



newbraindag

21 april 2001

20 oktober 2001

nonnenwater 8  
2801 VA gouda

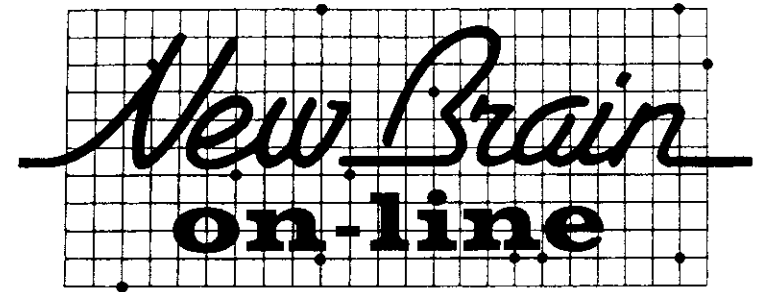
en het hele jaar door  
<http://www.hobby.nl/~newbrain-gg>

*NewBrain*-  
gebruikersgroep  
postbus 94494  
1090 GL amsterdam

# *New Brain* on-line

29  
april 2001





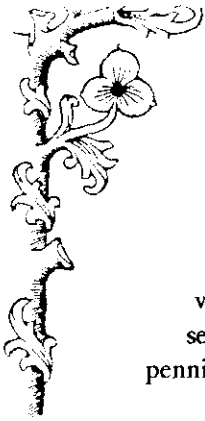
### ten geleide

in deze negenentwintigste uitgave van newbrain on-line heten we een nieuwe newbraingebruiker welkom: wim jansen laat zien dat de newbrain nog niet afgedaan heeft; en het beste daarbij is dat hij er meteen een stukje voor ons lijfblad over geschreven heeft

verder treft u hier een aantal interessante beschrijvingen aan van robots die aan de elektuurwedstrijd meegedaan hebben, en heeft abraham vreugdenhil nieuwe toepassingen bedacht voor het b+-bord

het volgende nummer van newbrain on-line is het dertigste. zo'n jubileumnummer leent zich er uitstekend voor om met een waaier van gevarieerde artikeltjes een overzicht te geven van wat er allemaal gebeurt in de gebruikersgroep. maar dat kan alleen als u allen een (kort) stukje kopij verzorgt

menno stevens



**NewBrain-**  
gebruikersgroep  
postbus 94494  
1090 GL amsterdam



voorzitter: jan wubben, (010) 4557698  
secretaris: maarten floor, (020) 4964374  
penningmeester: menno stevens, (020) 6924137  
dré jansen, (0174) 414199  
albert stuurman, (030) 2280163

postgiro 2505800 tnv hcc newbrain-gebruikersgroep

de newbrain-gebruikersgroep is een onderdeel van de  
hcc hobby computer club  
de molen 24, 3994 DB houten  
inschrijvingsnummer kvk utrecht 82311

website <http://www.hobby.nl/~newbrain-gg/>

landelijke newbraindag 20 oktober 2001  
in het clubhuis van de afdeling gouda  
nonnenwater 8, 2801 VA gouda

**newbrain on-line**

redactie: menno stevens, (020) 6924137

kopij voor nummer 30 graag naar het adres van de gebruikersgroep  
of per e-mail aan [mennostevens@hetnet.nl](mailto:mennostevens@hetnet.nl)  
geplaatste artikelen mogen alleen voor niet-commerciële doeleinden  
en onder bronvermelding worden overgenomen

# nieuw erbij

Toen ik tijdens de HCC-dagen 2000 een NewBrain-computer zag, die een grijp- en verplaatsrobot bestuurd, zag ik het licht! Zo klein en dankzij een Z80-hart zo krachtig, daar wil ik ook eens iets leuk mee doen. Zoals de vaste kern van HCC NewBraingebruikersgroep mij kent van de 6500 gg, en altijd bezig met home- en andere computers voor het besturen van Mecano-modellen, moet het er maar eens van komen om een NewBrain-modelbesturing te maken.

De keus van het model was niet moeilijk, want ik had net een klein miniatuur-draaibankje ter beschikking gekregen. Een klein model en een kleine computer met een kleine monitor, dat leek mij een goede combinatie. Toen ik Jan Wubben daarover raadpleegde, was hij direct bereid mij te helpen en stelde een computer en handboeken beschikbaar. Van een HCC-Roboticalid, Abraham Vreugdenhil, kreeg ik een PIO-interface te leen en een lesje in het programmeren.

Nu ben ik bezig proefprogramma's te schrijven en het lukt mij al om stappenmotoren te besturen met een SMC 800 interface, gekoppeld aan de PIO van de Z80. Voor HCC-leden die aan dit project voor de HCC-dagen willen meewerken, of goede raad willen geven, geef ik hierbij mijn adres: H. Kuipers Rietbergpad 17, 2286 PR Rijswijk, telefoon 070-3935695 (geen e-mail).

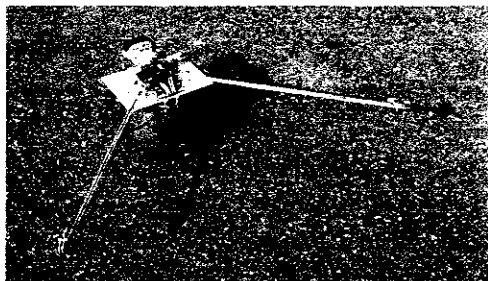


W. G. Jansen

# b+al.ans

De gedachtegang van B+AL.ANS is om een karretje te maken dat *op* een bal staat en zich in balans weet te houden. De uitdaging ligt hier op meerdere terreinen, te weten sensorenkeuze, mechanica, elektronica en het programma. Op welke wijze kun je meten of een object in balans is. Er zijn inclinometers (hellinghoek meters), kwikschakelaars en giroscofen. Alle te duur en ingewikkeld. Zelf een sensor maken valt ook niet mee.

Uiteindelijk viel de keuze op de GP2D02 van Sharp. Dit is een afstandsensor die werkzaam is tussen de 20 en 80 cm. Als deze sensor naar de grond gericht wordt, heb je een afstand



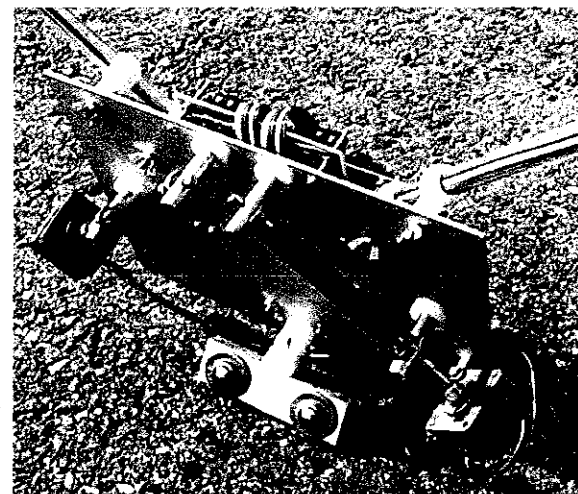
in de beginpositie van rond de 40 cm. Als je twee sensoren onder een hoek van 90 graden monteert, heb je zo de X- en de Y-as waaraan je twee motoren kunt koppelen. Elke motor werkt zo met zijn eigen sensor. Naarmate de afwijking van de gemeten afstand groter wordt, moet de stappenmotor sneller gaan draaien. Evenzo met de richting. Als de sensor te hoog zit moet de motor de ene kant op, zit hij te laag dan moet de motor de andere kant op. Op zich is het programma de eenvoud zelve.

Als uitgangspunt is een B+- bordje gekozen met daarachter een L297 als stappenmotor stuurlogica decoder en een L298 als een dubbele vermo-

gens H-brug die direct een stappen motor kan aansturen. De combinatie van deze twee IC's is mooi, omdat er stroombegrenzing mogelijk is en de stappenmotor op stroom in plaats van spanning te regelen is. Het resultaat hiervan is dat je bij een twee maal zo hoge spanning als geschikt voor de motor juist een goede snelle werking krijgt van de stappenmotor.

