

ROBO-

PTT Post

Port betaald
Port payé
Pays-Bas

BITS-25-

Jaargang 7, nummer 2, Juni 2004



Robosapien



Robotica gebruikersgroep

Afz. HCC Robotica, p.a. A. Vreugdenhil, Regulierenstraat 11, 2694 BA 's-Gravenzande

Inhoud

Field Robot Event 2004	p. 4
PICBasic -2-	p. 5
Co-Evolutie	p. 10
Reparatie zwenkwiel Cybot	p. 15
SHARP GP2D02	p. 16
De robot van PEP	p. 20
TNO-Rover van de Robotica-GG	p. 24
Internet	p. 28
Agenda	p. 28

Colofon

ROBOBITS is een uitgave van de Robotica-GG, en wordt naar alle leden van de gebruikersgroep opgestuurd. De oplage is 600 exemplaren. De Robotica-GG is een onderdeel van de Hobby Computer Club.

Redactie adres: A. Vreugdenhil,
Regulierenstraat 11,
2694 BA 's-Gravenzande.

E-mail: a.vreugdenhil@hccnet.nl.

Tekst aanleveren in Word of platte tekst in ascii en afbeeldingen er "los" bij in TIF of JPG formaat.



Re(d)actie

We mogen dit nummer van Robobit wel gevarieerd noemen. Van zeer theoretische tot zeer praktische artikelen. Van programmeer omgeving tot hoe werkt een sensor, van de opzet van een eenvoudige robot tot een geavanceerde robot die HET duel met de giganten der TNO aan moet gaan. Maar, bijna al deze artikelen zijn van leden die onze clubbijeenkomsten bezoeken en daar actief mee doen en denken. Dat maakt onze robotica-gg zo bijzonder vind ik. Veel leesplezier.

Abraham Vreugdenhil

Bestuur

Voorzitter

B.T.J.A. Buiskool
Pilotenlaan 11
7943 CH MEPPPEL
0522-241444

Technisch adviseur

Ing.H.M.A. van Bodegom
Stadionlaan 180
7552 VE HENGELLO OV
074-2434147
ing.h.m.a.van.bodegom@hccnet.nl

Secretaris

A.J. Janssen
Galjoenstraat 65
3534 PD UTRECHT
030-2444944
lex.janssen@hccnet.nl

Lid

P. Smits
Lijtweg 302
2341 HB OEGSTGEEST
071-5156090
psmits.1@hccnet.nl

Penningmeester

A. Vreugdenhil
Regulierenstraat 11
0174-420361
2694 BA S'GRAVENZANDE
a.vreugdenhil@hccnet.nl



Juni 2004 3

Field Robot Event 2004



Field Robot Event 2004: 17th and 18th of June 2004!

Wageningen.- Voordat de Crobhunter het maisveld ingaat, controleert de maker (een student van de Universiteit van Amsterdam) de robot voor de laatste keer. De Wageningse universiteit zet robots in voor de zogeheten precisielandbouw. Bij wijze van proef sturen de onderzoekers twintig robots van Europese en Amerikaanse universiteiten een maisveld op. De robots moeten planten apart verzorgen en de opbrengst opkrikken. Daarnaast kunnen ze een belangrijke functie vervullen in de biologische landbouw, doordat ze het gebruik van bestrijdingsmiddelen terugdringen. De deelnemers hebben de robot zelf ontworpen en opgebouwd.

Foto ANP

<http://www.fieldrobot.nl/html/main.html>

PICBasic -2-

PIC microprocessoren programmeren met behulp van PICBasic.

Deel 2:

In nr -24- van Robobit's heb ik beschreven hoe ik tot de keuze van PICmicro en de PICbasic Pro compiler ben gekomen. Daarnaast heb ik het over een prima editor gehad, die veel hulpmiddelen biedt bij het maken van je programma, en het daarna compileren daarvan. Aan het eind van al deze arbeid zitten we met een xxx.hex file die vervolgens nog in je PIC micro "gepompt" moet worden! Daarvoor hebben we dus een programmer nodig en software om die programmer aan te sturen. Als je dan even op het internet rondsnuift dan bemerk je al gauw dat daar een enorme variëteit in is. HVP of LVP, seriële programmers of parallel programmers, in circuit of gewone programmers met ZIF voet, al deze kretten komen voorbij! Het is daarbij wel belangrijk om te kijken of met de programmer de PICmicro's geprogrammeerd kunnen worden die je wilt gaan gebruiken of wellicht in de toekomst wilt gaan gebruiken. Ook is het belangrijk of het platform van waaruit je werkt door de programmer ondersteund wordt (Win XP of Linux bijv.). Dan is er nog een groot verschil in prijzen. Ook kun je zelf wat gaan bouwen, er zijn diverse schema's te vinden voor de bouw van een PIC programmer. Ook de aanstuursoftware daarvan is op Internet te vinden. Ons clublid Kristof Stevens werkt met zo'n zelfbouw programmer namelijk de Trivial HVP programmer in combinatie met de FPP software van David Tait. Dit kost slechts enkele euro's aan materiaal en de software is freeware. Ik heb echter gekozen voor de Wisp 628 die door Wouter van Ooijen is ontwikkeld. De reden dat ik voor deze programmer heb gekozen is divers. Hier een opsomming van mijn beweegredenen:

