](http://www.vego.nl)

# Inhoud

<b>1</b>	<b>Inleiding</b>	<b>4</b>
<b>2</b>	<b>De mechanische constructie</b>	<b>8</b>
	Inleiding	8
	Model 1: een zéér groot loopwerk	8
	Model 2: een kleiner model	9
	Model 3: epoxy printplaat als basis	14
	Model 4: aluminium als basis	15
	Alternatieven	16
	De bedrading in het loopwerk	17
	Epiloog	18
<b>3</b>	<b>De motorbesturing</b>	<b>19</b>
	Inleiding	19
	De motorbesturing in de praktijk	21
	Epiloog	23
<b>4</b>	<b>Onze robot leert dansen</b>	<b>24</b>
	Inleiding	24
	De robot gaat dansen	26
	Epiloog	29
<b>5</b>	<b>Onze robot leert luisteren</b>	<b>30</b>
	Inleiding	30
	De praktijk	30
	Epiloog	32
<b>6</b>	<b>Onze robot leert voelen</b>	<b>33</b>
	Inleiding	33
	De sensoren	34
	De elektronica	36
	De sensormontage	38
	Epiloog	40
<b>7</b>	<b>Onze robot krijgt ogen</b>	<b>41</b>
	Inleiding	41
	Het laadstation	42
	De ontvanger	43
	De elektronica	45
	Epiloog	47
<b>8</b>	<b>We gaan combineren</b>	<b>48</b>
	Inleiding	48
	De schakeling	48
	Epiloog	49
<b>9</b>	<b>Onze robot krijgt <math>\mu</math>P-hersenen</b>	<b>50</b>
	Inleiding	50
	De processorprint	51
	Nog enkele tips	54
	Epiloog	55

<b>10</b>	<b>Installeren van C-Control BASIC</b>	<b>56</b>
	Inleiding	56
	De koppeling met de PC	56
	CCBASIC programmeren	57
	Epiloog	58
<b>11</b>	<b>Een alternatieve motorbesturing</b>	<b>59</b>
	Inleiding	59
	Het aangepaste schema	60
	De R-2R netwerken	64
	Onze robot beweegt softwarematig	65
	Epiloog	66
<b>12</b>	<b>Obstakel detectie</b>	<b>67</b>
	Inleiding	67
	Epiloog	69
<b>13</b>	<b>Stappenmotor besturing</b>	<b>70</b>
	Inleiding	70
	Stappenmotoren	71
	De stappenmotor driver SAA1027	75
	Toepassingen van stappenmotoren	78
	Epiloog	79
<b>14</b>	<b>Archie, de man van staal</b>	<b>80</b>
	Inleiding	80
	Armen	80
	Benen	81
	Epiloog	82
<b>15</b>	<b>Een lesje CCBASIC</b>	<b>83</b>
	Inleiding	83
	Belangrijke CCBASIC-principes	83
	Epiloog	86
<b>16</b>	<b>Robotica informatie</b>	<b>87</b>
<b>17</b>	<b>De printen uit dit werkboek</b>	<b>90</b>
	Het maken van de printen	90
	De printontwerpen	91
	Downloaden via het Internet	93
<b>18</b>	<b>Aanvullingen bij de derde druk</b>	<b>94</b>

# 1 Inleiding

- Wat is een robot?** Archie, de man van staal, was één van de eerste robots. Hij zag eruit als een ijzeren mens en bestond slechts in een stripverhaal. Tegenwoordig kan een robot elk uiterlijk hebben afhankelijk van de functie waarvoor hij gebouwd is. Het begrip robot is daardoor een beetje ondoorzichtig geworden. **Mechatronica** is dan ook een betere verzamelnaam: mechanische bewegingen, gestuurd door elektronica. Onder deze definitie valt echter ook de modelspoorbaan. Bovendien kunnen afstandbestuurde modellen eronder worden verstaan. We trekken de definitie dan ook wat nauwer aan:  
***De elektronica moet onderdeel zijn van het mechatronisch apparaat.***  
Bovendien moet de robot bepaalde taken zelfstandig kunnen uitvoeren. Sensoren vormen het waarnemingsapparaat waarmee hij contact onderhoudt met de buitenwereld. Binnen deze definitie zijn de robots in dit boek opgebouwd.
- Robot op wieltjes** Verwacht géén lopende en pratende elektronische mannetjes of honden. De ontwikkeling van dergelijke robots laten wij met plezier aan Sony over. Wij ontwerpen een robot op wieltjes. Oftewel een voertuig dat over een plat oppervlak kan rijden, aangedreven door een paar elektromotoren. Een robot die u kunt besturen met geluid of met licht en die in staat is bepaalde problemen "intelligent" op te lossen. In een bepaald stadium van zijn ontwikkeling is onze robot bijvoorbeeld zo slim dat hij niet van de tafel afrijdt. Iets later is hij in staat zijn accu's zelfstandig op te laden. Dát soort dingen kunt u van dit boek verwachten.
- Geen voorgekauwde kost** Verwacht van dit boek ook geen volledig voorgekauwde kant-en-klare nabouwontwerpen á la Elektuur. Dit boek is een **Werkboek!** Wij geven u suggesties hoe u de mechanica van een rijdende robot zo gemakkelijk mogelijk kunt bouwen. Wij bewijzen dat in eerste instantie ingewikkelde problemen als "*hoe kan een rijdende robot ontdekken dat hij van de tafel gaat afrijden en op tijd gepaste maatregelen treffen?*" in wezen tamelijk eenvoudig zijn op te lossen.  
Wij hopen dat het enthousiasme waarmee aan dit project is gewerkt overslaat op u, lezer, en de start wordt van een interessante en leerzame hobby: **Mechatronica!**
- Een uniek project** Alle bouwbeschrijvingen die in elektronica boeken worden beschreven, vereisen alleen wat handigheid met de soldeerbout en de kniptang. Dit project eist heel wat meer handigheid en ervaring van de potentiële nabouwer! Natuurlijk staat het knutselen met elektronica voorop. Maar aan de basis staat een knap stukje fijnmechanica. Zonder dit loopwerk is de elektronica nutteloos. Er zijn diverse bouwpakketten van robots in de handel. Tóch hebben wij de voorkeur gegeven aan een compleet zelfbouwproject.
- Tandwieltjes, boutjes, palletjes en wieltjes** Dat betekent dat u aan de slag moet met miniatuur motoren, tandwieltjes, boutjes, palletjes en wieltjes. Laat u zich daardoor echter niet afschrikken! Zoals uit hoofdstuk 2 blijkt, is iedere handige knutselaar in staat zélf een soepel bewegende robot te maken.
- Hoe werkt de robot?** Het prototype heeft twee motoren, die individueel de twee voorwielen aandrijven. Groot voordeel van dit systeem is dat er geen ingewikkelde mechanische constructie nodig is voor het besturen van de robot. De

motoren zijn rechtstreeks, al dan niet via een vertragingskastje, gekoppeld aan de wielen. De robot werkt dus als een tank: draaien de beide motoren in dezelfde richting, dan rijdt de robot vooruit. Poolt u beide motoren om, dan zal de robot achteruit rijden. Wordt slechts één motor gestuurd, dan zal de robot een zeer scherpe bocht maken rond het wiel dat niet wordt aangedreven.

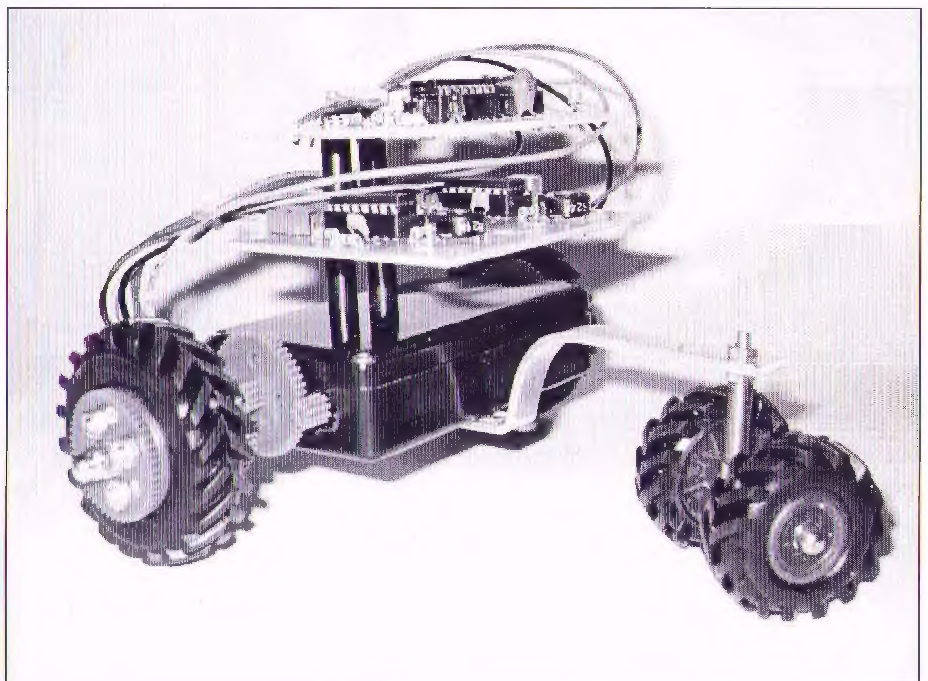
**Een eenvoudig instapmodel**

Dit lijkt misschien een beetje primitief, en in feite is het dat ook, maar toch heeft een op een dergelijke manier bewegende robot zijn charmes. Bovendien is het de bedoeling de lezers van dit boek warm te maken voor "mechatronica" en dan is een eenvoudig "instapmodel" een voor de hand liggende keuze.

**Hoe een en ander er uitziet**

Op onderstaande foto is één prototype van de robot in volle glorie te aanschouwen. Duidelijk is het chassis of loopwerk te bewonderen dat de motoren en de batterijen of accu's bevat. Boven op dit loopwerk worden de besturingsprinten gemonteerd. De onderste print is de absoluut noodzakelijke basisprint, waarmee de twee motoren worden bestuurd. De bovenste print is een van de digitaal werkende printen, die de robot "intelligentie" geeft.

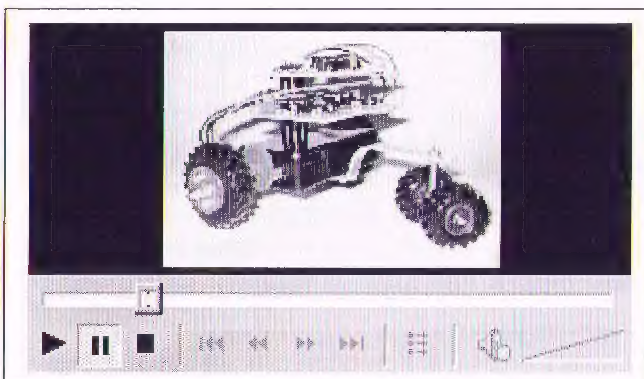
**De robot in volle glorie, voorzien van twee besturingsprinten**



**MPEG-filmpje op internet**

Wie het gedrag van deze robot in het echt wil aanschouwen kan terecht op de internetsite van de uitgever:  
[www.vego.nl/hobby/robot/robot.htm](http://www.vego.nl/hobby/robot/robot.htm).

**De robot is in actie te zien op de internetsite van de uitgever**



Op deze pagina staat een aanklikknop waarmee u een MPEG-filmpje van de robot in actie kunt bekijken.

**Wat kan de robot?**

De mechanica van een robot maken is één zaak, er nadien iets mee doen is een tweede. Natuurlijk is de vraag naar nuttigheid van een dergelijk project volledig misplaatst. Het is de bedoeling dat u ervaring opdoet met iets dat tegenwoordig met een groot woord "mechatronica" wordt genoemd, de combinatie van fijnmechanica en elektronica. In de volgende hoofdstukken wordt een aantal voorbeelden gegeven van elektronische besturingen die met dit project mogelijk zijn.

**Motorbesturing**

Via een eenvoudige motorbesturing kunt u de twee motoren in feite in "tri-state" aansturen: motor uit, motor draaien in de ene richting, motor draaien in de nadere richting. Als u er rekening mee houdt met het feit dat er twee motoren in de robot aanwezig zijn, ontstaan dus in principe negen "vrijheidsgraden".

**De "vrijheidsgraden" in de beweging van de robot**

LINKER WIEL	VOER TUIG	RECHTER WIEL
vooruit	Vooruit	vooruit
vooruit	Rotatie om rechter wiel rechtsom	stilstand
vooruit	Pirouette rechtsom	achteruit
stilstand	Rotatie om linker wiel linksom	vooruit
stilstand	Stilstand	stilstand
stilstand	Rotatie om linker wiel rechtsom	achteruit
achteruit	Pirouette linksom	vooruit
achteruit	Rotatie om rechter wiel linksom	stilstand
achteruit	Achteruit	achteruit

**Dansende robot**

Nadat de fundamentele motorbesturing klaar is, kan men de robot op diverse manieren aansturen. Bij het project "dansende robot" worden de drie besturingsmogelijkheden van de twee motoren digitaal aangestuurd uit twee eenvoudige digitaal naar analoog omzetteren die op hun beurt weer worden gestuurd uit een pulsgenerator. Een en ander heeft tot gevolg dat de robot een voorgeprogrammeerd bewegingspatroon gaat uitvoeren dat het best te omschrijven is als een wilde dans.

**Reageren op geluid**

In een volgend project wordt een schema ontwikkeld, waarmee de twee motoren van de robot reageren op het omgevingsgeluid. Via een elektret microfoon wordt dit geluid opgevangen, versterkt en omgezet in een digitaal signaal dat de DAC's uit het vorige project aanstuurt. De robot is dan te "besturen" met bijvoorbeeld handgeklap.

**Reageren op optische sensoren**

In het volgend project wordt de robot voorzien van primitieve zintuigen. Vier infrarode sensoren zorgen ervoor dat de robot bijvoorbeeld niet van de rand van een tafel valt, maar onmiddellijk omkeert als een van de sensoren een "afgrond" ontdekt.

**Robot zoekt zelf zijn voeding**

Het betere werk! In dit project wordt de robot voorzien van "ogen" die op zoek gaan naar de laadbron voor zijn accu's. Het apparaatje wordt uitgerust met twee "voelsprietten" die als bestek dienen en waarmee het elektronisch diertje contact kan leggen met de elektroden van de acculader.



- Microprocessor besturing** In het laatste project leren wij u hoe u uw robot kunt besturen met een microprocessor. Klinkt verschrikkelijk ingewikkeld, maar dat valt best wel mee. De elektronica zit namelijk al op een printje dat kant-en-klaar door de fabrikant wordt geleverd. Aansluiten op de rest van de elektronica en programmeren maar!
- Vervolgens?** Vervolgens niets meer. Of... héél veel! Wie de in het kort besproken hoofdstukken doorneemt en aan het experimenteren slaat zal vaststellen dat in feite alleen de grenzen van de eigen verbeelding bepalen of er een vervolg komt op óns verhaal.
- Follow up op internet** Op de internetsite van de uitgever is een extra pagina geopend voor dit robotproject. Iedereen die wat te melden heeft, nieuwe prototypes heeft ontworpen of nieuwe elektronica heeft bedacht voor het tot leven wekken van dit beestje kan hierop terecht.  
De URL is: [www.vego.nl/hobby/robot/robot.htm](http://www.vego.nl/hobby/robot/robot.htm)  
Wij zijn zéér benieuwd!

## 2 De mechanische constructie

### Inleiding

#### Het loopwerk als basis

De basis van iedere robot is de mechanische constructie oftewel het loopwerk. Dit loopwerk bestaat uit het chassis, de wielen, de aandrijving, de voeding en de mechanica die nodig is om de robot te kunnen besturen. Zo'n loopwerk kan in principe heel eenvoudig zijn, maar natuurlijk moet men wél over de nodige vaardigheden beschikken om de bouw tot een goed einde te brengen.

#### Een uitdaging

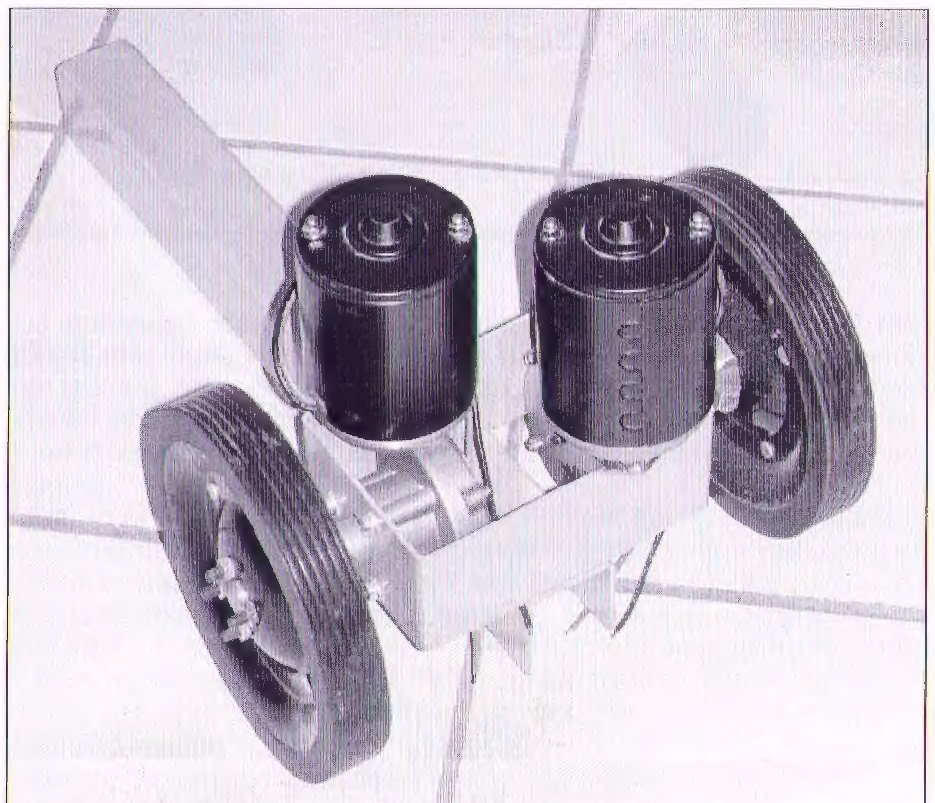
In ieder geval is het maken van een robot loopwerk voor iedere fijnmechanische knutselaar een échte uitdaging! Zeer handig en eigenlijk onontbeerlijk is het bezit van een aantal catalogi van gespecialiseerde leveranciers. Of het nu aankomt op de keuze van motoren, wielen, tandwielen en constructiematerialen, zonder uitgebreide documentatie over wat allemaal leverbaar is, komt men niet ver. Een van de snelle leveranciers is Conrad uit Enschede, die in haar algemene elektronica catalogus een dik hoofdstuk inruimt voor onderdelen die men nodig heeft als men modellen bouwt. Zie voor meer informatie [www.conrad.nl](http://www.conrad.nl). Ook de speciale internetsite [robot.pagina.nl](http://robot.pagina.nl) is een bron van informatie.

### Model 1: een zéér groot loopwerk

#### Ruitenwisser motoren

Er bestaat niet één robot, maar ieder ontwerp kan aangepast worden aan de individuele wensen en voorkeuren. Als eerste model van een loopwerk wordt een vrij grote uitvoering getoond. Hoe het loopwerk van deze robot er dus uit kán zien is geschetst in onderstaande figuur. Op de foto zijn twee elektromotoren prominent aanwezig, voorzien van een tandwielkast die een behoorlijke vertraging oplevert.

*Een uit de kluiten gewassen loopwerk voor de robot*



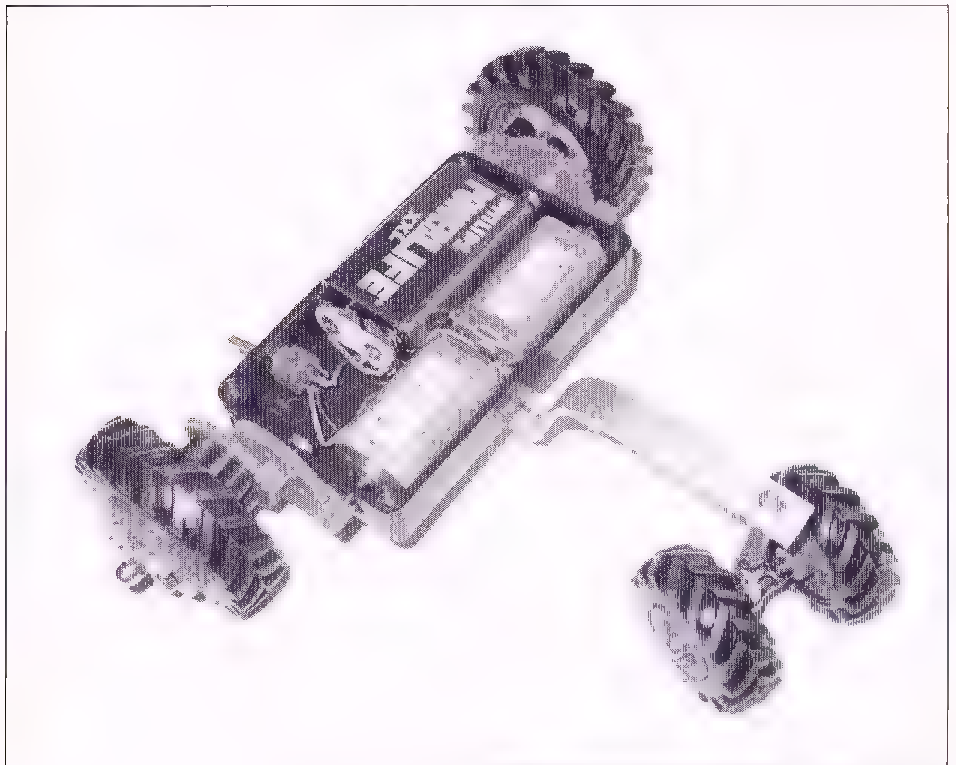
Op de uitgaande as is een wiel vast gemonteerd. De motoren zijn in dit geval (sloop) ruitenwischer motoren van een flinke personenauto en de wielen hebben een diameter van 18 cm. Dit voortbewegingsmechanisme vraagt een behoorlijke stroom en heeft een flinke 12 V accu nodig. Toch geeft dit model het principe van het loopwerk uitstekend weer.

## Model 2: een kleiner model

### Terugschalen naar mA-bereik

Om kosten, stroom en gewicht te drukken en de hanteerbaarheid te vergroten kan men het loopwerk terugschalen naar het mA-bereik. Dit model wordt wat uitgebreider beschreven en vormt de basis van de elektronische besturing, die in de volgende hoofdstukken aan de orde komt. Twee van de allergeedkoopste 9 V speelgoedmotortjes, plastic tandwielletjes, een handjevol M3 boutjes, moertjes en busjes, een 9 V blokcel en tenslotte een passend kastje is alles dat nodig is. Alle onderdelen zijn in de reguliere elektronica-handel te krijgen voor een zeer bescheiden bedrag. Als men ook nog de moeite neemt bij de speelgoedwinkel langs te gaan voor een paar knappe wielletjes, mag het resultaat er zijn.

*Een tweede uitvoering van een universeel loopwerk, maar nu uitgevoerd met kleine speelgoedmotortjes*



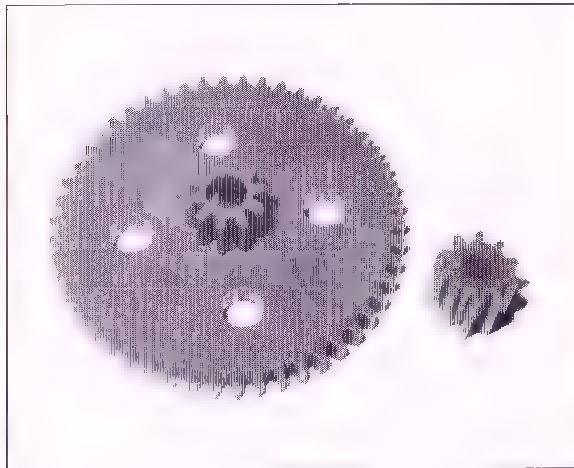
### De vertraging voor de motoren

De meeste standaard motoren hebben een veel te hoog toerental. Motortjes mét ingebouwde vertraging zijn heel bruikbaar en in vele soorten te koop, maar ze zijn behoorlijk duur. Mocht de prijs geen bezwaar zijn en fijnmechanisch knutselen niet uw sterkste kant is, dan gaat natuurlijk de voorkeur naar uit naar een exemplaar mét ingebouwde vertraging.

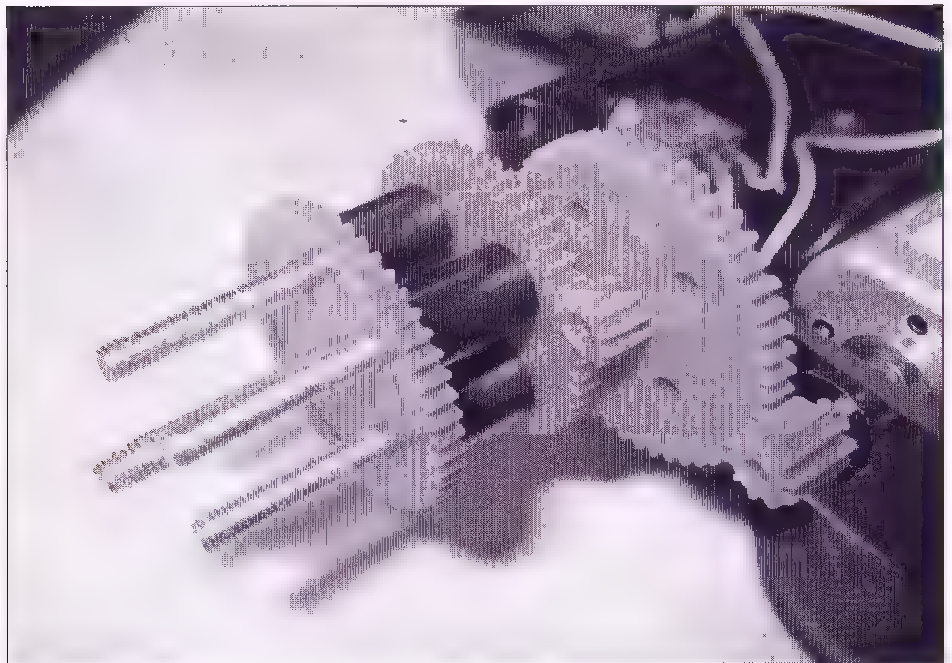
Maar in principe is het niet zo moeilijk om zélf een vertragingkast te construeren. Let er bij de keuze van een motortje zonder vertraging op dat de as een diameter heeft van 2 mm. Dat is namelijk de maat van de kleine plastic tandwielletjes die in het hobbycircuit te koop zijn.

Het kleine tandwiel kan dan met gepaste kracht voorzichtig op de as van de motor worden geduwd. Het grote tandwiel heeft een boring van 3 mm. Let goed op de tandwielconfiguratie op de foto van de volgende pagina! Deze geeft een mooi voorbeeld van een goed geconstrueerde, maar eenvoudig in elkaar te knutselen vertraging.

**Dergelijke kleine plastic tandwielletjes worden toegepast voor het maken van een vertragingskast**



**Een voorbeeld van een zélf in elkaar geknutselde vertragingskast**



**Stabiliseren van de wielen**

Omdat de wielen op schroefdraad gelagerd zijn, hebben ze bij één van de draairichtingen de neiging naar buiten te bewegen. Omdat het eerste vertragungswiel opgesloten is door het tandwiel op de wielas, is dit op zich geen probleem. Het kan echter gebeuren dat de tanden van de twee grote tandwielen gaan happen. Verbuig de wielas dan een beetje zodat de tanden van het tandwiel op de wielas het eerste vertragungswiel raken vlakbij het kleine tandwiel op dat eerste vertragungswiel. In dat geval kunnen de tanden nooit met elkaar in aanraking komen. Het gaat om enkele tienden van millimeters.

**De montage van motoren en vertragung**

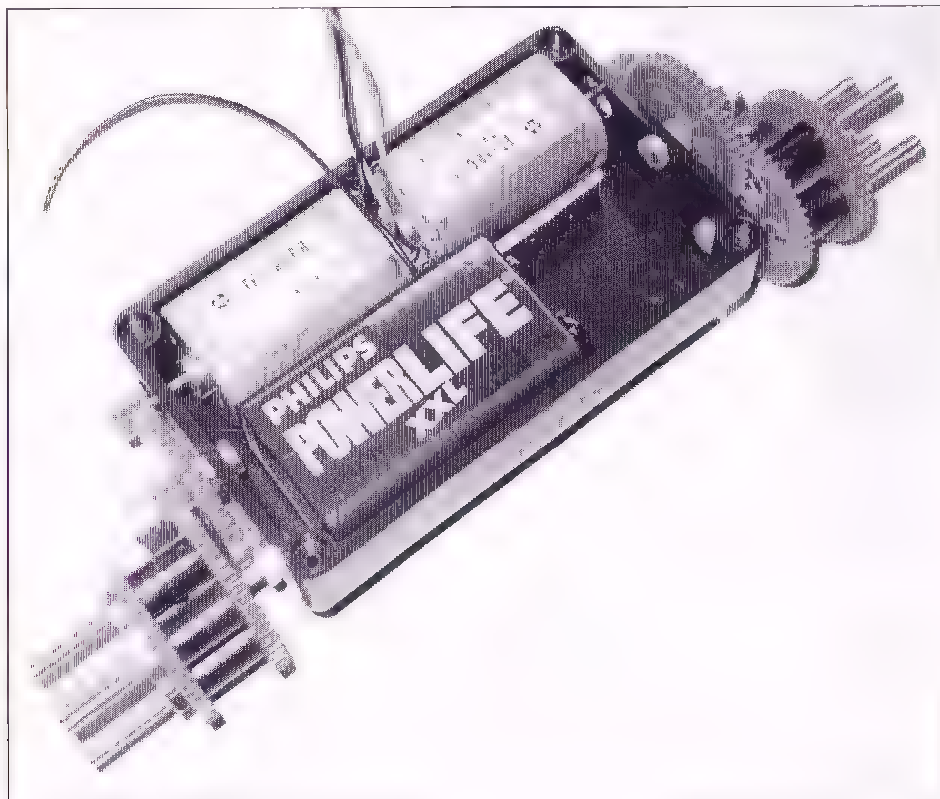
De motoren zijn met dubbelzijdig schuimtape in de behuizing gefixeerd. Het eerste vertragungstandwiel draait los over een M3 boutje. De schroefdraad geeft weliswaar niet de meest optimale lagering, maar het apparaat hoeft niet naar Rome. De positie van het boorgat in de behuizing voor de vertragungstandwielen is van groot belang. Om zeker te weten dat het op de goede plaats uitkomt, kan men een tandwiel als boormaal gebruiken. Zorg ervoor dat het tandwiel goed tegen het huis aanligt en bovendien spelingvrij ingrijpt in het kleine wiel. Boor nu via het gat van het tandwiel het gat in de behuizing. Steek een M3 boutje van binnenuit door het gat, zet dit aan de buitenzijde goed vast met een moertje en schuif het tandwiel erop. Dat hoort nu soepel te lopen. Mocht dat niet het geval zijn, boor het gat in de behuizing dan op

tot 3,5 mm. De bout heeft nu een beetje ruimte en kan tijdens het vastzetten geïdentificeerd worden. Hetzelfde gebeurt met het tweede wiel. Mik het zo uit, dat de boutkoppen in de behuizing geen hinder ondervinden van eventueel aanwezige verstijvinggrillen. Zoals op de foto is te zien, is het eerste tandwiel iets hoger en het tweede tandwiel iets lager gemonteerd. In een wat ruimer kastje is dat allemaal minder kritisch. Het tweede tandwiel draait eveneens op een M3 bout die aan de kast is vastgezet. Deze bout moet langer zijn. Op de foto is te zien dat het tandwiel op deze as met een tweede tandwiel gekoppeld is met afstandsbussen, om zo een dubbel lager te vormen. Op de vier bouten wordt straks een passend wiel gemonteerd. Let erop dat de vier boutkoppen in het tandwiel verzonken zijn.

### Het loopwerk is klaar

Dat is in feite alles wat nodig is om een goed werkend loopwerk voor een robot te maken. Het volledig loopwerk is voorgesteld in onderstaande foto.

### *Het gemonteerde loopwerk van het tweede model*



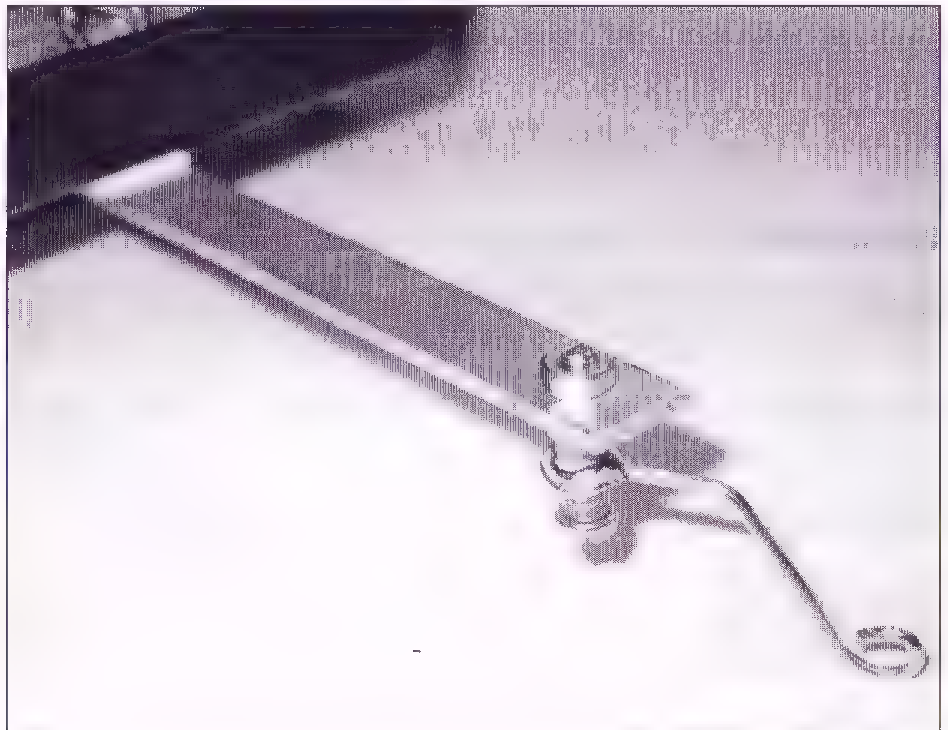
### Het staartmechanisme

Zoals het er nu uitziet, sleept het loopwerk over de grond. Er zijn immers alleen maar twee voorwielen aanwezig! Men kan een heel eenvoudig staartmechanisme maken dat alle bewegingen van het voertuig moeiteloos kan volgen. Hiervoor wordt tegen de bodem van het bakje een strip vastgezet met twee boutjes met verzonken kop aan de binnenzijde van de kastbodem. Hierdoor blijft het bakje aan de binnenzijde glad. Aan het eind van de strip komt een boutje waarover een messing M3 busje is geschoven. Hieraan is een sleepstangetje gesoldeerd. Een moertje onder de strip (niet zichtbaar op de foto op de volgende pagina) wordt zodanig aangedraaid dat het busje nog juist goed kan ronddraaien. Aan de andere kant van de strip komt een tweede boutje dat het geheel opsluit.

### **De wielen van de robot**

De sleephaak heeft aan de voet een oogje. Mocht het allemaal wat mooier moeten, dan kan men een écht staartwiel bouwen. Wielen zijn natuurlijk zonder meer noodzakelijk! En daarvoor moet men in de speelgoedwinkel zijn.

***Een mogelijke uitvoering van een "sleephaak"***

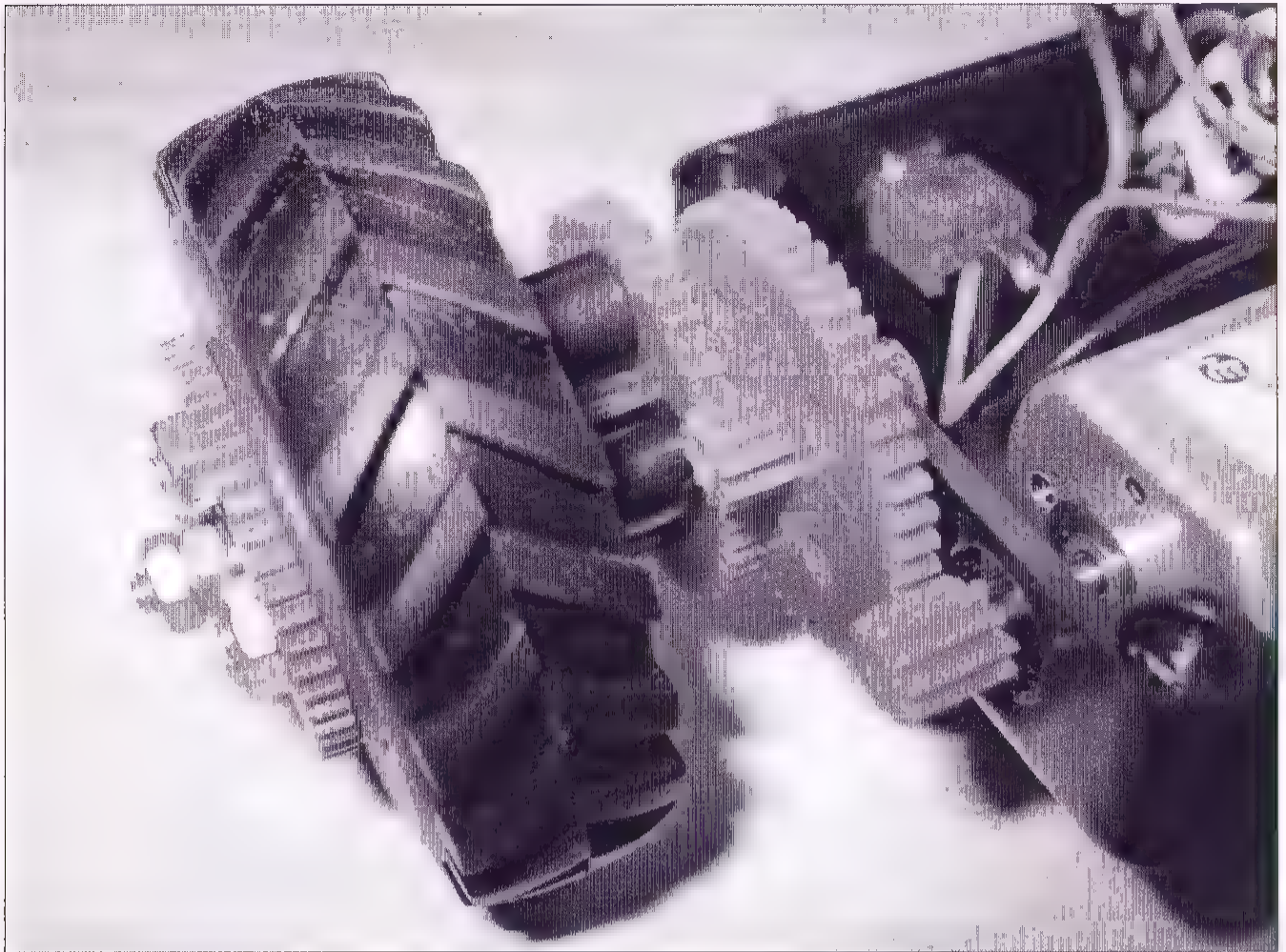


Op de afdeling voertuigen zijn goedkope plastic landbouwtrekkertjes met zeer realistisch uitgevoerde wielen te koop. Zoek een plastic trekker uit met wielen van bruikbare afmetingen (kost EUR 6,00 of daaromtrent). Een bruikbaar wiel voor het loopwerk is voorgesteld in onderstaande foto.

***Een wiel dat van een goedkope plastic trekker is gesloopt is érg bruikbaar voor dit robotmodel***



Er bestaan ook prachtige metalen modellen die ook plastic wielen hebben, maar dat is eigenlijk jammer. Bovendien kosten die een veelvoud van de plastic uitvoering. Boor de asgaten op tot 3 mm. De grote wielen worden op de hoofdassen gemonteerd. Het kan zijn dat hiervoor eventueel aanwezige verstijvingribben in de wieltjes verwijderd moeten worden. Zoals de foto op de volgende pagina laat zien, worden de hoofdwielen ingeklemd tussen de afstandsbussen en het buitenste tandwiel. De asbout steekt door het centrale gat en wordt voorzien van twee tegen elkaar vastgezette moertjes die voorkomen dat het wiel er af loopt. Later is het de bedoeling dat de moertjes met wat nagellak of borgmiddel worden beschermd tegen lostrillen. Maar dat is eerst aan de orde als alles goed werkt!

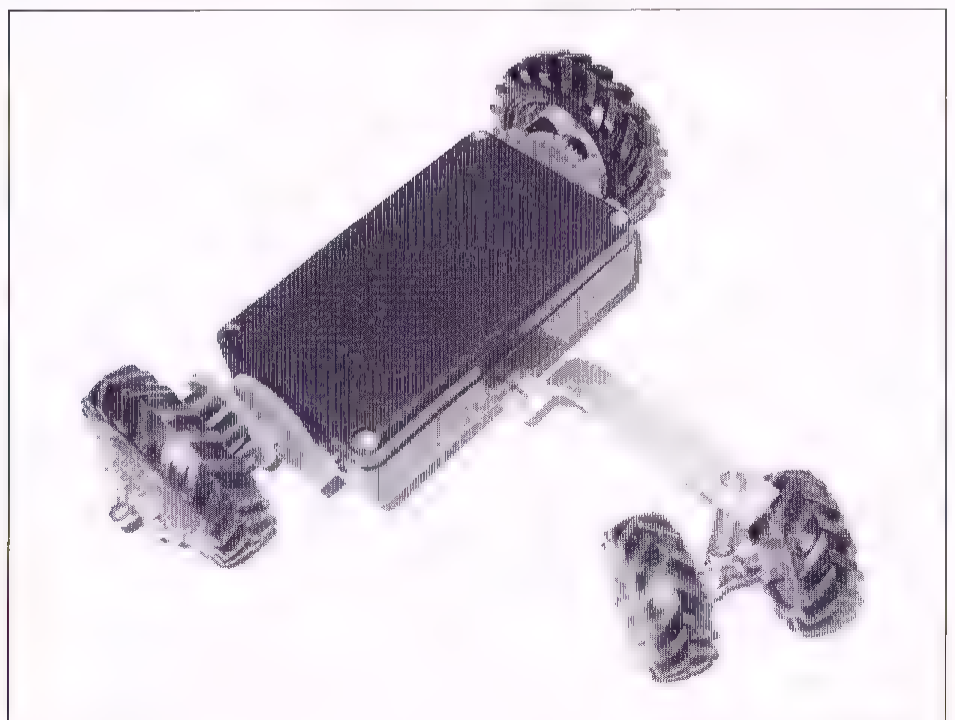


***Het loopwerk wordt voorzien van wieltjes, afkomstig van landbouwtrekkertjes***

De praktijk leert dat een en ander tijdens de bouw toch weer af en toe even los moet. Een borgmoertje is ook een goede oplossing.