Het programma is geschreven in basic en wordt gecompileerd met behulp van *Bascom-LT*. Eerst wordt de afstand tot de vloer gemeten. Dit is de referentie-afstand. Daarna wordt iedere keer opnieuw de afstand gemeten. Deze afwijking in de afstand wordt op maximaal 8 begrensd. Als rekenformule wordt dan (16 - afwijking) genomen en deze waarde wordt in de Int.0 subroutine bekeken.

Elke 125  $\mu$ s wordt de Int.0 aangesproken en wordt een teller verhoogd. Als deze teller groter is dan de waarde (16 - afwijking), dan wordt *of* een lage positie van de puls hoog gemaakt, *of* een hoge positie van de puls laag gemaakt. Deze pulstrein is de aansturing voor de stappenmotor en bepaalt dus de snelheid. Tevens wordt dan de teller weer op nul gezet.



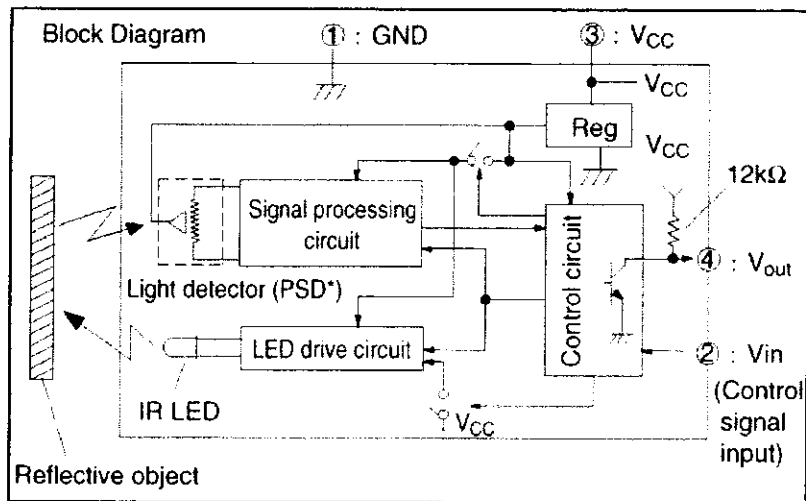
Mechanische constructie: Fred Sollart

Ontwerp, elektronica en programma: Abraham Vreugdenhil

# gp2d02

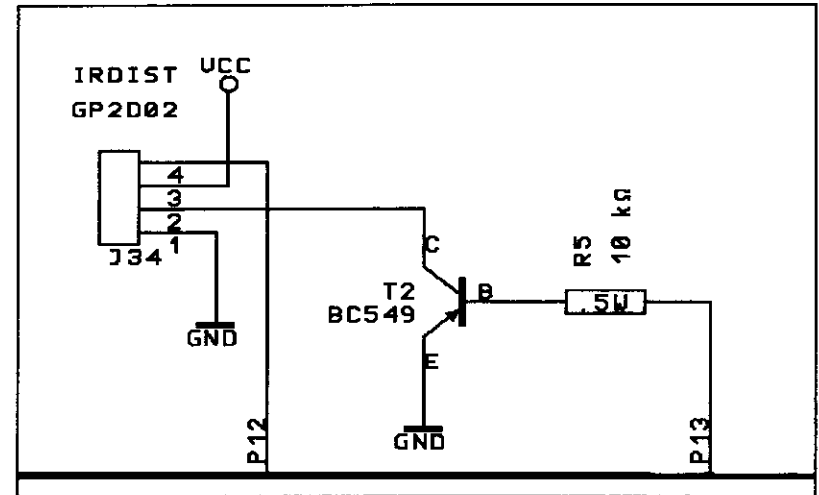
## de sharp gp2d02 afstandmeetsensor uitlezen met behulp van het b+-bordje

Tijdens het bouwen aan mijn B+ALANS zocht ik een methode om te kunnen meten, hoe de balans niet meer horizontaal was. Uiteindelijk kwam ik op de afstandmeetsensor GP2D02. Deze sensor meet tussen de 20 cm en de 80 cm en geeft digitaal de afstand weer. Mij leek dat de ideale sensor, alleen was de prijs een (klein) struikelpunt. Ruim 38 piek vroegen ze er bij *Conrad* voor. Er met een bestelling eerst een gekocht om te proberen of ze aan mijn wensen en eis zouden voldoen.



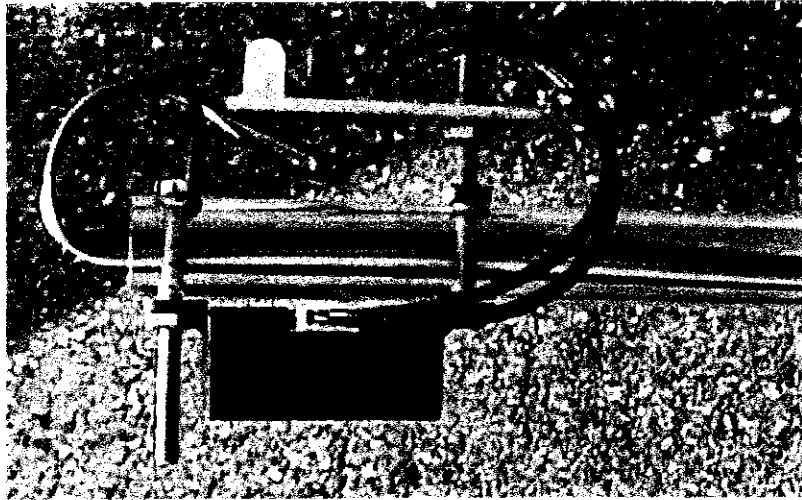
6

Uit de bijgeleverde *datasheet* blijkt, dat je hem met 3 volt (!) moet aansturen. Een vreemde waarde. Gewoon op 5 volt geprobeerd en het werkte dus *niet*. Hij zou hier snel van defect raken werd mij later verteld. Wat later hoorde ik dat Paul Wiegmans ook deze sensor gebruikt op zijn sumo-robot genaamd *Geesink*. Een e-mailtje later had ik een schema met daarin een transistor hoe deze sensor aan te sluiten.



Gebouwd en het werkte nog niet. Wat bleek (na het bestuderen van de signalen met een scoop), het signaal is na de transistor geïnverteerd. Programma aangepast en toen begon het ergens op te lijken. Uit onderstaand programma is aardig af te leiden, hoe je de sensor kunt gebruiken. Eerst maken we de signalen laag en als we de de meting starten dan wachten we eerst 10 msec en dan tasten we telkens de uitgang af in *Lusje*, totdat hij hoog wordt. Dan geven we een puls op de ingang en we kijken of de uitgang hoog of laag is. Dan geven we weer een puls op de ingang en kijken of hij dan weer hoog of laag is. Dit doen we 8 keer en we weten hoe de 8 bitjes er uitzien. De waarde ligt in theorie tussen 0 en 255, maar in de

7



praktijk tussen 255 voor een korte afstand en 60 voor een lange. Dit getal kunnen we naar de seriële poort van de computer sturen. Dit doen we in een van de laatste regels met het *print*-commando. Daarna beginnen we weer opnieuw.

Als je meerdere sensoren in een systeem gaat gebruiken, moet je erop letten, dat de sensoren *niet* gelijk zijn. Elke sensor geeft een iets andere waarde per afstand af. Op zich is dat niet erg, als je het maar weet.

Ook deze sensor is leuk om te gebruiken in een robot of een ander automatiseringssysteem dat je met behulp van je B+-bordje maakt. Veel plezier ermee. Zijn er vragen, dan hoop ik dat je me weet te vinden.