- * Goedkoop, print plus componenten kost ongeveer 20 euro, opgebouwde en geteste print ongeveer 30 euro.
- * In circuit programmer (PICmicro programmeren op zijn plaats in de schakeling, dus zonder hem te demonteren)
- * Ontwerp is eenvoudig en firmware is gratis en upgradebaar. (voor ondersteuning toekomstige "nieuwkomers".)

PICBasic -2-

- * Wouter woont enkele kilometers van mijn huis, dus leverancier vlak bij huis!
- * Als iets niet lukt, of er zijn anderszins vragen, is hulp in moerstaal vlak bij huis(mij is goede aftersales gebleken!)

Na installatie van de benodigde software is de programmer gereed voor gebruik. Op de website van Wouter kun je een test hex file halen voor de PICmicro die je hebt. Als je dan een LED aansluit volgens het schema wat daarbij gegeven wordt, dan zal na upload van de hex file in de PICmicro deze LED gaan knipperen. Slechts het dubbelklikken op de xxx.hex file is voldoende de toepassing te starten. Vervolgens bevestig je door het drukken op een knop dat je deze xxx.hex file in de PICmicro wilt laden en de rest gaat vanzelf. De cirkel is nu dus rond, we kunnen een programma schrijven, compileren en uploaden naar de PICmicro. Het is nu tijd om de PICBasic compiler eens nader te gaan bekijken! Zoals al eerder gezegd is maakt PICBasic grotendeels gebruik van dezelfde instructieset als de Basic Stamp2 (en ATOM). Er zijn echter ook verschillen! Voor de exacte verschillen verwijst ik je naar de manual die van Internet te downloaden is. Het is bijvoorbeeld mogelijk een INCLUDE file mee te laden in je programma waardoor je dezelfde pin benamingen kan aanhouden als bij de Basic Stamp. Dit wordt echter door een ieder afgeraden. Het is ook niet nodig als we ons een klein beetje verdiepen in de interne architectuur van onze PICmicro. Ik gebruik momenteel de 16F877A. Dit is een 40 pins Flash PICmicro met 8Kb FLASH code, 256 bytes data EEprom, 368 bytes RAM, 33 I/O pinnen waarvan 8 analoog (10 bit A/D) en een eigen UART en dan nog 2 x PWM en I2C enz. Deze processor heeft 4 poorten van elk 8 bits en 1 poort van 3 bits. Elke poort heeft 2 registers, een TRIS (TRI STATE) en een ADDRESS. Eenmaal de poort geconfigureerd met het TRIS register (welke bit is input en welke output), kan er rechtstreeks gelezen of geschreven worden naar het poort adres. Hebben we zoals het geval is bij de 16F877A vijf poorten dan hebben we dus een TRISA, TRISB tot en met TRISE register. Een voorbeeld van een programma voor een LED wat moet gaan knipperen op portB.0 (pin 33 van het IC):

PICBasic -2-

DEFINE OSC 20 ‘standaard is 4Mhz Kristal, ik gebruik 20Mhz

TRISB=%11111110 ‘poort B configuratie B0 is output de rest input
 ‘% teken staat voor binaire code

MAIN:

 HIGH PORTB.0 ‘zet poort B.0 hoog dus LED aan
 PAUSE 1000 ‘wacht 1000 milliseconde(1 seconde)
 LOW PORTB.) ‘zet poort B.0 laag dus LED uit
 PAUSE 1000 ‘wacht 1000 milliseconde(1 seconde)

GOTO main

END

Natuurlijk kun je deze poort adressering ook nog in een variabele verpakken, de code wordt dan als volgt :

DEFINE OSC 20 ‘standaard is 4Mhz Kristal, ik gebruik 20Mhz

TRISB=%11111110 ‘poort B configuratie B0 is output de rest input
 ‘% teken staat voor binaire code