Op onderstaande foto is een dubbel staartwiel afgebeeld. Een miniatuur landbouwtrekker heeft twee grote en twee kleine wielen. De laatste twee vormen een prachtige staartsectie. Het staartwiel lijkt op een constructie die bekend is van opleggers. Twee messing busjes zijn haaks op elkaar gesoldeerd.

***Het tweede model is klaar voor de elektronica***



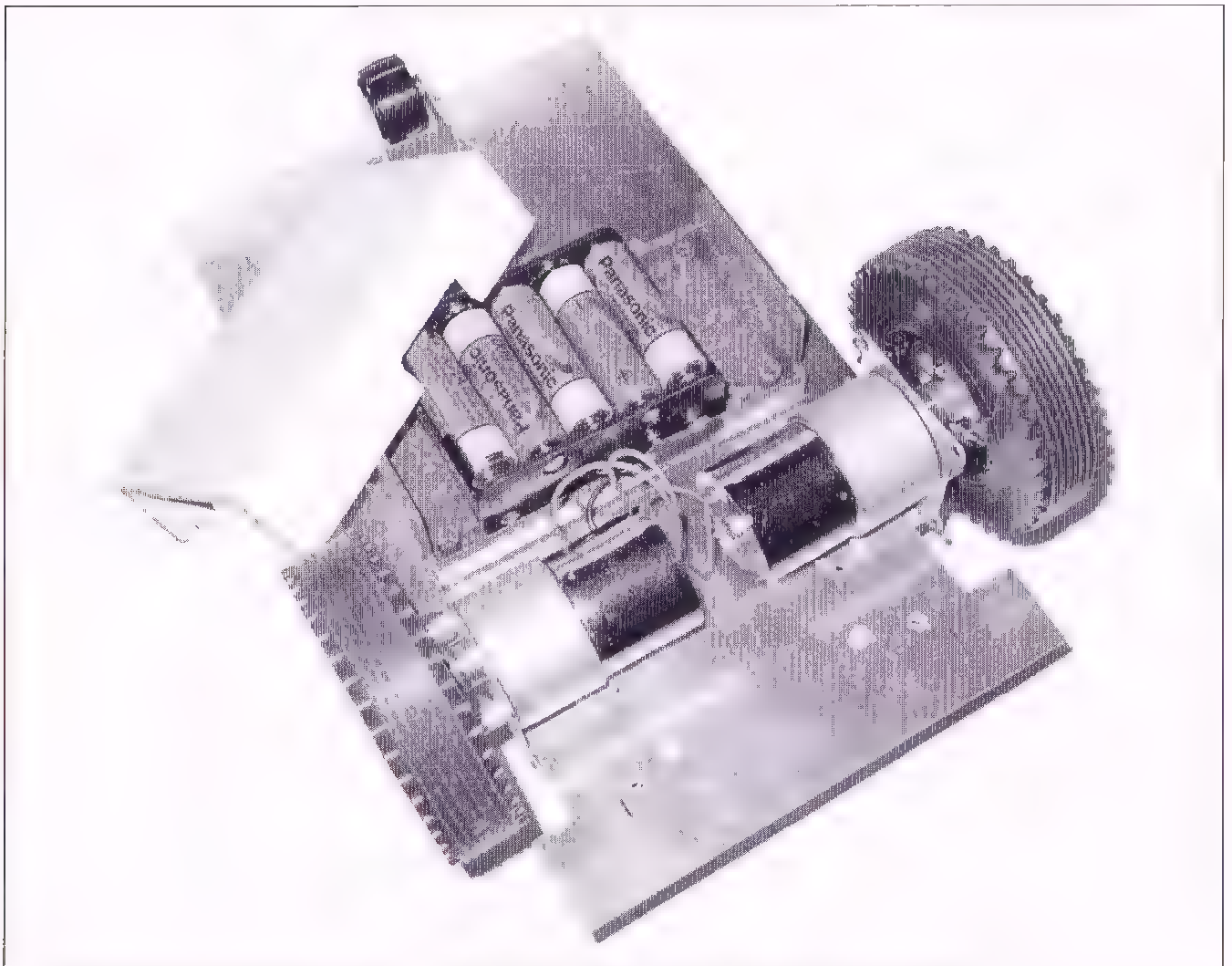
Om die werkzaamheid te vereenvoudigen, kan men met een stukje koperdraad een "broekenmannetje" aanleggen zoals betonvlechters dat doen. Een lange M3 bout gaat door de wielassen en de horizontale bus. De verticale bout wordt met twee moertjes, één aan beide zijden van de strip, zodanig afgesteld dat het geheel vrij kan draaien zonder al te veel speling. De staartwielen volgen moeiteloos elke beweging, ook achteruit rijden vormt geen enkel probleem. Deksel erop en klaar is het loopwerk!

### Model 3: epoxy printplaat als basis

#### Een loopwerk van printplaat

Wie het werken met metaal niet zo ziet zitten kan ook dit derde model nabouwen, waarbij het "chassis" uit niets meer bestaat dan een plaatje epoxy printplaat! Voor dit project wordt een speciaal chassis gemaakt op basis van gewone enkelzijdige printplaat. Op deze plaat worden alle onderdelen vastgeschroefd. Om het nog gemakkelijker te maken wordt bij dit derde model uitgegaan van motoren met een aangebouwde vertraging. Afhankelijk van de grootte van de wielen kiest men een toerental in de buurt van 59 omwentelingen per minuut bij kleine wielen of 31 omwentelingen per minuut als de wielen iets groter zijn uitgevalen. Een snelheid tussen 10 en 20 centimeter per seconde is heel geschikt.

Hoe dit model er na een paar uurtjes knutselen uitziet blijkt uit de onderstaande foto.

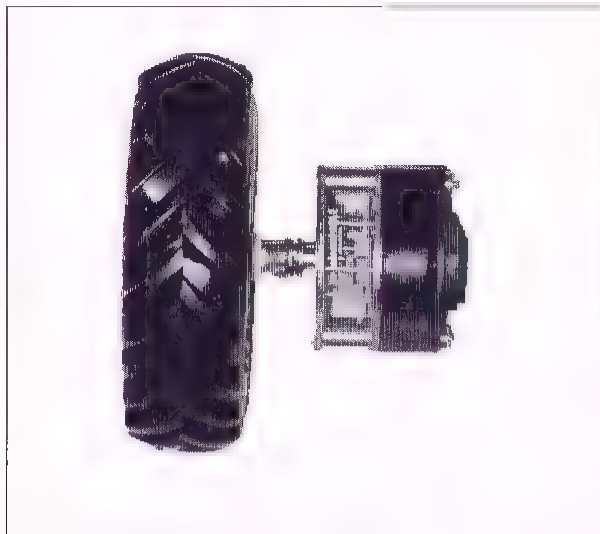


**Het derde model is klaar voor de elektronica**

Hoe de grote aandrijfwielen aan de as van de motor worden bevestigd blijkt uit de opname op de volgende pagina.



*Het bevestigen van de grote aandrijfwielen op de as van de motor met ingebouwde vertraging*



**De voeding**

Aan de onderzijde vinden de NiCad accu's een plaats. Een houder voor 10 stuks is in een montagekastje geplaatst. Aan de bovenzijde plaatsen we ook een wat grotere montagekast voor de besturingselektronica.

**Model 4: aluminium als basis**

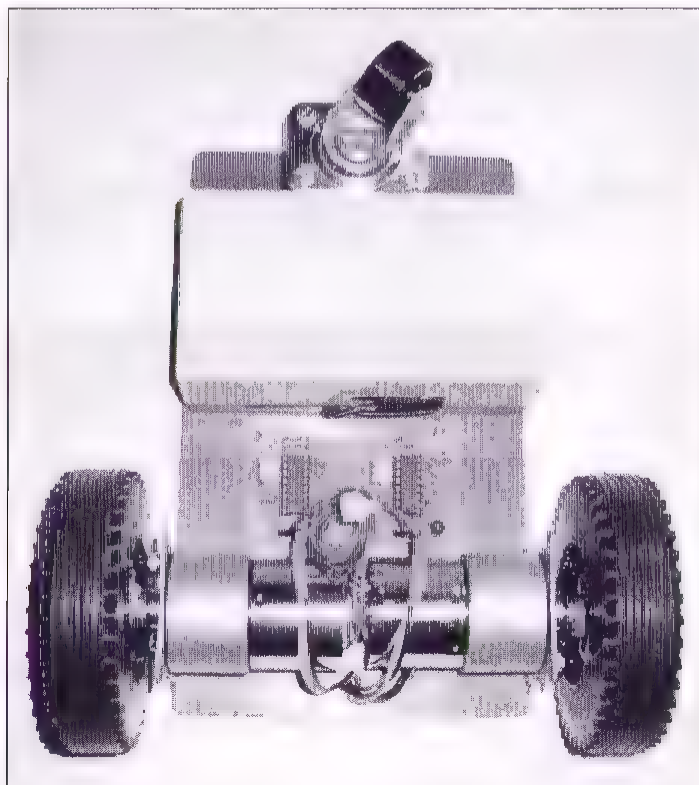
**Aluminium is een érg geschikt metaal**

Aluminium is niet alleen lichter dan ijzer, maar bovendien beter te bewerken. Zo kan aluminium plaat bijvoorbeeld gemakkelijk worden gesneden door de plaat aan weerszijden met een scherp hobbymes in te kerven en nadien voorzichtig te breken door de plaat op de rand van een tafel te leggen en het overhangende stuk voorzichtig heen en weer te buigen. In iedere doe het zelf zaak treft men aluminium platen in diverse maten aan.

**De constructie van het chassis**

Alle onderdelen kunnen, net zoals bij het epoxy-model, op de aluminium basisplaat worden gemonteerd.

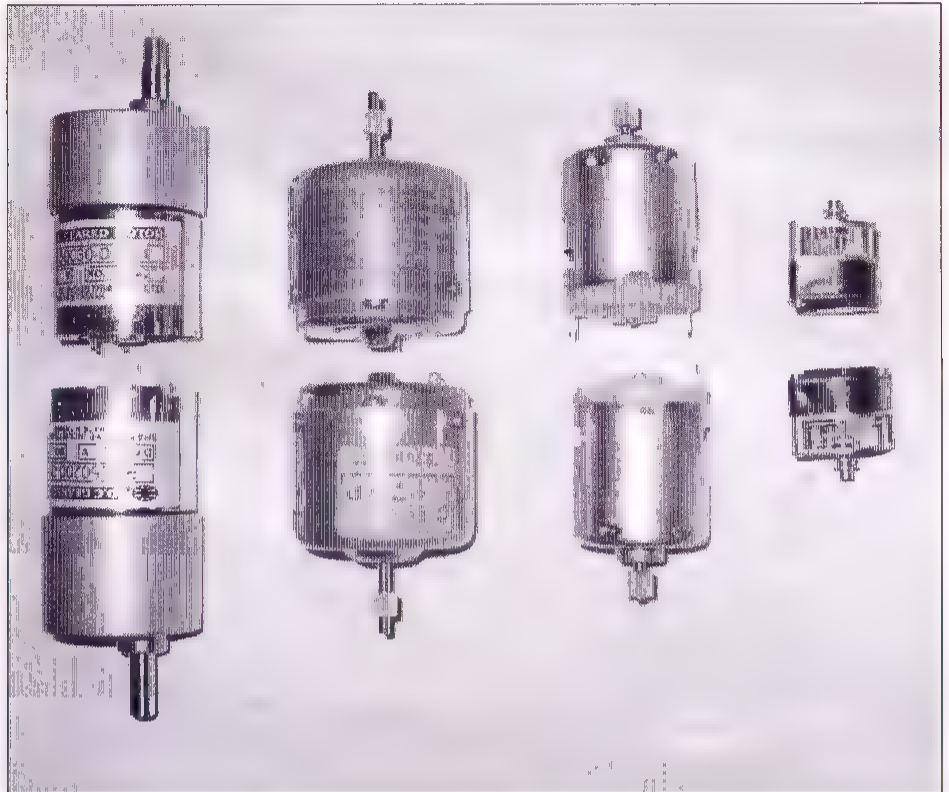
*Een chassis op basis van aluminium plaat*



## Alternatieven

Natuurlijk zijn er vele alternatieve mogelijkheden, afhankelijk van budget en verkrijgbaarheid. De onderstaande foto toont diverse motoren met en zonder vertraging in verschillende prijsklassen. Een bezoekje aan de elektronica shop of de modelbouw winkel zal verhelderend werken.

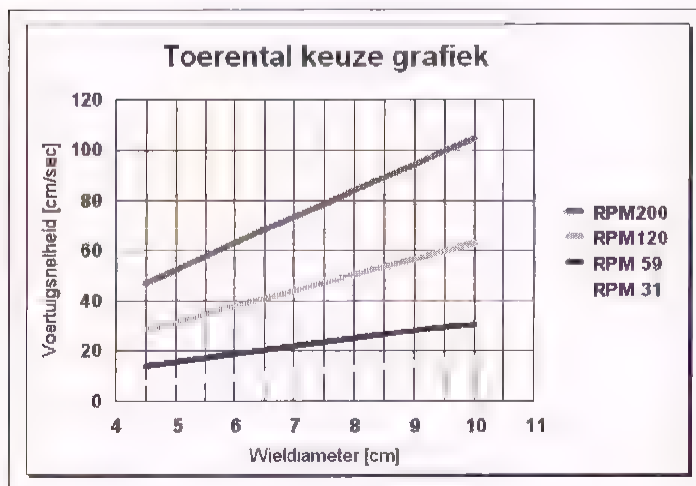
*Diverse motoren in verschillende prijsklassen, die bruikbaar zijn voor de bouw van een robot loopwerk*



### Snelheid versus toerental

Ook wat betreft de uiteindelijke "rijnsnelheid" van het model heeft men alle persoonlijke vrijheid. Het zal duidelijk zijn dat er een wiskundig verband bestaat tussen het toerental waarmee de wielen worden aangedreven, de diameter van de wielen en de robotsnelheid. In onderstaande grafiek is dit verband aangegeven voor vier verschillende toerentalen. Let op! Dit zijn natuurlijk niet de toerentalen van de motorassen, maar deze van de aandrijfwielen. Alleen als motoren mét ingebouwde vertraging worden toegepast waarvan de assen rechtstreeks op de aandrijfwielen zijn bevestigd, is de motorsnelheid gelijk aan het toerental van de aandrijfwielen. Het begrip "RPM" staat voor "omwentelingen per minuut".

*Het verband tussen de omwentelingssnelheid van de aandrijfwielen, de diameter van deze wielen en de snelheid van de robot*



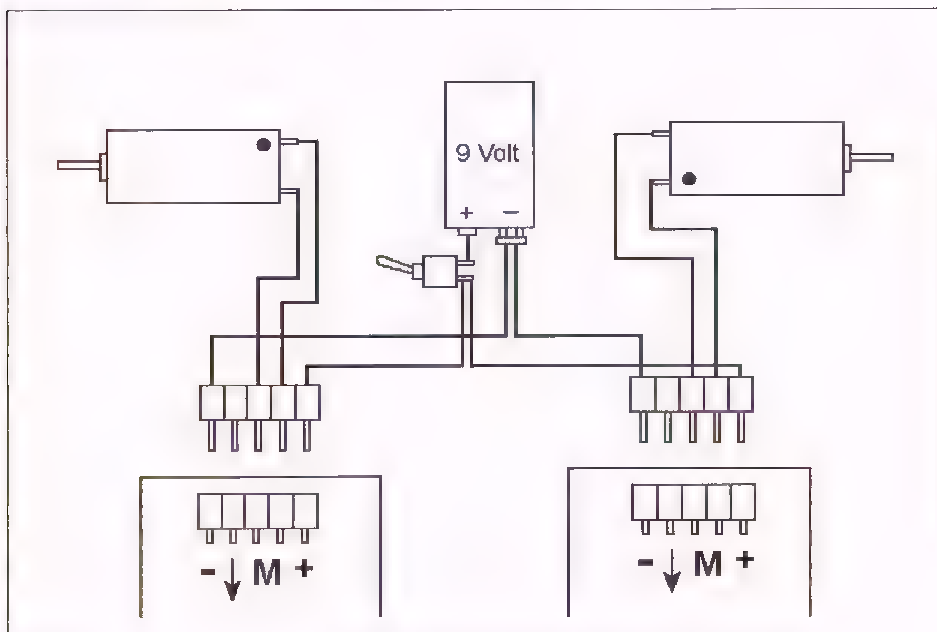
Hogere toerentallen zijn geschikt voor race-robots en grote ruimten. Ze genieten bepaald niet de voorkeur op een studentenkamer. Heel geschikte motortjes zijn te vinden in de diverse modelbouw/elektronica zaken.

## De bedrading in het loopwerk

### Afzonderlijke bedrading

De motoren worden afzonderlijk bedraad. Uit het mechanische deel komen dus twee maal twee motordraden en twee batterijdraden. Om straks de aansluiting op de elektronica simpel te houden, kan gebruik worden gemaakt van een vijfpolige aansluitplug waar de draden al aan zitten, zie onderstaande schets. De plug zélf wordt straks in het bijbehorende contact op de elektronicaprint gestoken. De losse draaduiteinden worden direct aan de motoren en de batterij gesoldeerd. Let wel, per motor wordt zo'n vijfpolige aansluiting gebruikt. De batterij wordt op beide kabels aangesloten. Op elke connector blijft één draad ongebruikt. Dat wordt straks de signaalingang. Het is handig de batterij aan te sluiten via een batterijclip. Nog handiger is het daarnaast ook een schakelaar op te nemen in een voedingsleiding van de batterij. De schakelaar vindt een plaats in de behuizing, zodanig dat hij aan de buitenzijde bediend kan worden.

### De fundamentele bedrading in het loopwerk



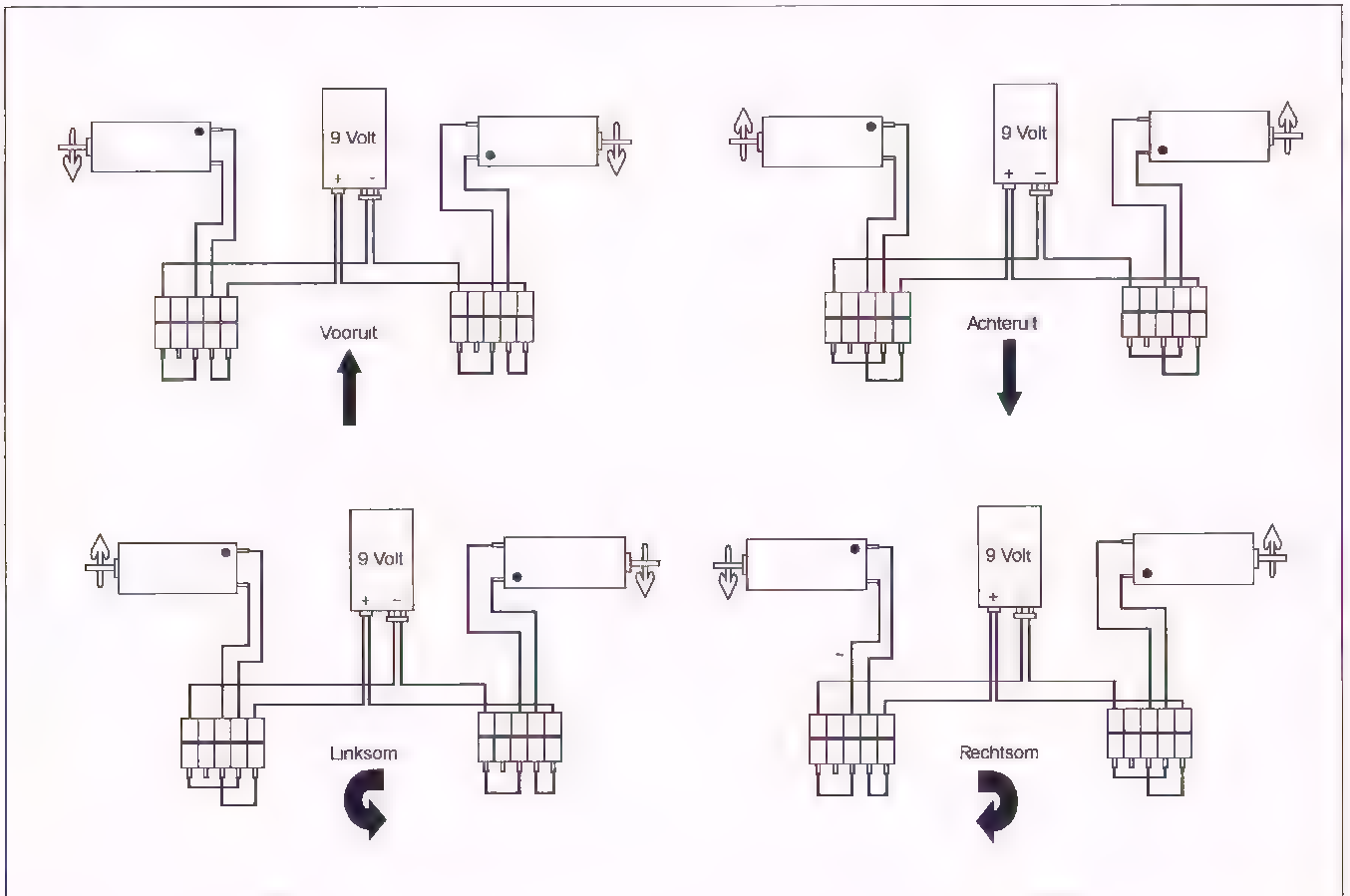
De stekers gaan maar op één manier in de op de printjes gesoldeerde connectoren. Let er goed op dat ze in de juiste stand gemonteerd worden. Mocht er geen gebruik gemaakt worden van connectoren maar van vaste bedrading, dan kan dat natuurlijk ook.

### De rijrichtingen

Het schema op de volgende pagina geeft een overzicht van de vier mogelijke rijrichtingen en de manier waarop de twee motoren hiervoor worden gestuurd.

### Slotopmerking

De overbrengingen, wieldiameter, motortoerental en voedingsspanning zijn allemaal factoren die de beweging beïnvloeden. Kleinere wielen maken het platform minder snel maar krachtiger. Mocht de snelheid veel te hoog zijn, kan eventueel een extra vertragingstandwiel gemonteerd worden. Bij tegengestelde motor draairichting maakt het loopwerk een fraaie pirouette. Als één van de motoren stil staat, draait het loopwerk om het stilstaande wiel. Het geheel mag best wel pittig reageren.



***De vier bewegingsrichtingen van de motoren***

In een volgend hoofdstuk komen alle mogelijke basisbewegingen aan de orde. Maar pas als het mechanische deel van het loopwerk naar behoren werkt, wordt het tijd om de soldeerbout warm te stoken!

**Epiloog**

Op het schema staat een deel van de omtrek van de besturingsprintjes. Dat maakt nieuwsgierig. In het volgende hoofdstuk worden de motoren elektronisch bestuurd.

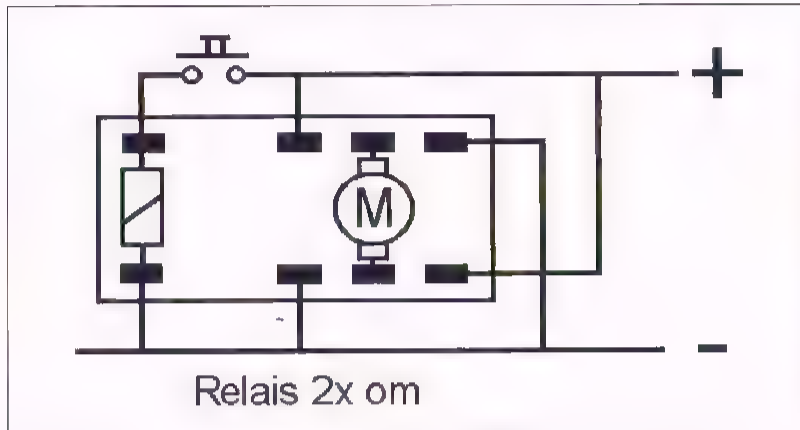
# 3 De motorbesturing

## Inleiding

**Werken met een relais?**

Met een relais is op eenvoudige wijze de draairichting van een gelijkstroom elektromotor om te keren. Daarvoor zijn twee omschakelcontacten nodig. De onderstaande tekening toont de aansluitgegevens aan de onderzijde van zo'n relais.

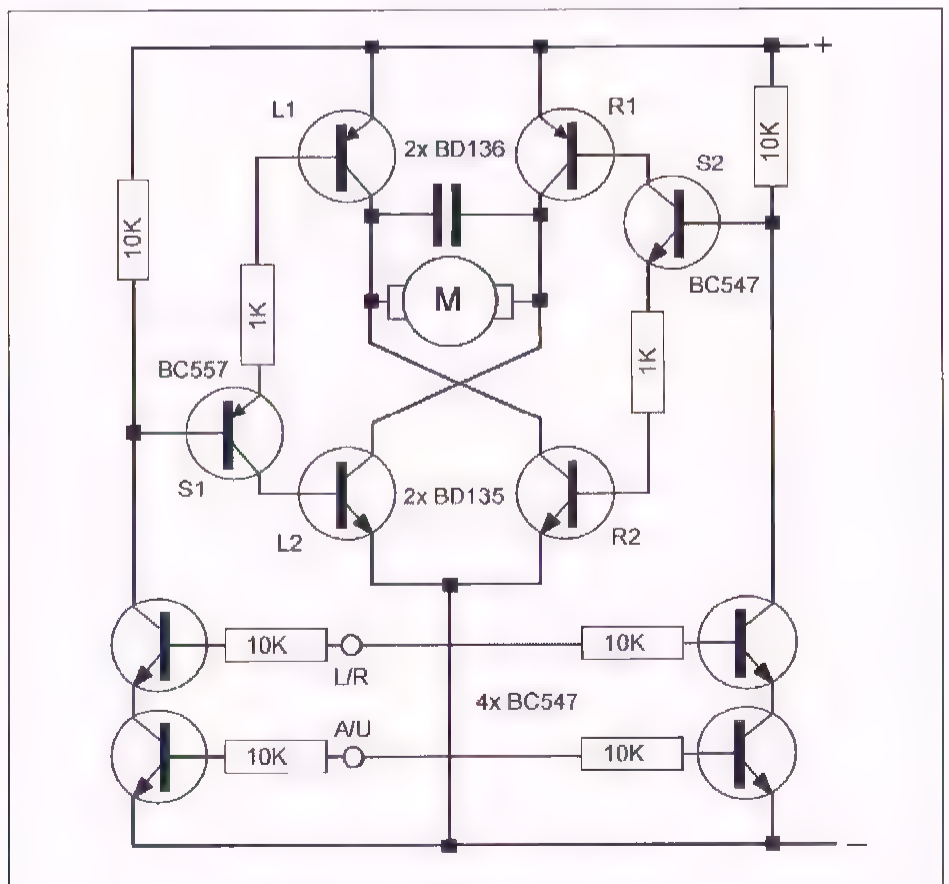
*Het besturen van de robotmotoren met relais*



**Brugschakeling met transistoren**

Toch is dit natuurlijk een beetje ouderwets! Een goed alternatief is een brugschakeling met transistoren, zoals voorgesteld in onderstaand schema. Stel men legt de links/rechts-stuuring L/R aan massa en de aan/uit ingang A/U aan de plus. Stuurtransistor S2 gaat dan geleiden en S1 blokkeert. L1 en L2 blijven dicht en R1 en R2 gaan open. De motor draait rechtsom. Omgekeerd, als L/R aan de plus ligt, stuurt S1 open en geleiden L1 en L2. De draairichting van de motor keert om.

*Het besturen van de motoren met een brugschakeling met transistoren*



**Gevaarlijke schakeling**

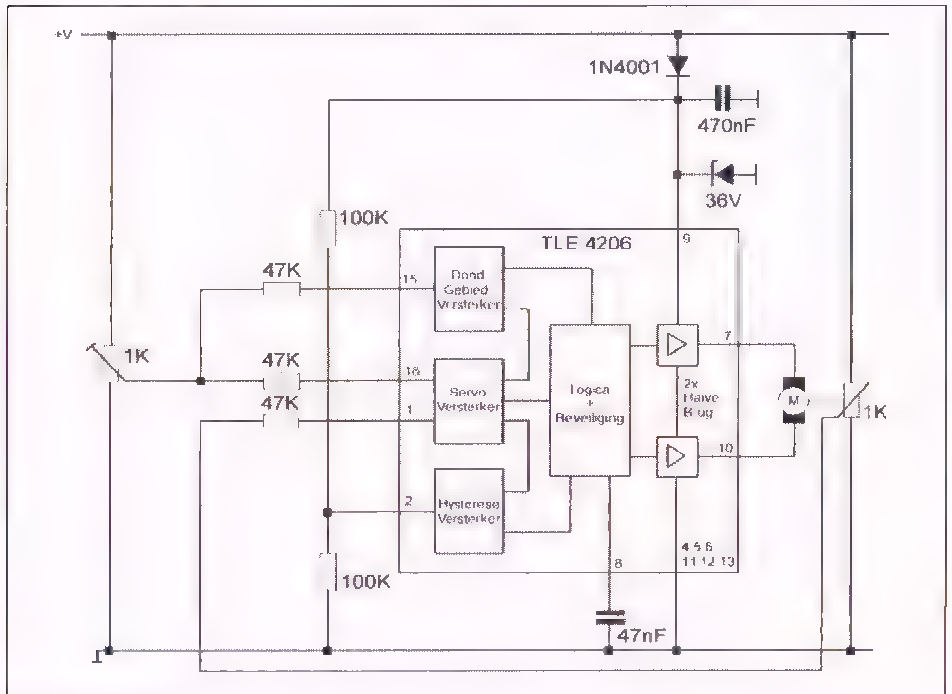
Stel dat onder deze conditie een klein stoorpulsje (bijvoorbeeld van de motor) op de basis van S2 komt. Heel even staan dan alle transistoren open en zijn vervolgens allemaal stuk. Dit verschijnsel heet stroomdoorslag en dit soort H-bridgen is er berucht om. Om deze reden heeft de industrie een aantal IC's ontwikkeld, waarbij de getekende H-brug wél toegepast wordt, maar dan aan alle kanten beveiligd tegen stroomdoorslag. Omdat deze IC's over het algemeen ontwikkeld zijn voor industrieel gebruik, hangt er een prijskaartje aan dat voor hobby-gebruik minder interessant is.

**Besturen met de TLE4206**

Het zoeken naar een goedkoper alternatief leverde een IC op dat bedoeld is voor de hoogtevastelling van koplampen in auto's. Het bezit alle interne beveiligingen om stroomdoorslag te voorkomen, maar heeft een voor servomotoren typische configuratie met een potentiometer. De schakeling zorgt ervoor dat de stand van de regelknop op het dashboard (een potentiometer) vergeleken wordt met de stand van een potentiometer die aan de koplampunit is gekoppeld. De motor zorgt ervoor dat de standen overeenkomen.

Voor iets meer dan € 3,00 krijgt men de beschikking over dit zeer handige IC, de TLE4206, een 16 potig juweeltje dat er in zijn basischakeling uitziet als getekend in onderstaand blokschema.

**De basisschakeling rond de TLE4206**



Het IC wordt gefabriceerd door Siemens, een oerdegelijke en oeroude Duitse fabrikant die tegenwoordig blijkbaar als "Infineon" door het leven wenst te gaan.

De linker potentiometer zit in de auto, de rechter is mechanisch gekoppeld aan de lampverstellingsmotor. Om te voorkomen dat de motor om het nulpunt gaat jitteren, kan een dood gebied worden ingesteld (bovenste versterkerblok) met externe weerstanden. Datzelfde geldt voor de hysteresis (onderste versterkerblok). De servoversterker zorgt ervoor dat de motor probeert beide potentiometerwaarden gelijk te maken. De twee halve bruggen hebben de functie zoals in de eerder beschreven transistoruitvoering. De condensator aan pen 8 onderdrukt storingen die voortkomen uit stroompieken. In de voeding zit nog een ontstoorcondensator, een beveiliging tegen spanningspieken (zenerdioden) en een diode die beschermt tegen ompolen van de voedingsspanning. Voor de toepassing als robotbesturing zijn deze dioden niet

noodzakelijk. De linker potentiometer wordt ook niet gebruikt, zodat een stuursignaal rechtstreeks kan worden aangesloten op het punt waar nu nog de looper zit. De rechter potentiometer wordt vervangen door twee gelijke weerstanden die het nulpunt vastleggen.

**De TLE4206 in de robot**

De overgang van een bepaalde beweging naar een andere gaat erg abrupt. Dat gedrag past goed in een robot, maar de bewegingen zijn wel erg hoekig. De TLE4206 stuurt de motor aan door middel van puls-breedte modulatie. De aansturing zoals die in de volgende hoofdstukken wordt beschreven is digitaal. Er wordt daarbij alleen gebruikt gemaakt van de drie toestanden:

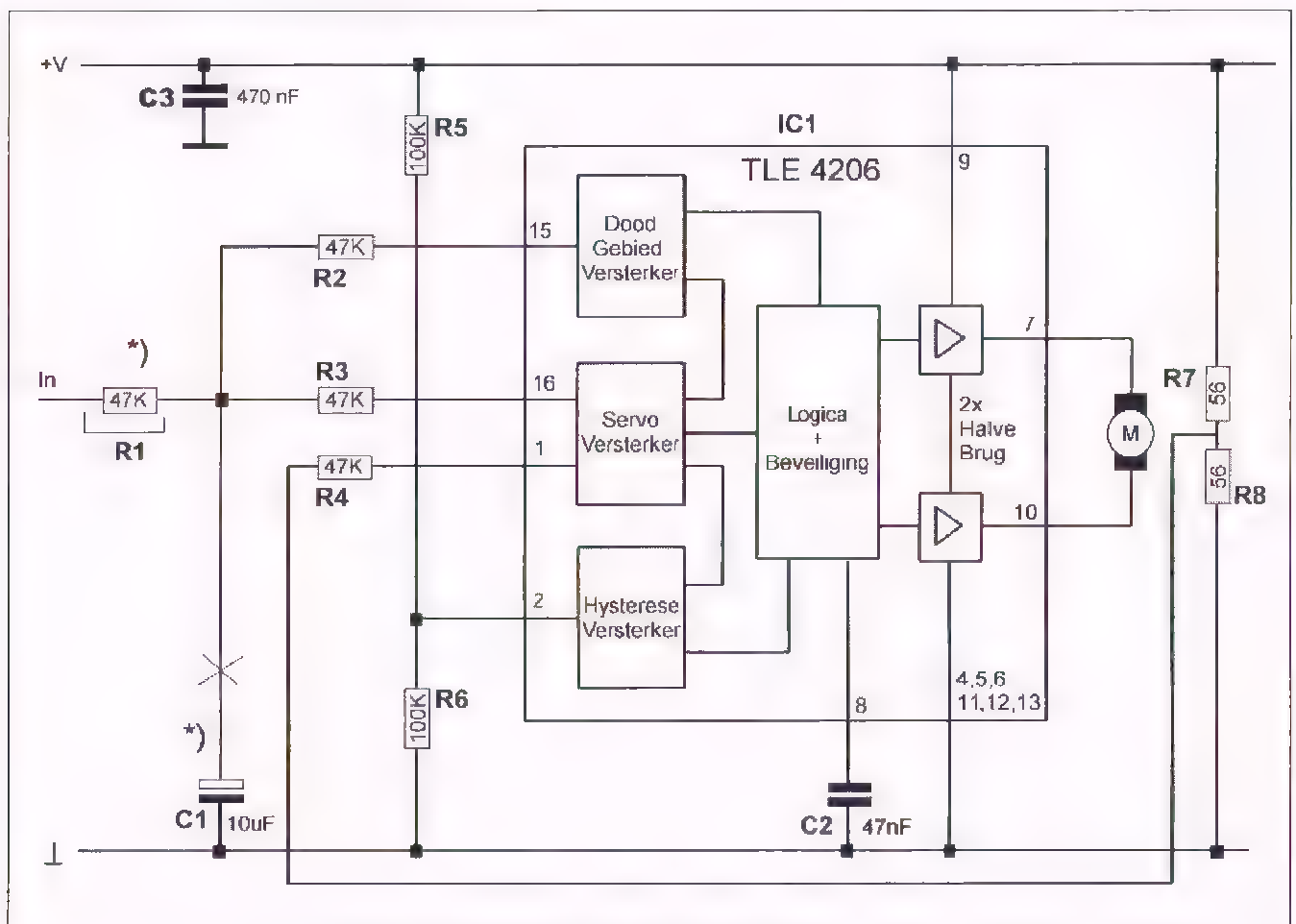
- motor vol vermogen linksom;
- motor stop;
- motor vol vermogen rechtsom.

Alle tussenliggende snelheden worden genegeerd. Toch kan men dat deel van het regelgebied gebruiken om de bewegingen van de robot wat meer af te ronden.

**De motorbesturing in de praktijk**

**Het volledig schema**

Het volledig schema van de besturing van de robotmotoren is getekend in onderstaand schema. Let wel, dit schema bestuurt één motor en wordt dus dubbel uitgevoerd.



**De praktische schakeling van de motorbesturing**

De reeds benoemde meer natuurlijke beweging wordt gerealiseerd met een weerstandje en een condensator als vertraging aan de ingang. Let wel, dit is een optie. Als de voorkeur uitgaat naar een Spartaans bewegende robot, dan wordt de condensator \*) weggelaten en de weerstand \*) door een draadbrug vervangen. Dat is ook de beste manier om te beginnen. De bewegingen zijn dan duidelijk gescheiden.

## ONDERDELENLIJST

### WEERSTANDEN, 1/4 W, 5 %

R1,R2,R3,R4	47 kΩ
R5,R6	100 kΩ
R7,R8	56 Ω

### CONDENSATOREN

C1	10 μF, 16 V printelco
C2	47 nF, MKH
C3	470 nF, MKH

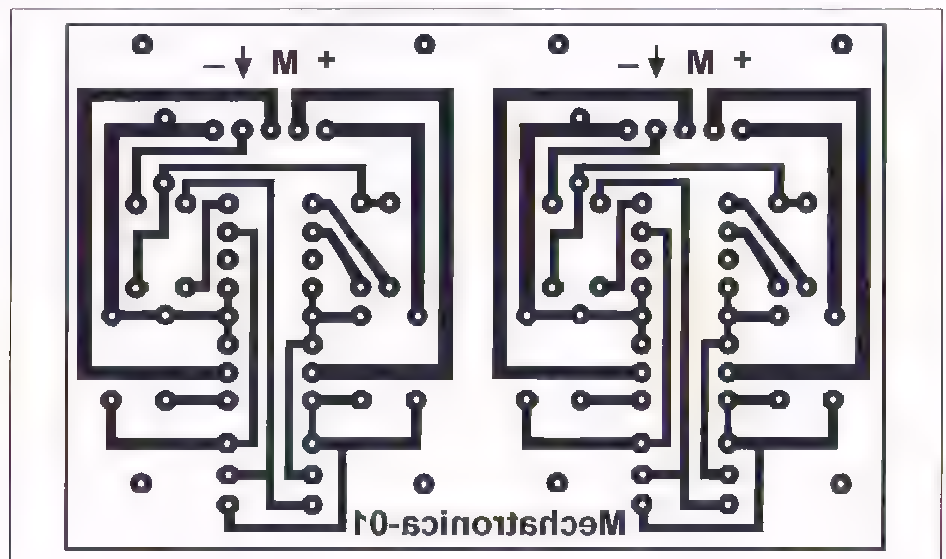
### HALFGELEIDERS

IC1	TLE4206
-----	---------

#### De bouw van de schakeling

De schakeling is zó eenvoudig dat het geheel eventueel ook op gaatjesboard betrouwbaar is te bouwen. Maar, in onderstaande figuur is een printontwerpje voorgesteld, waar alle onderdelen voor het besturen van de twee motoren een plaats vinden.

#### De print voor de motorbesturing



#### Let op

De print is hier niet op schaal 1/1 getekend.

**Over het maken van de printen uit dit werkboek treft u in het laatste hoofdstuk alle noodzakelijke informatie aan.**

#### Opmerkingen

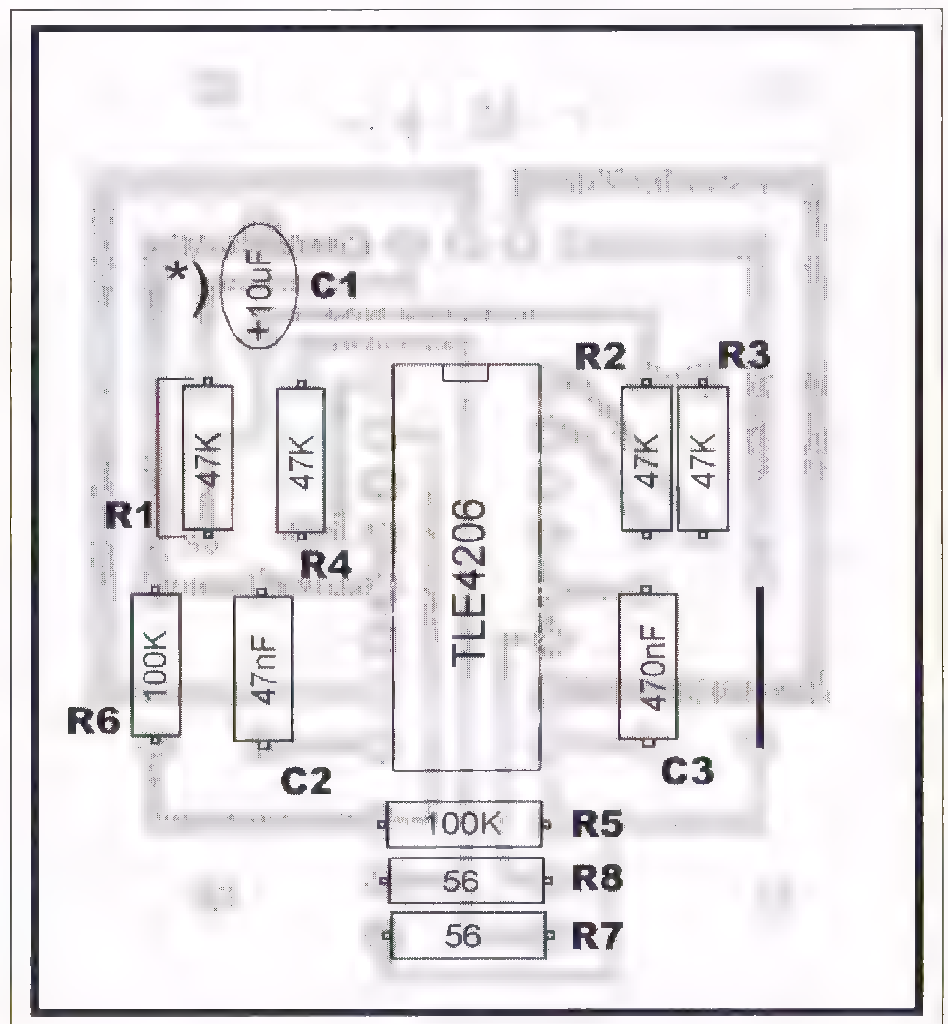
Het zal wel duidelijk zijn dat er twee van dergelijke printjes moeten worden gebouwd, voor iedere motor één. Maar, met deze printen zijn we er nog niet. De ingang heeft in feite drie hoofdcondities:

- Als hij (bijna) aan nul ligt draait de motor een bepaalde kant op.
- Als de ingang op de voedingsspanning ligt, draait de motor in de andere richting. Zorg ervoor dat deze richting overeenkomt met een achteruit rijdende beweging! Als dat niet het geval is moet de motor omgepoold worden.
- Op halve voeding staat hij stil.