Abraham Vreugdenhil  
 a.vreugdenhil@hccnet.nl  
 0174-420361 (een beller is sneller)

```
' SHARP GP2D02 ontvanger op het B+-bordje door AV
',
' P4.4 = uitgang sensor
' P4.5 = ingang sensor
',
```

```
$baud = 4800           ' communicatiesnelheid
$crystal = 12000000   ' kristal
$romstart = &H8100    ' begin B+-bordje
```

```
Dim In As Byte
Dim Dat As Byte
Dim Buffer As Bit
Dim In1 As Byte
Dim T As Byte
Dim Tt As Byte
```

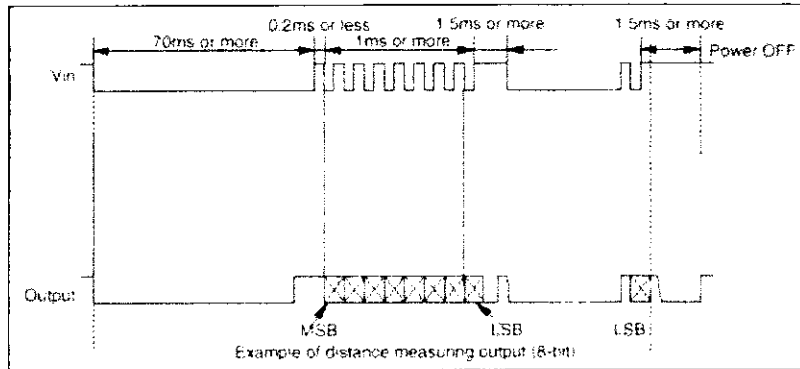
```
P4 = 0
Begin:
```

```
P4.4 = 1           ' start meting
P4.5 = 1
Waitms 10         ' wacht even
```

```
Lus:
If P4.4 = 0 Then   ' ongev. 70 mSec.
  Goto Lus
End If
```

In = 0

```
For Dat = 1 To 7          ' eerste 7 bits inklokken
  P4.5 = 0
  For Tt = 1 To 5 : Next Tt ' korte wachttlus
  P4.5 = 1
  For T = 1 To 5 : Next T  ' < 200 uSec.
  Buffer = P4.4 : In1 = 0  ' lees uitgang in
```



```
If Buffer = 1 Then      ' als 1 dan +1
  In1 = 1
End If                  ' anders + 0
In = In + In1          ' optellen
Rotate In , Left       ' schuif naar links
```

Next Dat

```
P4.5 = 0
For Tt = 1 To 5 : Next Tt
P4.5 = 1
For T = 1 To 5 : Next T
Buffer = P4.4 : In1 = 0
```

```
If Buffer = 1 Then
```

```
  In1 = 1
```

```
End If
```

```
In = In + In1
```

' laatste optellen en dan niet  
' meer schuiven

```
Waitms 1
```

```
P4.5 = 0
```

```
Waitms 2
```

' stop meting

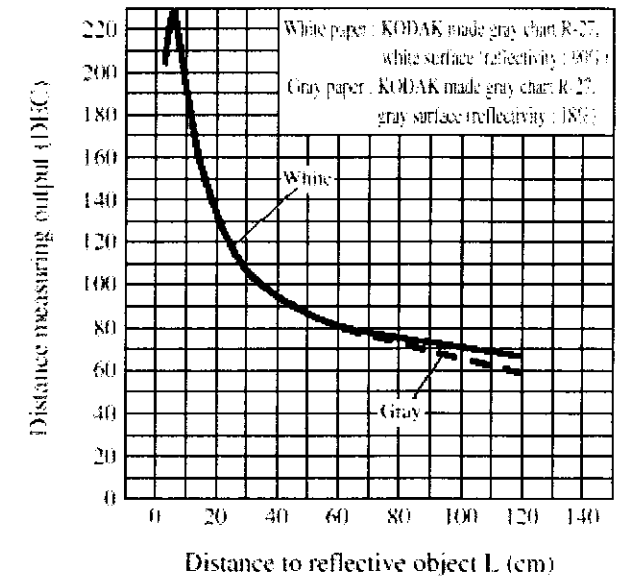
```
Print In
```

' stuur waarde naar monitor

```
Goto Begin
```

' opnieuw

```
End:
```



# lego

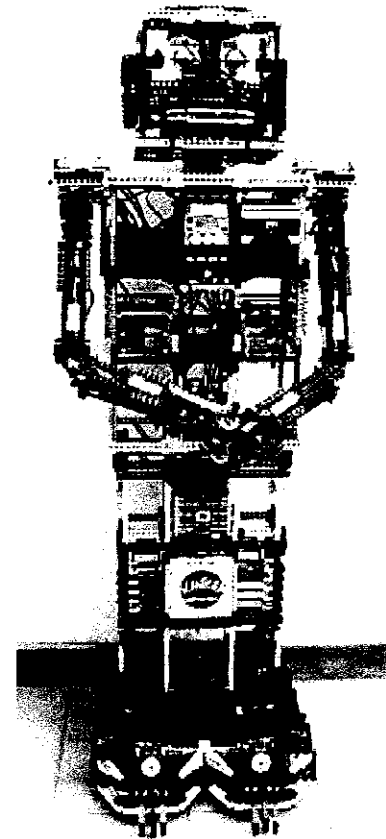
## lego-robots met menselijke trekjes

Sinds enkele jaren loopt er in België een robot rond met heel wat menselijke trekjes: *Robaticus Legorius*. Zo kan hij zijn hondje uitlaten, je vriendelijk goedendag knikken en zijn tenen krullen. Sinds vorig jaar heeft hij gezelschap gekregen van een goede vriend, *Mega-Teek*. Dit is een robot die een simulatie is van de bewegingen van een echte teek en die in de toekomst echt zou moeten kunnen rondhollen. Beide robots hebben enorme afmetingen om gewoon uit Lego-steentjes gemaakt te zijn, maar toch is het zo.

Omdat beide robots twee of drie RCX'en gebruiken, zal ik beginnen met een woordje uitleg over een RCX. Een RCX is een programmeerbare Legosteentje. Per RCX zijn er drie sensor-ingangen (voor druk-, positie-, temperatuur- en lichtsensoren) en drie inwendige sensoren (teller, timer en RCX-berichtenhouder). De RCX-berichtenhouder is nodig voor de communicatie tussen twee of meer RCX'en. Per RCX is er een luidspreker voor tonen en zijn er ook drie motoruitgangen voor onder andere lampjes en motoren.

Het programmeren kan gebeuren met de *RCX-code*, een speciale programmeertaal voor de RCX ontworpen door Lego, of door het gebruik van andere programmeertalen (basic...). Nadien wordt het programma naar het interne geheugen van de RCX gestuurd door middel van infrarood. Er kunnen maximaal 5 programma's opgeslagen worden in een RCX.

## robaticus legorius



Ik zal eerst beginnen met een beetje geschiedenis over de menselijke robot Robaticus Legorius. In de zomervakantie van 1998 heeft mijn vader me gezegd, dat ik al mijn *Lego Technic*-modellen moest afbreken en er één groot ding van moest maken; zo zou er eindelijk een einde komen aan de dagen dat ik me verveelde. Zo ben ik begonnen na te denken over een robot die ik zou kunnen maken van al die Lego-steentjes. Tijdens de kerstexamens heb ik mijn ideeën op papier gezet en nog wat ideeën bijgevoegd: het moest een grote menselijke robot worden met heel wat menselijke trekjes. In februari van het jaar 1999 ben ik dan begonnen te bouwen.

Mijn eerste gedacht was te beginnen met de voeten; deze zijn toch de basis van een mens, de basis waarop het lichaam rust. Maar dit viel tegen, de voeten wilden niet lukken. Ik had namelijk nog altijd het gedacht om de voeten veringen te geven tegen schokken, maar dan zouden steeds de veringen ingezakt zijn en niet meer veren door het eigen gewicht van de robot. Bovendien moest Robaticus Legorius vooruitrijden met *CyberMaster*. Maar de motoren waren niet krachtig genoeg.



Dan ben ik maar begonnen met het lichaam, dat ging me beter af. Eerst de pompjes onderaan inbouwen, het *Lego Technic Control Center* erboven voor het fronsen van de wenkbrauwen en het knikken van het hoofd. Daarboven het hart, bestuurd met de *Scan-Unit*, met de pneumatische ventielen. Het volledige pneumatische circuit is te zien op één van de afbeeldingen. Langs beide zijden van het lichaam zijn er 6 ventielen voorzien. Een voor het kiezen tussen een compressorpomp of een drukpomp om de lucht te leveren. Dan nog een voor het op-en-neergaan van de armen, twee voor het buigen van de armen en nog een voor het openen van de hand. Het zesde ventiel is een ventiel waarmee de tong kan bestuurd worden (rechts) of kan gekozen worden tussen niezen of winden laten (links). Deze twee laatste ventielen worden van lucht voorzien door een tweede drukpomp die in verbinding staat met een luchttank. Immers het niezen is niets anders dan het leeg laten lopen van de luchttank.