LED VAR PORTB.0

MAIN:

 HIGH LED ‘zet poort B.0 hoog dus LED aan
 PAUSE 1000 ‘wacht 1000 milliseconde(1 seconde)
 LOW LED ‘zet poort B.0 laag dus LED uit
 PAUSE 1000 ‘wacht 1000 milliseconde(1 seconde)

GOTO main

END

Het overkwam mij dat ik bovenstaand programma compileerde en met behulp van

Juni 2004 7

PICBasic -2-

de WISP628 programmer in de PICmicro “pompte”, en dat mijn LED mooi niet ging flinkeren terwijl dit wel lukte met de bij de WISP628 meegeleverde test hexfile! Vele mailtjes later bleek dat dit uiteindelijk kwam door een verkeerde configuratie setting! Bij sommige programmers start er eerst een beginscherf op waarin je deze settings kunt doen. Bij de WISP628 echter niet, hierbij moeten ze voor de compilatie meegegeven worden in het programma. Het betreft hier ondermeer de settings voor de klok frequentie, de Watchdog en read beveiliging. Deze moet je bij PICBasic eerst aanpassen in de rechtstreeks in de programma directory(C:\pb) staande gebruikte pic.inc file te veranderen!

Voorbeeld: 16F877A.INC

```
NOLIST
ifdef PM_USED
    LIST
    include 'M16F87xA.INC' ; PM header
    device pic16F877A, xt_osc, wdt_on, pwrt_on, lvp_off, protect_off
    XALL
    NOLIST
else
    LIST
    LIST p = 16F877A, r = dec, w = -302
    INCLUDE "P16F877A.INC" ; MPASM Header
    __config __XT_OSC & __WDT_ON & __PWRTE_ON & __LVP_OFF &
                                     __CP_OFF
    NOLIST
endif
LIST
```

De kristal instelling van 4Mhz (XT_OSC) veranderen in 20 MHz (HS_OSC)! Speciaal voor PICBasic zijn er verschillende boeken geschreven, zoals bijvoorbeeld Programming PIC Microcontrolers With PICBasic van Chuck Hellebuyck. Maar ook de boeken Amphibionics en Insectronics van Karl Williams zijn zeer de

PICBasic -2-

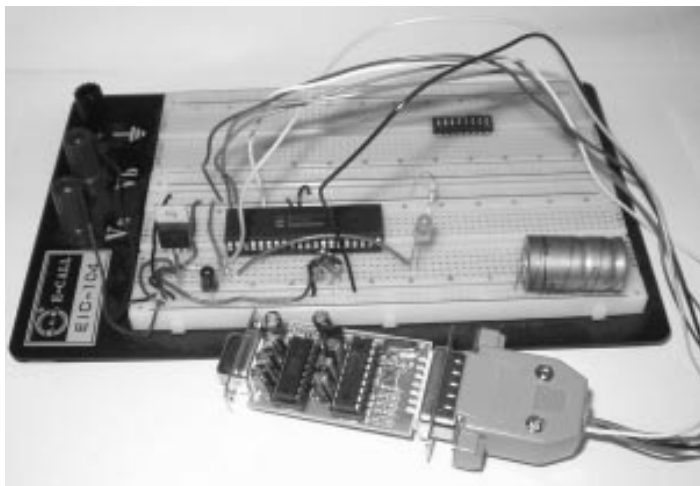
moeite waard! Maar ook alle sourcecode die op Internet te vinden is voor de Basic Stamp en ATOM is met kleine aanpassingen geschikt te maken voor deze compiler.

Weblinks:

Van alle in het artikel genoemde software zijn free of demo versies te downloaden.

Ook kun je met name veel leren op de website van Reynold Electronics, met bijzondere aandacht voor alles rondom de Micro-

bot waar PICBasic source code en beschrijvingen van te downloaden zijn.



<http://www.voti.nl/>

<http://www.thinkbotics.com>

<http://www.elproducts.com>

<http://www.melabs.com/>

<http://www.rentron.com/>

<http://www.mecanique.co.uk/>

<http://home.hccnet.nl/hj.de.gans>

Website Wouter van Ooijen.

Website Karl Williams.

Website Chuck Hellebuyck.

Maker PICBasic.

Reynolds Electronics.

Maker MicroCode.

Mijn website met meer informatie.

Co-Evolutie

Een verslag voor het vak Autonome Systemen

Herman Eldering Herman@Eldering.net

4th May 2004

Abstract

Kennis van evolutie is voor meerdere disciplines van belang. Aan de hand van drie artikelen vormt dit verslag een introductie voor co-evolutie. Zowel kunstmatige als natuurlijke co-evolutie wordt besproken. Tevens worden de beperkingen van genetische algoritmes besproken in vergelijking met algemenere evolutionaire algoritmes.