De fabrikant heeft het IC zo uitgevoerd dat de nul iets boven massa moet liggen. Als het IC met logica wordt aangestuurd, wordt automatisch aan die voorwaarde voldaan. Als de sturing echt aan een harde massa ligt kan dat duiden op kortsluiting in de bedrading. Uit veiligheidsoverwegingen wordt de motor in dat geval niet aangestuurd.



**De componenten-  
opstelling van de  
motorbesturing**



**Epiloog**

In het volgende hoofdstuk wordt duidelijk hoe de motorsturing, die in feite tri-state is, toch met een digitaal signaal kan worden aangestuurd.

# 4 Onze robot leert dansen

## Inleiding

**Ter herinnering**

De motorbesturingsprint uit hoofdstuk 3 kent drie toestanden:

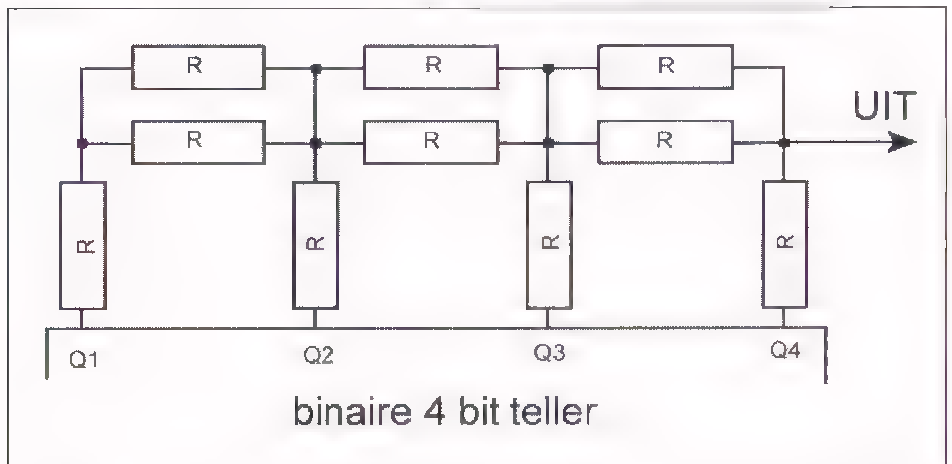
- stuurspanning 0 V: motor beweegt vooruit;
- stuurspanning 4,5 V: motor staat stil;
- stuurspanning 9 V: motor beweegt achteruit.

Het ligt voor de hand dat men digitale technieken kan gebruiken voor het aansturen van de twee motoren van de robot. Digitale schakelingen kennen echter maar twee condities, "L" of "H". Hierop zal een truc verzonnen moeten worden.

**Eenvoudige DAC**

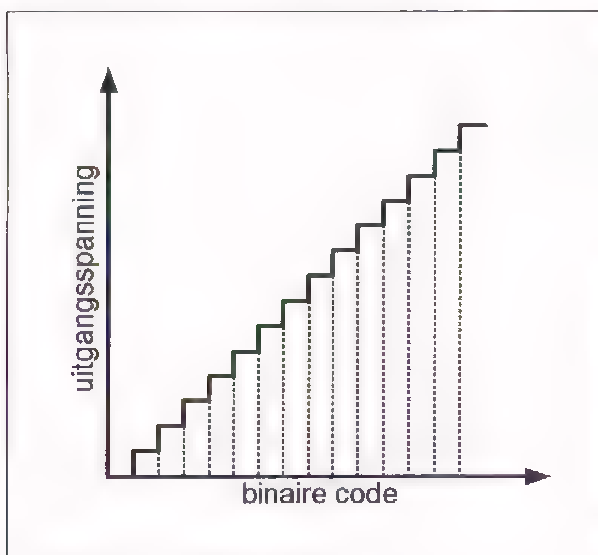
Door diverse digitale signalen aan een digitaal naar analoog omvormer (DAC) aan te bieden, kan men de "L" en "H" van de digitale elektronica omzetten naar analoge stuursignalen voor de motorbesturing. Een van de eenvoudigste systemen is gebruik te maken van een zogenaamde R-2R omzetter. Het prinsipeschema van een dergelijke omzetter is zeer eenvoudig en getekend in onderstaande figuur.

**Het principe van een R-2R omzetter**



Q1 tot en met Q4 zijn de uitgangen van een binaire teller. De uitgang van het zogeheten R-2R netwerk voert een analogo signaal dat evenredig is met de tellerstand, zie de volgende grafiek. In dit geval wordt het uitgangssignaal in 16 stapjes gedeeld ( $2^4 = 16$ ). Een R-2R netwerk gedraagt zich dus als digitaal analoog converter (DAC).

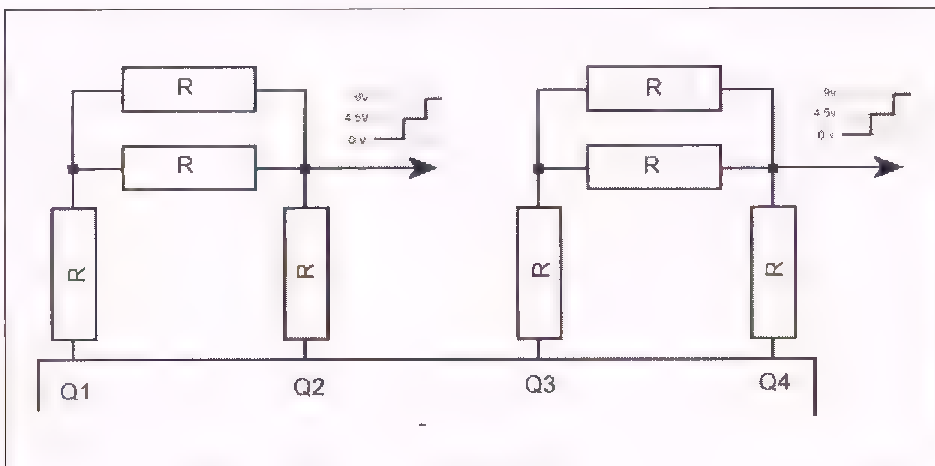
**De uitgangsspanning van het R-2R netwerk**



**DAC's en de robot**

In de volgende schakeling is het R-2R netwerk ingekort en tweemaal uitgevoerd. Het uitgangssignaal is nu beschikbaar op drie niveaus: nul, de halve voedingsspanning en de hele voedingsspanning. Als men uitgaat van een 9 V blokcel komen deze drie niveaus overeen met 0 V, 4,5 V en 9 V. Precies wat nodig is om alle gewenste ingangssignalen voor de TLE4206 op te wekken!

*De drie analoge niveaus voor het besturen van de motoren kunnen op deze manier uit één vier bit teller worden gegenereerd*



Q1 en Q2 worden gebruikt voor de besturing van het éne wiel en Q3 en Q4 voor het andere wiel.

**Waarheidstabel van de besturing**

De waarheidstabel van de besturing van één motor is getekend in de onderstaande waarheidstabel. Uiteraard worden Q1/Q2 voor de tweede motor vervangen door Q3/Q4.

*De waarheidstabel voor het besturen van één van de motoren uit twee binaire signalen*

Q1	Q2	UIT	WIEL
0	0	0V	vooruit
1	0	4,5V	stilstand
0	1	4,5V	stilstand
1	1	9V	achteruit

**Robot bewegingen**

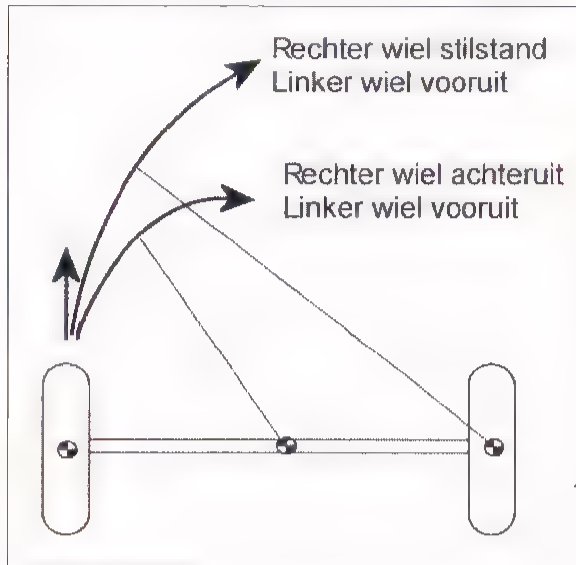
Als men de waarheidstabellen voor de beide wielen combineert is een negental voertuigbewegingen te realiseren (inclusief stilstand), zie onderstaande tabel.

*De negen bewegingsrichtingen oftewel "vrijheidsgraden" van de robot als de twee motoren met 2 x 2 bit worden aangestuurd*

LINKER WIEL	VOER TUIG	RECHTER WIEL
vooruit	Vooruit	vooruit
vooruit	Rotatie om rechter wiel rechtsom	stilstand
vooruit	Pirouette rechtsom	achteruit
stilstand	Rotatie om linker wiel linksom	vooruit
stilstand	Stilstand	stilstand
stilstand	Rotatie om linker wiel rechtsom	achteruit
achteruit	Pirouette linksom	vooruit
achteruit	Rotatie om rechter wiel linksom	stilstand
achteruit	Achteruit	achteruit

Het voertuig kan dus in beide richtingen om zijn as, maar ook om elk van de wielen draaien. In feite zijn er dus drie "middelpunten", waar de beweging rond plaatsvindt. Dit is voorgesteld in het volgende scheidingsdiagram. Aan beweeglijkheid dus geen gebrek. Nu nog de aansturing!

**De drie "middelpunten" op de vooras, waar de robot rond kan bewegen**



### De robot gaat dansen

**Het principe**

Ter demonstratie van de beweeglijkheid van het mechanische deel van de robot wordt een eenvoudige schakeling voorgesteld, die alle mogelijke bewegingspatronen doorloopt. Er zijn negen basisbewegingen. Met een simpele 4 bit binaire CMOS-teller, zoals de 4520, moet het dus mogelijk zijn de robot in beweging te krijgen. Een 4 bit binaire teller kent zestien toestanden die men meestal voorstelt door de decimale notatie "0" tot en met "15". Decimaal "0" komt dan overeen met binair "L-L-L-L", decimaal "15" met binair "H-H-H-H". Als de Q1 en Q2 uitgangen worden gebruikt voor de besturing van het linker wiel en Q3 en Q4 voor de besturing van het rechter wiel, ziet het verband tussen de stand van de teller en de bewegingsrichting van de robot er uit zoals voorgesteld in onderstaande besturingstabel.

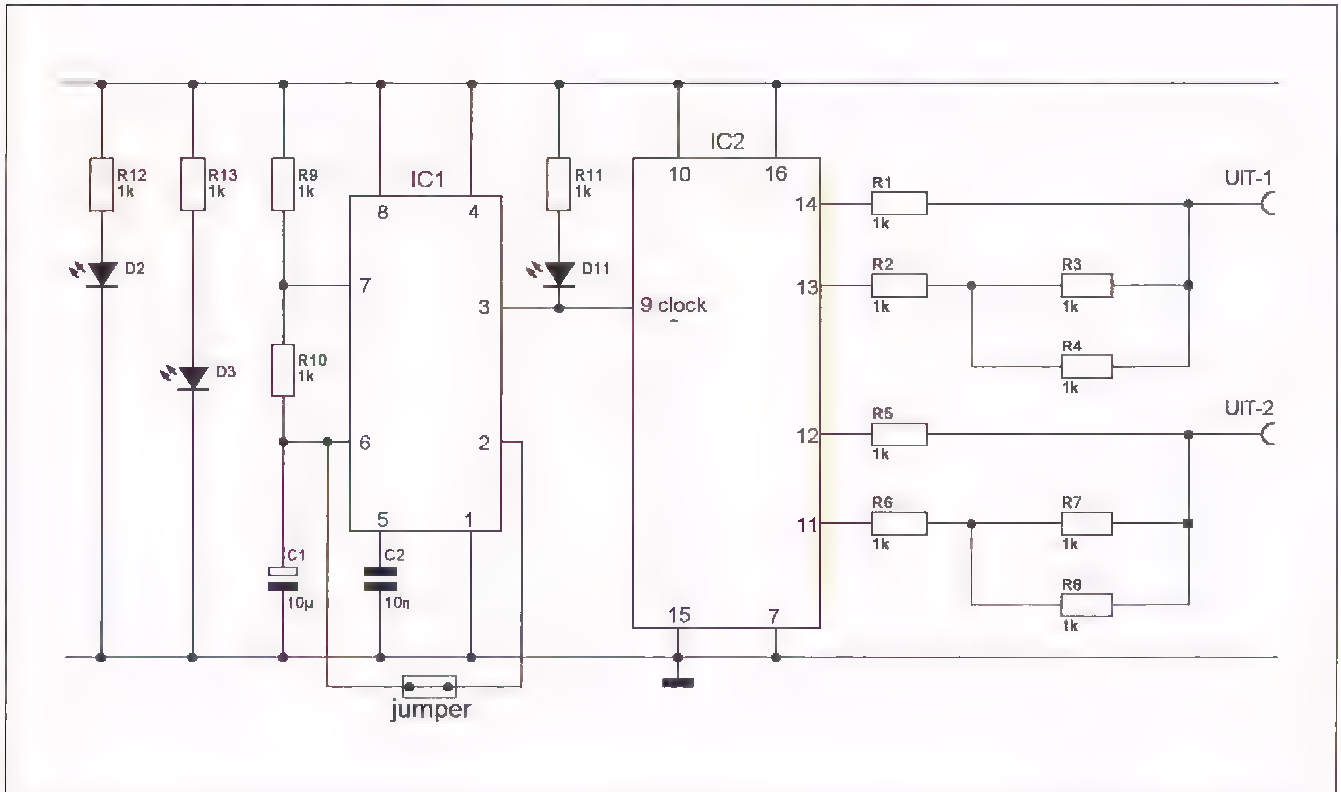
**Het verband tussen de toestand van de 4 bit brede teller en de bewegingsrichting van de robot**

STAP	LINKER WIEL		RECHTER WIEL		BEWEGING ROBOT
	Q1	Q2	Q3	Q4	
0	0	0	0	0	Vooruit
1	1	0	0	0	Rotatie om linker wiel linksom
2	0	1	0	0	Rotatie om linker wiel linksom
3	1	1	0	0	Pirouette linksom
4	0	0	1	0	Rotatie om rechterwiel rechtsom
5	1	0	1	0	Stilstand
6	0	1	1	0	Stilstand
7	1	1	1	0	Rotatie om rechterwiel linksom
8	0	0	0	1	Rotatie om rechterwiel rechtsom
9	1	0	0	1	Stilstand
10	0	1	0	1	Stilstand
11	1	1	0	1	Rotatie om rechterwiel linksom
12	0	0	1	1	Pirouette rechtsom
13	1	0	1	1	Rotatie om linker wiel rechtsom
14	0	1	1	1	Rotatie om linker wiel rechtsom
15	1	1	1	1	Achteruit

Een paar situaties komen dubbel voor, maar voor deze demonstratie is dat uiteraard geen probleem. Als telpuls generator kan men (hoe kan het anders) een onvolprezen 555 toepassen.

**Het schema**

Het schema van de besturing voor de dansende robot is getekend in onderstaande figuur. In de eenvoud herkent men de ware meesterontwerper! Géén microcontrollers of te programmeren geheugens, maar twee ordinaire IC'tjes. Waaruit blijkt dat eenvoudige mechatronica niet zo moeilijk is als het woord klinkt.



**Het volledig schema van de besturing van de dansende robot**

IC1 is de 555, alhier op de overbekende manier geschakeld als astabiele multivibrator. De frequentie van de schakeling wordt bepaald door de onderdelen R9, R10 en C1. De uitgangspulsen op pen 3 sturen de clock-ingang van IC2, de 4520. De vier Q-uitgangen van de teller sturen de twee R-2R netwerkjes R1 tot en met R8 aan. De 4520 bevat twee tellers. Hier wordt er slechts één gebruikt. Dat lijkt verspilling, maar CMOS IC's zijn zo goedkoop dat de keuze voor een meer gecompliceerde teller waarvan er maar één in een huisje zit niet handig is. Op het printje zijn tevens een rode en een groene LED D2/D3 gemonteerd, die stuur- en bakboord aangeven. Een blauwe LED D1 geeft de 555 pulsen weer, zodat gevolgd kan worden waar de dansende robot in de telcyclus zit.

**Belangrijke opmerking**

Tussen pen 2 en pen 6 van de 555 is een draadbrug aangebracht. In deze schakeling **moet deze worden aangebracht**, maar er zijn ook experimenten (zie hoofdstuk 5 bijvoorbeeld) waar deze draadbrug moet worden verwijderd en waarbij de 555 op pen 2 wordt getriggert met een extern signaal. De 555 werkt dan als monostabiele multivibrator.

**De bouw van de schakeling**

De print voor de schakeling is getekend in de figuur op de volgende pagina. Hiervoor gelden dezelfde opmerkingen als voor de print van hoofdstuk 3. De componentenopstelling volgt uit de daaropvolgende figuur.

# ONDERDELENLIJST

## WEERSTANDEN, 1/4 W, 5 %

R1-R13 ..... 1 k $\Omega$

## CONDENSATOREN

C1 ..... 10  $\mu$ F, 16 V printelco  
 C2 ..... 10 nF, MKH

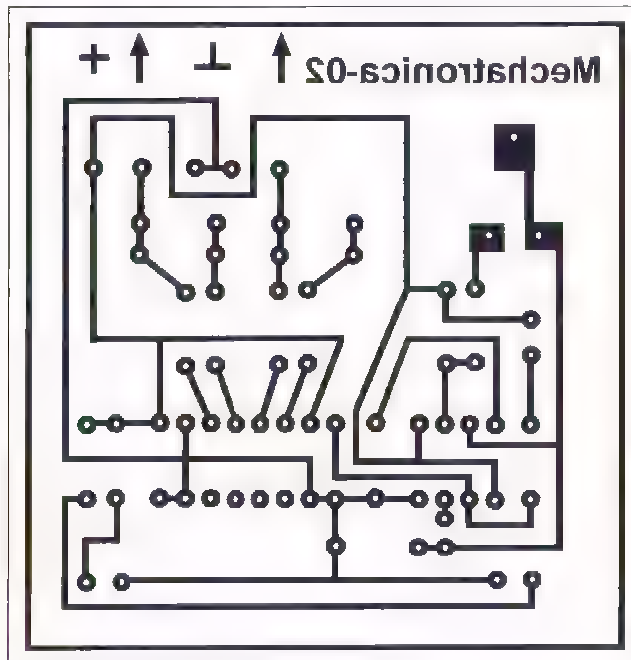
## HALFGELEIDERS

D1 ..... LED, 5 mm rood  
 D2 ..... LED, 5 mm geel  
 D3 ..... LED, 5 mm blauw  
 IC1 ..... 555, mini-DIL  
 IC2 ..... CD4520

## DIVERSEN

1 x ..... IC-voetje 8 pennen  
 1 x ..... IC-voetje, 16 pennen  
 7 x ..... printsoldeerlipje

*De print van de schakeling*

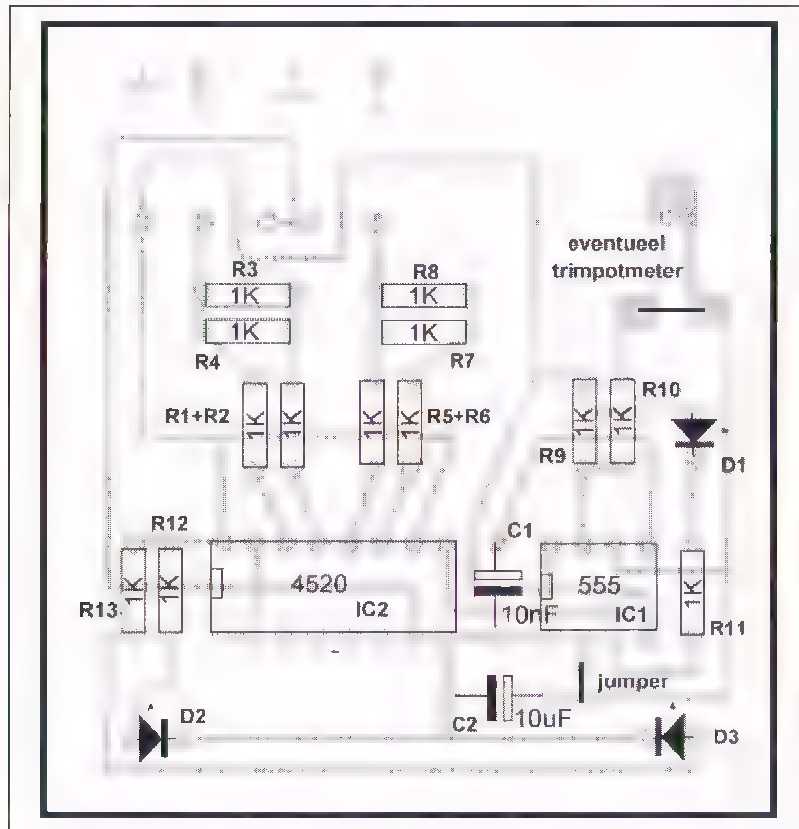


Rechts van C2 zit het draadbruggetje dat bij deze schakelingen de pennen 2 en 6 van de 555 doorverbindt. Zet in de twee gaatjes printsoldeerpenne en verbindt deze met een draadje. Op deze manier kan deze verbinding gemakkelijk worden verwijderd als dit noodzakelijk is. Denk er aan dat pen 2 dan wordt gebruikt voor de externe besturing van de 555. Tussen weerstand R10 en pen 6 van de 555 zit ook een draadbruggetje. Eventueel kan men dit draadbruggetje verwijderen en rechtsboven op de print een liggende instelpotentiometer van 4,5 k $\Omega$  en met als afmetingen 10 x 5 mm insolderen. Men kan dan experimenteren met de frequentie van de timer. Vervang dan wél R10 door een weerstand van 220  $\Omega$ .

### Montage op het loopwerk

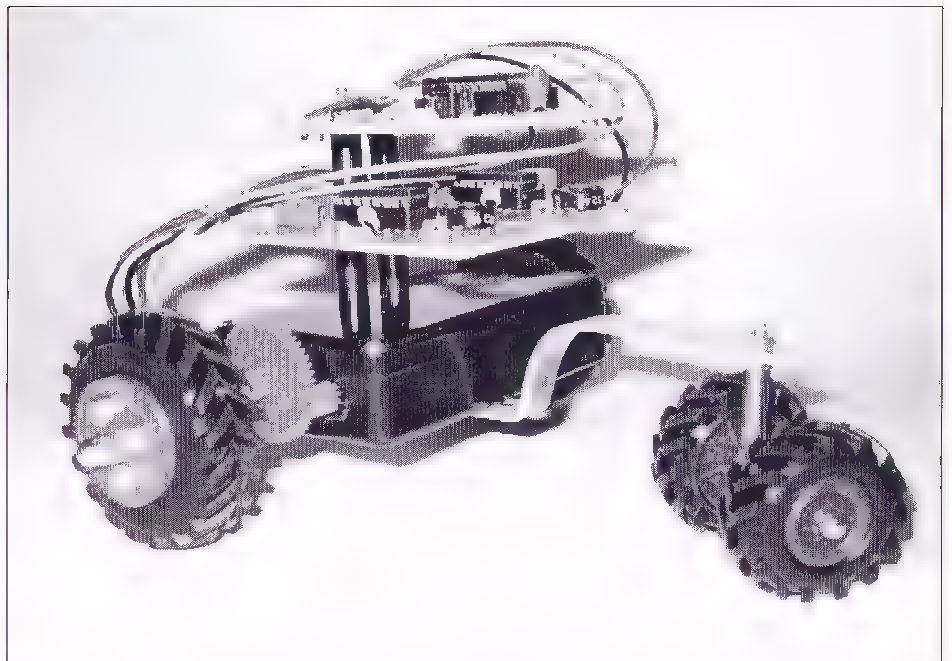
De printjes van de hoofdstukken 3 en 4 zijn op twee lange M3 boutjes geschoven die hun verzonken kop aan de binnenzijde van het deksel hebben. Afstandsbusjes voorkomen sluiting.

**De componenten-  
opstelling van de  
schakeling**



Onderstaande foto geeft een impressie van de compleet gemonteerde dansende robot.

**De compleet  
gemonteerde  
dansende robot**



**Epiloog**

Als uw huisdier op eigen beweging door de kamer danst, is al aardig wat bereikt. De mechanica werkt en wordt aangestuurd door de elektronica. Een volwaardig mechatronica-project, dus. In het volgende hoofdstuk wordt een geluidssensor toegevoegd, waardoor uw robot niet meer zijn interne klokrilme volgt, maar zijn bewegingen afstemt op het ritme van muziek of handgeklap.

# 5 Onze robot leert luisteren

## Inleiding

**Besturen met handgeklap**

Om een tango te kunnen dansen moet je naar muziek kunnen luisteren, of in ieder geval het ritme ervaren. Een volgend leerzaam project is dus de voorgeprogrammeerde beweging door middel van de 555 en de teller uit hoofdstuk 4 zó uit te breiden dat de robot reageert op muziek. Of, met andere woorden, de bewegingsrichting van de robot verandert als er opeens een krachtig geluid in de kamer weerklinkt. Dat kan muziek zijn, maar het is ook indrukwekkend om de robot te "besturen" met handgeklap.

**Het principe**

Wie wil luisteren moet oren hebben. In dit hoofdstuk gaan we onze robot dat ook voorzien van zo'n orgaan. Dat moet uiteraard een elektronisch "oor" zijn en wat ligt meer voor de hand dan gebruik te maken van een microfoon. Er zijn diverse soorten microfoons in de handel, maar uiteraard kiezen wij voor een klein en goedkoop elektret microfoonkapseltje. Dergelijke kapseltjes zitten in iedere GSM-telefoon en zijn zeer gevoelig. Precies wat wij nodig hebben om onze robot te laten reageren op zijn omgevingsgeluid.

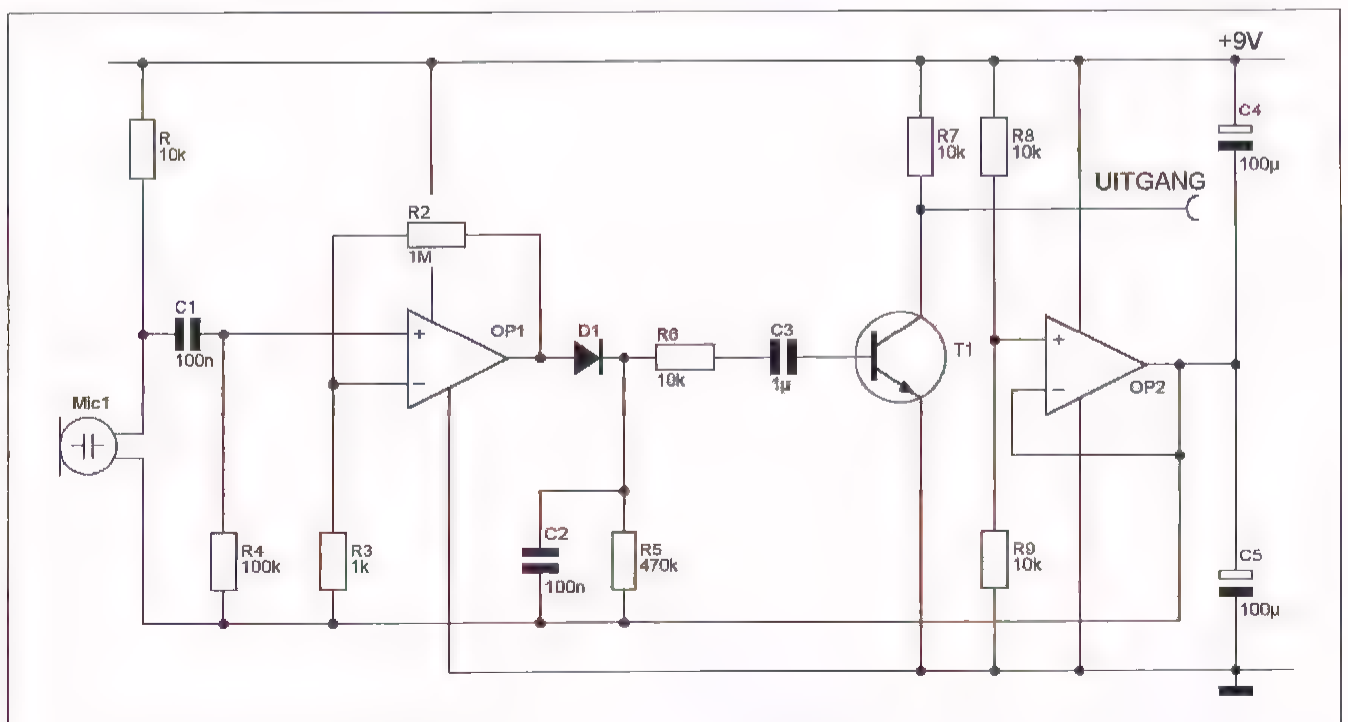
Natuurlijk moet het signaaltje van de elektret microfoon worden versterkt. Daarvoor gebruiken we een op-ampje. Via een transistortrap maken we van het analoge geluidssignaal een digitale pulstrein, die de 555 uit de schakeling van hoofdstuk 4 triggert.

## De praktijk

**Het schema**

Het uitbreidingsschema is voorgesteld in onderstaande figuur. De elektret microfoon Mic1 levert een uitgangsspanning af van ongeveer 10 mV als hij naar het omgevingsgeluid luistert. De weerstand R1 naar de plus zorgt voor de benodigde voorspanning voor de elektret. Het signaaltje van de microfoon wordt capacitief doorgekoppeld naar de versterker.

**Het compleet schema van de uitbreidingsschakeling**





Deze versterker OP1 staat behoorlijk op z'n tenen. Via de terugkoppeling R2/R3 is de gesloten lus versterking ingesteld op 1.000! Vormt waarschijnlijk lekker, maar dat is in deze toepassing geen probleem.

De diode D1 richt het signaal gelijk zodat de condensator C2 opgeladen wordt. De tweede versterker OP2 laat een klassieke truc zien om met één batterij aan een dubbele voedingspanning te komen. De niet-inverterende ingang wordt door middel van de spanningsdeler R8/R9 ingesteld op de helft van de voedingspanning. Omdat de op-amp is geschakeld als spanningsvolger zal ook de uitgang op die spanning staan. De uitgang vormt dus een virtuele massa, zodat de voeding opgedeeld wordt in -4,5 V en +4,5 V. Het is geen probleem hier de motorvoedingsbatterij te gebruiken. Bij hardnekkige storing kan over de beide voedingselco's C4 en C5 nog een ceramisch condensatortje van 100 nF gesoldeerd worden, aan de koperzijde van de print. De transistor T1 zorgt voor aanpassing aan het nulsignaal van de reeds beschreven tellerprint.

<b>ONDERDELENLIJST</b>	
<b>WEERSTANDEN, 1/4 W, 5 %</b>	
R1,R6,R7,R8,R9 . . . . .	10 kΩ
R2 . . . . .	1 MΩ
R3 . . . . .	1 kΩ
R4 . . . . .	100 kΩ
R5 . . . . .	470 kΩ
<b>CONDENSATOREN</b>	
C1,C2 . . . . .	100 nF, MKH
C3 . . . . .	1 μF, MKH
C4,C5 . . . . .	100 μF, 16 V printelco
<b>HALFGELEIDERS</b>	
D1 . . . . .	1N4148
T1 . . . . .	BC107, BC548
OP1,OP2 . . . . .	741, mini-DIL
<b>DIVERSEN</b>	
Mic1 . . . . .	elektret microfoon
2 x . . . . .	IC-voetje, 8 pennen
6 x . . . . .	printsoldeerlipje

**De bouw van de schakeling**

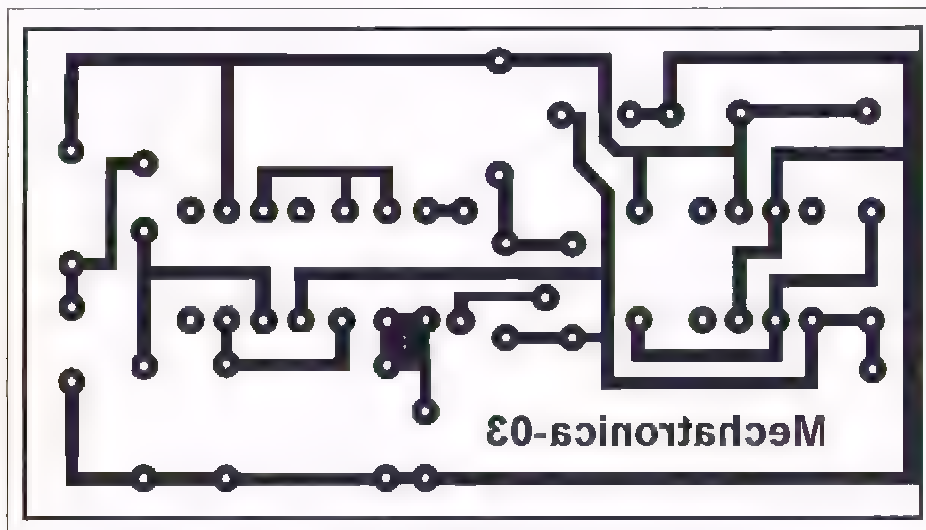
De print voor de schakeling en de componentenopstelling zijn voorgesteld in de twee figuren op de volgende pagina.

**Integratie in de robot**

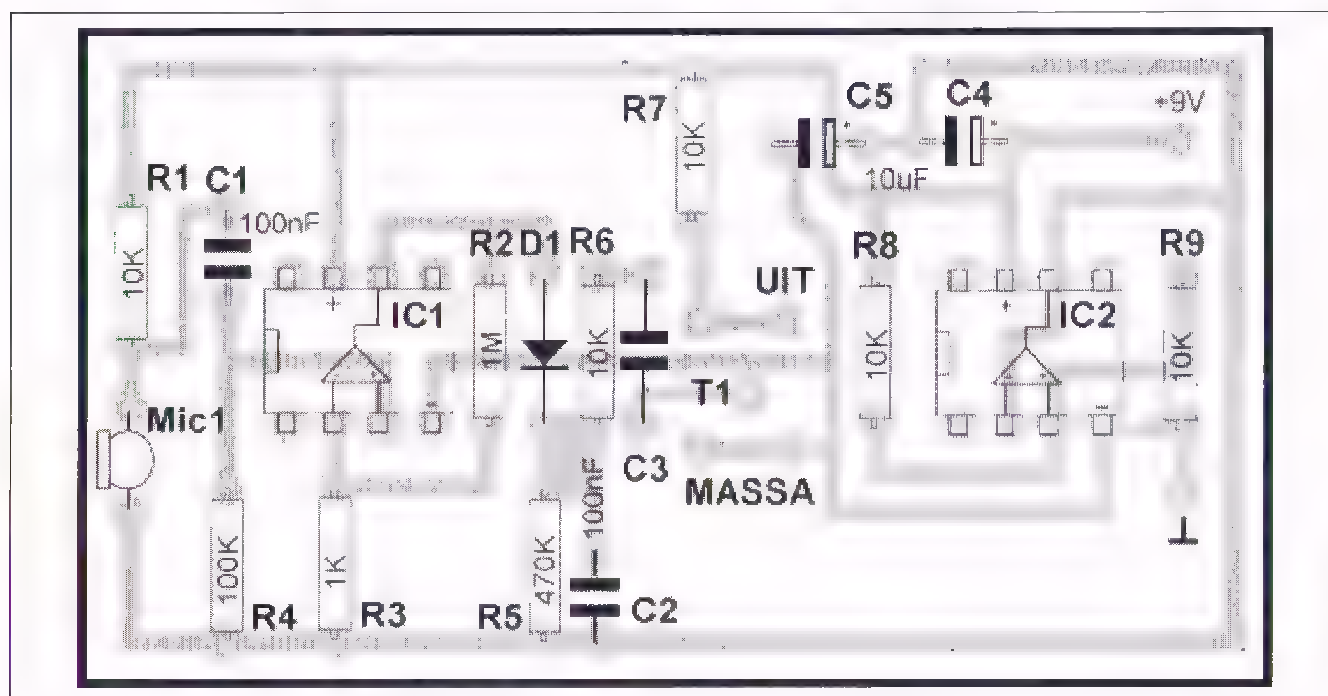
De uitgang van de print wordt aangesloten op pen 2 van de 555 op de print van hoofdstuk 4. De draadbrug tussen de pennen 2 en 6 op deze print moet verwijderd worden. Uiteraard worden ook de voedingslijnen doorverbonden.

Elke keer als er in de handen geklapt wordt, springt de teller meerdere stappen vooruit. Dit komt omdat het klappgeluid meerdere pulsen kan opleveren. Voor het effect is dit geen bezwaar, wel integendeel. Indien gewenst kan de 555-schakeling zo opgezet worden dat hij een puls geeft van enkele tienden van seconden, zodat de aansturing storingsvrij is.

De print voor de schakeling



De componentenopstelling van de schakeling



### Epiloog

De geluidsschakeling is leerzaam om enige ervaring op te bouwen met de mogelijke bewegingspatronen die het chassis uit kan voeren aan de hand van stuursignalen. Een robot die naar handgeklap kan luisteren is echter niet erg bijzonder. Reageren op geluid is ook niet echt een robotfunctie. Het is meer een verkapte afstandsbediening en zoals gesteld in de inleiding hoort dat niet binnen de definitie waar bij dit project van wordt uitgegaan. Daarom wordt het tijd voor het inbouwen van intelligentie. In de volgende hoofdstukken zal de robot met behulp van sensoren zijn eigen wereld gaan verkennen.

# 6 Onze robot leert voelen

## Inleiding

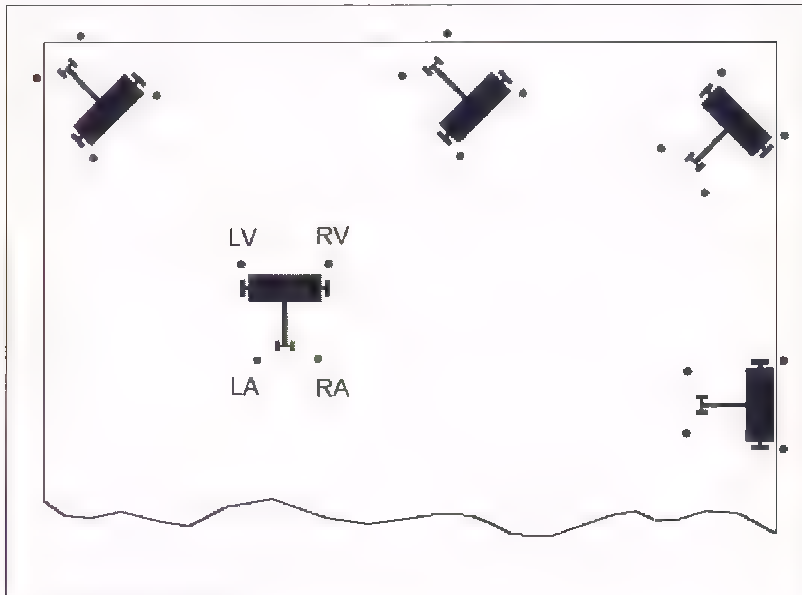
**Tafelrand detecteren**

In de voorgaande hoofdstukken is geïllustreerd hoe een robot door vastgelegde bewegingspatronen geleid kan worden. In dit hoofdstuk wordt een meer gerichte beweging afgedwongen door het activeren van een voelsensor. Als het allemaal lukt kan de robot zich aan het eind van dit hoofdstuk vrijelijk over een tafel bewegen zonder er af te vallen. Omdat er veel bewegingsmogelijkheden zijn die bewaakt moeten worden, worden vier sensoren geplaatst om te voorkomen dat een wiel zich buiten de tafelrand begeeft.

**Bewegingen bij de tafelrand**

Voor we ons in de bouw van de sensor en de elektronica gaan verdiepen, lijkt het verstandig eens na te denken over het gewenste bewegingspatroon als de tafelrand ontdekt wordt. Alle mogelijkheden zijn geschetst in onderstaande figuur.

*De mogelijke bewegingspatronen van de robot bij het benaderen van een tafelrand*



De vier sensoren worden geïdentificeerd door de codes LV, RV, LA en RA. Afkortingen van de namen LinksVoor, RechtsVoor, LinksAchter en RechtsAchter. De plaats van de sensoren moet zo gekozen zijn dat er een maximale controle over de gewenste beweging bestaat. De voorwielen kunnen niet zijdelings. Het is voldoende ze aan de voorzijde te beschermen. De twee sensoren LV en RV worden dan ook iets voor de voorwielen geplaatst. De achterwielen kunnen zich achteruit bewegen, maar ze kunnen ook zijwaarts zwenken. Om beide bewegingen af te dekken plaatsen we twee sensoren LA en RA ergens naast en achter de beide wielen.

**De waarheidstabel**

De volgende stap is het opstellen van een waarheidstabel, die aansluit bij de ingangen van de beschreven motorbesturingsprint (hoofdstuk 3) en de daarbij behorende bewegingen. Als linksachter de tafelrand verlaat (positie midden boven), moet het linkerwiel vooruit, terwijl het rechterwiel stil mag blijven staan. Als eerste benadering beschrijven we de beste beweging voor het over de rand gaan van elke cel apart. En vervolgens voor combinaties van twee cellen. We vullen in de linker kolommen van de reeds eerder beschreven bewegingstabel de celstanden in. Hierbij is een "0" in orde maar een "1" een beweging buiten de tafelrand. Het gemakkelijkst is de tabel in te vullen aan de hand van de

beweging in de laatste kolom. De regels waarbij het voertuig stil staat zijn weggelaten. Een pirouette levert ook geen goede ontsnappingsstrategie op. Die mogelijkheden worden óók weggelaten. We verwijderen in gedachten de tellerprint uit hoofdstuk 4 en leiden een simpele tabel af, aan de hand waarvan we de bewegingen statisch kunnen programmeren, zie onderstaande figuur.