Na het bovenlichaam kwamen de benen aan de pas en tenslotte nog de voeten voor een tweede poging. Deze poging ging me beter af, er werden geen veringen meer gebruikt en *CyberMaster* werd gebruikt voor de bewegingen van de tenen. Ondertussen had ik mijn eerste *RCX* gekocht en deze ingebouwd voor de bewegingen van het hoofd: de mond bewegen, samen met het flapperen van de oren, de ogen licht geven en het hoofd draaien.

Op 27 november 1999 was het dan zover: *Robaticus Legorius* werd geboren, hij was volledig afgewerkt. Rond deze tijd won hij ook zijn eerste wedstrijd: een uitvinderwedstrijd voor kinderen onder de 16 jaar. Hij was één van de elf gelukkigen en de uitvinder mocht met de klas naar *Technopolis* in Mechelen. Zo werd hij een beetje bekend en kon ik demonstraties gaan houden met mijn robot in mijn lagere school.

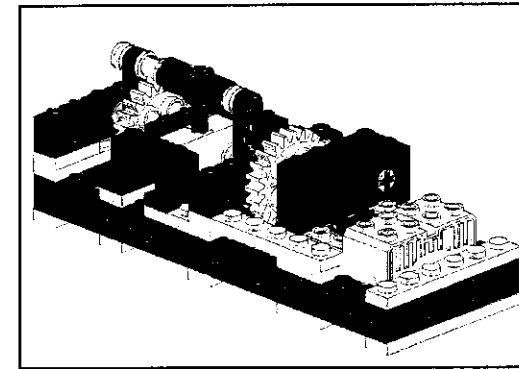
In februari 2000 was *Robaticus Legorius* eindelijk helemaal geprogram-

meerd en heb ik hem ingeschreven voor een wedstrijd via internet in Amerika. Na een maand kreeg ik bericht dat hij gewonnen had en kreeg ik mijn tweede prijs toegestuurd: de *Droid Developer Kit* van Lego.

Doordat ik ondertussen weer wat blokjes had verzameld, kon ik mijn robot uitrusten met een nieuw hart-long-bloedvaten systeem. Dit is een systeem waarbij de longen in en uit bewegen en het hart 2 tellen slaat en 1 te rust. In de linker borstholte kun je het bloed in uit zien stromen. Ten-

slotte heb ik hem ingeschreven voor de *Elektuur-robotwedstrijd*, die hij uiteindelijk ook won in de categorie onder de 18 jaar.

Eind december 2000 ben ik begonnen met de veranderingen aan de voeten. Hierdoor werd het slechte (en kwetsbare)



*nieuwe mogelijkheid voor een schakelaar*

re) systeem voor het bewegen van de tenen vervangen door een nieuw en sterker systeem. De *CyberMaster* unit werd eveneens verwijderd. Zo is er plaats vrijgekomen om in de toekomst eventueel een *RCX* te installeren. Deze *RCX* zou dan gebruikt worden om *Robaticus Legorius* vooruit te laten rijden en te laten draaien.

Oorspronkelijk werden er verschillende programmeerbare units gebruikt (*CyberMaster*, *Scan*, *RCX*, *Lego Technic Control Center*), maar geleidelijk aan schakel ik meer over op de *RCX*. Ik zal je nu ook nog even een gedacht geven over de afmetingen van *Robaticus Legorius* en mijn toekomstplannen vertellen.

Robaticus Legorius is 115 cm hoog en weegt ongeveer 7 kg; er werd een 7000-tal blokjes gebruikt, waaronder 8 motoren, 7 druksensoren, 1 lichtsensor. In het geheel werd er ook ongeveer gebruik gemaakt van 10 meter pneumatische slangetjes.

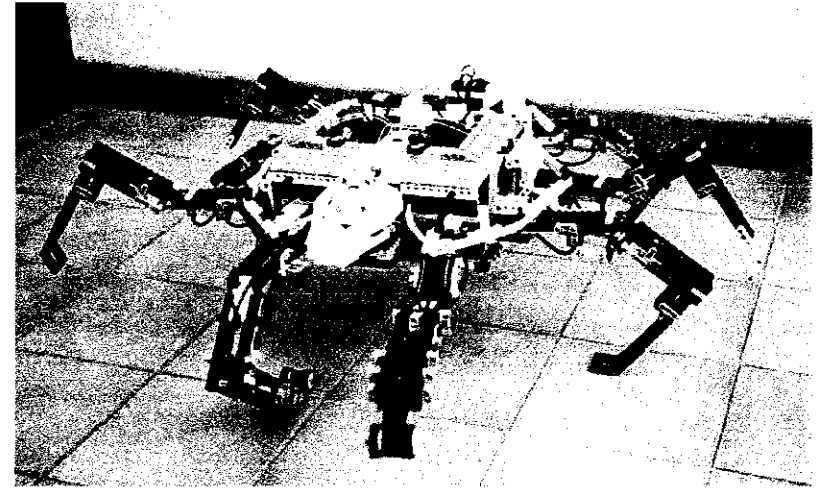
In de toekomst wil ik proberen Robaticus Legorius nog uit te rusten met twee benen om echt vooruit te wandelen. Dit zal naar het voorbeeld zijn van een Canadees die al twee wandelende benen heeft ontwikkeld.

### mega-teek

Mijn tweede robot is Mega-Teek: de simulatie van de bewegingen van een echte teek, maar dan in het megagroot. Ook hier zal ik beginnen met een stukje geschiedenis.

Ooit had ik in een natuurtijdschrift een klapperdief zien staan. Ik vond dit een machtig beestje en had dan maar het gedacht dit zelf eens te maken. Ik ben weer plannen beginnen te tekenen en ideeën op te schrijven. Ondertussen kwam er ook kruisspin aan te pas. Hieraan heeft Mega-Teek het oplichtende kruis op zijn rug te danken, evenals het aantal poten. Toen tenslotte een goede kennis langskwam, die heel wat over teken afwist en vond dat mijn nieuwe robotje op een teek leek, heb ik hem Mega-Teek gedoopt.

Op 24 augustus 2000 was het dan zover en werd Mega-Teek geboren; hij was ook volledig afgewerkt. Het programma liet nog wat op zich wachten, maar uiteindelijk verliepen alle bewegingen naar wens. In september heb ik hem dan ook ingeschreven voor de robotwedstrijd van Elektuur, waar hij moest strijden tegen zijn vriend Robaticus Legorius. Uiteindelijk heeft Mega-Teek verloren en de tweede prijs in de wacht gesleept.

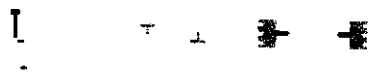
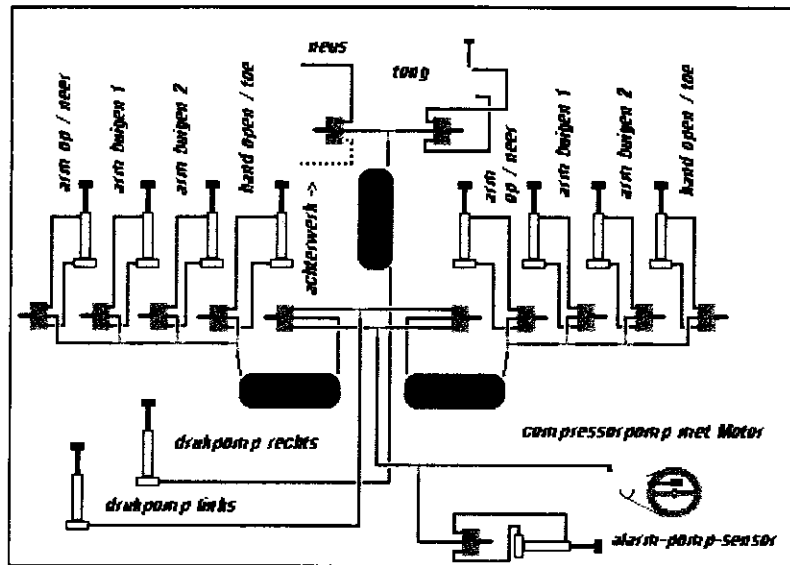


Na de wedstrijd had ik een idee hoe ik hem iets lichter kon maken: door de ventielen iets te verkleinen. Dit haalde niet veel uit, maar zo werd er toch al minder plaats ingenomen. Op demonstraties had ik ook gemerkt dat de bewegingen niet vlot verliepen. Dit kwam doordat de drie RCX'en slecht georiënteerd waren en de infraroodstraling naar boven ging en oploste in de lucht, voordat ze werd ontvangen door een andere RCX. Thuis had ik dit probleem niet, doordat de hoogte van mijn kamer lager was. Tijdens de wijzigingen aan de ventielen heb ik dan ook dit probleem verholpen door ze anders te oriënteren, met de infraroodzenders naar elkaar toe.