LV	RV	LA	RA	LINKER WIEL		RECHTER WIEL		VOERTUIG
				Q1	Q2	Q3	Q4	
0	0	X	X	0	0	0	0	Vooruit
0	1	0	0	0	1	0	0	Rotatie om linker wiel linksom
1	0	0	0	1	1	1	0	Rotatie om rechterwiel linksom
0	0	0	1	0	0	0	1	Rotatie om rechterwiel rechtsom
0	0	1	0	0	1	1	1	Rotatie om linker wiel rechtsom
1	1	0	0	1	1	1	1	Achteruit

**Het programmeren van de beweging bij het bereiken van een tafelrand**

De vier linker kolommen geven de uitgangsconditie van de sensoren aan, terwijl de paartjes Q1-Q2 en Q3-Q4 de gewenste wielbewegingen aangeven. Let erop dat de combinatie "0-1" een stilstaand wiel oplevert. "0-0" betekent vooruit en "1-1" staat voor achteruit. Een "X" betekent dat het signaal er niet toe doet.

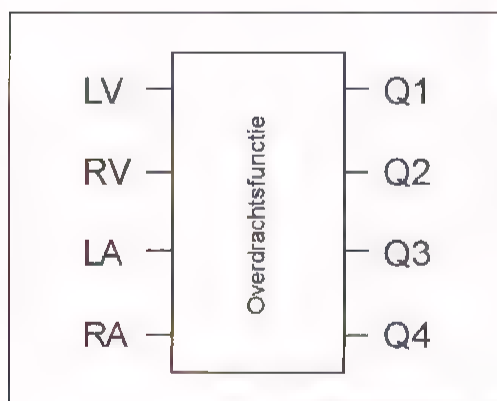
**Optimaliseren**

We kunnen de tabel optimaliseren als we gebruik maken van het feit dat Q1-Q2 "1-0" hetzelfde oplevert als Q1-Q2 "0-1". In beide gevallen staat het wiel stil. Hetzelfde geldt voor Q3-Q4. Met andere woorden, we mogen elke combinatie van een "0" en een "1" omwisselen, zolang we de combinaties "0-0" en "1-1" maar intact laten.

**Elektronische vertaling**

We moeten dus een schakeling verzinnen die, met de vier linker kolommen als ingangssignaal, op de uitgang de inhoud van de vier rechter kolommen geeft. Blokschematisch kan dit worden voorgesteld door het schema van onderstaande figuur.

**Het logisch probleem van deze schakeling teruggebracht tot de basis**

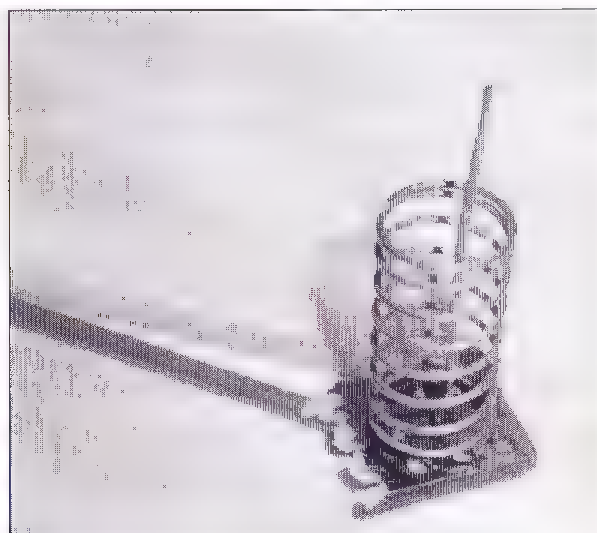


**De sensoren**

**Een mechanische sensor**

Een bruikbare mechanische sensor bestaat uit een balpenveertje om een tasterdraad heen, zie de foto op de volgende pagina. Als de tasterdraad bewogen wordt, maakt hij contact met de balpenveer. Door de draad als sleepcontact te gebruiken, kan men detecteren of er nog vaste grond is. Als hij contact met de bodem verliest, wordt de schakelaar onderbroken.

**Deze mechanische sensor detecteert het verlies van contact met een vaste bodem**

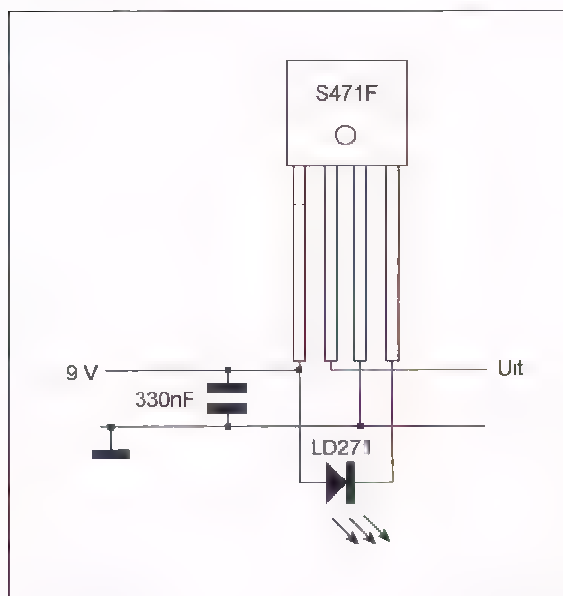


De centrale draad dient, evenals de veer, van verenstaal te zijn. Door de sensordraad lang en licht gebogen uit te voeren lijkt hij op een voelspriet van een insect. Een leuke bijkomstigheid!

**Een elektronische oplossing**

Ondanks de eenvoud van deze contactsensor, wordt in dit hoofdstuk toch gekozen voor een elektronische oplossing. Voor het "voelen" van afstanden wordt meestal ultrasoon geluid gebruikt. Een veel nieuwere techniek bestaat uit een infrarood zender die een pulstrein uitzendt en vervolgens "kijkt" of deze weer ontvangen wordt. Naar beneden gericht ziet de sensor dus of de tafel er nog is, of dat hij zich al boven de afgrond bevindt. De sensor in kwestie is de S471F van Sharp. Hij is niet kieskeurig wat de voeding betreft en werkt tussen 4,5 V en 16 V. De fabrikant adviseert wél een condensator van 330 nF over de voeding te plaatsen.

**Het schema van de tafelrand detector**



Deze sensor is een juweeltje voor weinig geld. Sharp heeft dit IC ontwikkeld voor grootschalig gebruik in consumentenelektronica als printers, faxen en dergelijke. Daardoor werd een zeer lage prijs mogelijk gemaakt. Jammer dat de IR-LED niet geïntegreerd is. Die moeten we, samen met de ontstoringcondensator over de voeding, zélf toevoegen. Omdat het IC de LED intern aanstuurt via een stroombron is een serieweerstand niet nodig. Het IC ziet eruit als een transistor met vier pootjes.

Aan de platte zijde bevindt zich een venstertje waarachter de ontvanger zich schuil houdt. De sensor stuurt een externe IR-LED (piekgolflengte 940 nanometer) aan. Een gewone LD271 is hiervoor zonder meer geschikt.

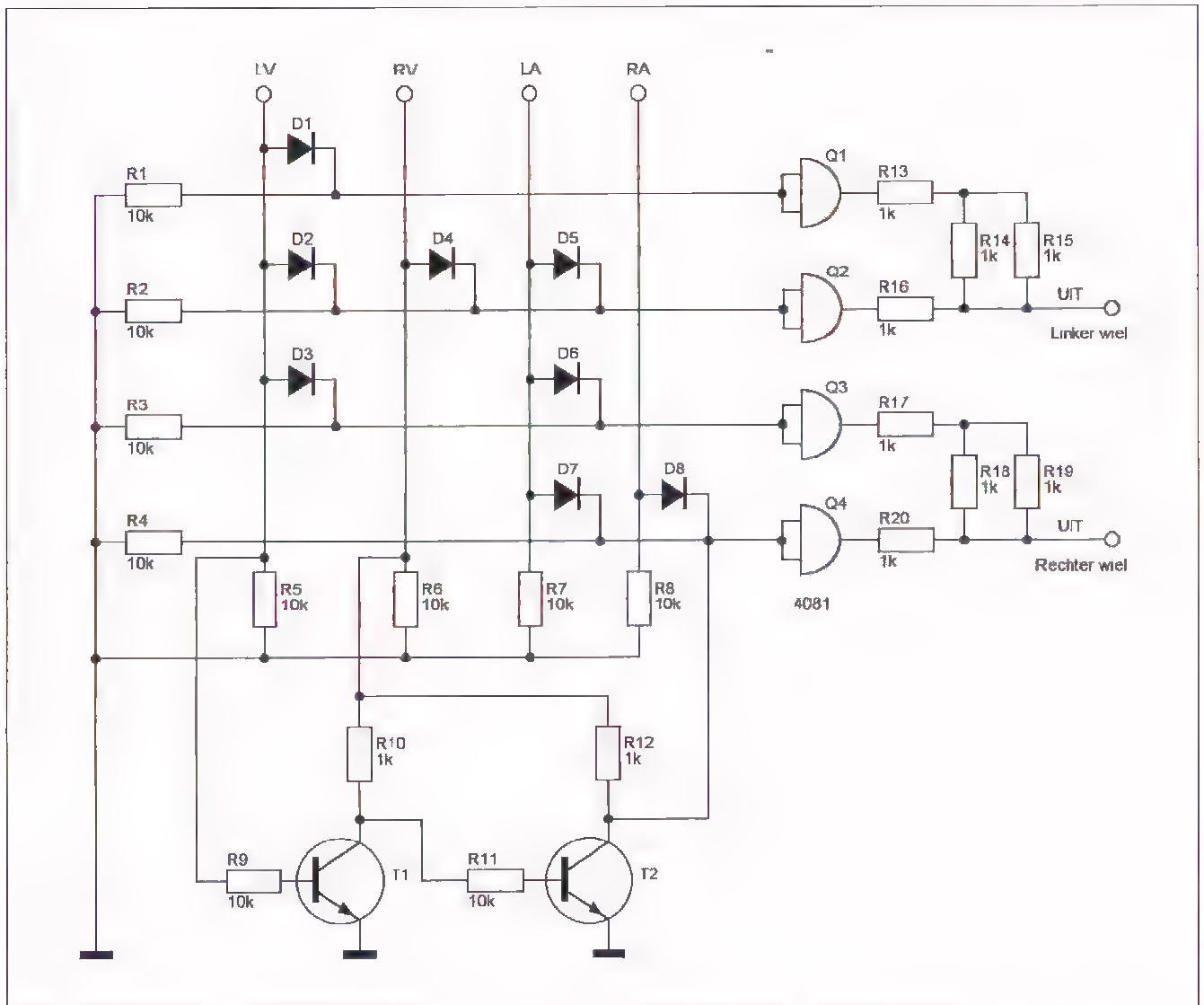
**Licht detecteren**

Zolang het gereflecteerde pulstreintje waargenomen wordt is de sensoruitgang "0". Zodra de ontvangst wegvalt wordt de sensoruitgang "1". Door deze signaaltechniek is de sensor ongevoelig voor lichtstoringen van buiten. Wél moeten het ontvangstvenstertje en de LED-zender natuurlijk niet in elkaars gezichtsveld staan.

**De elektronica**

**Het schema**

Met een simpele diode/transistor-matrix kan de eerder opgestelde tabel in een statische decodering worden omgezet. Het volledig schema van deze "intelligente" besturing is getekend in onderstaande figuur.



**De elektronische vertaling van de waarheidstabel van de taferrand detectie**

De matrix van vier horizontale lijnen (de uitgangsspanningen voor de motoren) en de vier verticale lijnen (de uitgangsspanningen van de sensoren) wordt afgesloten met de reeds bekende eenvoudige R-2R digitaal naar analoog omvormer. In deze schakeling is deze DAC uitgevoerd met de vier poorten die in een CD4081 aanwezig zijn. Als geen sensor aangestuurd is of alleen de achterste twee samen, dan rijdt het voertuig vooruit. Voor de laatste regel hebben we een iets ingewikkelder schakeling nodig. Dat is namelijk de enige conditie

waarbij twee sensoren tegelijk signaal afgeven. Deze situatie wordt gedecodeerd door een twee-transistor schakeling, die door RV onder spanning wordt gezet en dit hoge signaal doorkoppelt naar de ingang van Q4 als LV "1" wordt. Alle lijnen worden door een weerstandje in rust naar nul getrokken. Q1+Q2 en Q3+Q4 sturen weer de laddernetwerkjes aan die we aanvankelijk achter de teller hadden geplaatst. De twee uitgangen sturen rechtstreeks de ingangen van de motorbesturingsprinten uit hoofdstuk 3 aan.

## ONDERDELENLIJST

### WEERSTANDEN, 1/4 W, 5 %

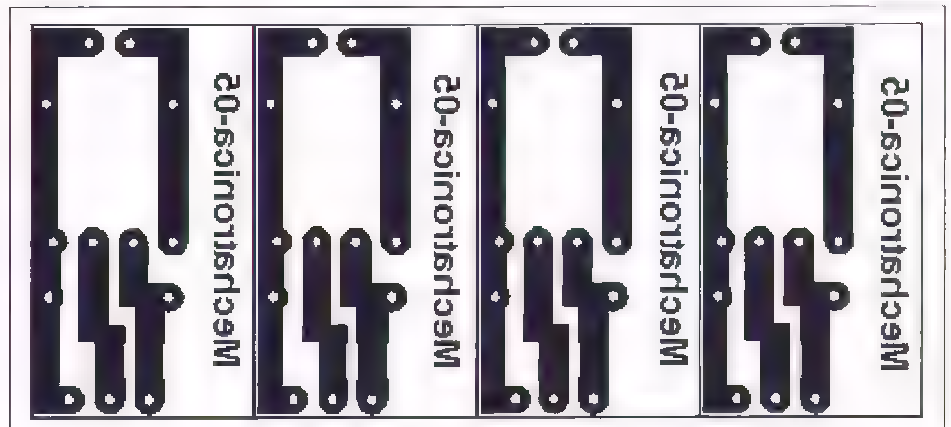
R1-R8,R9,R11	10 kΩ
R10,R12,R13-R20	1 kΩ

### HALFGELEIDERS

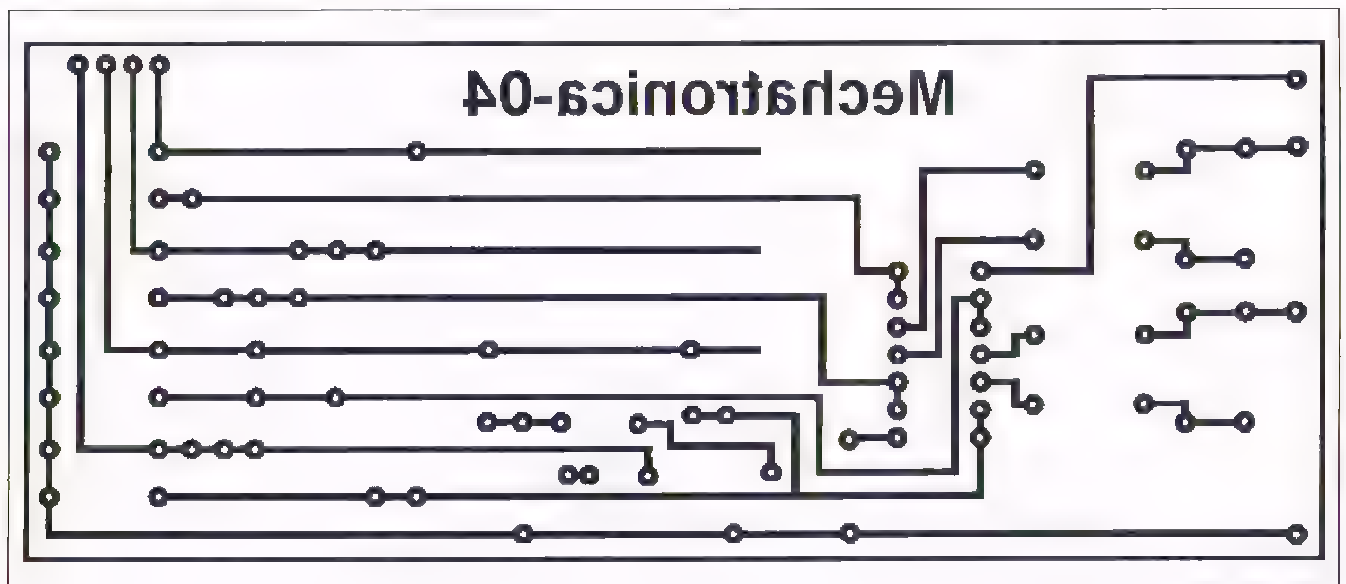
D1-D8	1N4148
T1-T2	BC107,BC548
IC1	CD4081

**De printen** Ook hier past alles weer op printjes, zie de afbeeldingen op deze en volgende pagina's.

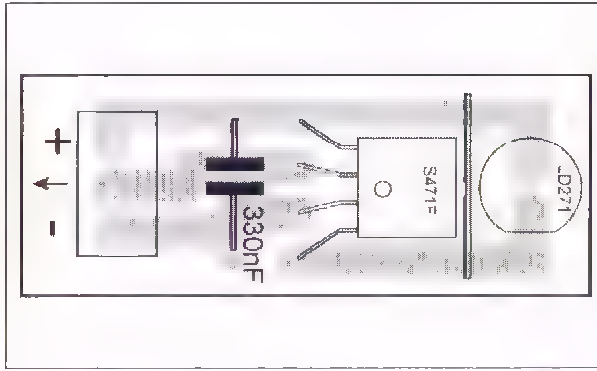
*De print voor de sensoren*



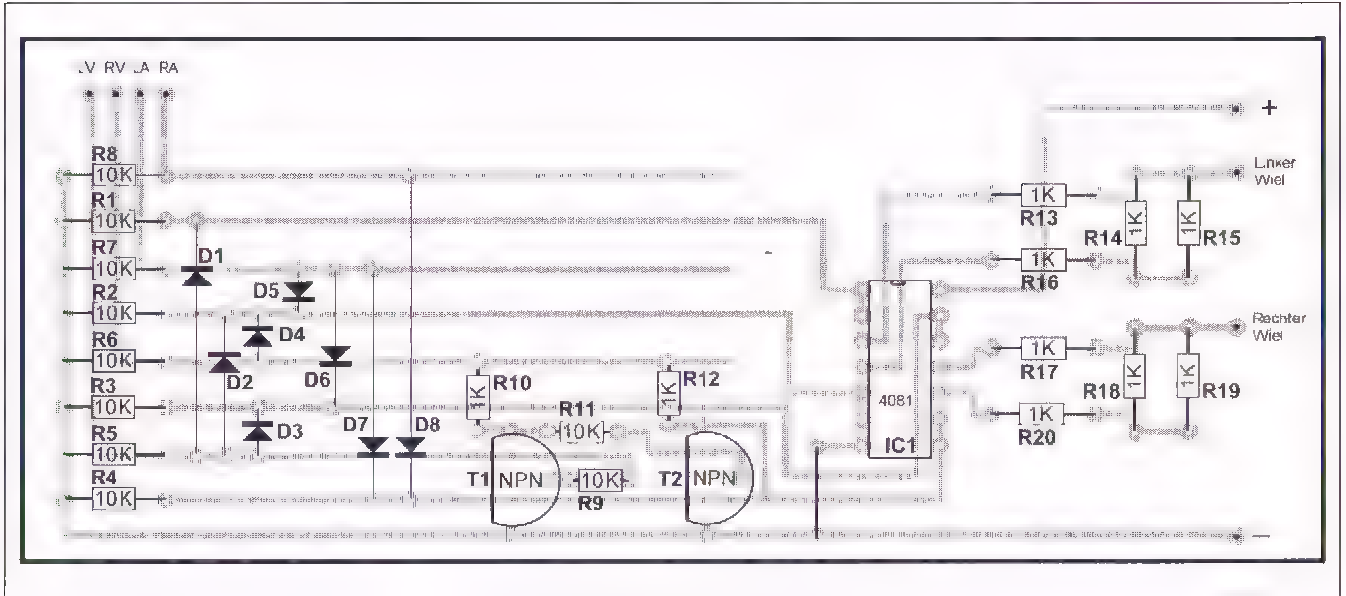
*De print voor de besturingselektronica*



**De sensor is op de print gemonteerd**



**De componenten-opstelling van de besturingselektronica**



### De sensormontage

**Goede afscherming noodzakelijk**

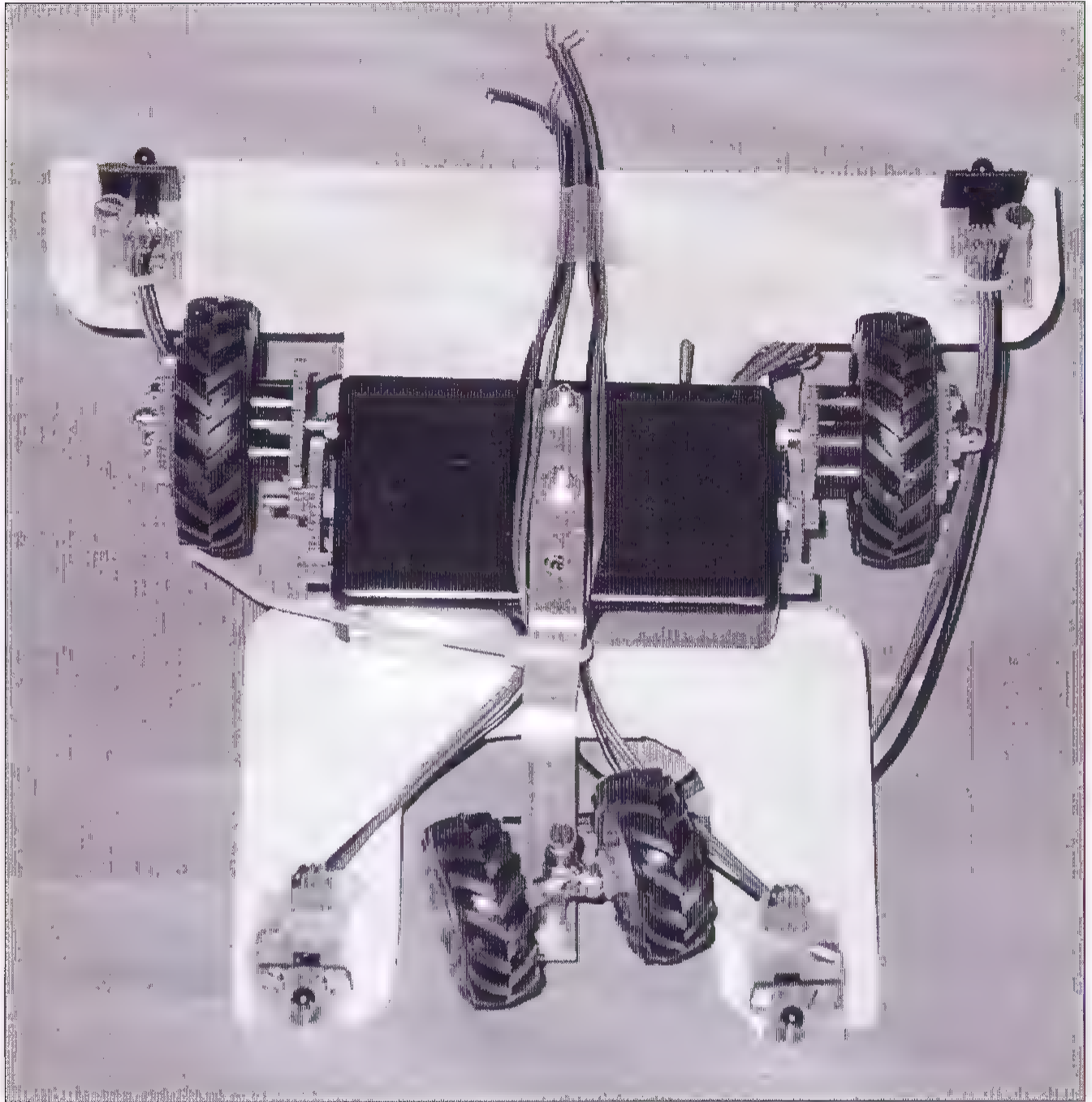
De drie onderdelen nemen nauwelijks printruimte in beslag. De decoderprint met vier sensoren is zo eenvoudig, dat overwogen kan worden het geheel op gaatjes-board op te bouwen. Als er toch een print gemaakt wordt is het onhandig de sensorprintjes apart te etsen. Op de printtekening zijn er vier gecombineerd. Omdat de print uitsteekt en onderhevig kan zijn aan stoten, is hij stevig uitgevoerd. Bovendien is extra ruimte aanwezig waarmee men de print op het chassis kan bevestigen. De draden lopen via een driepolige printconnector. Het vreemde onderdeel tussen het IC en de LED is een stukje printplaat, vastgesoldeerd aan soldeerpenen die verbonden zijn met de kopersporen op de print. Om kortsluiting te voorkomen is het koper in het midden onderbroken.

**Detail van de montage van een sensorprint op het chassis**





De LED reflecteert zoveel licht naar alle zijden dat dit schotje absoluut noodzakelijk is. De onderbreking van de koperlaag wordt afgeplakt met zwarte tape of, als dat bij de hand is, voorzien van een kwastje zwarte verf.



***De montage van de vier sensorprintjes op de onderzijde van het chassis van de robot***

Het gedeelte van het schotje zonder koper laat namelijk zoveel IR-licht door dat overspraak onvermijdelijk is. Hoe de vier sensorprintjes onder het chassis van de robot worden bevestigd blijkt uit de bovenstaande foto. Deze prent laat een eenvoudige oplossing zien. Een kunststof plaatje met uitsparingen voor de wielen geeft voldoende flexibiliteit, ook voor het monteren van toekomstige uitbreidingen. Door gaatjes in het platform te boren, kan de bedrading bovendeaks gevoerd worden. De kans dat draden tussen de tandwielen komen is dan minimaal. Het voertuig op de foto is nog niet in die fase van montage. Het moge duidelijk zijn dat we tegen de onderzijde van het voertuig aankijken.

**Opmerking**

Uit de praktijk is gebleken dat een klein vuiltje al roet in het eten kan gooien. Als de robot zich als een lemming over de tafelrand stort, kan

het nuttig zijn de sensoren op vuil te controleren. Het kan ook zijn dat het IR-licht via reflectie de sensor bereikt. Een goede afscherming van de lichtbron is echt noodzakelijk.

## Epiloog

Onze robot lijkt al aardig compleet. De robot "voelt" de rand van de tafel en reageert adequaat. Vier sensoren en een decoderprint zorgen voor de gewenste reactie als de tafelfrand op welke manier dan ook bereikt wordt. Nog een klein probleempje moet worden opgelost. Zoals de robot nu geconfigureerd is zal hij bij het randje van de tafel ervoor zorgen dat hij van de afgrond wegrijdt. De sensor ziet dan weer vaste grond onder zich en de robot rijdt onmiddellijk weer vooruit. We zullen dan ook enige vertraging moeten inbouwen zodat het loopwerk een terugtrekkende beweging maakt van voldoende omvang. Die vertraging zit al in de motor stuurprint ingebouwd. Afhankelijk van de reactiesnelheid van het onderstel, dus eigenlijk de overdrachtsfunctie tussen de motor-as en de wielen, kan de vertragingcondensator op de motor stuurprint aangepast worden.

Plaats de sensoren op voldoende afstand van de wielen. Dat geeft het voertuig de gelegenheid op tijd tot stilstand te komen.

Vervolgens zal ongetwijfeld de vraag rijzen of het geheel steeds opnieuw weer gesloopt moet worden als de batterij aan vervanging toe is. Zowel de elektronica printen, het montagedek en het kastdeksel moeten daarvoor los. Het antwoord op deze vraag is ontkennend. We maken de robot zo intelligent dat hij zélf voor zijn voeding zorgt. Hoe, dat zien we in het volgende hoofdstuk.

# 7 Onze robot krijgt ogen

## Inleiding

### Het betere werk

In de voorgaande hoofdstukken hebben wij vrij volledig uitgewerkte projecten aangeboden. Nabouwen, aansluiten en het werkt. Dit wordt vanaf nú anders. In dit hoofdstuk geven wij u bijvoorbeeld een idee hoe u de robot kunt voorzien van "ogen". Als voorbeeld worden deze "ogen" gebruikt voor het opzoeken van een laadstation voor de accu van de robot. Maar u kunt het geschetste principe natuurlijk voor een heleboel andere toepassingen gebruiken. U zou bijvoorbeeld via LED's een "route" kunnen uitzetten op de vloer en de robot deze route laten zoeken en volgen. Het beschreven principe is ook hiervoor bruikbaar, maar u zult uw eigen fantasie moeten aanspreken om een en ander naar een praktisch bruikbaar systeem om te zetten.

### Elektronica als richtlijn

Vandaar dat u de beschreven elektronica eerder als richtlijn dan als kant-en-klare nabouwoplossing moet opvatten. Bepaalde basiszaken weet u inmiddels al. U weet hoe u uit vier digitale signalen Q1 tot en met Q4 de twee stuursignalen voor de motoren moet afleiden. Lees hoofdstuk 4 er nog maar een op na. Dergelijke basale zaken komen nu niet meer aan de orde.

### Batterijen vervangen door NiCad's

Een beetje robot zoekt zijn eigen kostje op als hij honger krijgt. Hij moet dan wel ingewanden hebben die hem in staat stellen voedsel op te nemen. Bij een robot is dat eenvoudig. We halen de 9 V blokcel eruit en vervangen die door een oplaadbaar type. In dit hoofdstuk gaan we uit van een gewone NiCad. Verder hebben we een externe lader nodig waaraan de robot zijn honger kan stillen. Als we die zaken opgelost hebben, bespreken we hoe de robot uit eigen beweging op zoek gaat naar zijn voedselbron.

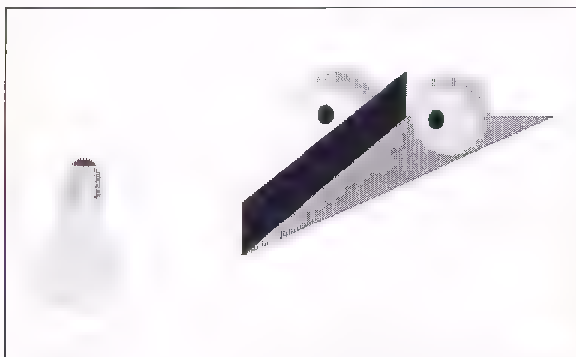
### NiCad laden

Het laden van een NiCad is geen simpele zaak. De laadcurve is complex en een automatische lader die de laadtoestand van de NiCad in de gaten houdt en de laadstroom aanpast vraagt al gauw om een ingewikkelde schakeling. Hoewel interessant, zullen we in dit boek geen geavanceerde lader ontwikkelen. We grijpen terug naar een alternatief. Een NiCad wordt over het algemeen geladen met een laadstroom die ongeveer een kwart van zijn capaciteit is. Een 9 V blokcel heeft een capaciteit in de buurt van 120 mAh. De laadstroom mag dan ongeveer 30 mA bedragen. Door een weerstand in serie met de laadspanning te zetten, begrenzen we de stroom op die waarde.

### Licht als lokmiddel

We gebruiken infrarood licht als lokaas voor de robot. Zodra een IR-LED brandt, zal de robot zich een weg naar deze lichtbron zoeken. Om die taak goed uit te kunnen voeren geven we hem een paar "ogen", zie onderstaande figuur.

*Door de robot uit te rusten met twee "ogen" kan hij op zoek gaan naar de lichtbron*



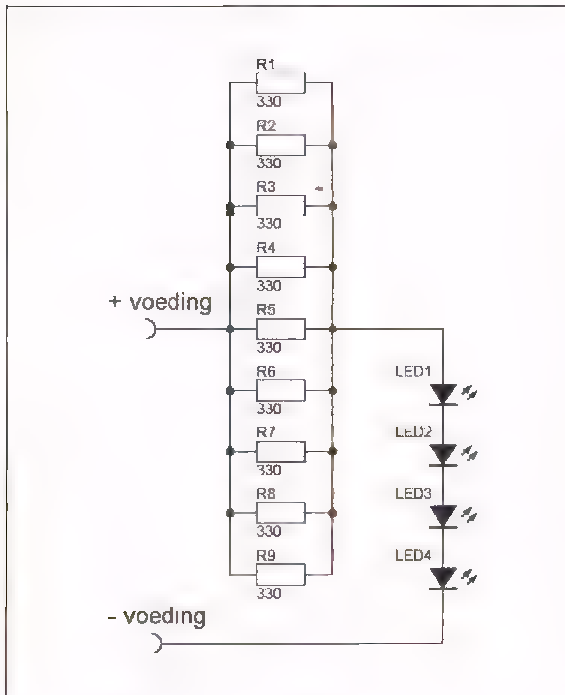
Door een schotje tussen de ogen te plaatsen kan hij de positie van de lichtbron inschatten. In de figuur ziet alleen het rechter oog licht. Het linker oog valt in de schaduw. Dat is voor de robot het signaal iets naar rechts bij te sturen. Door overshoot zal de robot zich enigszins waggelend naar de lichtbron begeven. De schakeling die dit alles verzorgt bestaat uit weinig meer dan twee infrarood transistoren en een eenvoudige zender.

## Het laadstation

### De lichtbron

Om de ontvanger eenvoudig te houden stoppen we wat meer energie in de zender. We gebruiken vier in serie geschakelde hoogvermogen IR-LED's van 375 mW/sr bij 250 mA, zie onderstaand schema.

*De robot wordt naar het basisstation gelokt door het licht dat door vier IR-LED's wordt uitgestraald*



De stroom door de hoogvermogen LED's is 250 mA. Dat is wat de voeding betreft geen probleem. De zender is immers onderdeel van het "grondstation" dat uit het net wordt gevoed. De negen weerstanden zijn normale kwart watters. Die kunnen uiteraard ook vervangen worden door één 2 W weerstand van 36 Ω.

### Het laadstation

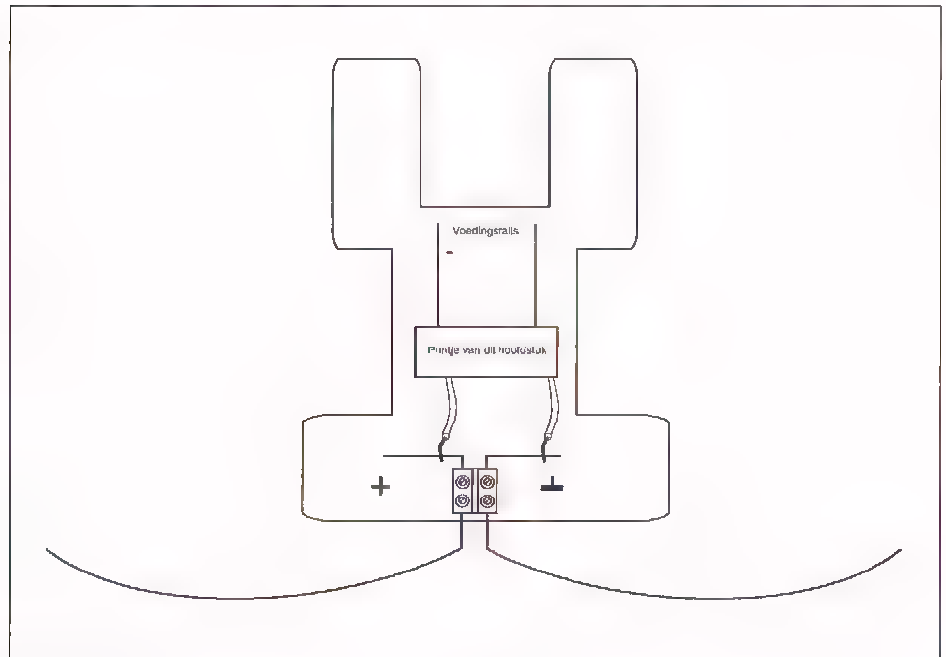
De "zender" oftewel het laadstation verdient nog enige aandacht. Als we vier uur willen laden kunnen we een elektronische tijdvertraging bouwen met heel grote elco's of een aantal digitale tellers. Veel eenvoudiger is een simpele schakelklok zoals ze tegenwoordig al voor minder dan EUR 6,00 in de bouwmarkten te koop zijn. Een 9 V netsteker van 500 mA completeert het geheel. Als de netsteker een andere spanning afgeeft, moet de serie weerstand van de zender-LED's evenredig aangepast worden. We maken de zender vier uur actief en zetten hem dan een tijdje uit.

In de tijd dat de zender geen licht geeft scharrelt de robot wat rond. Zodra de accu leeg begint te raken moet hij aan de voeding. De lengte van het "speelkwartier" is sterk afhankelijk van de motorstromen en de accuconditie. De speeltijd moet dan ook éénmalig bepaald worden. Het kan dus zijn dat de robot een half uur rondscharrelt en vier uur lang eet. Deze verhouding lijkt niet erg gunstig, maar het diertje kan dit ritme heel erg lang volhouden. We moeten wel nog het bestek aanreiken. Zonder bestek geen maaltijd.

**Het contact**

De robot is bedoeld als een echt experimenteerplatform. Verschillende printen worden eraan gehangen die allemaal voeding nodig hebben. Door een soort voedingsrail te maken lossen we een aantal problemen in één keer op. Een kroonsteentje komt aan de voorkant op de montageplaat waarop de vier tafelrandensensoren zich bevinden. Aan de achterzijde komen twee stukjes dik montagedraad die als voedingsrail dienen, terwijl aan de voorzijde twee voelsprietten komen. Die kunnen het best uit verenstaal gemaakt worden, zie onderstaande figuur. De voelsprietten dienen als bestek. Door het laadstation met dergelijke identieke sprietten uit te rusten, maar dan verticaal, kan de robot contact maken met zijn maaltijd.

*Twee “voelsprietten” uit verenstaal worden aan de voorzijde van het robotchassis bevestigd en maken contact met twee identieke sprietten aan het laadstation*



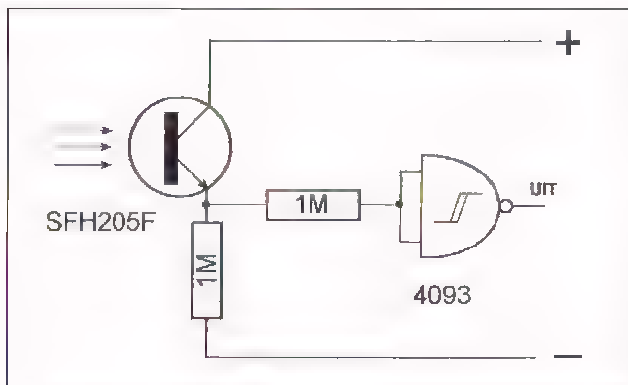
De twee voelsprietten worden aangesloten op de ingang van het printje dat in dit hoofdstuk wordt besproken. De twee voedingsrails worden verbonden met de uitgang van het printje. De twee accudraden worden rechtstreeks op de voedingsrail gesoldeerd. Hetzelfde doen we met de voedingsdraden van alle gebruikte en nog te gebruiken printjes.

**De ontvanger**

**Het principe**

Het principe van de ontvanger is de eenvoud zelve, zie onderstaande figuur. Een infrarood gevoelige transistor van het type SFH205F vangt het licht van de LED's op. Als er voldoende licht op de transistor invalt gaat deze volledig geleiden en staat de voedingsspanning over de weerstand van 1 MΩ.

*Het basisschema van een “oog” van de robot*



We bouwen de ontvangerschakeling twee maal. Door te kiezen voor het type SFH205F krijgen we een ingebouwd daglicht filter cadeau. Hierdoor wordt het geheel minder beïnvloed door omgevingslicht. Een MOS-IC met schmitt-trigger NAND-poorten maakt logische nullen en enen van het ontvangen signaal. De twee overgebleven poortjes worden niet gebruikt. De ingangen worden aan massa gelegd. Let wel, de uitgang is laag als de ontvanger licht ziet.

**De logica** Uiteraard moeten we vervolgens weer een logische schakeling ontwerpen, die uit de twee signalen van de "ogen" op de juiste manier de vier digitale signalen Q1 tot en met Q4 afleidt voor het besturen van de twee motoren.

Als laadspanning geconstateerd wordt, is het zaak dat de robot stil blijft staan. Dat betekent dat de motorprint sturingangen Q1 en Q2 tijdens het laden ongelijk moeten zijn. Hetzelfde geldt voor Q3 en Q4.

Als de linker IR-ontvanger licht ziet, maar de rechter niet, dient het voertuig zich naar links te bewegen. Zodra laadstroom gedetecteerd wordt gaat de robot op de plaats rust. Ziet geen van de cellen licht, dan is de voeding niet meer actief en kan de robot over gaan tot de orde van de dag.

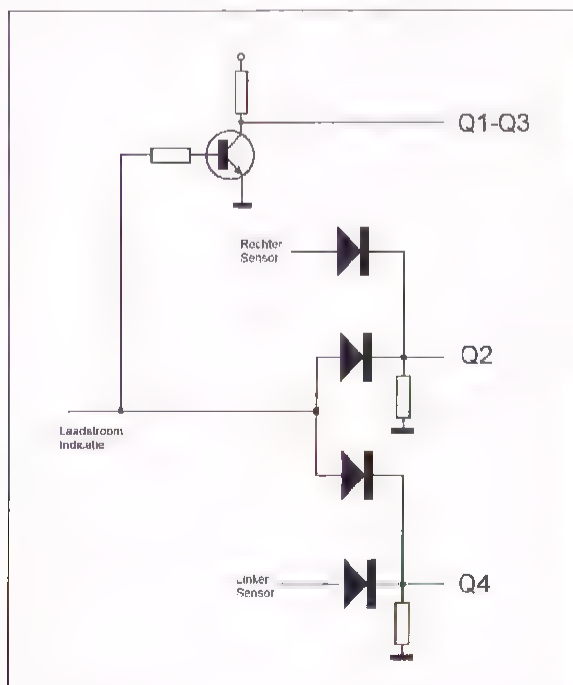
Hiervoor kan weer op de reeds bekende manier een waarheidstabel worden opgesteld, zie onderstaande figuur.

LS	R	LADEN	LINKER WIEL	RECHTER WIEL	VOERTUIG		
			Q1	Q2	Q3	Q4	
0	0	0	0	0	0	0	Vooruit
0	1	0	0	1	0	0	Rotatie om linker wiel linksom
X	X	1	0	1	0	1	Stilstand
1	0	0	0	0	0	1	Rotatie om rechterwiel rechtsom

**De waarheidstabel voor de besturing van de twee motoren**

Een schakeling die bovenstaande tabel tot leven brengt is eenvoudig, zie onderstaand schema. We passen, zoals we eerder gedaan hebben, weer een statische manier van programmeren toe. Een viertal dioden en wat weerstanden om de schakelingangen Q1 t/m Q4 laag te houden is alles dat we nodig hebben.

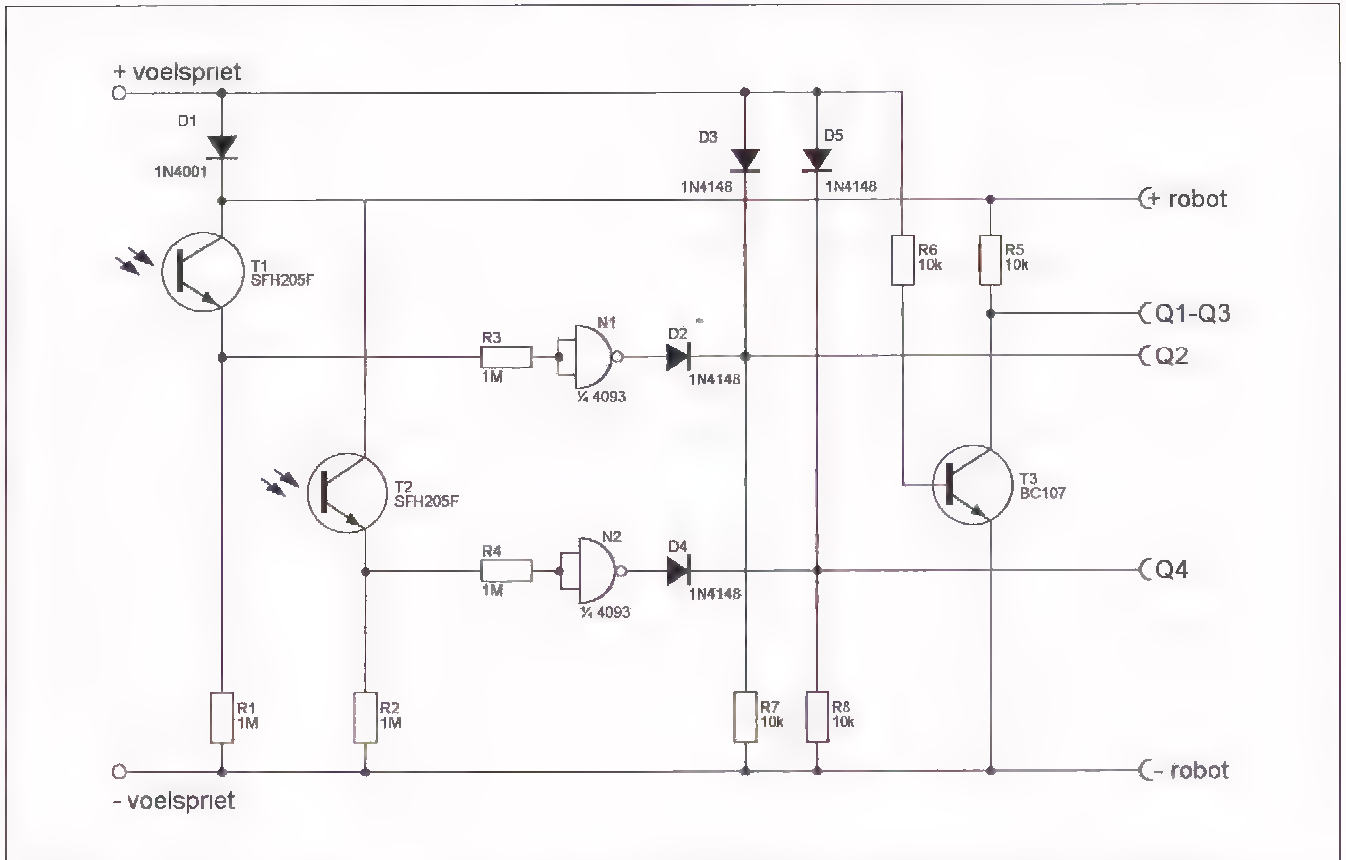
**De diodematrix die de waarheidstabel tot leven wekt**



Let wel, de linker en rechter sensor zijn laag als ze actief zijn, terwijl de laadindicator een hoog signaal af dient te geven als er laadstroom gedetecteerd wordt. Als het licht wegvalt, en daarmee ook de laadstroom, zijn alle Q-uitgangen hoog en de robot rijdt achteruit om zich van de lader te verwijderen.

## De elektronica

Het schema De volledige elektronica is samengevat in onderstaande figuur.



**Het volledig schema van de toegepaste elektronica**

De vier uitgangen Q1 tot en met Q4 moeten met de reeds vaker toegepaste R-2R schakeling omgezet worden in besturingssignalen voor de twee motorprinten. Deze R-2R netwerken zijn niet op de print aanwezig, omdat we in het volgende hoofdstuk een elektronische omschakelaar bespreken die wordt gestuurd door de Q-uitgangen. Als u die omschakelaar nabouwt moet de R-2R omzetter op de uitgang van de elektronische omschakelaar worden aangesloten. Als u de print zelfstandig gebruikt, moet u de weerstandjes even als "spinnenweb" achter de print zetten.

## ONDERDELENLIJST

**WEERSTANDEN, 1/4 W, 5 %**

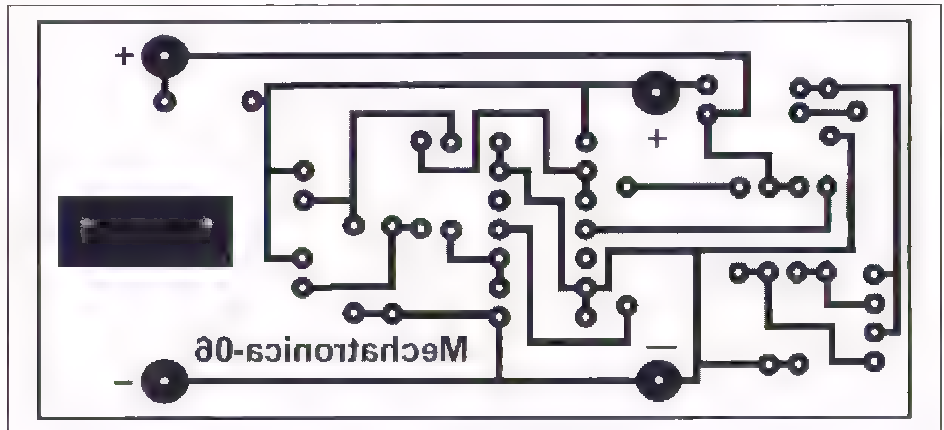
R1,R2,R3,R4 . . . . . 1 MΩ  
 R5,R6,R7,R8 . . . . . 10 kΩ

**HALFGELEIDERS**

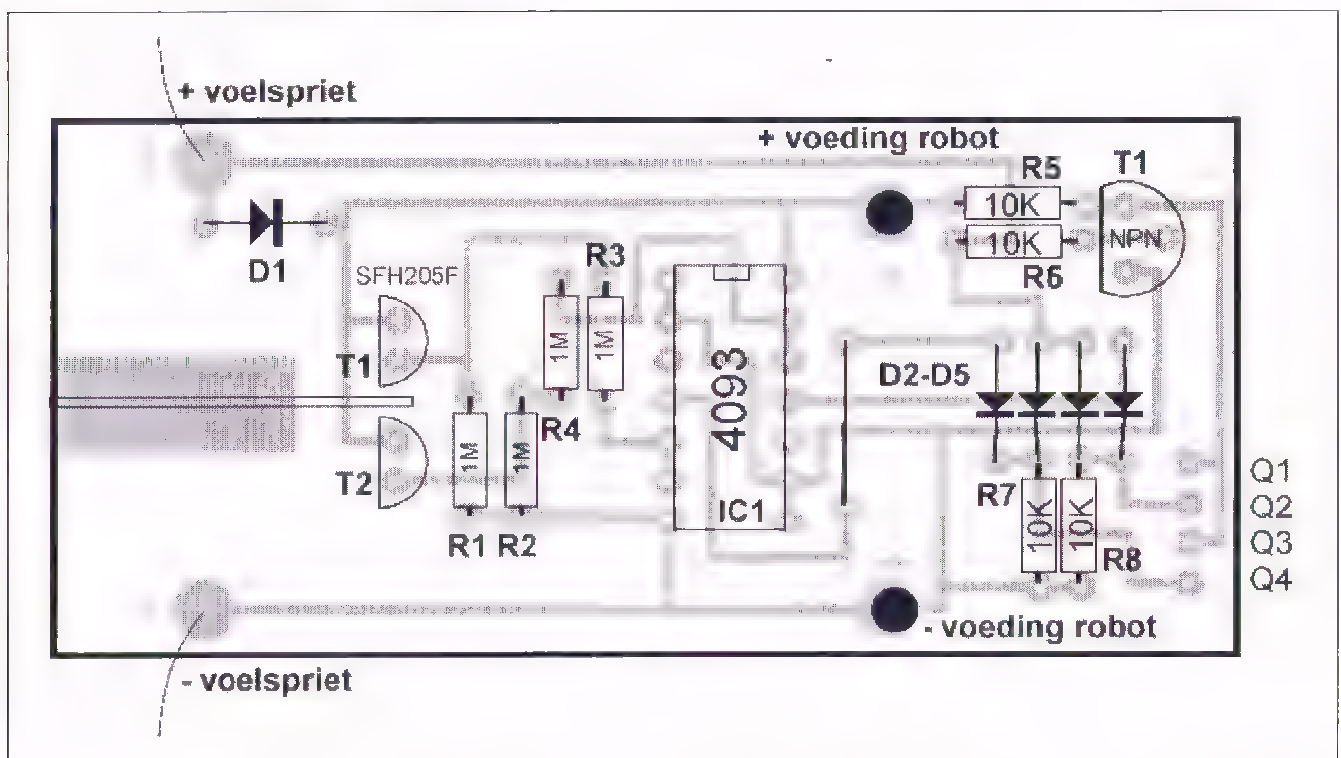
D1 . . . . . 1N4001  
 D2-D5 . . . . . 1N4148  
 T1,T2 . . . . . SFH205F  
 T3 . . . . . BC107, BC548  
 N1,N2 . . . . . CD4093

**De bouw** Ook voor deze schakeling is weer een print ontworpen, zie de twee volgende figuren.

**De print voor de schakeling**



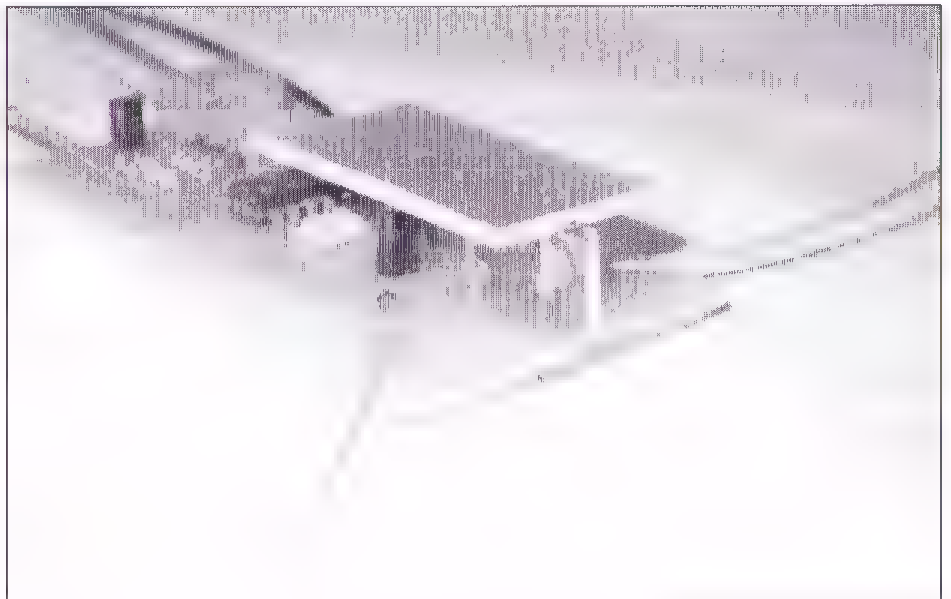
**De componentenopstelling voor de print**



Op de print is een kopervlakje aangebracht, zie de foto op de volgende pagina. Loodrecht hierop, precies tussen de IR-transistoren, wordt een stukje printplaat met twee draadjes vastgesoldeerd, precies zoals we al eerder deden bij het afschermen van infrarood licht sensoren. De schakeling hoort tot op ongeveer een halve meter probleemloos te reageren. Eventueel kan de waarde van de beide 1 MΩ weerstanden naar de massa verhoogd worden om de gevoeligheid te verhogen. Afhankelijk van de hoeveelheid invallend vreemd licht kan tot een weerstandswaarde van 10 MΩ gegaan worden. Wel moeten de ontvangsttransistoren in dat geval goed afgeschermd worden. Hiertoe kan op het lichtschot een stukje printplaat gesoldeerd worden, evenwijdig aan de print. Zo ontstaat een goede afscherming voor van boven invallend licht.



***De afscherming rond de twee IR-gevoelige transistoren***



## **Epiloog**

We kunnen de laadcyclus het best eerst uitvoeren zonder dat de tafelandetectorprint aangesloten is. We maken, onafhankelijk van de tafelandetector, een opstelling die de robot naar de voeding dirigeert. In het volgend hoofdstuk zullen we beide functies combineren.

# 8 We gaan combineren

## Inleiding

### Afzonderlijke schakelingen combineren

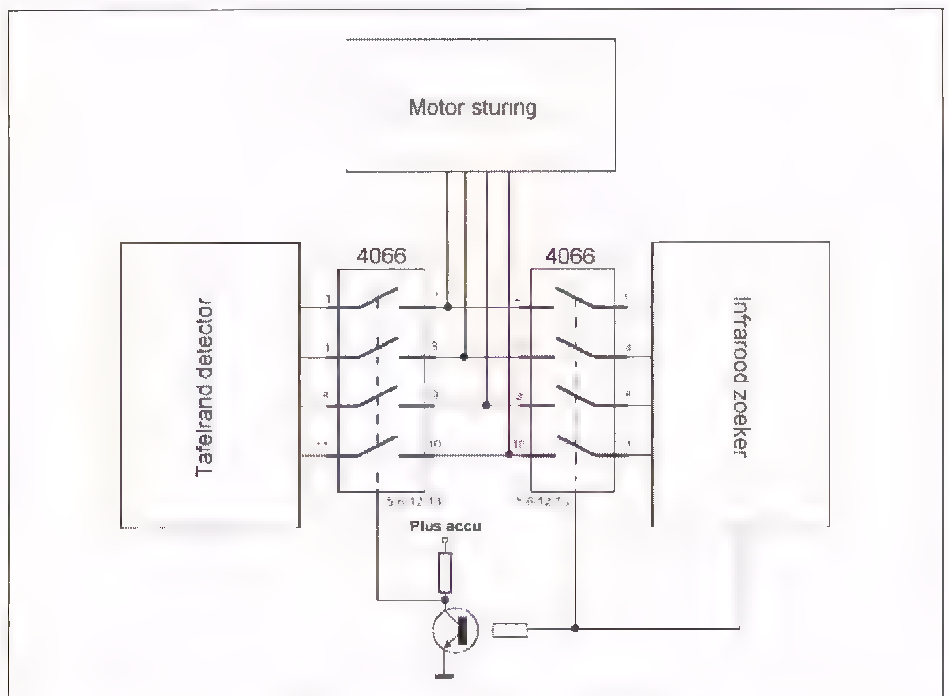
In de vorige hoofdstukken hebben we een scharreelprogramma ontworpen dat er voor zorgt dat de robot niet van de tafel stort. Bovendien hebben we een maaltijdmodule ontwikkeld. Beide schakelingen zijn afzonderlijk uitgevoerd. Het is natuurlijk veel aardiger als de robot zélf kan bepalen welk programma van toepassing is en zélf een keuze maakt.

## De schakeling

### CMOS-schakelaars

Onderstaande schakeling maakt gebruik van CMOS-schakelaars om tussen de twee programma-modulen te kiezen. Als er infrarood licht gedetecteerd wordt sluiten de schakelaars in de rechter 4066 en zoekt de robot de lichtbron op. Valt het IR weg na de laadcyclus, dan worden de schakelaars in de rechter 4066 geopend en in de linker gesloten. De robot heeft gegeten en gaat weer op pad.

*Met behulp van elektronische schakelaars kunnen de motoren door een van de schakelingen uit de vorige hoofdstukken worden bestuurd*



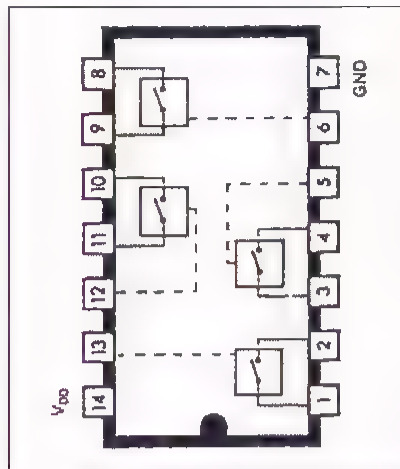
### De 4066

We gebruiken elektronische schakelaars uit de CD4xxx-serie, om precies te zijn het type CD4066. Eén zo'n IC bevat vier enkelpolige elektronische schakelaars. Iedere schakelaar wordt bediend door een eigen stuursignaal. Is de spanning op deze stuuringang "L", dan staat de schakelaar open. Is de stuurspanning gelijk aan de voedingsspanning, dan sluit de schakelaar. In principe is de werking van dit IC dus te vergelijken met de AAN/UIT-schakelaar waarmee wij het licht bedienen. Alleen kunnen de schakelaars uit de CD4066 maar kleine spanningen en stromen schakelen. Maar voor het omschakelen van de besturingsignalen van onze robot is dit geen probleem.

### Aansluitgegevens

Wij geven geen print voor deze schakeling, het geheel kan op een stukje gaatjespertinax worden opgebouwd. De aansluitgegevens van de CD4066 (bovenaanzicht!) zijn gegeven in onderstaande figuur. De GND (pen 7) wordt met de massa van onze robotelektronica verbonden, de  $V_{DD}$  (pen 14) met de accuspanning.

**De aansluitgegevens  
van de CD4066 in  
bovenaanzicht**



## Epiloog

In de achter ons liggende hoofdstukken hebben we een zeer beweeglijk chassis een stukje zelfstandigheid gegeven door aan vier stuur-ingangen logische signalen aan te bieden. Deze signalen werden bepaald door sensoren. De stuurprogramma's werden vrij statisch vastgelegd in een diode-decoder. Als we de complexiteit van de robot willen opvoeren is dat een omslachtige methode, die voor elke functie een aparte decoder vraagt en bovendien een schakeling om tussen de diverse programma's te kunnen schakelen.

Als kennismaking met het bewegende onderstel en de mogelijkheden hiervan was dat een uitstekende benadering. Nu de complexiteit gaat toenemen zijn nu toch zover dat we de robot op een meer intelligente manier willen aansturen. En dan ontkomen we niet aan de microprocessor. In de volgende hoofdstukken zullen we zien hoe we met een low cost microprocessor alle logische beslissingen in software beschrijven. Afhankelijk van de programmeervaardigheid kan de robot extra slim gemaakt worden. Maar schrik niet, we nemen kleine stapjes met reeds geschreven software.

Door het wijzigen van de software kunnen functies, en daarmee het robotgedrag, relatief eenvoudig gewijzigd worden.

# 9 Onze robot krijgt $\mu$ P-hersenen

## Inleiding

### Intelligentie in BASIC

Vanaf nu beperkt uw vaardigheid zich voornamelijk tot het boren van gaatjes en het vastschroeven van onderdelen. Na het eerste deel niet iets om echt bang voor te zijn. Daarnaast solderen we enkele eenvoudige printjes, tenzij die nog op de werktafel liggen als resultaat van het eerste deel. De belangrijkste en meest gecompliceerde print, deze met de microprocessor, is volledig voorgemonteerd. De belangrijkste nieuwe vaardigheid die in dit deel aan bod komt betreft het programmeren van een microprocessor. Om ook de beginnende programmeur/elektronicus snel op weg te helpen, is gekozen voor een systeem dat zich laat aanspreken met BASIC, de eenvoudige programmeertaal die een ieder zo onder de knie heeft.

### Programmeren via de PC

In de volgende hoofdstukken gebruiken we een microprocessor. Een microprocessor wordt geprogrammeerd en dat betekent dat we de beschikking moeten hebben over een computer. Geen 2,4 GHz model, in feite is elke computer (ook een tweedehands afdankertje) geschikt, zolang er maar een Windows-versie op draait.

### Welke microprocessor?

In microprocessorland is de keuze overweldigend. Voor elk type zijn ontwikkelkits beschikbaar en de logische bouwblokken volgen elkaar snel op. Enkele jaren geleden was 8 bit de standaard, nu zijn 16 bit of zelfs 32 bit processors meer gebruikelijk.

We doen voor dit boek een behoudende keuze. We kiezen voor een microprocessor die al vele jaren in de handel is (niet de modernste dus) en die in een eenvoudig apentaaltje (BASIC) geprogrammeerd kan worden. We gaan er in dit werkboek overigens van uit dat programmeren voor de gemiddelde lezer nog abracadabra is.

Het gekozen IC wordt ondersteund met ontwikkelkits, CD's en boeken (zij het in de Duitse taal) voor het geval u de smaak te pakken krijgt. Bovendien is er een printje beschikbaar waarop alle functies gemakkelijk toegankelijk zijn. De kit wordt geleverd door Conrad ([www.conrad.nl](http://www.conrad.nl)) en heet daar C-Control Basic (bestelnummer 95 05 72). De kosten?

Ongeveer € 50,00. Let op, er is ook een M-uitvoering. Die is wel goedkoper, maar daar moet nog veel aan gesoldeerd worden. Er is ook een uitvoering waarbij de functies (schakelaars, motortjes, LED's, sensoren) als symbooltjes op een ontwerpvel met elkaar verbonden kunnen worden. Een geniale vondst, maar toch blijken eenvoudige programmeeregels beter te begrijpen, met name als we na wat langere tijd de draad weer willen oppakken en het schema ingewikkelder wordt. Bovendien geeft programmeren een goede training voor het geval we later nog eens echt willen programmeren in C+ of zo en gebruik willen maken van modernere elektronica.

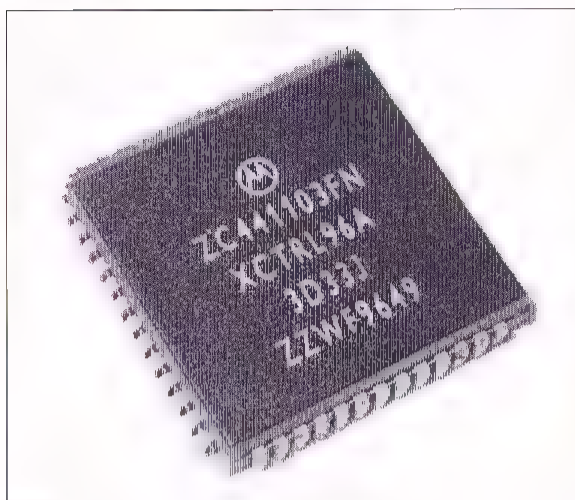
### De MC68HC05B6

De gebruikte processor is een 8 bit versie MC68HC05B6 van Motorola met een niet-vluchtig geheugen van 8 kB (1 kilobyte is 1.024 bit, waarbij een bit een "0" of een "1" kan voorstellen). Dat lijkt niet veel in een tijd dat we geheugens in MB (1.000.000 byte) en GB (1.000.000.000 byte) uitdrukken, maar voor ons doel meer dan we ooit zullen gebruiken. Niet-vluchtig wil zeggen dat het programma onthouden wordt, ook als de spanning er af is. Als de robot geslapen heeft weet hij dus nog precies hoe zijn wereld er uit ziet en zal hij nog steeds reageren op sensoren op de wijze die hem aangeleerd is.

**In- en uitgangen**

Onze microprocessor heeft 16 in- en/of uitgangen (naar wens te kiezen) die schakelen naar 5 V (= "1") en 10 mA kunnen verwerken. Bovendien hebben we de beschikking over 8 analoge ingangen en 2 analoge uitgangen. Deze kunnen 8 bit brede informatie verwerken. Dat wil zeggen dat ze een spanning van 5 V kunnen verdelen in 2 tot de macht 8 ( $2^8$ ), ofwel 256 stapjes. De processor kan dus acht spanningen meten (bijvoorbeeld sensorsignalen) en twee spanningen genereren (bijvoorbeeld een lamp én een motor regelen).

**De MC68HC05B6 microprocessor van Motorola wordt de hersenen van onze robot**

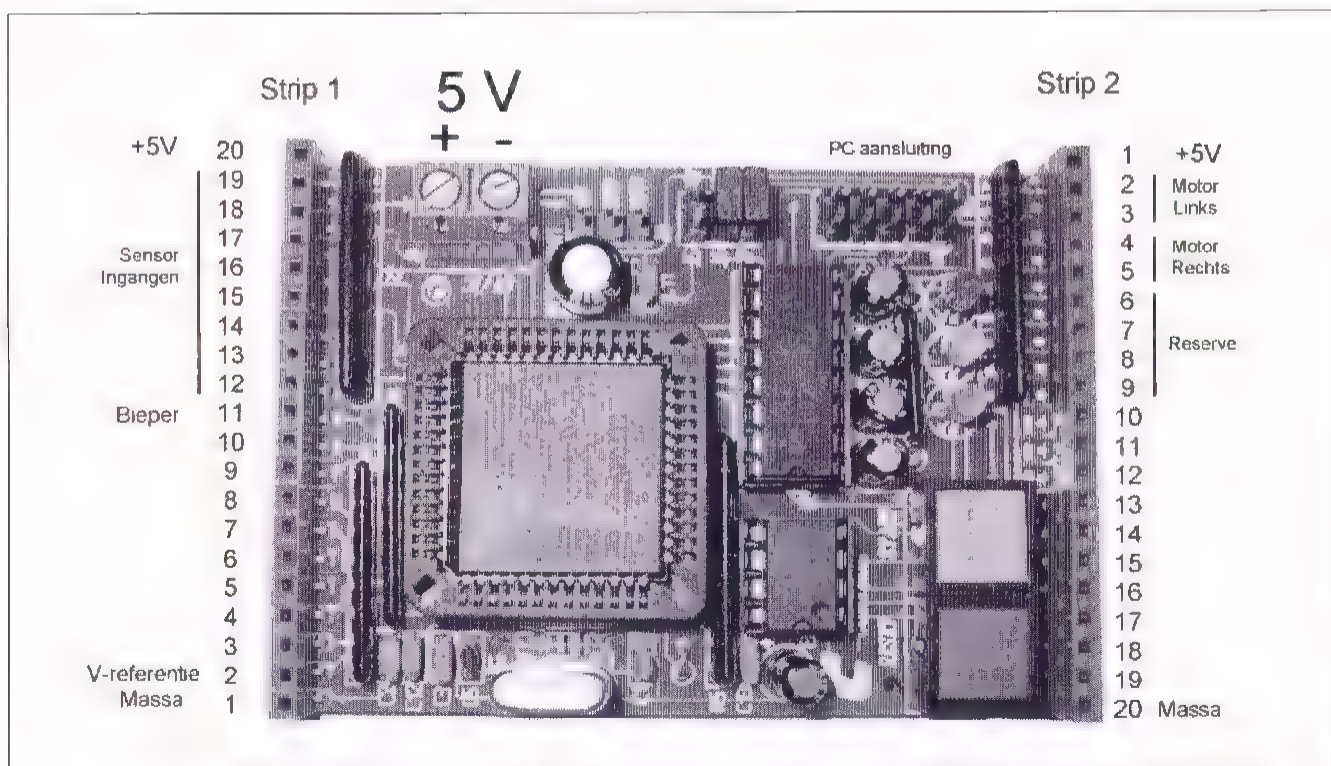


Hij heeft nog veel meer functies aan boord waarvan we de meeste in dit boek echter niet zullen gebruiken.

**De processorprint**

**Inleiding**

De processor zélf is, met wat omringende elektronica, door de leverancier al op een keurig printje gezet.



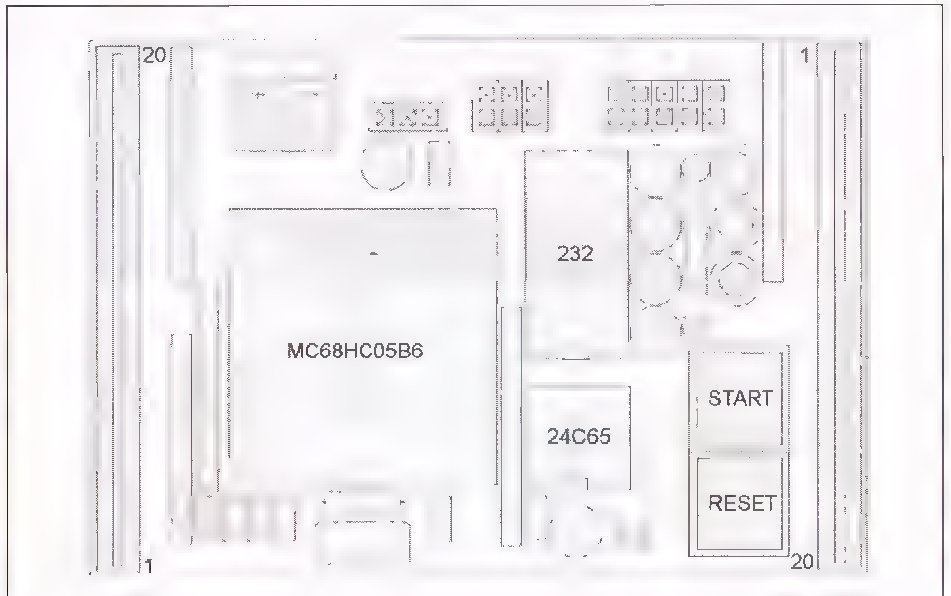
**De processor is, met wat andere elektronica, op een printje gezet en kan dus zó worden ingezet zonder soldeerwerk**

Aan weerszijden zit een aansluitconnector met 20 contacten. Daarmee krijgen we toegang tot alle signalen die de microprocessor levert. Boven op de print zit een kroonsteentje waar we de + en de - van de 5 V voeding kunnen aansluiten. Rechts boven zit een connector waarmee we de print met onze PC kunnen verbinden voor het programmeren van de processor.

**Schematische voorstelling**

Omdat we teveel worden afgeleid door alle details in een foto wordt zo'n processorprint meestal schematisch voorgesteld en zien wij duidelijk de belangrijkste onderdelen, zoals de twee drukknoppen "START" en "RESET" en de connectoren.

**De schematische voorstelling van de processorprint**



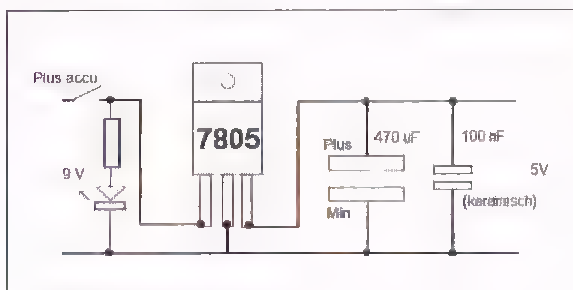
**Het toegankelijk maken van de processorprint via gaatjesboard**

De processormodule heeft, zoals reeds geschreven, twee parallelle rijen van 20 aansluitingen met een normale gaatjesprint steek. Het beste kunnen we gebruik maken van een gaatjes printplaat met eilandjes. De contacten in de connectoren van de processorprint lopen door. De processorprint kan daardoor op twee pennenstrippen worden gestoken. De pennen van deze strippen moeten wel iets langer zijn dan gebruikelijk. Een lengte van 15 mm is geschikt. Als naast deze strippen met lange pennen een uitvoering met normale, kortere, pennen geplaatst wordt op de posities van de digitale poorten, kunnen we later aansluitingen solderen, zonder dat de processorprint in gevaar komt. Enkele hoofdfuncties zoals de 5 V regelaar brengen we eveneens op de gaatjesprint aan. Let bij het solderen heel goed op en controleer alle lassen met een vergrootglas. De afstand tussen de kopersporen is bijzonder klein en een kortsluiting is snel gemaakt.

**De 5 V voeding**

De benodigde 5 V processorspanning wordt afgeleid van de accu's voor het loopwerk. De regeling is klassiek en neemt niet veel ruimte in op de print.

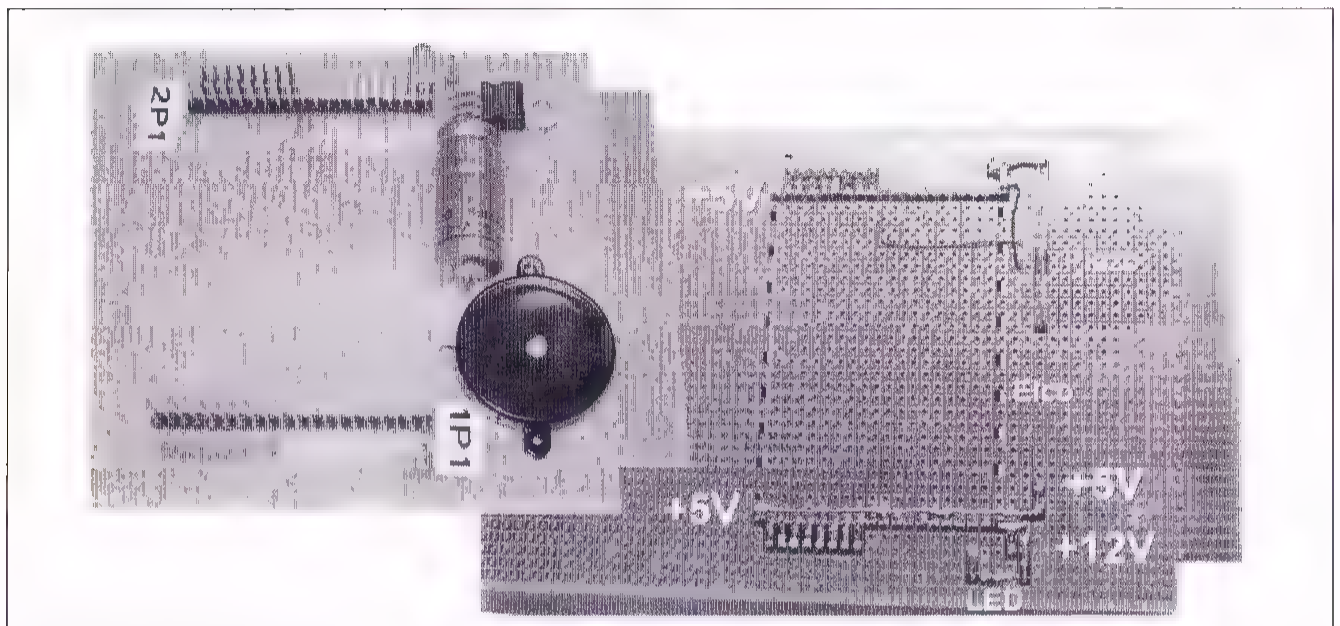
**Met maar vijf onderdeeljes zetten we de 9 V van de accu's van het loopwerk om in een mooie 5 V spanning voor de processorprint**



De weerstand links boven dient voor de stroombegrenzing van de lichtgevende diode (LED) en heeft een waarde van  $470 \Omega$ .

**De kant-en-klare montageprint**

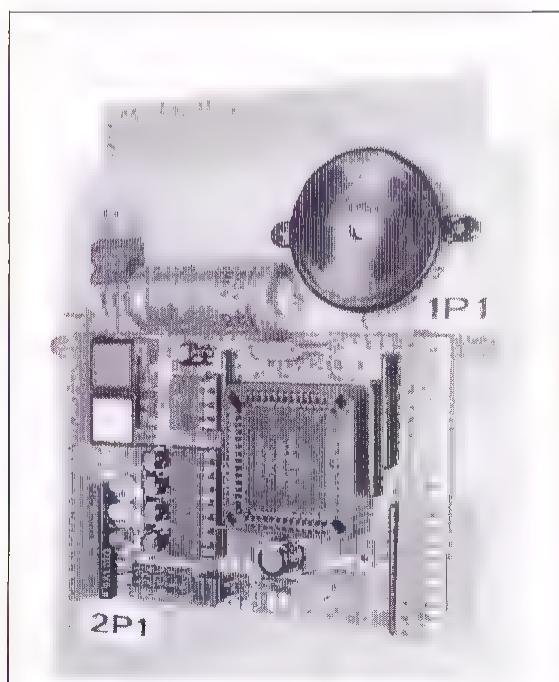
De stippellijnen geven doorverbindingen aan op de processorprint zelf. Een LED op de hoofdoeding en een piëzo-zoemer (een type zonder ingebouwde elektronica) maken het geheel compleet. Het is niet verstandig de hoofdschakelaar op de gaatjesprint te monteren. Daarvoor is een dergelijk plaat te kwetsbaar. Het is beter de hoofdschakelaar een plaatsje te geven op de chassisplaat. Onderstaande foto geeft een impressie van de boven- en onderzijde van de gaatjesprint vóórdat de processorprint werd aangebracht. Duidelijk zijn de twee connectoren te zien, waarop we laten de processorprint monteren.



**De gaatjesprint is klaar voor het ontvangen van de processorprint**

Met de processorprint over de aansluitpennen geschoven ziet het er redelijk indrukwekkend uit. Let erop dat het voedingskroonsteentje op de processorprint niet gebruikt wordt. Alle noodzakelijke verbindingen zijn al op de gaatjesprint gemaakt.

**De processorprint is op de connectoren van onze gaatjesprint aangebracht**



**De aansluitgegevens**

Via het gaatjesboard krijgen we op een veilige manier toegang tot alle in- en uitgangssignalen van onze microprocessor. Maar dan moeten we natuurlijk wél weten wat die veertig contactjes te betekenen hebben. Welnu, in onderstaande tabel zijn de functies van alle pennen van de twee connectoren samengevat.

Connector 2		Connector 1	
Pen 20	Massa	Pen 1	Massa
Pen 19	Reset	Pen 2	V ref
Pen 18	DCF-77	Pen 3	A/D [1]
Pen 17	D/A [2]	Pen 4	A/D [2]
Pen 16	D/A [1]	Pen 5	A/D [3]
Pen 15	Groene led	Pen 6	A/D [4]
Pen 14	Rode led	Pen 7	A/D [5]
Pen 13	Gele led	Pen 8	A/D [6]
Pen 12	Intern	Pen 9	A/D [7]
Pen 11	Intern	Pen 10	A/D [8]
Pen 10	Start	Pen 11	Zoemer
Pen 9	Poort [8]	Pen 12	Poort [9]
Pen 8	Poort [7]	Pen 13	Poort [10]
Pen 7	Poort [6]	Pen 14	Poort [11]
Pen 6	Poort [5]	Pen 15	Poort [12]
Pen 5	Poort [4]	Pen 16	Poort [13]
Pen 4	Poort [3]	Pen 17	Poort [14]
Pen 3	Poort [2]	Pen 18	Poort [15]
Pen 2	Poort [1]	Pen 19	Poort [16]
Pen 1	+ 5 volt	Pen 20	+ 5 volt

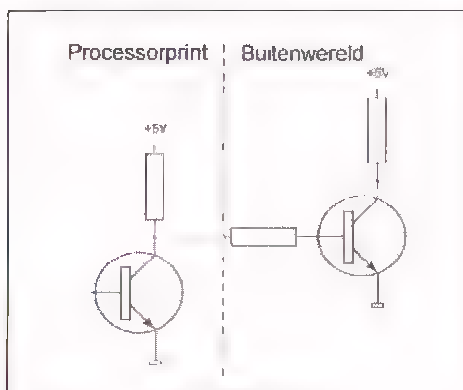
*De functie van de 40 pennen van de twee connectoren*

**Nog enkele tips**

**Afsluiting processoruitgangen**

Alle uitgangen van de microprocessor worden op de processorprint met een weerstand omhooggetrokken. Zolang er geen opdracht is geweest zijn ze dus hoog ("H"). Dat kan soms lastig zijn. De wielen van het loopwerk bijvoorbeeld, draaien dan zodra de spanning op de schakeling gezet wordt. Door een transistortrapje tussen de processoruitgang en het R-2R netwerk (zie hoofdstuk 11) te zetten, is de beginconditie laag ("L") en staan de wielen stil.

*Via een eenvoudige transistorinverter kunnen wij de processoruitgangen in rust "L" maken*

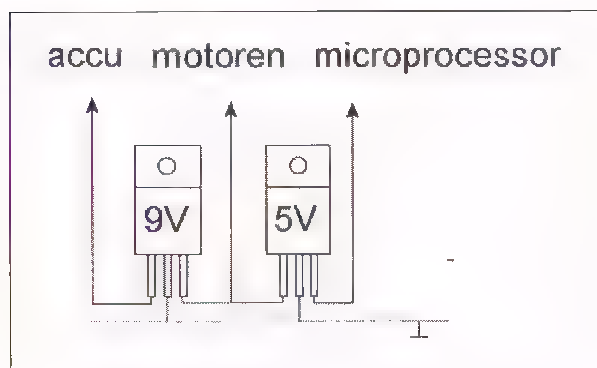




**De voedingsstabiliteit**

Een andere zaak is de voedingsstabiliteit. Naarmate de accu's leger worden daalt de voedingsspanning en worden de motorbesturingsprinten met een steeds lagere spanning gevoed. Tot nu toe was dat geen punt. Met microprocessorbesturing kan dat een probleem worden. De processor wordt immers gevoed uit een gestabiliseerde spanning van 5 V. Deze spanning blijft constant, ook als de accuspanning daalt. Zoals uit hoofdstuk 11 zal blijken, kan de dalende accuspanning een conflict veroorzaken met de stabiel blijvende processorvoeding. Als de hobby serieuze vormen dreigt aan te nemen, is het beter tussen de accu's en de schakeling een 9 V stabilisator op te nemen. Het euvel is hiermee verholpen.

*Met deze uitbreiding van de voedingsschakeling wordt niet alleen de processor stabiel gevoed, maar ook de overige elektronica van de robot*



**Epiloog**

Voor we nu verder gaan met het aansturen van het loopwerk, is het aardig eerst wat eenvoudige microprocessor ervaring op te bouwen. Het programmeren van de processor gaat op een heel eenvoudige manier. We sluiten een kabeltje aan tussen de processorprint en een seriële uitgang van onze PC waar Windows op draait. Maar voor dat werkt zullen we de software moeten installeren. Hoe dat gaat zien we in het volgende hoofdstuk.

# 10 Installeren van C-Control BASIC

## Inleiding

### Bij de microprocessor geleverde CD-ROM

De bij de microprocessor geleverde CD-ROM bevat programma's voor een aantal systemen. De CD-ROM start vanzelf op. Maak op de harde schijf alvast een tijdelijke map CControl aan. Een andere naam mag ook. Start vervolgens de CD-ROM en kies achtereenvolgens voor:

**C-Control I (/BASIC,/PLUS, Station)**

**C-Control/BASIC**

**Opslaan**

**Kies de zojuist aangemaakte directory**

Op deze map wordt vervolgens het bestand CCBasicSetup opgeslagen. De CD-ROM kan er uit.