Het pneumatisch systeem van Mega-Teek is nog uitgebreider dan dat van Robaticus Legorius, doordat alle bewegingen van de poten gecoördineerd worden door middel van het pneumatische systeem. Elke poot is uitgerust met twee pompen: een pomp voor het op-en-neerbewegen en een pomp voor het vooruitbewegen. De ventielen voor deze pompen worden omgeschakeld door motoren. De stroom voor de motoren is afkomstig van de

RCX'en. De stand van de ventielen wordt bepaald door positiesensoren (zie ook tekening).

In elke voorpoot zit er een druksensor, waarmee hij kan voelen of er zich een muur voor hem bevindt, of dat hij zich boven een gat in de grond bevindt. Als er dus een muur of een gat is, zal Mega-Teek draaien (linkerkant vooruit, rechterkant achteruit, of omgekeerd). Achteraan zit er ook nog een sensor om botsingen van achter te vermijden. Verder is er nog een



lichtsensor voor het ontsteken van het kruis op de rug of de ogen. Naast de lichtsensor is er ook nog een "OFF-druk" druksensor. Als ik op deze druksensor duw, zakken alle poten tot op de grond en worden de ventie-

len neutraal gezet. Voor al deze bewegingen wordt er gebruik gemaakt van drie RCX'en met druk berichtjesverkeer.

Voor het echt vooruitlopen is er nog altijd het gewichtprobleem. Dit zou kunnen verholpen worden door de batterijen in de RCX'en te vervangen door dunne elektriciteitsdraden aan te sluiten op de RCX'en en de stroom te leveren door het *Lego Technic Control Center* dat elektriciteit aftapt van het stroomnet. Door deze verandering zou Mega-Teek een 600 gram minder wegen.

Nu zal ik ook hiervan nog enkele cijfers en afmetingen geven:

2000 blokjes	4 positiesensoren
8 motoren	5 druksensoren
4 ventielen	1 lichtsensor
20 pompen	
61 cm lang (met uitgestoken poten: 82 cm)	
70 cm breed (met uitgestoken poten: 85 cm)	

Nu ben ik bezig mijn twee robotjes met virtuele Lego-steentjes te bouwen op de computer. Verder heb ik nog plannen voor nieuwe robots: een grote kermiswagen, een kat-en-muisspelletje met *Vision Command...* Verder zou ik willen meedoen aan andere robotwedstrijden.

En tenslotte nog een goede raad: bouw je eigen ideeën, niet die van een ander, zo heb je de meeste pret! Reacties (met als onderwerp de naam van de robot) naar [luc.verschueren@advalvas.be](mailto:luc.verschueren@advalvas.be).

Maarten Verschueren

# 3d-robot

## inleiding

De 3D-robot is een 6-assige robotarm. De arm wordt aangedreven door 6 stappenmotoren. De grijper wordt bediend door een servo. De hardware bestaat uit een zelfontworpen stappenmotorkaart. Deze kaart is gebaseerd op een Microchip PIC16F877 microcontroller. Op de kaart kunnen 6 stappenmotoren aangesloten worden. Daarnaast beschikt de stappenmotorkaart over 14 i/o-lijnen en een RS232-poort. Een pc-voeding voorziet de print en de motoren van energie.

De software van de microcontroller is geschreven in C (CCS). De positionering en het refereren worden zelfstandig door de stappenmotorkaart uitgevoerd. Via de RS232-poort kunnen de gewenste posities van de motoren ontvangen worden. De huidige posities worden teruggestuurd.

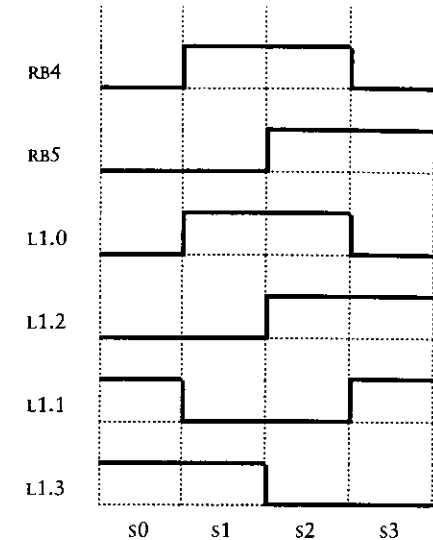
Op de pc draait een in Microsoft Visual C++ 6.0 geschreven programma. In dit programma worden de gewenste 3D-coördinaten ingevoerd. Ook kunnen de standen van de pols, grijperrotor en grijper opgegeven worden. De robot kan zowel handbediend als automatisch werken. In automatisch bedrijf kunnen 100 voorgesprogrameerde posities doorlopen worden.

## elektronische hardware

De hardware is getekend in ULTicap en ULTiboard. De bestanden staan op de bijgeleverde CD-ROM in de directory 'hardware'. De eindtrappen

van de stappenmotoren zijn allemaal gelijk. Ze worden elk door twee uitgangen van de controller bestuurd. De databits en de uitgangsströmen van motor 1 zijn in afbeelding 1 getekend. De spoelen L1.0, L1.2, L1.1 en L1.3 worden achtereenvolgens bekrachtigd.

Via J5 kan de controller met behulp van de *in-circuit debugger* (ICD) van Microchip geprogrammeerd worden. De kabel die bij de ICD geleverd wordt is echter gekruist. Voor de stappenmotorkaart is een 1-op-1 kabel nodig.



afbeelding 1: spoelstromen

J4 is een 3-draads RS232-poort. Via deze poort wordt met de pc gecommuniceerd.

tabel 1: overzicht van motoren			
motornummer	functie	A-uitgang	B-uitgang
1	roteren (a)	RB4	RB5
2	schouder links (b)	RD2	RD3
3	schouder rechts (b)	RD4	RD5
4	elleboog (c)	RD6	RD7
5	pols links (d)	RB1	RB2
6	pols rechts (e)	RD1	RD0

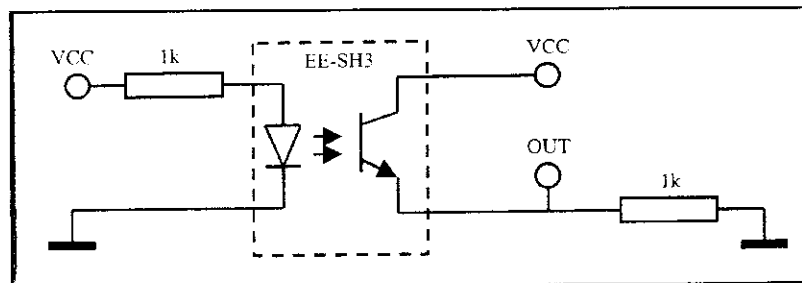
Via J13 zijn alle ongebruikte i/o-lijnen van de controller naar buiten ge-

J13 pin	functie	bit
1	dig. i/o; ADC (sensor)	RA0
2	dig. i/o; ADC (sensor)	RA1
3	dig. i/o; ADC (sensor)	RA2
4	dig. i/o; ADC (sensor)	RA3
5	dig. i/o (open collector)	RA4
6	dig. i/o; ADC (sensor)	RA5
7	dig. i/o; ADC (sensor)	RE0
8	dig. i/o; ADC (sensor)	RE1
9	dig. i/o; ADC (sensor)	RE2
10	dig. i/o; counter	RC0
11	dig. i/o; PWM (servo)	RC1
12	dig. i/o; PWM (servo)	RC2
13	dig. i/o; interrupt	RB0
14	dig. i/o	RC5
15	0..5 V	Pot
16	0..5 V	Pot
17	+5 V	Vcc
18	+5 V	Vcc
19	0 V	Gnd
20	0 V	Gnd

voerd. Op de pennen 1 tot en met 5 (poort A) zijn de referentiesensors van de stappenmotoren aangesloten (bij het refereren wordt eerst naar de posities van de sensoren gedraaid). Op pen 11 (RC1) is de servo van de grijper aangesloten. In tabel 1 staan de verschillende motoren met hun functie en de bijbehorende i/o-lijnen. In tabel 2 zijn de diverse functies van J13 beschreven.