### Installeren van het programma

Zoek de map op en start het setup-bestand door er (dubbel) op te klikken. Bevestig de vraag om door te gaan en druk vervolgens op "Weiter". We zien nu dat setup een map voor de installatie voorstelt.

We accepteren die keuze door "Weiter" te kiezen en nog een keer "Weiter" om de volledige installatie te kiezen. Nogmaals op "Weiter" en vervolgens op "Installieren". Een laatste druk op "Fertigstellen" en het programma is geïnstalleerd.

Kijk maar in C:\Program Files\Conrad Electronic\CCBasic\CCEW32d Daar staat de toepassing CCEW32D. Een klik rechts op de muis stelt ons in staat een snelkoppeling te maken die we naar het bureaublad slepen.

### Opstarten van het programma

Na het starten van het programma zien we een akelig leeg scherm. Gelukkig staat er ook een aantal voorbeeldprogramma's in de directory CCBasic. We openen in de map "station" het programma "schule.bas". Hier zien we een typisch voorbeeld van een BASIC-programma, zij het van commentaar in het Duits voorzien.

Daar gaan we uiteraard enige verandering in aanbrengen. De lijst kan bewerkt worden alsof het een normaal tekstbestand is.

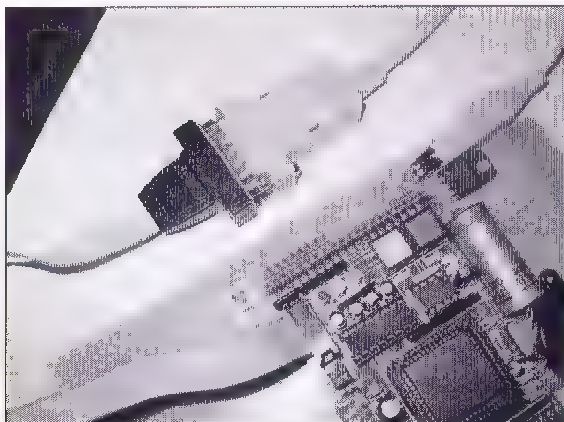
Let erop dat het kleinste foutje in een programmaregel een fout oplevert. Fouten in de commentaarregels vormen geen probleem, zolang er een ' voor staat.

## De koppeling met de PC

### Seriële kabel aansluiten

Sluit de meegeleverde seriële kabel aan op een COM-poort op de computer (COM1 of COM2). Het andere einde wordt verbonden met de lintkabel die op zijn beurt weer verbonden is met de print.

### Het aansluiten van de seriële kabel op de microprocessorprint



## CCBASIC programmeren

**Inleiding** Programmeren is een heel precieze klus. Door programma's van duidelijk commentaar te voorzien kunnen we ook later weer snel begrijpen wat het programma doet.

**Een eenvoudig voorbeeld** Omdat we nog niet veel hardware hebben, behelpen we ons even met de zoemer. Met onderstaand programma testen we de werking van de combinatie soft- en hardware. Type het onderstaande programma als nieuw bestand in CCBasic.

```
*****
'CCBasic
'De zoemer op de print wordt geactiveerd met de BEEP opdracht
'BEEP toonhoogte, tijdsduur, pause tot volgende beep
'BEEP 250000/frequentie, lengte * 20 msec, pause * 20 msec

#opnieuw 'Hier springt het programma aan het eind naartoe

beep 700,10,0
beep 600,10,0
beep 500,10,0
beep 400,10,0
beep 300,10,0
beep 200,10,0
beep 100,10,0
beep 200,10,0
beep 300,20,0
beep 400,20,0
beep 500,20,0
beep 600,20,0
beep 700,20,0
beep 800,20,0

goto opnieuw
*****
```

**Zoemer aansluiten** We sluiten de zoemer aan op pin 11 aan de rechter zijde van bovenaf geteld. De tweede draad van de zoemer mag aan de massa of aan de voeding hangen.

**Hoe het werkt** Het zoemertje op de print gedraagt zich een beetje gecompliceerd. Het eerste getal geeft de toonhoogte weer via een omrekening. Als we bijvoorbeeld 440 Hz willen horen (standaard muziektone A), dan berekenen we het in te vullen getal met de formule:

$$\text{GETAL} = 250.000 / \text{Frequentie} = 250.000 / 440 = 568$$

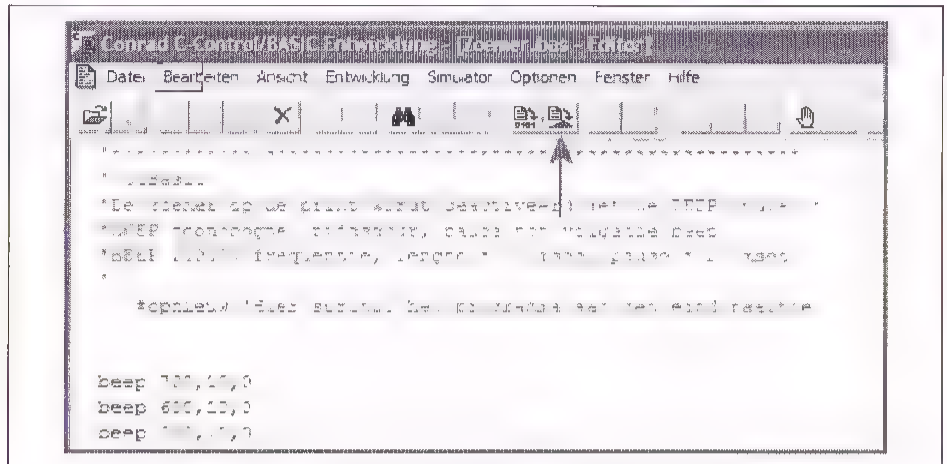
Hoe hoger de toon, hoe kleiner het getal. Het getal 700 in de eerste opdrachtregel geeft een frequentie van  $250.000 / 700 = 357$  Hz. Het tweede getal, gescheiden door een komma (na de komma mag een spatie gebruikt worden, hoeft niet) geeft de toonlengte weer in eenheden van 20 milliseconden.

Het getal 10 geeft een toonlengte van  $10 * 20 = 200$  msec, hetgeen overeenkomt met 1/5 seconde. Het derde getal geeft de pause aan tussen de volgende toon, ook weer in eenheden van 20 milliseconden. Als er nul staat, zoals hier, sluiten de tonen aan.

**Overbrengen naar de microprocessor** Nu het programma gereed is kunnen we het in voor de microprocessor begrijpelijke machinetaal overbrengen. Dit proces dat "compileren"

heet, wordt automatisch voor ons geregeld. Bovendien wordt tijdens dat proces het programma op fouten gecontroleerd. De compileertoets is met een pijl in de onderstaande afbeelding weergegeven.

**Met de compileertoets in de knoppenbalk wordt het BASIC-programma omgezet in een machinecode programma dat onze microprocessor kan begrijpen**



**Starten van het programma**

Als de aansluiting met de print in orde is en de print staat onder spanning, licht het rode LED'je op. Zodra dat weer dooft kan het programma gestart worden met de gele toets.

**Gaat er iets fout?**

Elektronica doet een aanslag op ons logisch denkvermogen, software doet dat nog meer. De combinatie van deze twee vormt een echte uitdaging en draagt wezenlijk bij tot het verhogen van het frustratie tolerantieniveau.

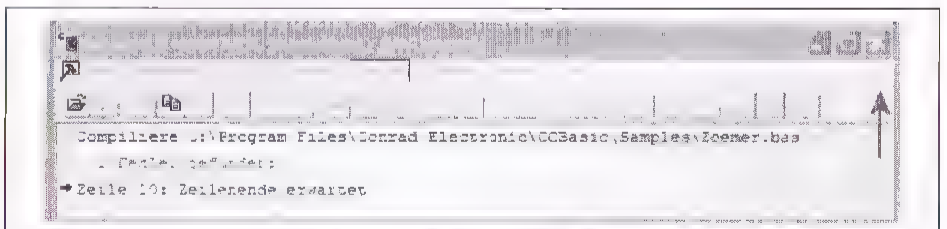
Een belangrijke eigenschap.

We laten ons dus niet ontmoedigen door tegenslagen. Als iets niet werkt is daar een oorzaak voor. Meet de voedingspanning op de processorprint na op de diverse pennen. Let erop dat de aansluitconnector 1 rechts zit en pen 1 bovenaan zit, dit in tegenstelling tot aansluitconnector 2 die onderaan begint met pen 1. Zit de zoemer inderdaad op pin 11 aan de rechter zijde van bovenaf geteld? De tweede zoemeraansluitdraad mag zowel aan massa als aan de +5 V gelegd worden. De uitgang van pen 11 geeft een blokgolf. Let goed op eventuele foutmeldingen en kijk of tijdens het overzetten de rode LED even brandt.

**Kleine oorzaken, grote gevolgen**

In onderstaand voorbeeld is in de eerste BEEP opdracht vóór het getal 700 een komma geplaatst. Dat is inderdaad op regel 10 als we alle regels (ook de lege) meetellen. We gaan terug naar het programma door het ónderste kruisje aan te klikken (zie pijl). Het rode kruisje sluit het programma zelf af.

**Het programma geeft een foutmelding in regel 10**



## Epiloog

Eerste programmeerklus geklaard? Probeer er dan eens een welluidend melodietje van te maken. Voor we verder gaan met de eigenlijke opdracht, het maken van een microprocessor gestuurd voertuig, komen we nog even terug op de besturing van de motoren.

# 11 Een alternatieve motorbesturing

## Inleiding

### De TLE4206 in een nieuwe schakeling

De aansturing van de motoren wordt ook bij de microprocessor besturing verzorgd door het IC dat in hoofdstuk 3 uitvoerig aan de orde is geweest, de TLE4206.

Even ter herinnering, de ingang heeft in feite drie hoofdcondities:

- Als hij (bijna) aan nul ligt draait de motor een bepaalde kant op.
- aan de voedingsspanning geschakeld draait de motor in de andere richting. Zorg ervoor dat deze richting overeenkomt met een achteruit rijdende beweging! Als dat niet het geval is moet de motor omgepoold worden.
- Op halve voeding staat hij stil.

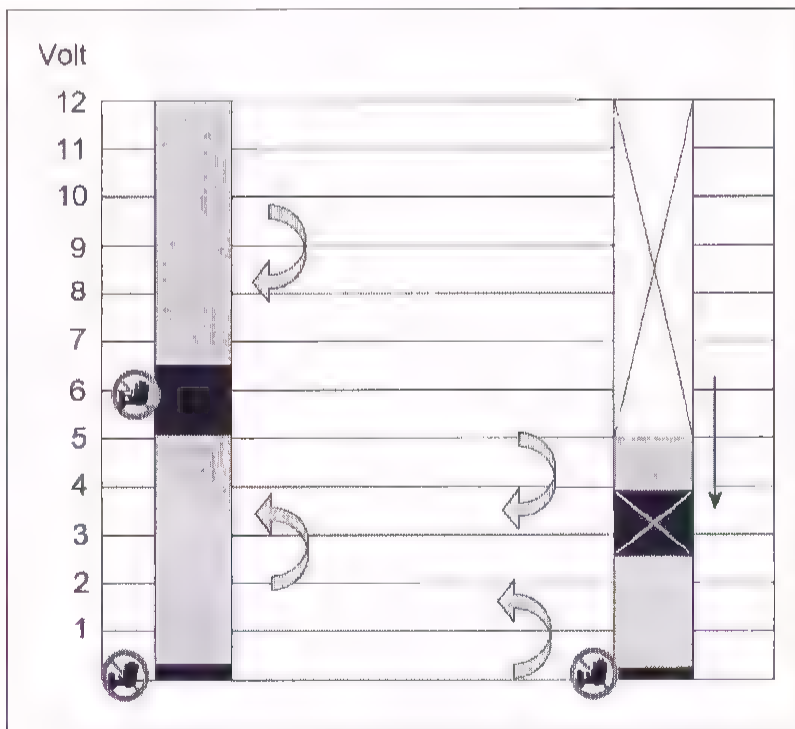
We hebben al gezien hoe we de motorsturing, die in feite tri-state is, toch met een digitaal signaal kunnen aansturen. Het verschil met hoofdstuk 3 is echter dat nu de ingangen Q1 tot en met Q4 door een programmeerbare microprocessor worden aangestuurd. En daar komt het eerste probleem om de hoek kijken.

### Spanningsverschillen

De motorstuurprint werkt vanaf 8 V, met 10 goed geladen NiCad accu's staat hij zelfs op 12 V. De processor werkt echter op 5 V. Het maximale digitale uitgangssignaal is bovendien slechts 4 V.

Hierdoor ontstaat een probleem dat in onderstaande figuur wordt verduidelijkt.

### Het ingangsspanningsbereik van de TLE4206 moet kunstmatig worden verlaagd



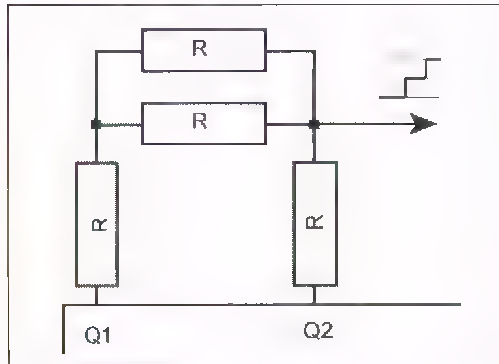
De linker kolom in de figuur laat de toestand zien zoals de ontwerper van het IC die bedoeld heeft. Boven 6 V aan de ingang draait de motor in een bepaalde richting, onder 6 V aan de ingang draait de motor in de andere richting. In het zwarte tussengebied staat de motor stil. Rond de nul zien we een dood gebied waarbinnen de motor ook stil staat. De microprocessor kan echter geen hogere signalen afgeven dan 4 V. Bovendien geeft een nul aan de uitgang een bijna echte nul.

**R-2R netwerk op de uitgangen**

Als we weer een R-2R netwerk gebruiken in combinatie met onze microprocessor zien we drie mogelijke condities ontstaan:

- Beide uitgangen hoog geeft ongeveer 4 V op de netwerkuitgang.
- Beide uitgangen laag geeft bijna 0 V op de uitgang.
- Eén van de uitgangen hoog en de andere laag geeft ongeveer 1,5 V tot 2,5 V op de uitgang.

*Ook nu moet de motorbesturingsprint via het welbekende R-2R netwerk worden aangestuurd vanuit de microprocessor uitgangen*



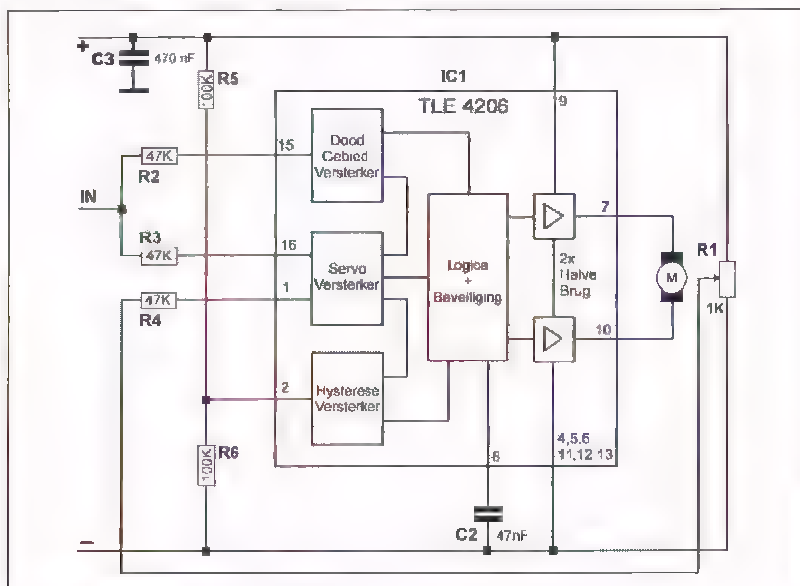
Deze spanningen liggen vast en zijn dus niet compatibel met de 6 V omschakeldrempel van de TLE4206. Gelukkig bestaat er een eenvoudig truukje om de TLE4206 geschikt te maken voor besturing met lagere spanningen op de ingang. In het originele schema van pagina 21 werd pin 1 gestuurd uit een spanningsdeler R7/R8 die deze pin op de helft van de voedingsspanning vastlegde. Als we die twee weerstanden vervangen door een instelpotentiometer kunnen we het schakelgebied omlaag draaien (rechter kolom in de figuur op pagina 59), zodat we met de spanning van 4 V aan de ingang juist in het rechts draaiend gebied zitten en met de spanning van 1,5 V tot 2,5 V juist in het links draaiend gebied. Het tussengebied gebruiken we niet omdat de processor daar, zonder extra ingrepen, geen spanningsniveau kan leveren. Als beide uitgangen laag zijn zitten we vanzelf in het dode gebied rond de nul. Dat wordt dan ook het nieuwe niveau waar de motor stil staat. De niet gebruikte delen op de spanningschaal in de rechterkolom zijn weggekruid.

**Het aangepaste schema**

**Nóg minder onderdelen**

Het oorspronkelijke schema van pagina 21 is dus niet meer bruikbaar als wordt gestuurd met een uit 5 V gevoede microprocessor.

*Het aangepaste schema van de motorbesturing als we met een microprocessor willen aansturen*



Het aangepaste schema op de vorige pagina blinkt uit door eenvoud. De twee weerstanden R7 en R8 zijn vervangen door een instelpotentiometer R1. Bovendien zijn de vertragende elementen R1 en C1 aan de ingang niet langer noodzakelijk. Een eventuele vertraagde reactie kan nu immers via de software worden ingebouwd.

<b>ONDERDELENLIJST</b>	
<b>WEERSTANDEN, 1/4 W, 5 %</b>	
R2,R3,R4 . . . . .	47 kΩ
R5,R6 . . . . .	100 kΩ
<b>INSTELPOTENTIOMETER, LIGGEND</b>	
R1 . . . . .	1 kΩ
<b>CONDENSATOREN</b>	
C2 . . . . .	47 nF
C3 . . . . .	470 nF
<b>HALFGELEIDER</b>	
IC1 . . . . .	TLE4206

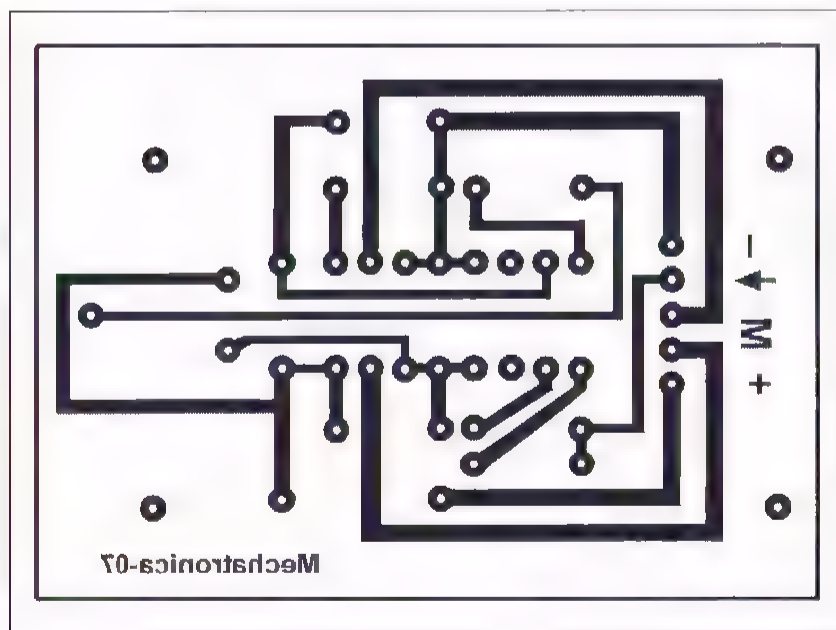
**De bouw van de schakeling**

Ook voor dit project hebben wij een printje ontworpen, waarvan er natuurlijk twee moeten worden nagebouwd. De componentenopstelling is getekend op de volgende pagina.

**De besturingsprinten in de robot**

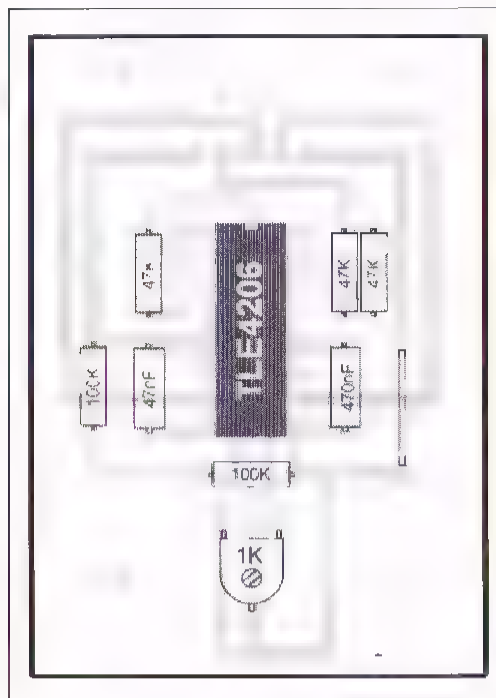
De twee motorbesturingsprinten kunnen op de reeds bekende manier onder het robotchassis worden gemonteerd. Wat aansluitingen betreft verandert er natuurlijk niets. Onze basisprint met de daarop gemonteerde microprocessorprint wordt dan aan de bovenzijde van het chassis gemonteerd.

**De print voor de schakeling**

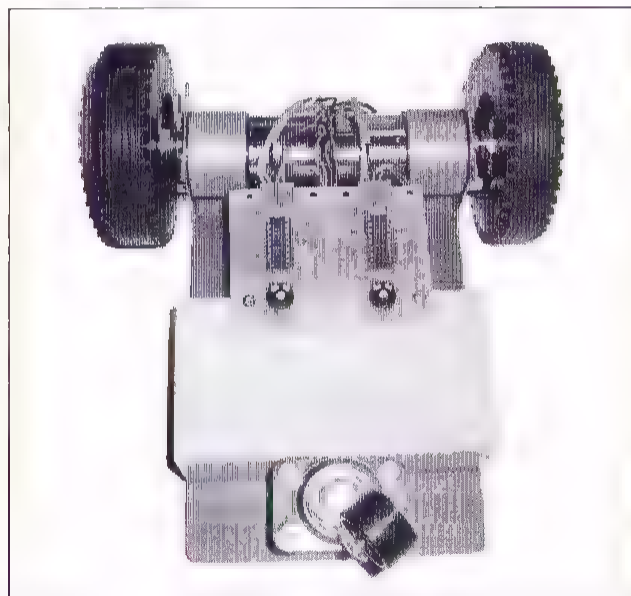


Nog even de aansluitingen aan de microprocessorprint. We maken een R-2R netwerkje op een heel eenvoudige manier. We solderen rechtstreeks aan de pennen van de digitale uitgangen, of gebruiken een verloopstekertje.

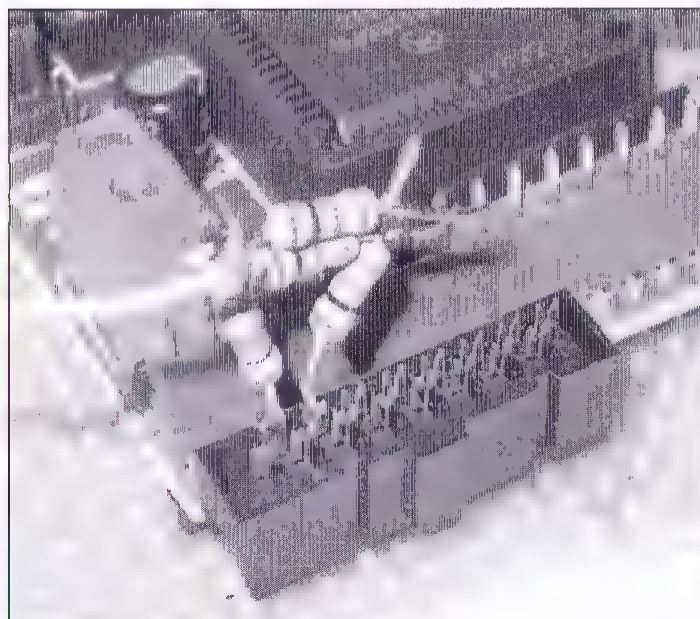
**De componentenopstelling van de aangepaste motorbesturing**



**De montage van de twee besturingsprinten voor de motoren aan de onderzijde van het robot chassis**



**Op deze manier kunnen we even snel een R-2R netwerkje op onze microprocessorprint aansluiten**





Het draadje aan de uitgang van het netwerkje gaat naar de motorstuuringang.

**Afregelen van de instelpotentiometer**

Het afregelen van de instelpotentiometer kan hardwarematig, maar ook met behulp van onderstaand programma.

```
*****
' Sluit een van de motorstuurdraden aan op de R-2R uitgang

define MotorRechts1 port[15]
define MotorRechts2 port[16]

#opnieuw 'aan het eind komt het programma hier terug

beep 500,20,0
'Beide uitgangen hoog betekent vooruit
MotorRechts1 = ON
MotorRechts2 = ON

'Beide uitgangen laag betekent stop
pause 100
MotorRechts1 = OFF
MotorRechts2 = OFF

pause 200

'Halve uitgangsspanning betekent achteruit
MotorRechts1 = OFF
MotorRechts2 = ON

pause 100

beep 300,20,0

'Idem met omgekeerde poorten betekent achteruit
MotorRechts1 = ON
MotorRechts2 = OFF

pause 100
goto opnieuw
*****
```

**Bespreking van het programma**

In de eerste regels krijgt de microprocessor te horen op welke aansluitpennen we een digitale uitgang willen hebben: [15] en [16]. Vervolgens zetten we beide uitgangen ON (=hoog). De motor hoort nu vooruit te lopen. In een tweede stap gaan beide uitgangen naar OFF (=laag) De motor stopt. De derde stap vraagt nauwkeurige afregeling van de potmeter op de motor print. Draai tegen de klok in tot de draairichting tegengesteld is aan die in stap twee. Het programma blijft de lus "goto opnieuw" doorlopen tot we het onderbreken. Let nog even op de PAUSE opdrachten. Ook dat getal moeten we met 20 milliseconden vermenigvuldigen. De PAUSE opdrachten bepalen hoelang de motor in de conditie die ervoor is vastgelegd blijft.

**En nu de tweede print**

Na geslaagde afregeling krijgt de tweede motorprint dezelfde behandeling. Het is handig het voertuig in deze fase op te bokken of op de rug

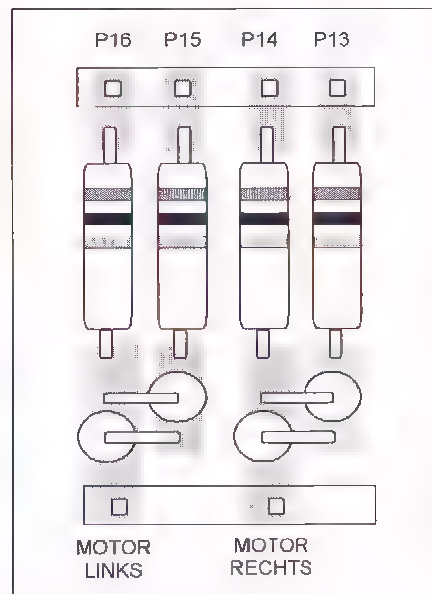
te leggen. Dat voorkomt dat het voertuig er vandoor gaat. Tien tegen een dat de tweede motor de verkeerde kant opdraait. Omdat de motor 180 graden gedraaid is moeten de aansluitdraden omgekeerd aangesloten zijn in vergelijking met de eerste motor.

## De R-2R netwerken

### Inleiding

Het lijkt voor de hand liggend de R-2R netwerkjes op de basisprint van de processor te solderen. Het is echter handiger hiervoor een aparte strook gaatjesprintplaat te gebruiken. Dat vermindert het beschadigingsrisico van de processor en maakt het solderen gemakkelijker.

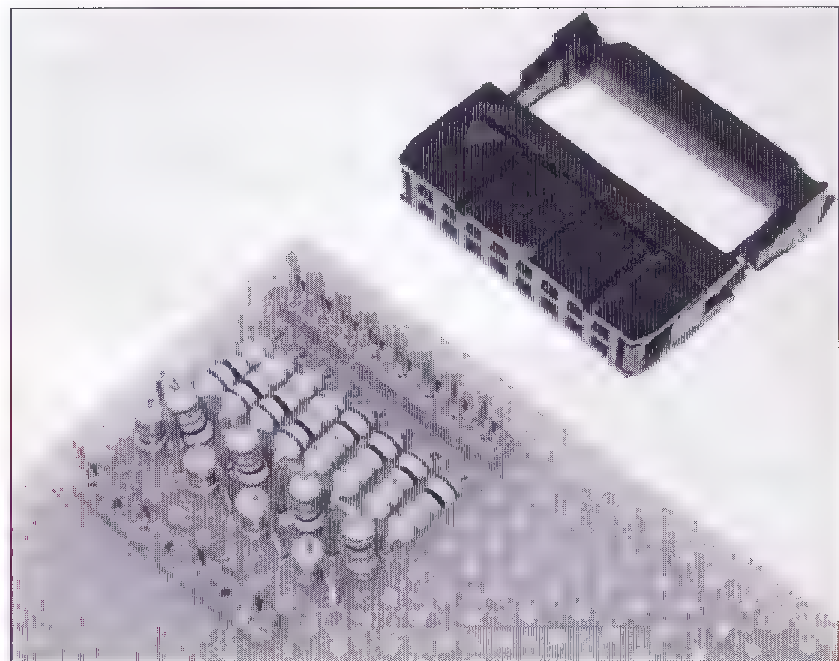
### *Twee R-2R netwerken op een strookje gaatjesprint*



### Gebruik van lintkabel

Pennenstrippen maken het gebruik van een platte lintkabel mogelijk. Gewoon verbindingdraadjes solderen mag natuurlijk ook.

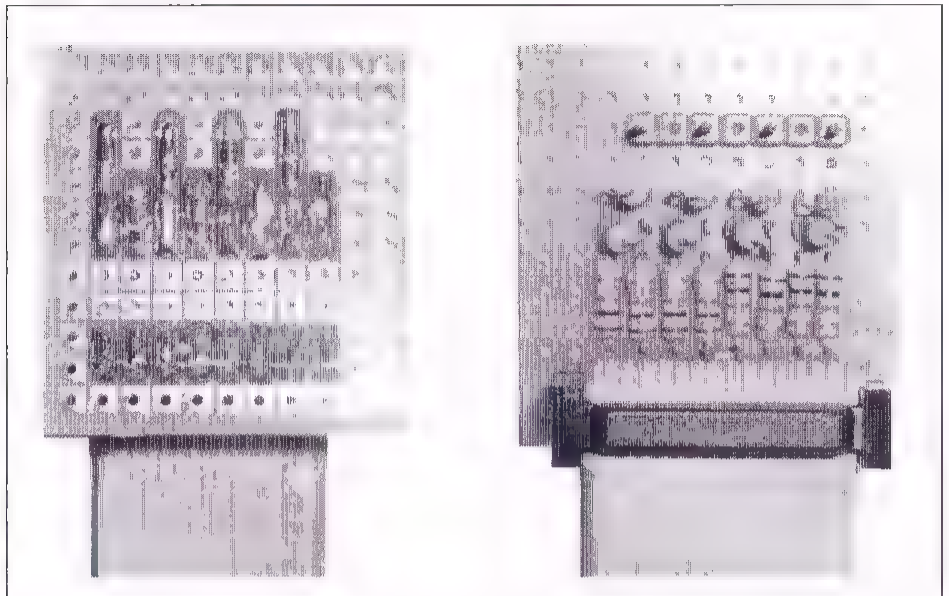
### *Een andere oplossing: de vier noodzakelijke R-2R netwerken worden hier met connectoren verbonden met de microprocessorprint en met de motorbesturingsprinten*



Bovenstaande foto geeft het netwerk voor de twee motoren. De eilandjes aan de onderzijde (grijze vlakjes) zijn doorverbonden. Pas op voor sluitingen aan de soldeerzijde, controleer met een loupe. Op de print is

alvast rekening gehouden met twee extra motorprinten. De op de afbeelding gebruikte connector heeft twee rijen, waarvan er maar één gebruikt wordt. Dat houdt in dat van de linkkabel om de andere ader niet gebruikt wordt. Even opletten dus bij het aansluiten van de connector aan de processorkant.

**Nog een oplossing: vier R-2R netwerken op één klein stukje gaatjesboard**



## Onze robot beweegt softwarematig

### Inleiding

In feite is het rijdend gedeelte van de microprocessorbestuurde robot hiermee gereed. Op basis van het voorbeeldprogramma kunnen alle basisbewegingen geprogrammeerd worden. Voordat we dit thema verlaten, zullen we eerst nog een hink-stapsprong programmeren om een goed idee te krijgen van de beweeglijkheid van dit zelfgebouwde mechatronica project.

### Het programma

Onderstaand programma demonstreert hoe we met behulp van software de bewegingen van de robot kunnen besturen.

\*\*\*\*\*

```
' C-Control Basic
' Hink-Stapsprong
'
```

```
define MotorRechts1 port[14]
define MotorRechts2 port[13]
define MotorLinks1 port[16]
define MotorLinks2 port[15]
```

#opnieuw 'aan het eind komt het programma hier terug

```
beep 500,40,0
'Halve uitgangsspanning betekent vooruit
MotorRechts1 = OFF
MotorRechts2 = ON
MotorLinks1 = OFF
MotorLinks2 = ON
pause 20
beep 400,20,0
```

```
'Beide uitgangen nul betekent stop
```

```

MotorRechts1 = OFF
MotorRechts2 = OFF
MotorLinks1 = OFF
MotorLinks2 = OFF
pause 20

beep 300,20,0

'Beide poorten hoog betekent achteruit
MotorRechts1 = ON
MotorRechts2 = ON
MotorLinks1 = ON
MotorLinks2 = ON
pause 20

beep 200,20,0

'Pirouette
MotorRechts1 = ON
MotorRechts2 = ON
MotorLinks1 = ON
MotorLinks2 = OFF
pause 20

beep 100,20,0

'Draai om wiel
MotorRechts1 = OFF
MotorRechts2 = OFF
MotorLinks1 = ON
MotorLinks2 = ON
pause 20

goto opnieuw
    
```

\*\*\*\*\*

**Opmerking** Het kan zijn dat vooruit en achteruit omgewisseld zijn. Kwestie van de motoren ompolen.

### Epiloog

In hoofdstuk 6 hebben we een voorziening getroffen zodat de robot niet van de tafelrand kon vallen. In het volgende hoofdstuk voeren we een botsingsensor in. Om het eenvoudig te houden doen we dat op de vooruitgaande beweging.

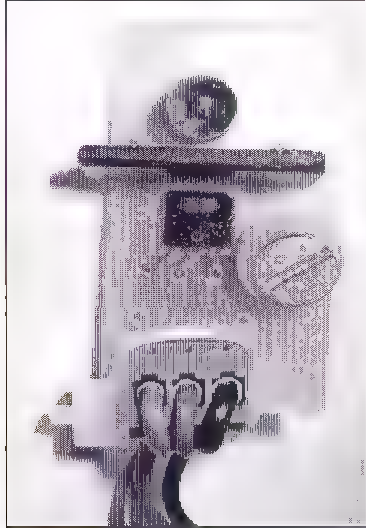
# 12 Obstakel detectie

## Inleiding

### Botsingen voorkomen

In dit hoofdstuk gaan we onze robot zo slim maken dat hij in staat is obstakels te ontdekken en er als het ware omheen te rijden. De sensor is identiek aan die in hoofdstuk 7. We laten hem nu echter niet omlaag kijken, maar recht vooruit zodat eventuele obstakels waargenomen kunnen worden.

*Het reeds bekende sensorprintje krijgt een nieuwe toepassing*



### Twee sensoren voor de voorwielen

Elk voorwiel krijgt er een en ze worden aangesloten op de digitale poorten 1 en 2. **Let er vooral op dat de voeding betrokken wordt uit de 5 V bron.** Een geschikte plaats om deze af te tappen is het kroonsteentje op de microprocessorprint. Een hogere spanning kan de processor beschadigen.

*De printjes worden op deze manier voor de voorwielen van onze robot gemonteerd*



## De software

### Kleine aanpassingen

De aanpassing in het BASIC-programma is niet moeilijk. We vertellen de processor dat de poorten 1 en 2 vanaf nu als ingangen gezien moe-

ten worden. Verder laten we het loopwerk terugdeinzen voor een obstakel

**Het programma**

```
*****
' C-Control Basic
' Obstakel Detectie
'

define MotorRechts1 port[14]
define MotorRechts2 port[13]
define MotorLinks1 port[16]
define MotorLinks2 port[15]

define SensorRechts port[1]
define SensorLinks port[2]

#opnieuw
gosub vooruit
'Let wel, de naderingssensor is normaal hoog
if SensorRechts = OFF OR SensorLinks = OFF then gosub terug

goto opnieuw

#vooruit
'Beide uitgangen hoog betekent vooruit
MotorRechts1 = OFF
MotorRechts2 = ON
MotorLinks1 = OFF
MotorLinks2 = ON
return

#terug
'Beide uitgangen hoog betekent achteruit
beep 500,30,0
MotorRechts1 = ON
MotorRechts2 = ON
MotorLinks1 = ON
MotorLinks2 = ON
pause 200
return
*****
```

**Programma bespreking**

De structuur van dit programma is duidelijk anders dan de eerdere voorbeelden. De rij instructies “vooruit” en “terug” zijn als subroutines uitgevoerd. Vanuit het programma wordt ernaar verwezen. Dit is een van de sterke punten van BASIC. Het is handig alle bewegingsmogelijkheden van het loopwerk als subroutine éénmalig te schrijven en die in elk programma via kopiëren/plakken op te nemen. Er kan dan vanuit het programma een beroep op worden gedaan. Er bestaat een vrucht aan BASIC-literatuur, ook eenvoudige boekjes voor beginners. Het omzetten naar het CCBASIC-dialect is niet al te moeilijk.

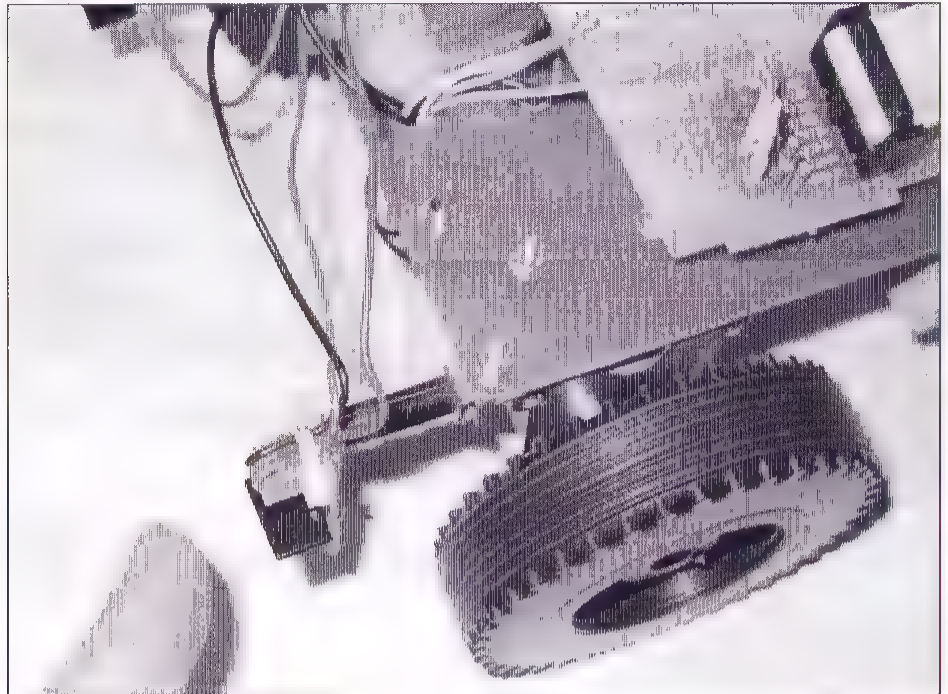
**Kleine veranderingen aanbrenge**

Met een dergelijk BASIC-programma kunnen we lekker gaan stoeien. Bijvoorbeeld kleine wijzigingen aanbrenge en kijken hoe onze robot er op reageert. Een kleine verandering kan zelfs héél grote gevolgen hebben. Wat gebeurt er bijvoorbeeld als we de een na laatste regel  
 pause 200  
 verwijderen?

**De robot wordt een trouw hondje**

Hij loopt dan niet een eindje achteruit, maar probeert een voorgehouden hand te volgen. Als de hand zich verwijderd, loopt snuffel erachter aan. Als de hand nadert doet snuffel stapjes terug.

*Door één regel BASIC te verwijderen wordt onze robot een trouw hondje dat achter onze hand aanloopt*



## Epiloog

We zijn nu in staat onze twee motoren te laten reageren op allerlei externe invloeden. Maar onze motoren werken niet soepel. Ze draaien of ze draaien niet. Niet écht handig, zo'n schokkende beweging!

Gelukkig bestaat daar een handige remedie tegen: stappenmotoren. In het volgende hoofdstuk zullen wij deze nuttige onderdelen gaan ontdekken.

# 13 Stappenmotor besturing

## Inleiding

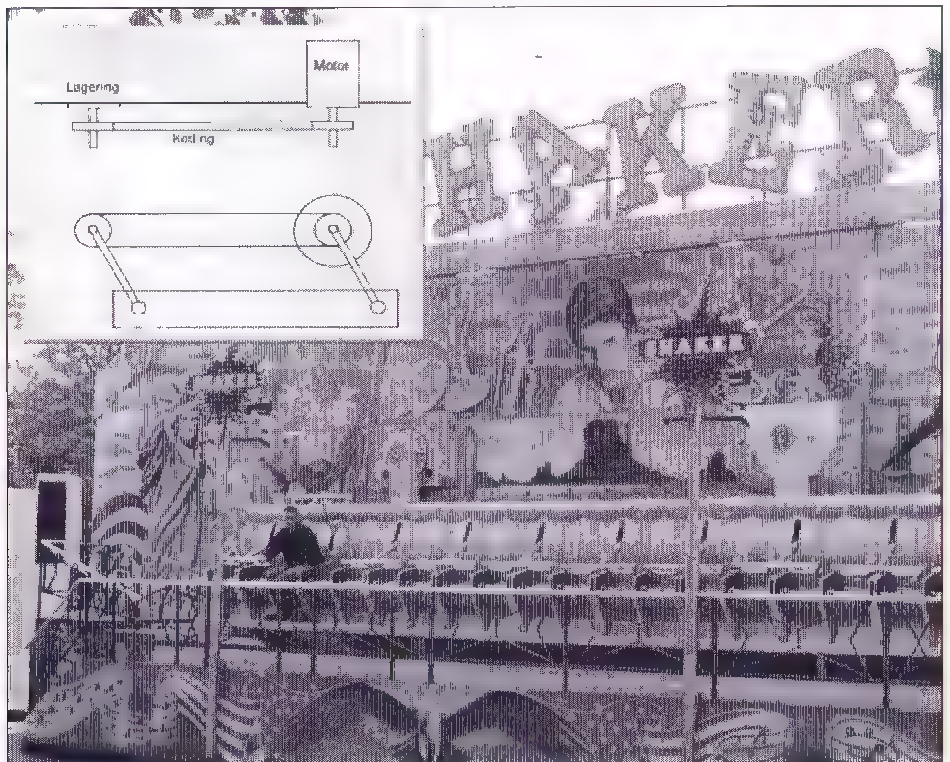
**Alternatieve toepassingen**

Met het manipuleren van een rijdend voertuig houden de mogelijkheden natuurlijk niet op. Veel robots komen nooit van hun plaats, denk maar aan las en spuit robots in de automobielindustrie. Met mechatronica komt een scala aan statische mogelijkheden tot onze beschikking. Kermisattracties leren ons veel over statische machines met gecompliceerde bewegingen.

**Voorbeeld 1**

De hieronder afgebeelde attractie lijkt te draaien op twee motoren die elk een as aandrijven. Dat is echter misleidend. Als er verschil optreedt tussen de standen van de assen wordt de boel uit elkaar gerukt. Beide assen zijn gekoppeld en worden aangedreven door één motor.

*Een kermisattractie die wij met onze kennis van mechatronica gemakkelijk zouden kunnen namaken*



Deze attractie is dan ook zonder meer uit te voeren met de kennis die we nu hebben van mechatronica. Eén motor wordt door reeds bekende programma gestuurd. Door gebruik van de BASIC-opdracht RANDOMIZE (zie verderop in de hoofdstuk 15) kunnen we de bewegingen door een toevalsgenerator laten bepalen.

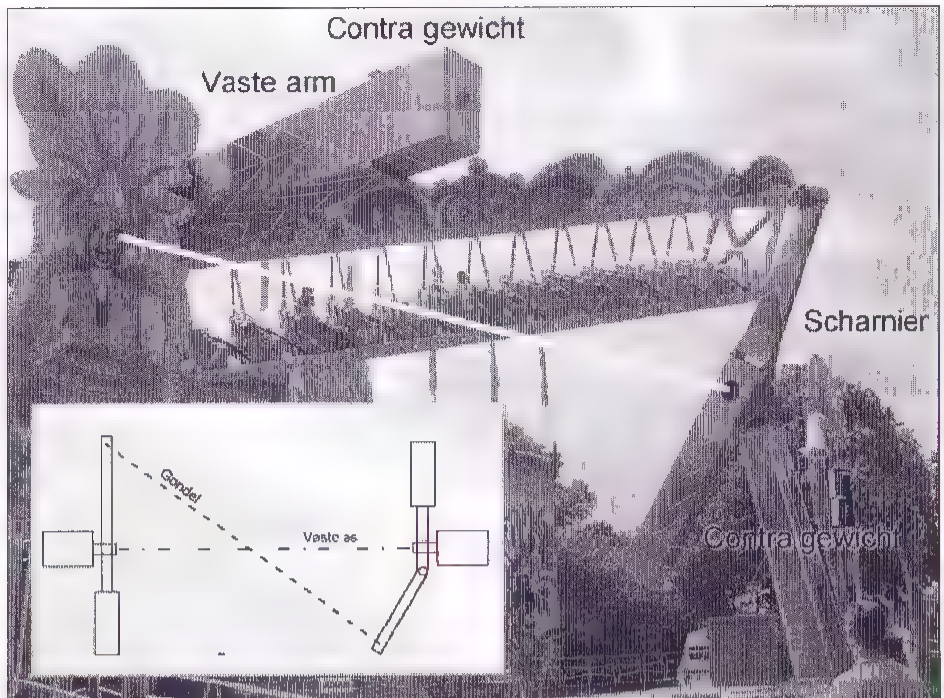
**Voorbeeld 2**

Een kermis attractie die wél gebruik maakt van twee onafhankelijke motoren is afgebeeld op de volgende pagina. Twee motoren kunnen linksom, rechtsom, snel, langzaam of stilstaan. Ook kan één van de armen langzaam draaien en de andere snel. De gondel voert daardoor een complex bewegingspatroon uit waar vooral de jeugd enorm van schijnt te genieten.

Als we de twee motoren van het chassis van onze robot slopen en tegenover elkaar zetten, de wielen eraf halen en een arm op de assen monteren, hebben we deze attractie in miniatuur. Let op het speciale scharnier in één van de armen. Dat zorgt ervoor dat afstandsverschillen tussen de uiteinden van de armen overbrugd worden.



**Een kermisattractie die door twee onafhankelijke motoren wordt aangedreven**



**Beperking van onze motoren**

Deze kermistoepassingen brengen een tekortkoming in onze aanpak aan het licht. De motoren in dit boek draaien met een constant toerental. Dat resulteert in schokkende bewegingen die voor kermisattracties minder geslaagd zijn. Ook is het precieze aantal omwentelingen niet bekend. Nauwkeurig positioneren is dan ook niet het sterkste punt.

## Stappenmotoren

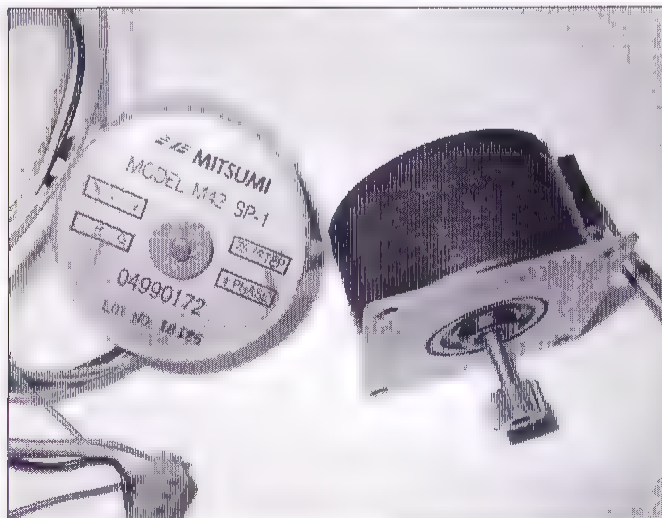
**Ideaal voor microprocessor besturing**

Het is duidelijk in welke richting onze toekomstige mechatronica projecten zich dienen te ontwikkelen. De in dit werkboek gebruikte microprocessoropstelling kan heel goed overweg met pulsen en frequenties. En dat legt een duidelijke koppeling in de richting van stappenmotoren. En ook die zijn tegen lage kosten beschikbaar in vele miniatuuruitvoeringen, eventueel compleet met stuurprint.

**Wat zijn stappenmotoren?**

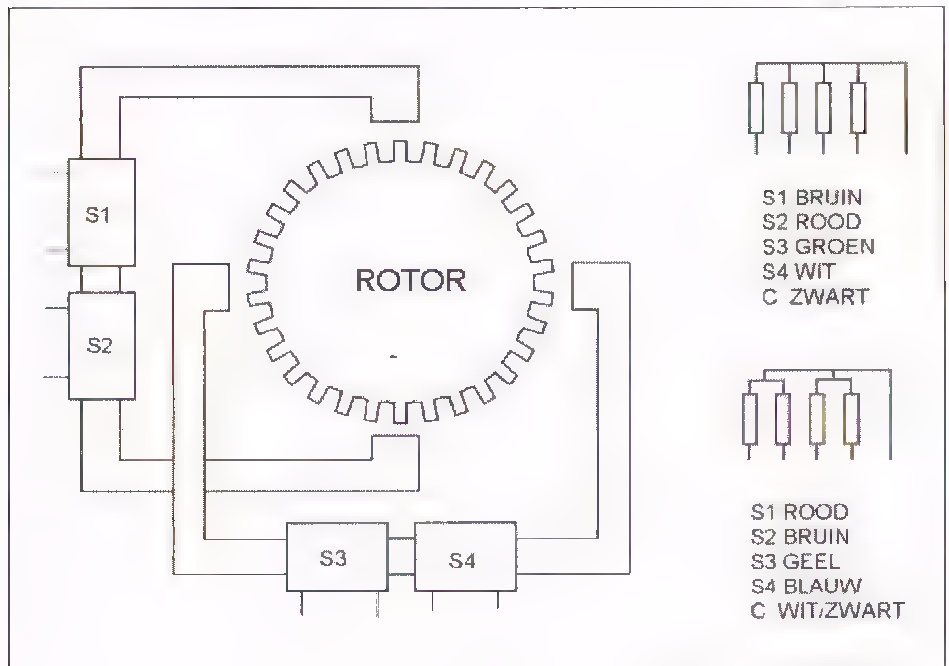
Om toch een beetje beeld te krijgen over de technische mogelijkheden zullen we een aantal basiszaken proberen te doorgronden. Stappenmotoren zijn motoren die niet continu draaien als zij onder spanning worden gezet, maar een kleine hoek verdraaien in de een of de andere richting als er een pulsformige spanning wordt aangeboden.

**Kleine stappenmotoren die ideaal zijn voor onze mechatronica experimenten**



De grootte van de beweging van de rotor (de hoekverdraaiing) is dus volledig te besturen door een bepaald aantal pulsen aan de stappenmotor aan te bieden. Op deze manier kunnen wij bijvoorbeeld bepalen dat de as van de rotor precies 125 graden moet verdraaien. Een stappenmotor, althans de gebruikelijke kleine typen, heeft vier stator spoelen die aan één zijde met elkaar verbonden zijn. Er komen dus vijf draden uit. Soms zijn deze al twee aan twee gekoppeld en af en toe zit er ook nog een aarddraad bij die met de behuizing is verbonden.

**De samenstelling en de aansluitgegevens van stappenmotoren**



Ook kunnen de gemeenschappelijke verbindingen per spoelenset apart naar buiten zijn gevoerd. Omdat alle spoelen dezelfde weerstand hebben, kan een universeelmeter helpen bij het uitzoeken van de gekleurde draden en hun interne verbindingen.

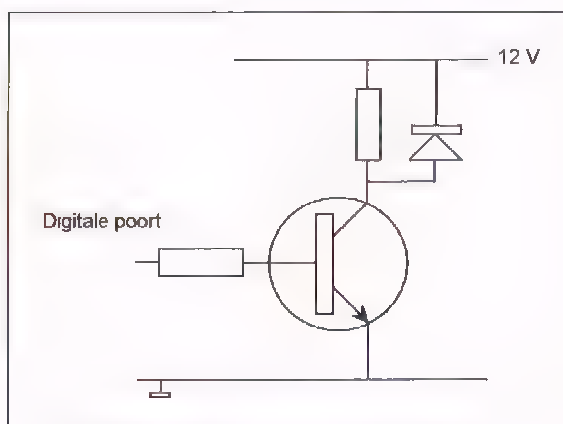
**Werking**

Er is steeds slechts één spoel bekrachtigd. Zodra de volgende spoel stroom krijgt, draait de rotor één tandje verder. Afhankelijk van de volgorde waarin de spoelen bekrachtigd worden draait de motor dus links of rechts om. Stappenmotoren zijn er in vele spanningen. Meestal ligt de stuurspanning tussen 12 V en 24 V.

**Aansturing van stappenmotoren**

Normaal gesproken worden stappenmotoren aangestuurd vanuit een driver-IC. We hebben op onze processorprint maar liefst 16 uitgangen. We zouden er dus in beginsel vier stappenmotoren mee moeten kunnen aansturen. Er is echter nog een hindernis te nemen.

**Met een dergelijke eenvoudige buffer kunnen wij de spoel van een stappenmotor uit onze microprocessor besturen**



De processor levert maximaal 10 mA bij 5 V. Er moet dus een buffertje tussen. De eenvoudigste buffer is een transistorschakelaar met ontstoringdiode, zie het schema op de vorige pagina

### Snelheid van besturing

Omdat een volledige omwenteling van de as flink wat stappen vraagt moeten we een behoorlijk hoge puls frequentie bereiken. Het is daarom interessant te weten welke frequentie de microprocessor kan leveren. Onderstaand programma gebruikt een afzonderlijk digitale uitgang voor elk van de vier spoelen en schakelt spoel voor spoel in en de andere weer uit.

```
*****
```

```
' C-Control Basic
' Stappenmotor
```

```
define uitgang1 port[14]
define uitgang2 port[13]
define uitgang3 port[16]
define uitgang4 port[15]
```

```
#opnieuw
gosub spoel1
gosub spoel2
gosub spoel3
gosub spoel4
goto opnieuw
```

```
#spoel1
uitgang1 = ON
uitgang2 = OFF
uitgang3 = OFF
uitgang4 = OFF
return
```

```
#spoel2
uitgang1 = OFF
uitgang2 = ON
uitgang3 = OFF
uitgang4 = OFF
return
```

```
#spoel3
uitgang1 = OFF
uitgang2 = OFF
uitgang3 = ON
uitgang4 = OFF
return
```

```
#spoel4
uitgang1 = OFF
uitgang2 = OFF
uitgang3 = OFF
uitgang4 = ON
return
```

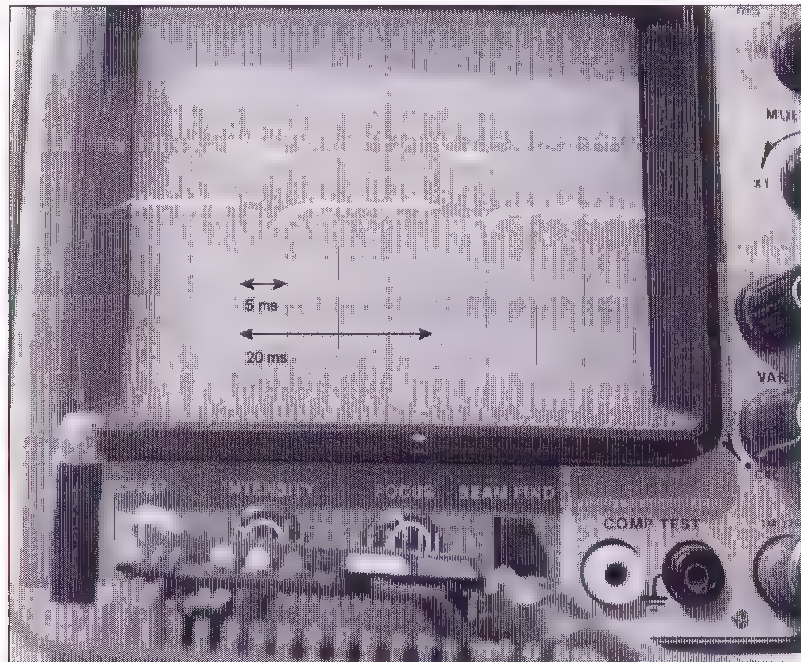
```
*****
```

### Het programma op de oscilloscoop

Op het oscilloscoopscherm zien we het beeld van één van de spoelen. Op de processorprint zien we aan de gele en rode LED dat het pro-

programma loopt. Iedere spoel is 5 milliseconde bekrachtigd en daarna 15 milliseconde uit.

**Het stuursignaal van één van de spoelen van de stappenmotor**



In die tijd zijn de andere spoelen één voor één actief. De totale cyclusduur is dus 20 milliseconde. De maximale puls-frequentie ligt daarmee op 50 Hz.

Een motortje met 48 stappen per omwenteling (7,5 graden per stap) kan dus één omwenteling per seconde maken. Niet indrukwekkend, maar ook niet per definitie onbruikbaar. Dat zal van de toepassing afhangen.

**De stappenmotor in de praktijk**

Het aansluitschema is de eenvoud zelve. Vier van de eerder geschetste buffertrapjes worden elk door een digitale poort van onze microprocessor aangestuurd. De vier spoelen komen in de collectorleidingen naar de plus. Dat is dan tevens de gemeenschappelijke aansluiting van de spoelen.

De looprichting wordt door het programma bepaald. Er komen dus nog vier subroutines bij met de volgorde 4-3-2-1 in plaats van 1-2-3-4 zoals in het voorbeeld programma.

Het aantal stappen kan opgegeven worden met de opdracht

```
*****
```

```
FOR teller = 1 TO 100
```

```
' Uit te voeren programma met gosubs
```

```
NEXT
```

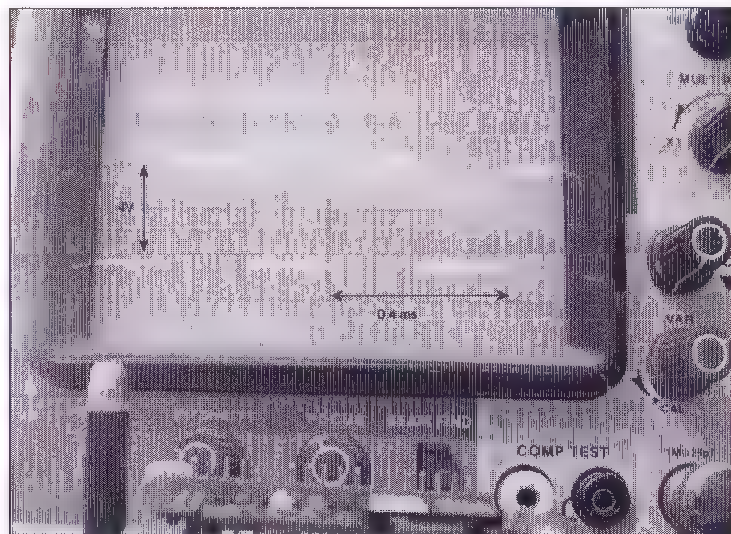
```
*****
```

Tussen de FOR en NEXT regel komt het voorbeeld programma te staan, dat dan 100 keer uitgevoerd wordt.

**Sneller met BEEP**

Toch is het resultaat een beetje teleurstellend. Veel programmeerwerk voor zo'n matige snelheid. Wat snelheid betreft doet de BEEP-uitgang van onze microprocessor het beter. De hoogste frequentie die zich nog stabiel gedraagt is BEEP 100, hetgeen overeenkomt met 2.500 Hz. Verdeeld over de vier spoelen is dat nog altijd meer dan 600 Hz, voldoende voor een respectabel toerental.

**Het resultaat van het sturen van de spoel van de stappenmotor met het BEEP-signaal van onze microprocessor**



Het is dan ook zeer interessant het BEEP-signaal nader te analyseren. Hierboven staat het signaal van pin 11 van de eerste connector bij BEEP 100 op het scherm van de oscilloscoop. Een mooi strak signaal. Maar, helaas maar één beschikbare uitgang. BEEP heeft nog het voordeel dat we precies weten hoeveel pulsen er afgegeven worden. BEEP 100, 1, 0 geeft gedurende 20 milliseconde een frequentie van  $250.000/100=2.500$  Hz. In een/vijfde seconde (20 ms) is dat dus 500 pulsen. Bij 7,5 graden as-rotatie zijn dat bijna 10 omwentelingen. BEEP heeft dus een aantal voordelen tegenover de reguliere digitale uitgangen:

- de snelheid is instelbaar;
- de snelheid is onafhankelijk van de programmagrootte;
- we kunnen een precies aantal pulsen doseren;
- de BEEP functie vraagt nauwelijks programmeerwerk.

BEEP heeft echter ook een nadeel. Voor het verdelen van de pulsen over de vier spoelen hebben we wat externe elektronica nodig. Maar wie is daar nog bang voor?

## De stappenmotor driver SAA1027

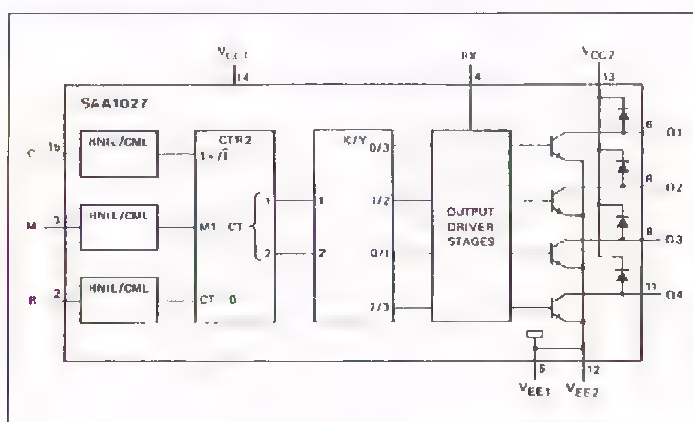
**Een interessant IC van Philips**

We vallen meteen met de deur in huis: de SAA1027 stappenmotor driver van Philips is een ideaal IC voor onze experimenten. Het IC is goedkoop en, hoewel al vrij oud, toch nog goed verkrijgbaar.

**Intern schema**

Het intern schema van de SAA1027 is voorgesteld in de volgende illustratie. Het IC heeft drie ingangen voor de digitale besturing en vier open-collector uitgangen voor de besturing van de vier spoelen van de stappenmotor.

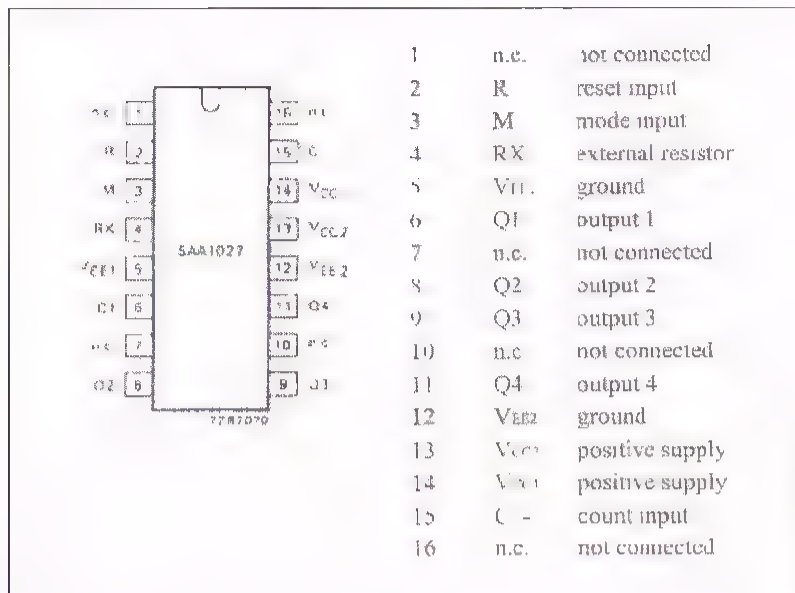
**Blokschema van de SAA1027 stappenmotor driver**



**Aansluitgegevens**

Het IC is ondergebracht in een DIL-16 behuizing. de aansluitgegevens en de functie van de zestien pennen volgen uit de onderstaande illustratie.

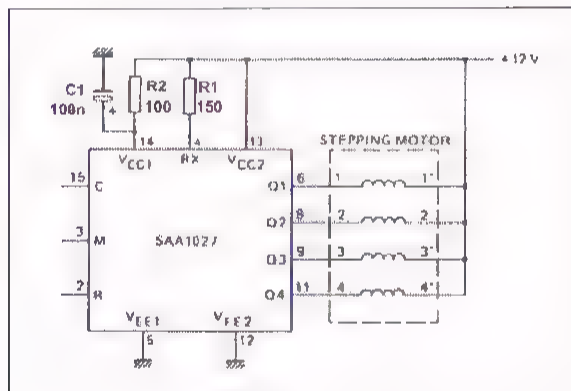
**Aansluitgegevens van de SAA1027**



**Het schema**

Het schema rond de SAA1027 is de eenvoud zelve: er zijn maar twee weerstanden en één condensator noodzakelijk. Een en ander is zo eenvoudig dat een onderdelenlijst achterwege kan blijven.

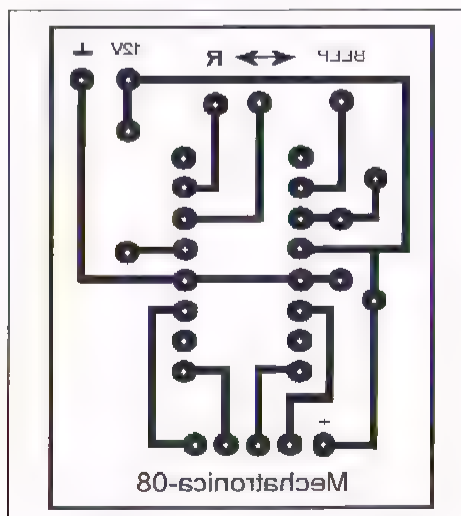
**Het basisschema rond de SAA1027**



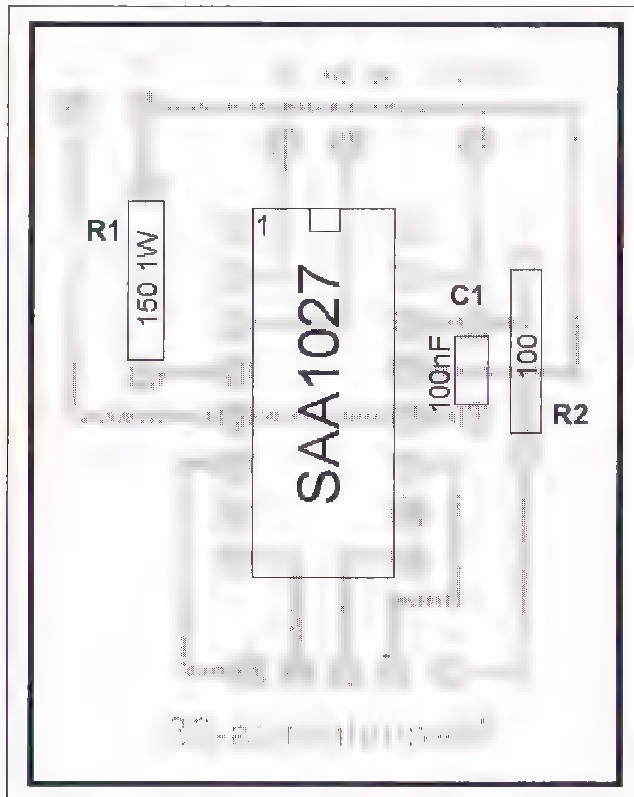
**Het printje**

Ook voor deze schakeling hebben we een printje ontworpen.

**De print voor de stappenmotor besturing**



**De componentenopstelling**



**Werken met de print**

Aan de hand van de componentenopstelling zullen we de functies van de diverse aansluitpennen nader onder de loupe nemen. De vier spoeluitgangen vinden we aan de onderzijde van de print. Bovendien treffen we de voeding aan. De draairichting en de pulsingang vinden we aan de andere zijde.

Met de linker weerstand van 150 Ω begrenzen we de spoelstroom op een veilige waarde. Deze weerstand moet wel een 1 W uitvoering zijn. De draairichting besturen we direct vanuit een digitale uitgang. Met BEEP bepalen we het aantal stappen.

**Het besturingsprogramma**

Hoe eenvoudig het is om met ons “apentaaltje” mooie mechatronica aan te sturen blijkt wel uit onderstaand programma.

```

*****
'aansturing SAA1027

define omkeer port[16]

#opnieuw

omkeer = OFF 'linksom
beep 100,20,0 '500 stappen

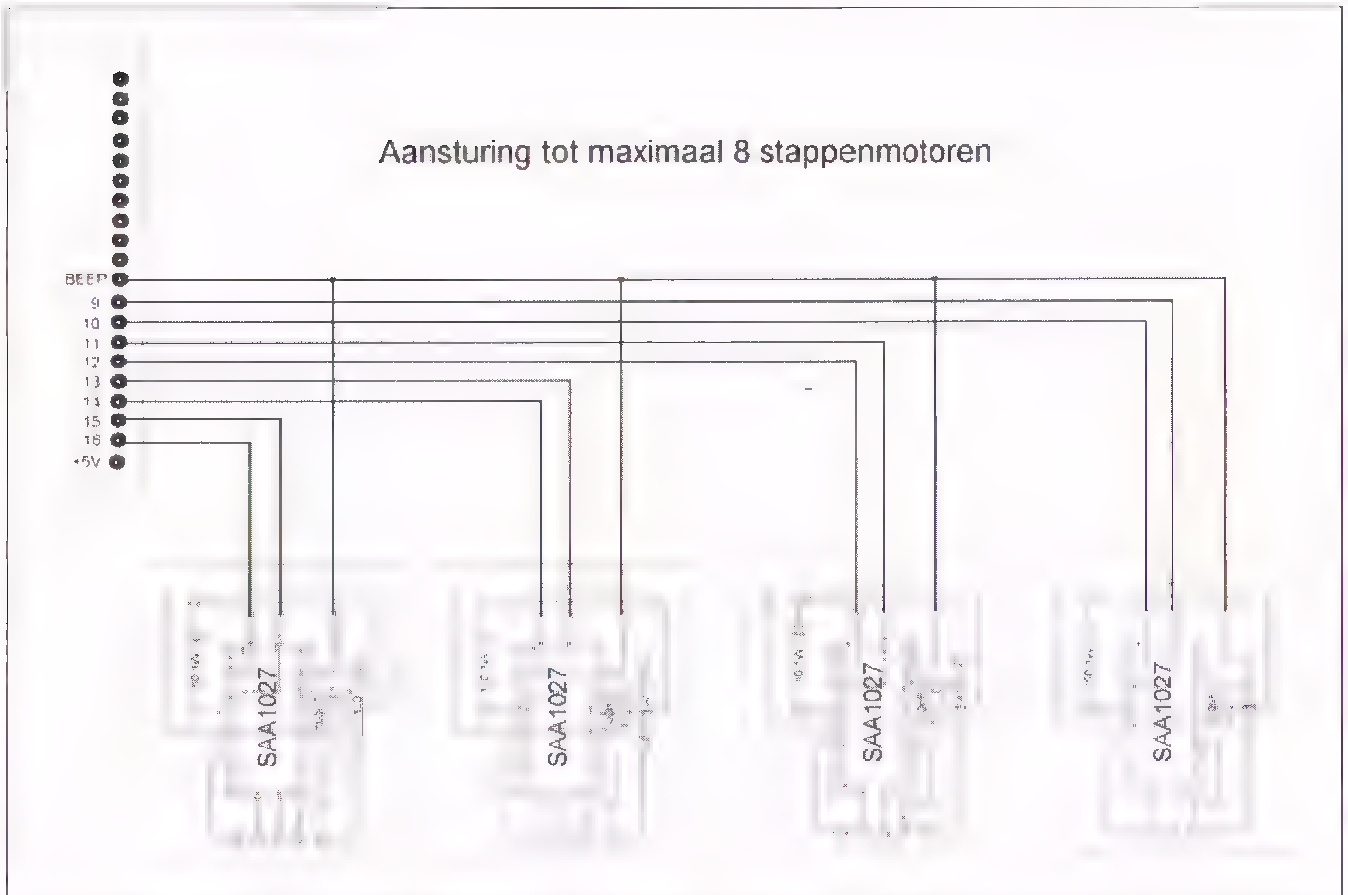
omkeer = ON 'rechtsom
beep 100,20,0 '500 stappen

goto opnieuw
*****
    
```

**Uitbreiding naar vier motoren**

Een kind kan de was doen. Of toch niet? Eén stappenmotor, is dat alles? Nee, want we kunnen op een heel eenvoudige manier tot maximaal acht stappenmotoren met onze microprocessor besturen. Op de SAA1027 zit nog namelijk een handige voorziening op aansluiting 2.

Dat is de RESET-aansluiting. Zolang die laag is, staat de teller stil. De BEEP-uitgang wordt aangesloten op alle stappenmotor printjes die we willen gebruiken. Per print hebben we dan nog twee logische poorten nodig. Eén voor de richting en één om de print te resetten en daarmee de betreffende motor te stoppen. Omdat we 16 digitale uitgangen hebben kunnen totaal 8 stappenmotoren aangesloten worden en daar kan een heel complexe beweging zeer precies mee aangestuurd worden.



Een schema waarmee we vier stappenmotoren met onze microprocessor kunnen besturen

**Opmerking** Let erop dat de BEEP-functie op de achtergrond werkt. Dat wil zeggen dat de pulsen afgewerkt worden los van andere opdrachten. Tussentijds stoppen is alleen maar mogelijk door de RESET-ingang van de stappenmotorprint laag te maken. We moeten er in het programma voor zorgen dat de BEEP-opdracht afgewerkt is, voordat de RESET weer hoog gemaakt wordt.

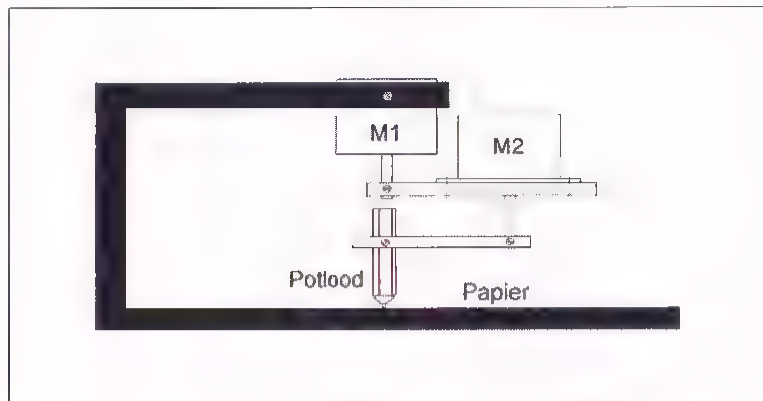
## Toepassingen van stappenmotoren

**Inleiding** Het is al diverse keren opgemerkt, maar we herhalen het hier toch nog maar eens: dit boek is een **WERKboek**, hetgeen wil zeggen dat wij niet alles voorkauwen, maar de lezer(es) suggesties doen om verder te experimenteren. We geven tot slot van dit hoofdstuk twee voorbeelden van mechatronica-projecten die ideaal zijn om met stappenmotoren te worden uitgewerkt.

**Spirograaf** Een aardige toepassing voor twee stappenmotoren is de spirograaf, een apparaatje waarmee wij mooie geometrische figuren op een vel papier kunnen tekenen. Motor 1 maakt één volledige omwenteling (anders komt het snoer van motor 2 in het gedrang). Tijdens die omwenteling roteert motor 2.



**De schematische opstelling voor een spirograaf**



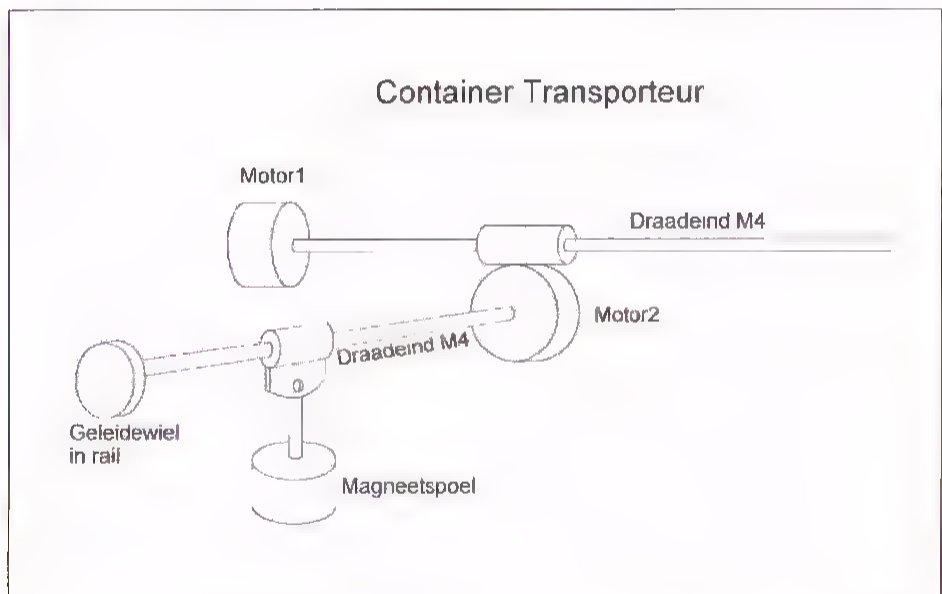
We kunnen hierbij met succes gebruik maken van sinusfuncties om aardige repeterende figuren te krijgen. Een prachtig programmeerlesje én onmiddellijk resultaat!

Mocht de stapgrootte van de motoren wat aan de groffe kant zijn, een tandwielletje ertussen verhoogt de resolutie van de tekening aanzienlijk.

**Hijsportaal**

Een portaal voor containertransport heeft een vaste motor 1 die een draadeind aandrijft. Op het draadeind twee messing moertjes die in een busje zijn gesoldeerd. Aan dat busje is tevens motor 2 bevestigd. Ook motor 2 drijft een draadeind aan waarop een dergelijke bus beweegt.

**Schematische voorstelling van de constructie van een container transporteur**



Aan déze bus hangt een hefmagneet. Het zal duidelijk zijn dat een stabiele mechanische uitvoering wat tijd vraagt. De afmetingen zijn vrij.

**Epiloog**

Er zijn talloze toepassingen te verzinnen van robots die door programmering intelligent gemaakt kunnen worden. Het hefsysteem hierboven kan bijvoorbeeld onthouden waar hij de containers gelaten heeft of kan detecteren of er een auto op de laadplaats staat. Het komt allemaal aan op creatief programmeerwerk en daar leer je logisch denken van. Wij zijn nu zover dat we een lopende robot kunnen ontwerpen: Archie, de man van staal!

# 14 Archie, de man van staal

## Inleiding

### Armen en benen

Zoals in de inleiding al is aangehaald, wordt bij het woord robot een bepaald beeld opgeroepen. Door films als "De Terminator" wordt dit beeld versterkt. Wie kent niet de scène waarin het laatste restje van een robot-arm toch nog probeert zijn slachtoffer te grijpen?

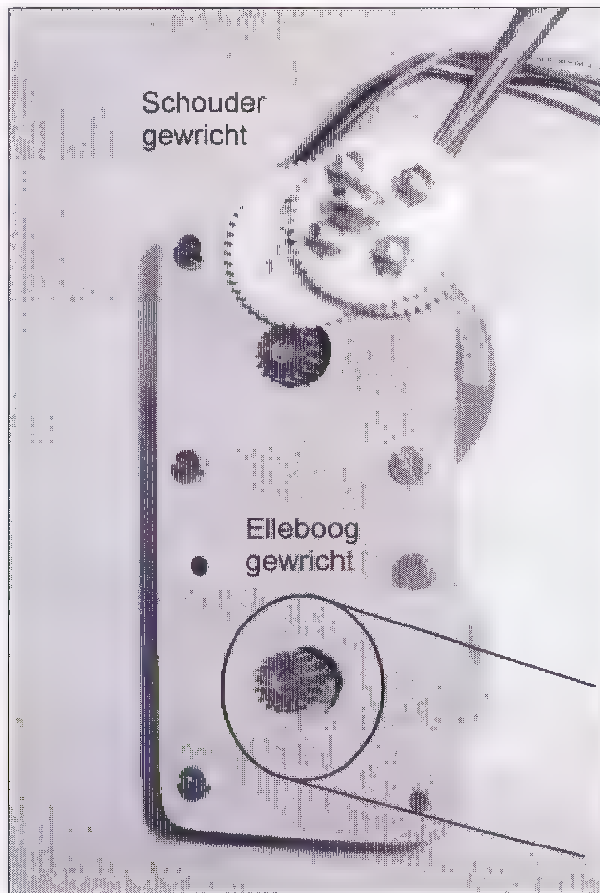
In dit laatste hoofdstuk maken we een reconstructie van een oerrobot die armen en benen kan bewegen. Of hij daarmee iets kan uitrichten zal uitsluitend van de programmeerkunsten van de lezer(es) afhangen. Omdat armen en benen zeer sterk gericht zijn op positioneren, gebruiken we stappenmotor-tjes.

## Armen

### Twee stappenmotoren

Een arm zou er als voorgesteld op onderstaande foto kunnen uitzien.

### *De basisconstructie van een robotarm*

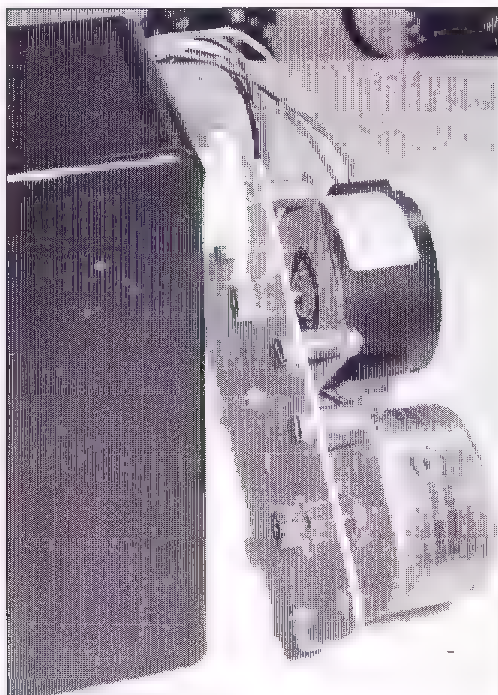


Stappenmotoren zijn behoorlijk sterk, maar het opheffen van een arm waarop nóg een motor is bevestigd wil niet lukken. Daarom bouwen we bij de schouder een vertraging in. Eén van de motor bevestigingsbouten wordt extra lang uitgevoerd. Zoals in de eerste hoofdstukken voeren we het tandwiel dubbel uit met afstandsbusjes ertussen. Met de vier boutjes wordt de arm tevens vastgezet aan het lijf. Voor de centrale bout boren we een ruim gat in het lijf, zodat de twee tegen elkaar ingedraaide moertjes vrij kunnen ronddraaien.

### Onderarm

Op de tweede motor-as wordt een onderarm (naar eigen inzicht) gemonteerd. In de foto op de volgernde pagina is de arm vastgezet aan het lijf.

**De arm is aan het chassis bevestigd**

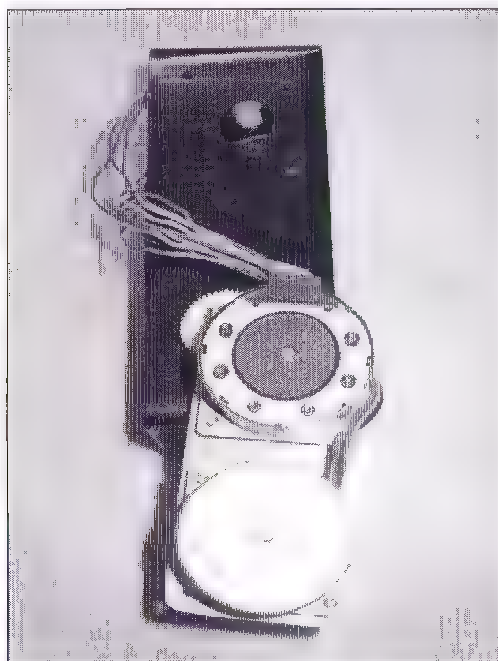


## Benen

**Schouder wordt heup**

Als we de tekst "schoudergewricht" vervangen door "heupgewricht" en "ellebooggewricht" door "kniegewricht" hebben we in principe een robotbeen. De voet kan scharnierend aan het onderbeen verbonden worden.