De sensoren zijn van het type EE-SH3. Dit zijn optische schakelaars, die bestaan uit een LED en

een fototransistor. Om het uitgangssignaal op een ingang van de micro-



afbeelding 2: sensorschakeling

controller aan te kunnen sluiten is de schakeling uit afbeelding 2 gebruikt. De weerstanden zijn in de aansluitkabel van de sensor opgenomen.

## mechanische hardware

Elke robot bezit ook een mechanisch gedeelte. Zo ook de 3D-robot. Zoals al eerder vermeld wordt de robot aangedreven door stappenmotoren. Al de gebruikte stappenmotoren zijn afkomstig uit oude naaldprinters. Voor het bedienen van de grijper is een servo op de arm gemonteerd. In afbeelding 3 is vereenvoudigd weergegeven hoe de arm aangedreven wordt. Motor A zwenkt 1de robotarm. De motoren B dienen voor het heffen van de schouder. Motor C bedient de elleboog. Motoren D en E bedienen gezamenlijk de pols en de grijperrotor. Als beide motoren omhoog draaien, dan zal de pols omhoog bewegen. Als de motoren tegengesteld draaien, dan zal de grijper roteren. Alle overbrengingen zijn uitgevoerd als getande riem. De servo trekt via staalkabels de grijper dicht. Het openen van de grijper gebeurt door middel van veren.

## microcontroller-software

De software van de controller zet de instructies van de PC om in posities van de stappenmotoren en de servo. De stappenmotoren kunnen op twee manieren aangestuurd worden:

*Hard wil zeggen:* er wordt geen gebruik gemaakt van acceleratie en deceleratie. De motor draait direct op de opgegeven snelheid en staat daarna direct weer stil.

*Soft wil zeggen:* er wordt gebruik gemaakt van acceleratie en deceleratie. De motor loopt langzaam aan (in ongeveer 0,5 seconde) en remt ook lang-

zaam weer af (ook in ongeveer 0,5 seconde). Deze manier van aansturen is gunstiger, omdat het traagheidsmoment van de assen nu minder van invloed is op het door de motoren te leveren koppel.

### mogelijke instructies voor de robot

Refereren (home):  $H<LF>$

Gewenste positie opgeven:

$Mxsssppppm<LF>$

x = motor (A, B, C, D of E)

sss = maximale snelheid in stappen per seconde

pppp = gewenste positie in stappen

m = mode: H = hard, S = soft

Stand van grijper opgeven:  $GFppp<LF>$

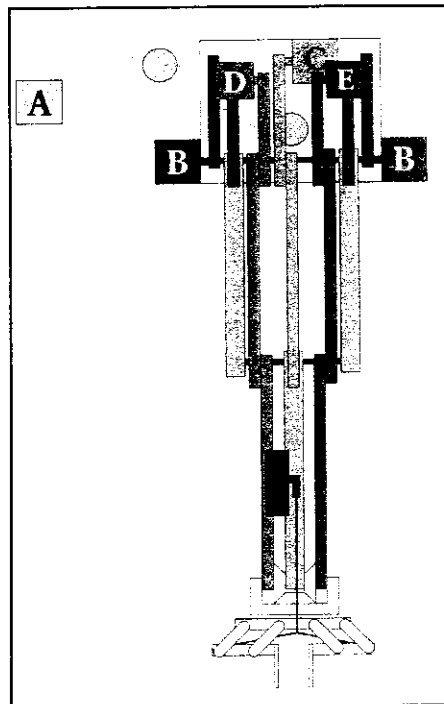
ppp = positie van servo in procenten

De reactie van de robot is als volgt opgebouwd:  $Mxssspppp<LF>$

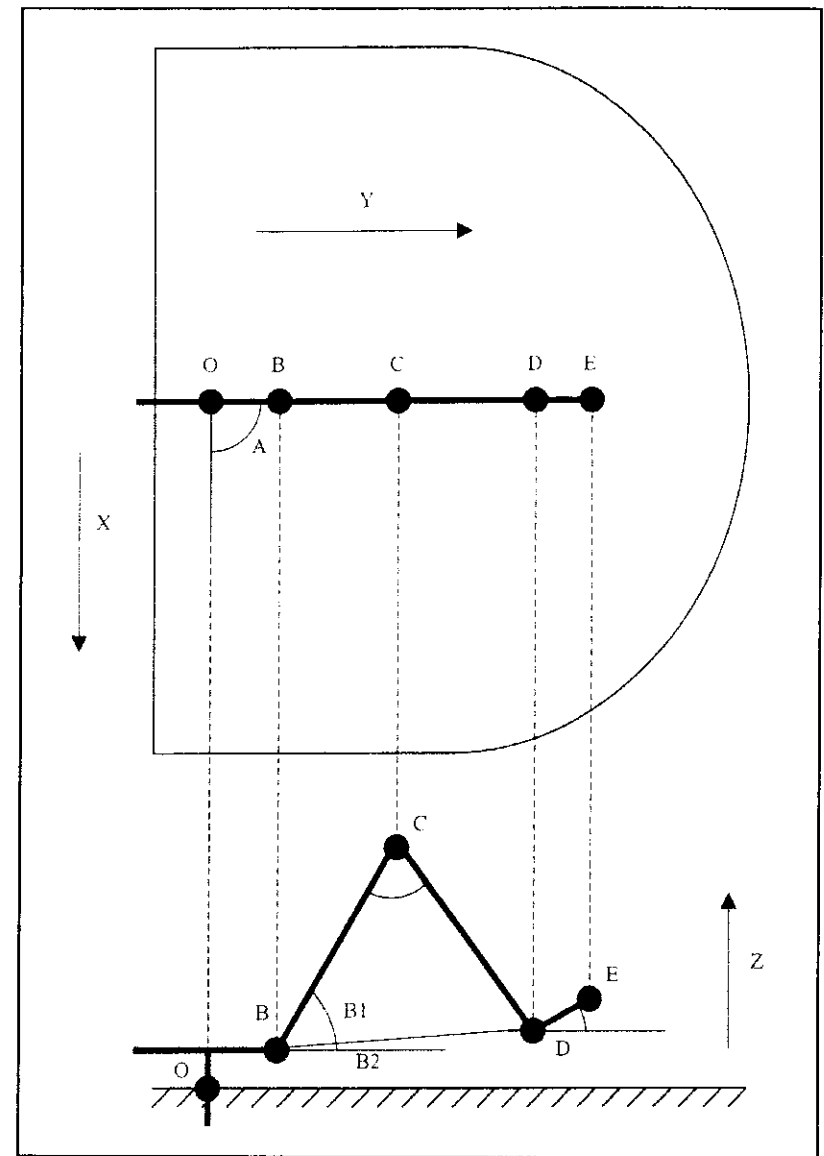
x = motor (A, B, C, D of E)

sss = maximale snelheid in stappen per seconde

pppp = huidige positie in stappen



afbeelding 3: mechanisch ontwerp



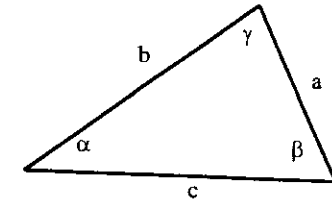
afbeelding 4: benamingen van de diverse hoekpunten