**De constructie van een robotbeen**



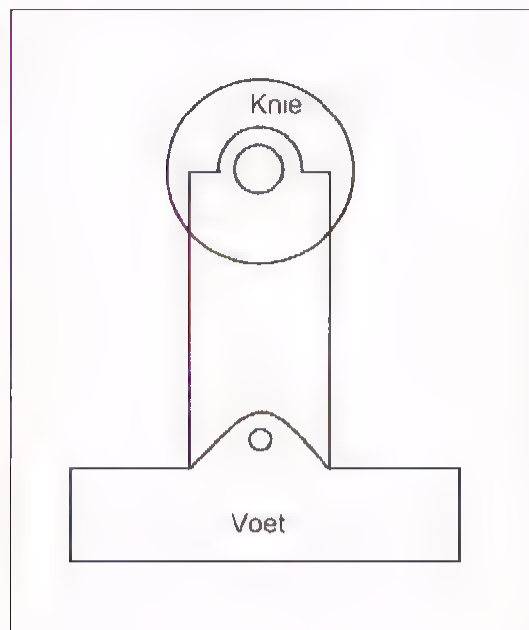
**Opmerking**

Het moge duidelijk zijn dat bij het andere arm/been paar het dubbele tandwiel op de linker motor bevestigingsbout wordt gezet. Het gat voor het tussenwiel wordt overgenomen terwijl het tandwiel spelingvrij tegen de vertanding van de motor en het dubbele wiel gedrukt wordt. Dat waarborgt een flexibele loop.

**En nu nog een voet**

Zie de schets op de volgende pagina. Zo ziet het eruit als de motoreenheid als been wordt gebruikt. Let wel, het bovenbeen ontbreekt hier. De schets geeft niet meer dan een suggestie.

**Een schets van de constructie van een onderbeen met een voet**



De voet roteert vrij, maar wordt met een veertje in de middenstand gehouden.

## Epiloog

We hebben in de voorafgaande hoofdstukken alle basisprincipes van mechatronica behandeld. U kunt nu uw fantasie de vrije loop laten en eigen projecten ontwikkelen.

Een enorm krachtig hulpmiddel hebben we maar terloops aangeroerd: CCBASIC. Een krachtige programmeertaal die we absoluut moeten beheersen als we onze projecten tot leven willen wekken.

# 15 Een lesje CCBASIC

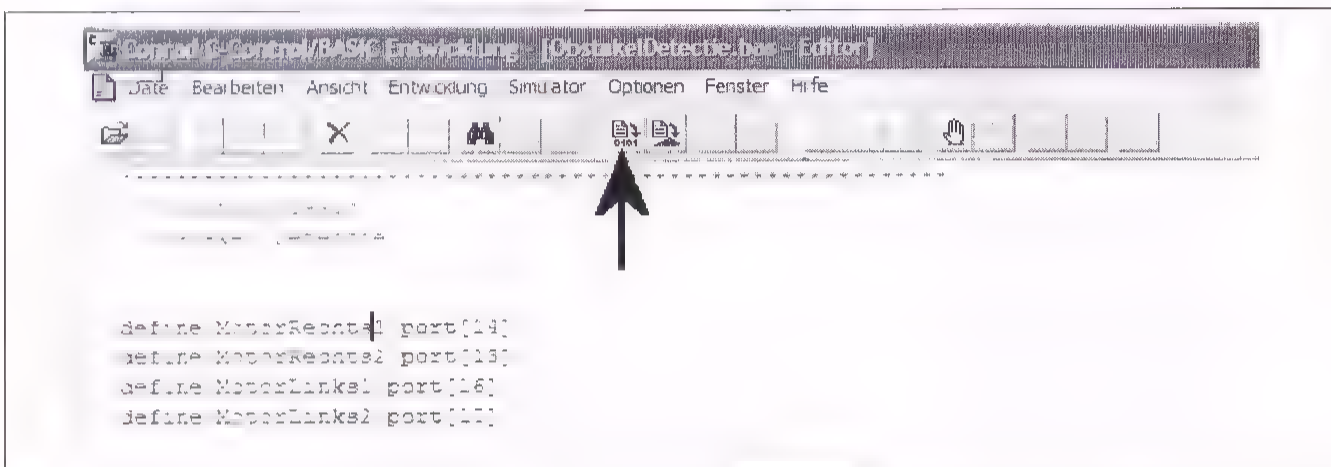
## Inleiding

**Een van de vele BASIC-dialecten**

Voor de beginnende BASIC-programmeur is het moeilijk inzicht te krijgen in de eigenaardigheden van het CCBASIC-dialect dat bij deze microprocessor hoort. Daarom zijn enkele wetenswaardigheden zoals die gelden voor de gebruikte processor hier verzameld.

**Experimenteren op het scherm**

Deze minicursus kunnen we het best volgen op het scherm van onze PC. Als het programmastukje gereed is, kan het namelijk direct op fouten onderzocht worden door te klikken bij de pijl.



*Deze knop doet hetzelfde als zijn rechter buurman, maar verzendt niets naar de processorprint*

Op deze manier kunnen we programma's uittesten zonder code naar de microprocessor te moeten zenden.

## Belangrijke CCBASIC-principes

**Declareren van variabelen en constanten**

Een CCBASIC-programma kan pas iets doen als het weet wie de spelers zijn. We hebben gezien dat we dat met behulp van de define opdracht laten weten.

Voor digitale in- en uitgangen hebben we dat al gedaan. Voor een analoge poort geldt iets dergelijks:

```
*****
define Temperatuurmeter AD[1]
'voor een analog naar digitaal omzetting
*****
```

Er zijn acht AD-converteren beschikbaar die gebruikt kunnen worden voor het meten van spanningen. Let wel, als de ADC's gebruikt worden, dan moeten we de spanningsreferentie met een spanningsbron verbinden die beslist niet hoger mag zijn dan 5 V. Als de gebruikte sensor slechts een signaal afgeeft van 1 V, dan leggen we 1 V aan pin 2 van connector 1. Geeft de sensor 5 V af, dan leggen we deze pin aan de voedingsspanning van 5 V. Op deze manier wordt steeds het spanningsbereik van de sensor uitgesmeerd over "de volle schaal" van de ADC. De gevoeligheid is 8 bit. De sensorspanning kan dus in  $2^8 = 256$

stapjes uitgelezen worden (digitale codes van 000 tot 255), een oplos-send vermogen van een half procent.

**Variabelen benoemen**

We zien dat we variabelen zélf een naam moeten geven (MotorRechts, Temperatuur, etc.). Maak daar dankbaar gebruik van om de structuur van uw programma's te verbeteren en de leesbaarheid ervan te ver-groten.

Een klein experiment: type de volgende regels.

\*\*\*\*\*

'Vastleggen van variabelen

```
meter = 1
centimeter = meter * 100
```

\*\*\*\*\*

Dat geeft dus aanleiding tot een foutmelding. Hoe het wél moet?

\*\*\*\*\*

'definitie van variabelen

```
define factor 100
define centimeter BIT[1]
define meter BIT[1]
```

```
meter = 12
centimeter = factor * meter
```

```
print centimeter
```

\*\*\*\*\*

Dat gaat al een stuk beter, geen foutmeldingen meer. We zien dat zelfs een constante aangekondigd moet worden. Dit heeft te maken met de manier waarop het geheugen ingevuld wordt. Achter BIT staat een 1. Afhankelijk van de grootte van de te verwerken getallen mag daar een getal staan tot 192. De laatste regel geeft een printopdracht. Die uit-komst komt niet op de PC-printer terecht, maar op de seriële ingang die we ook voor het programmeren gebruiken. Daar kan heel eenvoud-ig een kleine display op aangesloten worden.

**Iets over BIT, BYTE en WORD**

Maximaal kan de processor 192 bit grote getallen verwerken. Er gaan 8 bit in een byte en 2 byte in een word. De uitdrukkingen:

\*\*\*\*\*

```
define getal BIT[192]
define getal BYTE[24]
define getal WORD[12]
```

\*\*\*\*\*

geven alle hetzelfde aan. Er wordt gewerkt met getallen tot  $2^{192}$ . Het zal echter duidelijk zijn dat we hiermee behoorlijk wat geheugenruimte in-nemen. Het is daarom beter het getal aan de behoefte aan te passen.

**Werken met functies**

In berekeningen worden vaak functies gebruikt zoals de vierkantswortel, cosinus en dergelijke. Onderstaand programmavoorbeeld geeft een voorbeeld voor de vierkantswortel (functie SQR).

```
*****
'functies
define oppervlak BIT[1]
define zijde BIT[1]

oppervlak = 100
zijde = SQR(oppervlak)

print " De zijde is" zijde
```

\*\*\*\*\*

Maak opzettelijk kleine veranderingen en kijk hoe de foutendetectie die er feilloos uithaalt.

**Werken met RANDOMIZE**

Met de functie RANDOMIZE kunnen we een willekeurig getal oproepen zodat we enige onvoorspelbaarheid in de programma's kunnen aanbrengen.

```
*****
'Willekeurig getal
define getal BIT[1]
define willekeur BIT[1]
RANDOMIZE willekeur
getal = RAND
```

\*\*\*\*\*

RANDOMIZE initieert dus een willekeurig getal. RAND rekt hieruit een volgend willekeurig getal uit.

**Programmastructuur**

We hadden al gezien dat we meerdere malen voorkomende programablokken als subroutine kunnen vastleggen. Het programma wordt daar overzichtelijk van, vooral als de subroutines betekenisvolle namen meekrijgen.

```
*****

GOTO subroutine1

#subroutine1

'*****

is dus niet erg handig.

'*****

GOTO linkerdraai

#linkerdraai

'*****
```

maakt foutzoeken in programma's veel gemakkelijker.

**IF... THEN... ELSE...** Een veel gebruikte functie is IF... THEN... of IF... THEN... ELSE...

```
*****
```

```
'Voorwaarde
```

```
define temperatuur AD[1]
define max 35
define relais port[6]
```

```
#meting
```

```
IF temperatuur > max THEN relais = ON else relais = OFF
GOTO meting
```

```
*****
```

Het signaalrelais wordt aangetrokken bij temperaturen boven 35. Dat zijn geen graden, maar digitale getallen die de microprocessor levert. Die kunnen we met een formule laten omrekenen naar spanningen. Het digitale getal 255 komt bijvoorbeeld overeen met 5 V van de sensor en de sensor levert 5 V bij een temperatuur van 100 graden. Dan komt het resultaat 255 overeen met 100 graden en het getal 128 overeen met 50 graden.

## Epiloog

Tot zover deze beknopte BASIC-les. Mensen zijn niet altijd even geschikt voor logische processen. Programmeren betekent veel oefenen en hard nadenken waarom bepaalde zaken anders gaan dan verwacht.

Het is natuurlijk mogelijk dit werkboek te vullen met ellenlange BASIC-programma's. Dit werkboek is echter een robotica werkboek en geen BASIC-leerboek. De handleiding in de Duitse taal die bij CCBASIC geleverd wordt is niet overal even helder. Daar staat tegenover dat het uitproberen van een programma, met behulp van de ingebouwde foutzoeker, bijzonder snel en gemakkelijk gaat.

Nu het gebruik van de digitale poorten als in- of uitgang duidelijk is, kan vrijwel elke sensor, relais, lampje, LED of sirene aangesloten worden. Met 16 poorten kan heel wat gedaan worden. De liefhebber van vechtrobots kan twee motoren toevoegen waarmee de tegenstander omvergeworpen kan worden. Naast de digitale poorten kan CCBASIC ook met analoge signalen werken. De robot kan dan op afstand bijvoorbeeld temperatuurmetingen doen. Het volgen van een streep op de grond of een ingegraven draad maakt nauwkeurige positionering mogelijk.

In het door dezelfde uitgever uitgegeven boek "Vego's groot elektronische sensoren boek" (zie [www.vego.nl/5/01/5\\_01.htm](http://www.vego.nl/5/01/5_01.htm)) kan inspiratie opgedaan worden betreffende invoersignalen. Er zijn dus nog voldoende interessante mogelijkheden om vele avonden met mechatronica bezig te zijn. Als hobby, of als voorbereiding op een professionele carrière.



## 16 Robotica informatie

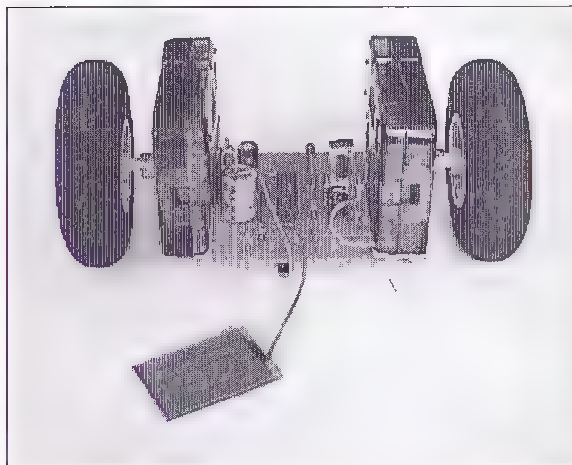
### Inleiding

Heeft u de smaak te pakken? Dan hebben wij ons doel bereikt! Robotica kan inderdaad uitgroeien tot een boeiende en leerzame hobby. Maar dan komt het er op de eerste plaats op aan zoveel mogelijk informatie te verzamelen over de onderwerpen mechatronica en robotica. Dat kan zowel schriftelijk als via het Internet

### Hobby Elektronica & Actueel IC-handboek

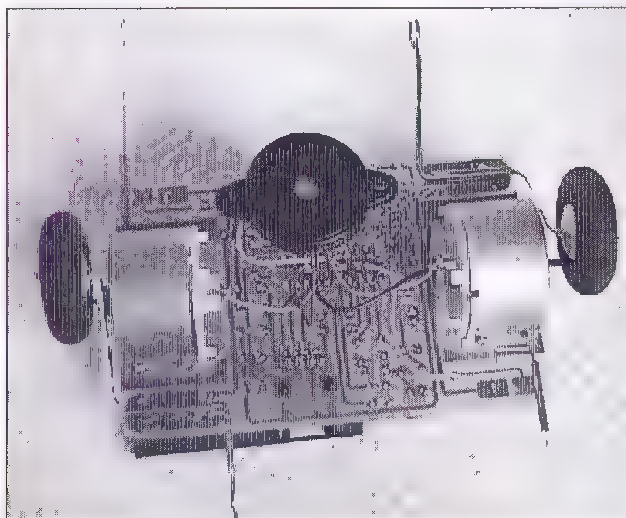
In de jaargang 2003 van het losbladig naslagwerk "Hobby Elektronica & Actueel IC-handboek" (zie [www.hobbyelektronica.vego.nl](http://www.hobbyelektronica.vego.nl)) worden twee volledig uitgewerkte robotprojecten voor zelfbouw beschreven. Het eerste, "Smiley", is een simpele zelfbouw robot die aangedreven wordt door eenvoudige elektronica. "Smiley" krijgt energie van een klein zonnepaneeltje. Zijn motoren zijn klokjes, aangepast om een stuk sneller te lopen dan normaal. De wielen zijn van een type dat wordt gebruikt voor het onderstel van modelvliegtuigen. Hij "voelt" zijn weg met 0,3 mm staaldraad, gebogen in cirkelbogen met het tegenoverliggende wiel als middelpunt.

*"Smiley", een robotproject voor de beginnende hobbyist*



Het tweede project, "Dizzy", is heel wat complexer. Deze robot wordt bestuurd door een zelfdenkende microprocessor. "Dizzy" laat zich zelden verrassen. Als je in zijn buurt komt, begint hij een "gesprek" in een merkwaardig eigen taaltje. Hij maakt ontdekkingsreizen door zijn "terrarium" en als hij honger krijgt, gaat hij naar zijn eettafel voor een lunchpauze van een uur of zo. Onder het eten blijft hij met veel commentaar reageren op zijn omgeving.

*De intelligente robot "Dizzy" wordt bestuurd met een ingebouwde microprocessor*



“Dizzy” heeft vijf verschillende zintuigen voor de waarneming van obstakels, accuspanning, laadstroom, gemoduleerd infrarood licht, daglicht en duisternis, beweging om hem heen en zijn eigen beweging. “Dizzy’s” voorbeeld in de dierenwereld is de degenkrab, een 400 miljoen jaar oude zeebewoner met een zwaardvormige staart. De robot heeft twee floretachtige voelsprietten. Al rijdend mept hij om zich heen naar obstakels, terugwijkend zodra een voelspriet contact maakt. Gemoduleerd infrarood licht wijst “Dizzy” de weg naar zijn eettafel. De acculader is verbonden met twee stukken printplaat. Een ervan maakt deel uit van de wand, de tweede ligt ernaast op de vloer van het terrarium. Wanneer een van “Dizzy’s” pootjes en een voelspriet contact maken met de twee platen, voelt de robot laadstroom en stopt hij. Een deel van zijn brein (dat meerdere taken tegelijk aankan) werkt als delta-V snellader. Zodra zijn 140 mAh NiMH-accu vol is, verlaat hij de eettafel. Hij eet gewoonlijk tussen de 1 en 2 uur per dag, verdeeld over 1 tot 4 maaltijden.

Als het donker wordt, valt “Dizzy” in slaap. Hij moet wel, want zijn belangrijkste sensor is afhankelijk van zichtbaar licht. In het donker kan hij geen beweging zien en dus niets meer doen. Je hoort alleen nog een zacht tikkend geluid. Een elektronisch gesnurk, veroorzaakt door zijn microprocessor. Elke vier seconden ontwaakt het brein heel kort uit een energiezuinige modus, om te zien of het nog steeds donker is. In de ochtendschemering houdt het tikken op en zie je de robot zich omdraaien om nog even door te dommelen, tot de nieuwe dag echt is begonnen.

**Micros & Robots**

“Micros & Robots” is een Frans tijdschrift dat zich volledig met robottechnologie bezig houdt. In ieder nummer worden diverse bouwbeschrijvingen van robot- en mechatronica-projecten gepubliceerd. Daarnaast veel informatie over commerciële bouwpakketten, speciale onderdelen en robottechnologie in het algemeen. Het Nederlandse bedrijf Arexx Engineering, gespecialiseerd in hobby-robots, heeft het initiatief genomen dit Franse tijdschrift naar Nederland te halen en het met een Nederlandstalige bijlage op de markt te brengen.

*Het tijdschrift “Micros & Robots” is een onmisbare informatiebron voor iedere robot hobbyist*



In die bijlage worden vrijwel alle artikelen vertaald, zodat dit prachtige tijdschrift ook voor de Nederlander en Vlaming toegankelijk wordt. Nader informatie op [www.arexx.nl](http://www.arexx.nl).

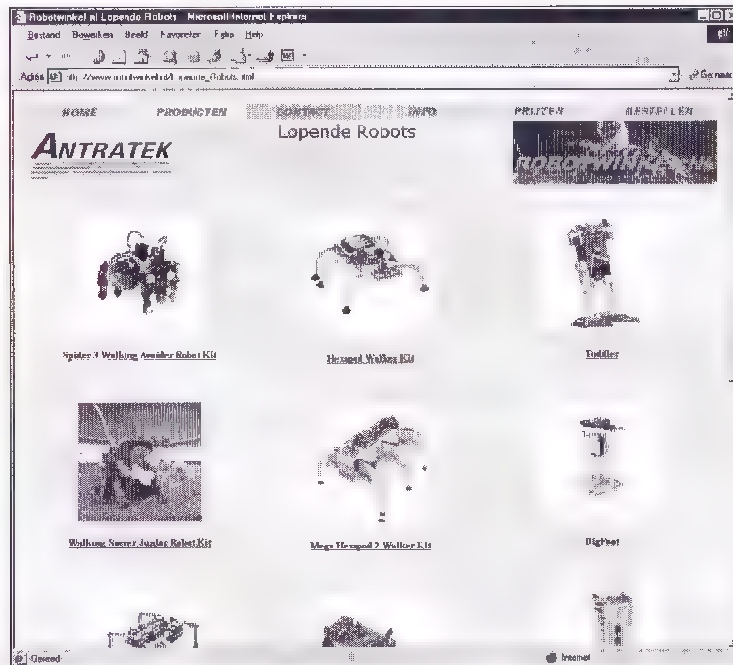
**Robotica.pagina.nl**

Een van de “dochteren” van startpagina.nl, die volledig gewijd is aan robotica. Dit is een zogenaamde portaal-site die links legt naar tientallen sites die iets met robotica te maken hebben.

**Robotwinkel.nl**

De Internet-site [www.robotwinkel.nl](http://www.robotwinkel.nl) biedt on-line tientallen complete bouwpakketten van robots aan. Er is voor ieder wat wils, van lopende robots over rijdende robots tot industriële robotarmen.

*Via de site [www.robotwinkel.nl](http://www.robotwinkel.nl) kunt u tientallen bouwpakketten van eenvoudige tot complexe robots bestellen*



**Arexx Engineering**

Dit reeds genoemde bedrijf doet meer dan een Frans tijdschrift vertalen. Op de site [www.arexx.nl](http://www.arexx.nl) worden tientallen zelfbouwrobots aangeboden, waaronder zeer leerzame via de PC te programmeren robotarmen.

*Een door Arexx Engineering op de markt gebrachte robotarm*

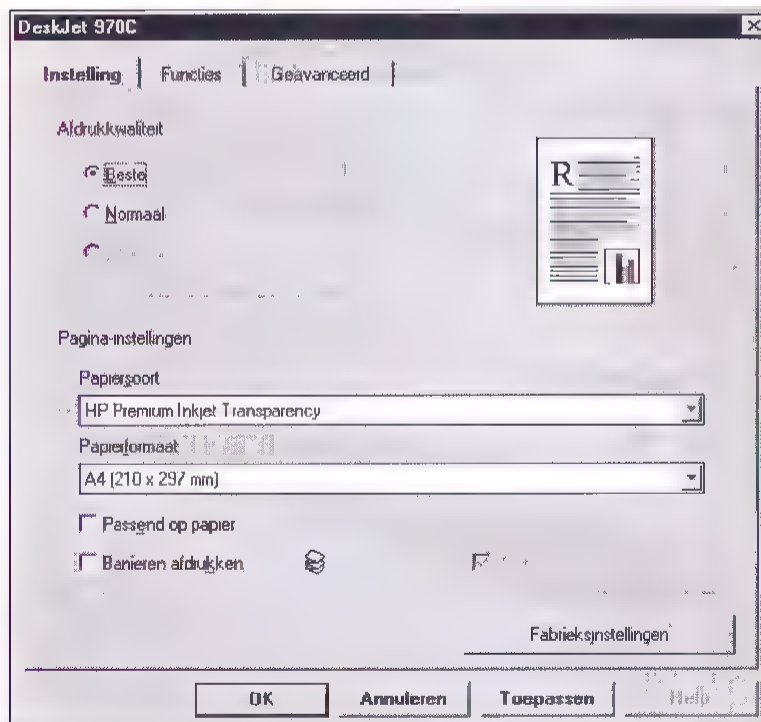


# 17 De printen uit dit werkboek

## Het maken van de printen

- Acht printen** In dit werkboek hebben wij de elektronica ondergebracht op acht printjes. Handig, maar hoe maakt u die?
- Ware grootte** Op de twee volgende pagina's hebben wij die acht printjes op ware grootte afgebeeld. Voor de duidelijkheid hebben wij er nog eens de afmetingen (breedte bij hoogte) bijgezet. Zo weet u in ieder geval hoe groot de printjes in het écht zijn.
- Scannen** De eerste stap is dat u de twee pagina's met deze printontwerpen uit dit boek haalt en deze scant. Doe dat met een resolutie van 600 dpi en in zwart/wit. U krijgt dan nog te behappen bestanden. Bewaar de scans als TIF.
- Knippen** Laad vervolgens een van die scans in een beeldbewerkingsprogramma, bijvoorbeeld Paint Shop Pro. Knip nu de gewenste print heel nauwkeurig langs de zwarte lijntjes uit de totale scan en bewaar de print als een afzonderlijke tekening.
- Printen** Druk deze tekening af op transparante folie op uw inkjet printer met de opgegeven afmetingen. Let er op dat u de papierkeuze op "Transparant" instelt en, indien mogelijk, de inktensiteit op maximum. Bij de meeste printerstuurprogramma's heet deze optie "beste printkwaliteit".

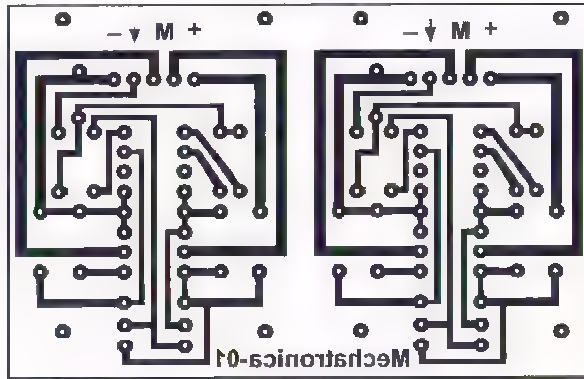
### Het instellen van een HP Deskjet op transparant papier en beste printkwaliteit



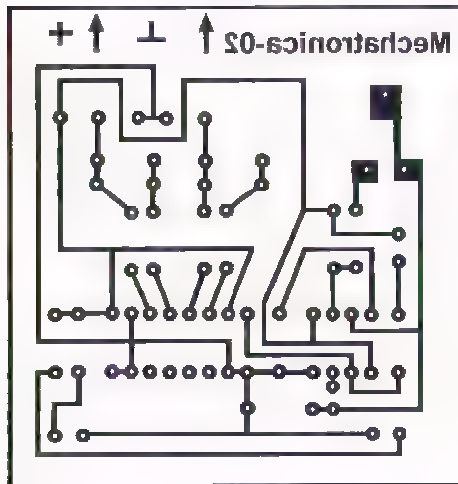
- Belichten** Leg de folie met de **bedrukte** zijde op een stukje fotogevoelige print en belicht deze op de gebruikelijke manier met UV-licht.
- Normale afwerking** De belichte print kan op de normale manier worden ontwikkeld, geëst en afgewerkt.

## De printontwerpen

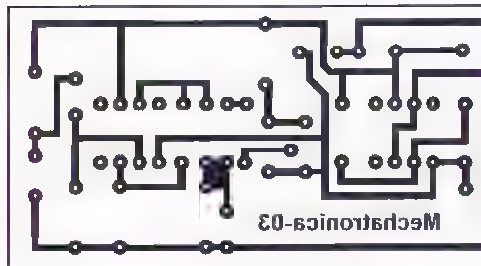
**Hoofdstuk 3**  
 pagina 22  
 afmetingen 75 x 48 mm



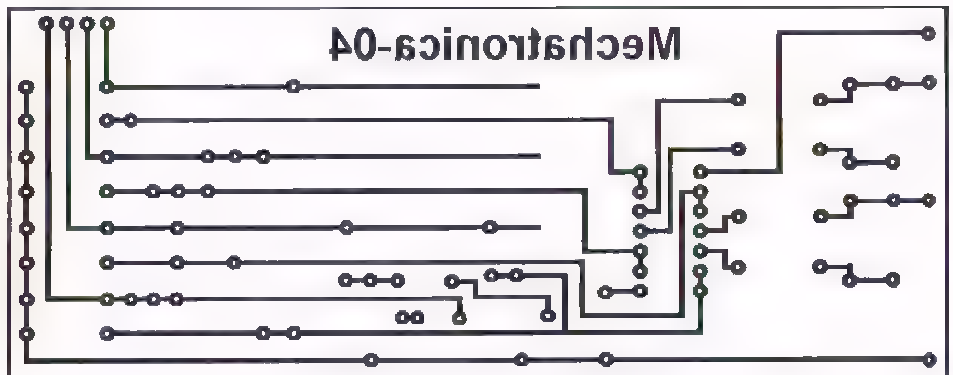
**Hoofdstuk 4**  
 pagina 28  
 afmetingen 59 x 62 mm



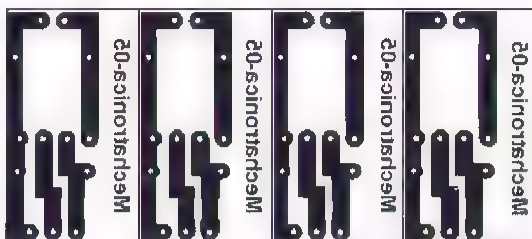
**Hoofdstuk 5**  
 pagina 32  
 afmetingen 62 x 34 mm



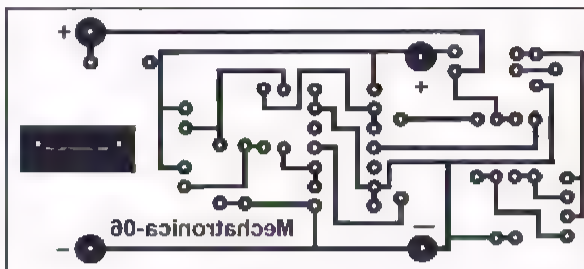
**Hoofdstuk 6**  
 pagina 37  
 afmetingen 120 x 48 mm



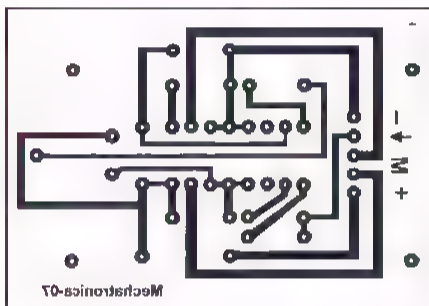
**Hoofdstuk 6**  
**pagina 37**  
**afmetingen 58 x 30 mm**



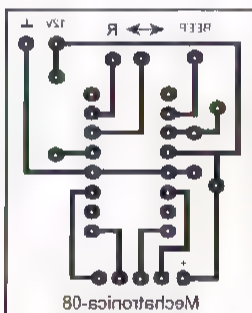
**Hoofdstuk 7**  
**pagina 46**  
**afmetingen 75 x 34 mm**



**Hoofdstuk 11**  
**pagina 61**  
**afmetingen 55 x 39 mm**



**Hoofdstuk 13**  
**pagina 76**  
**afmetingen 32 x 40 mm**

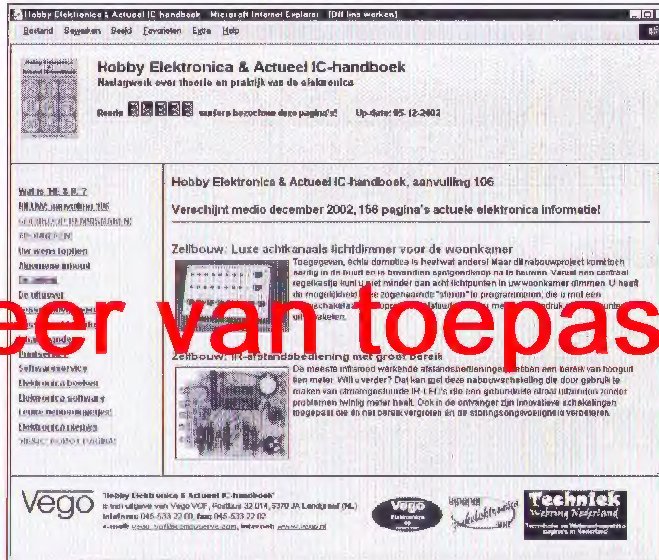


## Downloaden via het Internet

**“Hobby Elektronica & Actueel IC-handboek”**

De hoofdstukken 1 tot en met 7 van dit werkboek zijn ook verschenen in de jaargang 2002 van “Hobby Elektronica & Actueel IC-handboek”. Dit losbladig naslagwerk heeft een eigen uitgebreide Internet-site [www.hobbyelektronica.vego.nl](http://www.hobbyelektronica.vego.nl), waarvan alle gepubliceerde printontwerpen kunnen worden gedownload.

*De Internet-site van het losbladig naslagwerk “Hobby Elektronica & Actueel IC-handboek”*

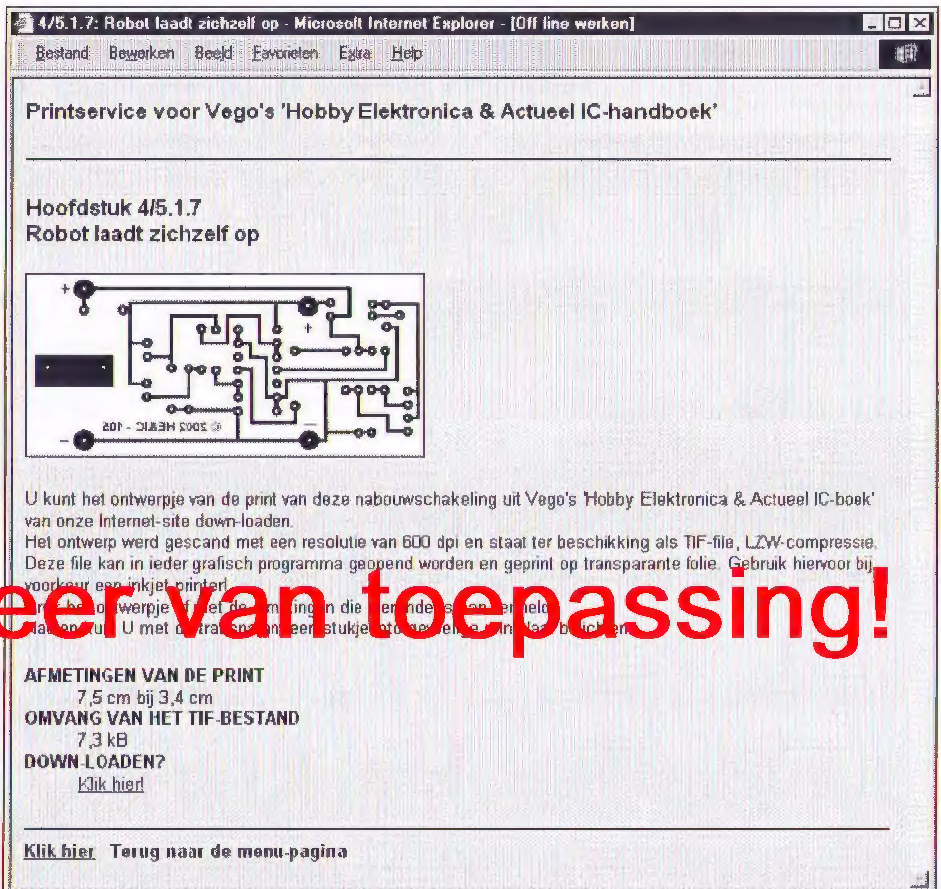


**niet meer van toepassing!**

**Printontwerpen downloaden**

Ga in het linker menu naar de optie “Printservice” en selecteer dan een van de aanvullingen 102 tot en met 105. In deze aanvullingen zijn namelijk de eerste zeven hoofdstukken van dit werkboek gepubliceerd. Klik dan op een van de robotprinten en download het ontwerpje vanuit onderstaande pagina.

*Via dit soort pagina's kunt u de eerste zes printontwerpen uit dit werkboek downloaden*



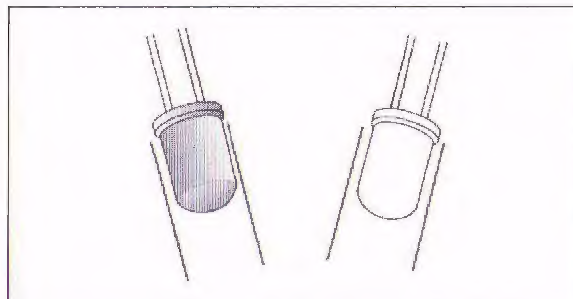
**niet meer van toepassing!**

# 18 Aanvullingen bij de derde druk

## Infrarood sensor S471F

De levenscyclus van functionele elektronica componenten wordt in het huidige ontwikkelingstempo steeds korter. Toch is op een aantal plaatsen in het boek bewust gekozen voor toegespitste componenten. Naarmate de verschijningsdatum van het boek verder achter ons ligt, wordt de kans groter dat bepaalde componenten niet meer verkrijgbaar zijn. De infrarood sensor S471F is bijvoorbeeld zo'n lot beschoeren. Gelukkig geeft het boek op pagina 43 een bruikbaar alternatief; het oog van de robot. Omdat deze schakeling geen gemoduleerd signaal gebruikt, moet wel enige aandacht gegeven worden aan de afscherming van storende lichtbronnen. Door de zender (een simpele infrarood LED met voorschakelweerstand) en de ontvanger (zoals geschetst onderaan pagina 43) beide te voorzien van een kartonnen koker (zie schets) wordt hetzelfde resultaat bereikt. Let erop dat zender en ontvanger elkaar goed zien.

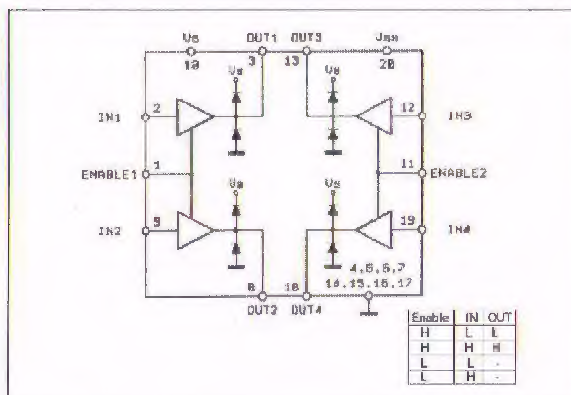
**Een alternatief voor de infrarood sensor S471F is gebruik te maken van een gewone infrarode LED en de fototransistor SFH205F**



## De TLE4206 motordriver

De projecten in het werkboek zijn gebaseerd op een motorsturing die gebruik maakt van het IC TLE4206. Hoewel deze bouwsteen prominent aanwezig was in de 2003 catalogus van Conrad, blijkt de verkrijgbaarheid problemen op te leveren. De zoektocht naar een minstens zo goed alternatief leverde een nieuw IC op, speciaal bedoeld voor aandrijving van stappenmotoren. De L293D (Conrad bestelnummer 17 40 03-44, telefoon 053-428.54.90) bevat vier kanalen waarvan de uitgang de ingang volgt.

**Het intern schema van de alternatieve motordriver L293D**



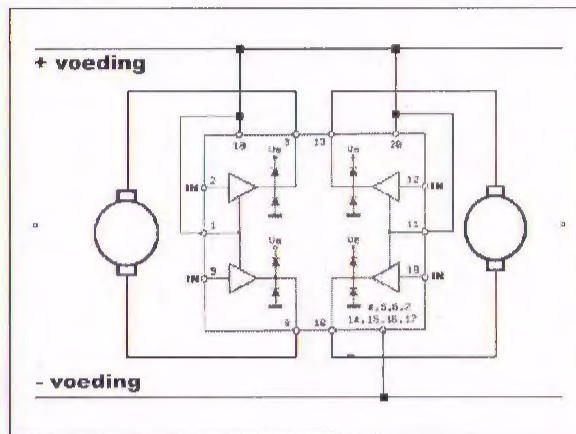
Door een motor aan te sluiten tussen twee uitgangen kan de draairichting bepaald worden door op de ingangen een logisch signaal aan te bieden. Als beide signalen hoog of laag zijn, staat de motor stil. Als de



ingangen verschillend zijn, bepaalt de polariteit de draairichting. De enable-ingangen gebruiken we niet. Die worden aan de plus gelegd. Kan het eenvoudiger?

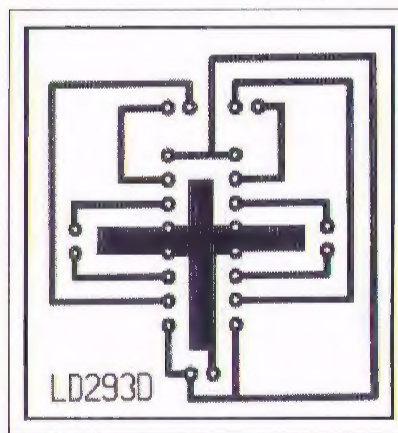
De aansturing van deze schakeling vraagt een kleine vereenvoudiging van de in het boek beschreven schema's. Alle weerstanden (tien stuks) die het R-2R netwerk vormen worden weggelaten. De uitgangen Q1 tot en met Q4 worden rechtstreeks verbonden met de ingangen van de L293D.

**De motoren kunnen rechtstreeks met de uitgangen van de L293D worden verbonden**

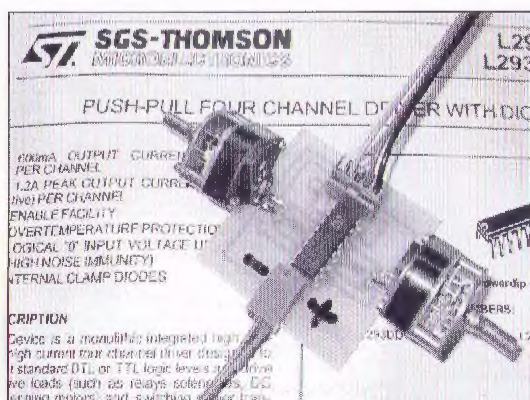


Het IC biedt de mogelijkheid de motoren en de logica apart te voeden. In dit geval is daar geen gebruik van gemaakt. Het kan natuurlijk wél. Ook het printje is kinderwerk. Met Abacoms Sprint Layout (zie de site van de uitgever, [www.vego.nl/abacom/02/02.htm](http://www.vego.nl/abacom/02/02.htm)) vraagt het ontwerp niet meer dan enkele minuten. Een afdruk met een inkjet printer op transparant materiaal levert direct de goede schaal op. Zoals de datasheet laat zien levert het IC 600 mA per kanaal. Vandaar dat de voedingsaarde op de print voorzien is van de mogelijkheid er een koelopervlak aan te solderen.

**Het alternatieve printje voor de motoraansturing (niet op schaal)**



**Het gemonteerde printje, voorzien van twee kleine motoren**



Willem H. M. van Dreumel

# Mechatronica en robotica

93 pagina's  
124 illustraties

NUGI 433

Verwacht in dit boek géén lopende en pratende elektronische honden. De ontwikkeling van dergelijke robots laten wij met plezier aan Sony over. Wij ontwerpen een robot op wieltjes. Oftewel een voertuig dat over een plat oppervlak kan rijden, aangedreven door een paar elektromotoren. Een robot die u kunt besturen met geluid of met licht en die in staat is bepaalde problemen "intelligent" op te lossen. In een bepaald stadium van zijn ontwikkeling is onze robot bijvoorbeeld zo slim dat hij niet van de tafel afrijdt. Iets later is hij in staat zijn accu's zelfstandig op te laden. Dát soort dingen kunt u van dit boek verwachten. Verwacht van dit boek ook geen volledig voorgekauwde kant-en-klare nabouwontwerpen. Dit boek is een **Werkboek!** Wij geven u suggesties hoe u de mechanica van een rijdende robot zo gemakkelijk mogelijk kunt bouwen. Wij bewijzen dat in eerste instantie ingewikkelde problemen als "*hoe kan een rijdende robot ontdekken dat hij van de tafel gaat afrijden en op tijd gepaste maatregelen treffen?*" in wezen tamelijk eenvoudig zijn op te lossen. Wij hopen dat het enthousiasme waarmee aan dit project is gewerkt overslaat op u, lezer, en de start wordt van een interessante en leerzame hobby:

**Mechatronica!**

Vego

ISBN 90-805610-7-X



9 789080 561076