## omrekenen van coördinaten naar motorhoeken

Omdat het voor het positioneren van de gripper makkelijker is om coördinaten in te geven dan motorhoeken, is besloten om de hoeken door de pe-software te laten berekenen. Dit wordt gedaan in de functie `CRobotDlg::CalcAngles()`. In afbeelding 4 is te zien hoe de verschillende armhoeken benoemd zijn. Een omschrijving van de variabelen is hieronder weergegeven:

O	oorsprong
B	hoek van de schouder
C	hoek van de elleboog
D	hoek van de pols
E	uiteinde van de gripper
R	radius in het horizontale vlak ( $OE_{\text{horizontaal}}$ )
OB_R	afstand tussen O en B in het horizontale vlak
OB_Z	afstand tussen O en B in de Z-richting
BC	afstand tussen B en C
BD	afstand tussen B en D
CD	afstand tussen C en D
DE	afstand tussen D en E
DE_R	afstand tussen D en E in het horizontale vlak
DE_Z	afstand tussen D en E in de Z-richting

Als X, Y, Z en D worden opgegeven, dan zijn A, B en C te berekenen. Hierbij wordt gebruik gemaakt van de sinus- en de cosinusregels.



Sinusregel: 
$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma}$$

Cosinusregels:

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos \alpha$$

$$b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos \beta$$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos \gamma$$

Berekenen van hoek A:

$$A = \arctan (Y / X)$$

Berekenen van hoek C met de cosinusregel:

$$R = \sqrt{X^2 + Y^2}$$

$$DE\_R = DE \cdot \cos(D)$$

$$DE\_Z = DE \cdot \sin(D)$$

$$BD = \sqrt{(R - OB\_R - DE\_R)^2 + (Z - OB\_Z - DE\_Z)^2}$$

De drie zijden van de driehoek BCD zijn nu bekend. Hoek C kan dan als volgt met de cosinusregel berekend worden:

$$C = \arccos \left( \frac{BC^2 + CD^2 - BD^2}{2 \cdot BC \cdot CD} \right) \quad (\text{cosinusregel})$$

Berekenen van hoek B met de sinusregel:

$$B_1 = \arcsin\left(\frac{CD \cdot \sin(C)}{BD}\right) \quad (\text{sinusregel})$$

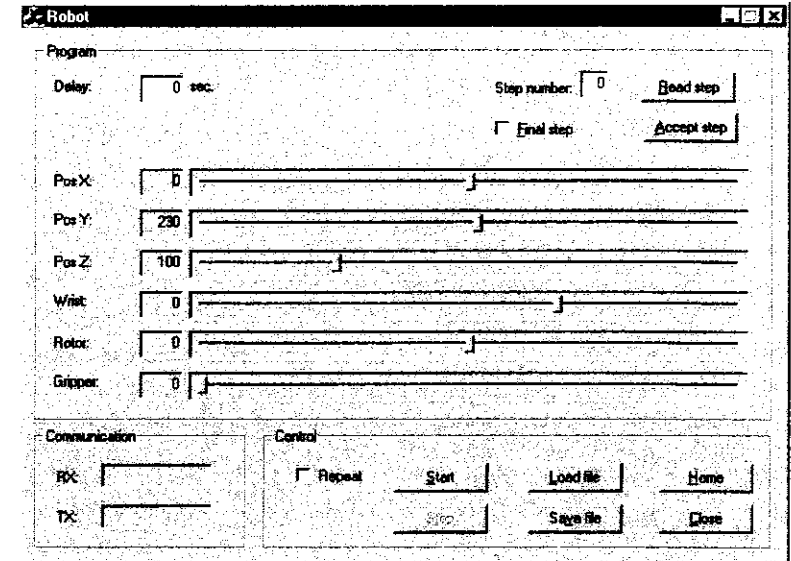
$$B_2 = \arcsin\left(\frac{Z-OB \cdot Z-DE \cdot Z}{BD}\right)$$

$$B = B_1 + B_2$$

### pc-software

De pc-software is eenvoudig te bedienen (zie afbeelding 5). De 3D-coördinaten van het uiteinde van de gripper kunnen direct via een schuifbalk ingegeven worden. Ook het ingeven van de hoeken van de pols en de gripper-rotor gaat op die manier. Wil men een bepaald bewegingspatroon opgeven, dan zijn hiervoor 100 stappen beschikbaar. Bij het invoervak *Delay* kan voor iedere stap opgegeven worden na hoeveel seconden de stap uitgevoerd moet worden. Als de juiste positie ingevoerd is (de robot beweegt mee tijdens het invoeren van een stap), dan kan de stap geaccepteerd worden door op de knop *Accept* te klikken. Bij de laatste stap moet het selectieblok *Final step* actief zijn.

Het stappenprogramma kan nu uitgevoerd worden door op *Start* te drukken. Als *Repeat* actief is, dan begint het programma steeds weer opnieuw. Het programma kan gestopt worden door op *Stop* te klikken. *Load file* en *Save file* zijn toegevoegd om gebruik te kunnen maken van een eerder ingevoerd stappenprogramma. *Home* zorgt ervoor dat de robot opnieuw de referentieposities van alle assen opzoekt. Dit is alleen nodig als de robot tegengehouden (*gecrasht*) is en de stappenmotoren daardoor stappen overgeslagen hebben.



afbeelding 5: pc-software

In de hoek *Communication* zijn nog twee velden *RX* en *TX* zichtbaar. Hier worden de door de pc verzonden en ontvangen instructies afgebeeld. Dit is gedaan om te kunnen zien of de communicatie nog actief is.

John van Maanen





# ibutton



## iButton lezen met het B+-bordje

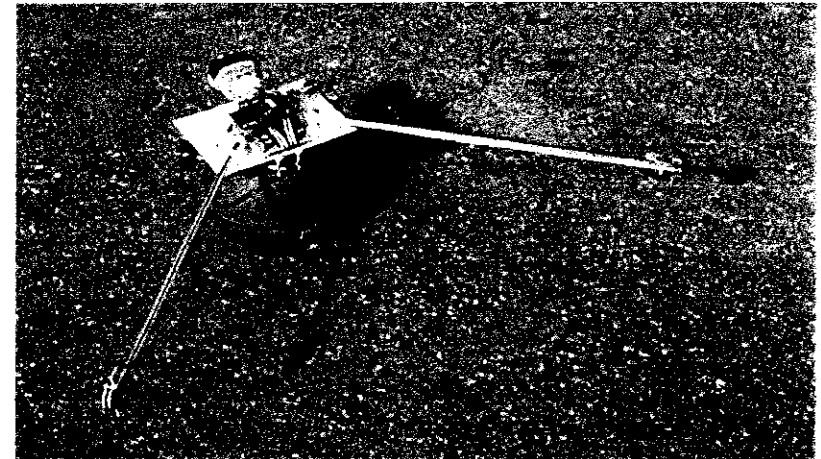
Soms wil je met behulp van het B+-microcontroller bordje een *identificatiesysteem* maken. Bepaalde mensen hebben wel toegang tot bepaalde zaken en anderen niet. Of je hebt toegang tot een bepaald niveau en anderen tot een hoger niveau. Kortom je wilt personen kunnen identificeren. Dit identificeren kan op verschillende manieren. De allermooiste is met behulp van persoonsgebonden (lichaams-) eigenschappen, zoals de iris, stem of vingerafdruk. Deze methoden zijn wat aan de kostbare kant voor de hobbyist. Ander systemen werken met een wachtwoord, maar dat kun je vergeten. Weer andere methoden zijn met behulp van een code die 'ingebakken' in een sleutel zit. Deze systemen kunnen heel eenvoudig en goedkoop zijn.

De website van *Dallas* voor meer info:  
<http://www.ibutton.com>  
Hier zijn alle soorten iButtons te vinden samen met de datasheets. Je hebt ze niet nodig, maar het is soms best interessant om meer van de achtergronden te weten.

Een van die mogelijke systemen kun je maken met behulp van de telefoonkaart lezer die in NewBrain on-line 26 (oktober 1999) is gepubliceerd. In elke telefoonkaart zit een uniek num-

mer verwerkt. Deze code (getal) kun je als unieke persoonlijke identificatiecode gebruiken.

Een andere wijze is de *iButton*. *Dallas semiconductors* heeft hiervoor een hele reeks componenten uitgebracht die over verschillende eigenschappen beschikken. Een iButton is een IC in de behuizing van een knoopcelbatterij van 17 mm doorsnede en 5 mm dik. De communicatie met de iButton gebeurt via het *1wire protocol*. Dit houdt in, dat er data en voeding over 1 ader naar de iButton gaat. Er bestaan verschillende iButtons, in sommige kun je data wegschrijven en uit anderen kun je alleen maar een unieke



code uitlezen. Een voorbeeld hiervan is de DS1990A-F5. In deze iButton zit alle elektronica om een 64 bit getal hieruit uit te kunnen lezen. En in deze code zijn we nu net geïnteresseerd.

De volgende vraag is hoe lezen we dit signaal cq. protocol uit. Hier zijn twee mogelijkheden voor. Ten eerste kunnen we aan de hand van de datasheets het protocol zitten uitpluizen en heel moeilijk deze gegevens omzetten in een programma. Maar er is een tweede oplossing. Nu, hier heb-

ben we de o zo handige *BascomLT* basiccompiler voor. Wat blijkt, in de standaarduitrusting van de commando's zitten een paar 1wire buscommando's. En hiermee is het een fluitje van een cent om een 1wire component uit te lezen.

De drie commando's die voor ons van belang zijn zijn de volgende:

```
1wreset      1wwriteY      X = 1wread()
```

De betekenis is als volgt: *1wreset* reset de 1wirebus en vraagt of er een component, de DS1990A in ons geval, aanwezig is. Zo ja dan krijgen we het volgende commando *1wwrite Y*. Hier is Y het commandonummer dat we aan het device willen geven, &H33 in ons geval, wat betekent dat we het ROM willen uitlezen. Nu kunnen we met behulp van het commando *X=1wread()* het eerste byte in de variabele X zetten. Door dit laatste commando 8 keer uit te voeren krijgen we de 8 bytes, zijnde de 64 bits, die in het ROM staan.

Dat is alles. Was alles maar zo makkelijk.

In het onderstaande programma hebben we een iButtonconnector op pin 1.0 aangesloten, en met een weerstand van 4,7 kOhm aan de plus 5 volt gelegd. Daarnaast is er een LCD-scherm aangesloten. De commentaren bij het programma moeten voor zich spreken. Als je een iButton tegen de houder houdt, dan wordt deze gedetecteerd en wordt de code uitgelezen en op het display getoond.

Een opmerking bij het programma is dat je in plaats van het wacht- en controle lusje van *Start*: ook van een interrupt gebruik kunt maken. Dit is voor de eenvoud niet gedaan. Lees ook de help file in *BascomLT* goed door: hier staan hele leuke toepassingen, oplossingen, ideeën en tips in.

Op de foto's is de DS2244T als test bord gebruikt in plaats van het B+-bordje; het programma draait op beide bordjes, dus naast deze twee genoemde ook op de kleine 89C2051 van Atmel.

Een laatste opmerking met betrekking tot het LCD-scherm: een scherm met 1 regel met 16 karakters wordt gezien door het programma (en het LCD-scherm zelf) als een scherm met 2 regels van 8 karakters die achter elkaar staan. Vandaar dat er eerst 8 karakters op het scherm gezet worden, dan wordt het commando *lowerline* gegeven, ten teken dat we naar de tweede regel gaan voor de resterende 8 karakters. Dit was in het begin voor mij heel verwarrend, totdat ik begreep hoe het werkt. Nu goed dus.

Ik wens jullie veel plezier met deze toepassing van het B+-bordje, *BascomLT* en de iButton.

Abraham Vreugdenhil

---

iButton.bas

6-1-2001

Een programma om de iButton van Dallas uit te lezen met behulp van het B+ bordje. De iButton wordt op p1.0 aangesloten. De DS1990A-F5 wordt gebruikt, er zitten een 8-bits familiecode, een 48-bits unieke code en een 8-bits CRC in. En dit gaan we uitlezen met behulp van *basic*. Binnen 6 milliseconde.

```
$romstart = &H8100      ' startadres B+
```

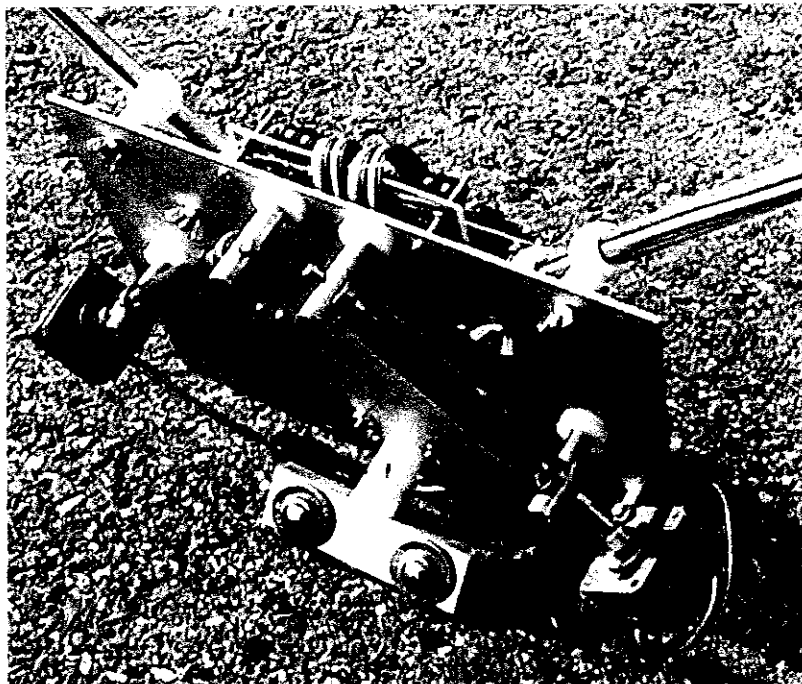
```
Config Lcd = 16 * 1     ' LCD definitie
```

```
Config 1wire = P1.0     'gebruik deze pin
```

```
Dim Ar(8) As Byte
Dim A As Byte , I As Byte
```

```
Lcd "iButton "
Lowerline
Lcd "lezer AV"           ' tekst op LCD
Wait 1
```

```
Start:
If P1.0 = 1 Then        ' wacht op contact
    Goto Start          ' maken iButton
End If
Err = 0
```



```
Lezen:
1wreset                 ' reset de 1wirebus
Print Err
1wwrite &H33            ' ROM-leescommando
For I = 1 To 8
    A = 1wread()        ' lees byte
    Ar(i) = A           ' zet in array
Next
If Ar(1) <> 1 Then      ' test fam. = Code 01
    Goto Lezen          ' = Oke
End If
If Ar(8) = &HFF Then   ' test checksum <> FF
    Goto Start         ' FF is sowieso fout
End If
```

```
For I = 8 Downto 1
    A = Ar(i) : PrintheX A; ' print in hex-formaat
Next
```

```
Print
Cls
For I = 8 Downto 5
    Lcdhex Ar(i)         ' halve regel op LCD
Next I
Lowerline
For I = 4 Downto 1
    Lcdhex Ar(i)         ' tweede helft op LCD
Next I
```

```
Goto Start
```



# inhoud on-line 29



- 3 nieuw bij newbrain /w g jansen/
- 4 b+al.ans /abraham vreugdenhil/
- 6 gp2d02: de sharp gp2d02 afstandmeetsensor uitlezen met behulp van het b+-bordje /abraham vreugdenhil/
- 12 lego-robots met menselijke trekjes  
robaticus legorius en mega-teek /maarten verschueren/
- 20 3d-robot /john van maanen/
- 30 ibutton lezen met het b+-bordje /abraham vreugdenhil/