1 Introductie

Co-evolutie is een evolutionair systeem waarin individuen elkaar beïnvloeden 5 . Meer specifiek, waar ze elkaars fitness functie beïnvloeden. De individuen die elkaars fitness functie direct (positief danwel negatief) beïnvloeden worden symbioten genoemd 6 . Door Morrison is onderstaand raamwerk voorgesteld 4 , dat in dit artikel gebruikt zal worden.

Vormen van symbiose

- Amensalisme(antibiose) - De interactie is in het nadeel van een populatie(amensaal), maar heeft geen invloed op de andere populatie.
- Commensalisme - Een populatie is gebaat bij de interactie(commensaal), de andere wordt er niet door beïnvloed.
- Competitie/Concurrentie -Beide populaties ondervinden nadeel van de interactie.
- Mutualisme/Coöperatie - Beide populaties hebben voordeel van de interactie. -
- Predatie - Een populatie(predators/roofdieren) heeft voordeel, de andere(prooi) nadeel van de interactie.
- Adaptisme - Individuen bevoor- of benadelen andere individuen van dezelfde populatie.

Daarnaast komt parasitisme voor in de literatuur, maar het gebruik hiervan is niet eenduidig. In de meeste gevallen wordt predatie bedoeld, maar er zijn ook gevallen waarbij het gebruikt wordt als er sprake is van commensalisme. Ook wordt er

Co-Evolutie

in de literatuur soms onderscheid gemaakt tussen populaties die van elkaar afhankelijk zijn om te overleven. Bij mutualisme, predatie en parasitisme is er dan sprake van afhankelijkheid. Bij cooperatie is er geen afhankelijkheid. Dit aspect wordt in de rest van het artikel buiten beschouwing gelaten.

Evolutionaire algoritmes zijn een belangrijk onderdeel van de kunstmatige intelligentie omdat het een volledig automatisch proces is. Dit heeft als voordeel dat het ontwikkelen minder tijd vraagt van de onderzoeker en dat de kans kleiner is dat verkeerde aannames van de onderzoeker het ontwikkelproces verstoren.

Als co-evolutie het ontwikkelproces kan versnellen en verbeteren kan dat van groot belang zijn voor het ontwikkelen van kunstmatige intelligentie. Maar het is daarnaast ook van groot belang voor andere onderzoeksgebieden zoals genetische manipulatie en het ontwikkelen van geneesmiddelen (bijvoorbeeld in verband met de toename in multi resistente virussen). Er zijn twee kwaliteits aspecten bij kunstmatige evolutie. Het eerste aspect is de efficiëntie, door sneller goede oplossingen te vinden zou de benodigde rekenkracht verminderd kunnen worden. Ten tweede is het van belang welke oplossingen door de evolutie gevonden kunnen worden.

De problemen bij kunstmatige evolutie

Kunstmatige evolutie kent een aantal problemen die bij vrijwel elke toepassing weer naar voren komen. Zo werkt een evolutionair algoritme niet goed als er niet sprake is van een helling in de fitness functie. Dit is het geval bij het bootstrap probleem 1, als bij de start van de evolutie alle individuen een fitness hebben van 0 en nagenoeg alle afstammelingen ook de fitness 0 hebben komt de evolutie moeilijk op gang. Het is een speciaal geval van het probleem dat er een vlak gedeelte (een plateau) in de fitness functie is met daarin slechts een klein gedeelte dat hoger is. Dit probleem is bekend als het naald op een plateau probleem 2.

De traditionele aanpak om het bovenstaande probleem op te lossen is het aanpassen van de expliciete fitness functie. Dit heeft echter als nadeel dat er meer werk door mensen verricht moet worden, en er is het risico dat de evolutie de verkeerde kant op gestuurd wordt 1. Een ander probleem bij kunstmatige evolutie is het gebrek aan diversiteit. Hierbij zijn alle individuen min of meer hetzelfde geëvolueerd. Het kan dan gebeuren dat de evolutie blijft steken in een lokaal minimum (of een lokaal maximum bij een fitness functies).

Co-Evolutie

2 Predatie in de computer

De paper van Nol_ en Floreano (1) beschrijft een aantal experimenten die ze hebben uitgevoerd. De experimenten hebben tot doel de rol van co-evolutie bij een jager en prooi robot vast te stellen. De centrale vraag die hierbij gesteld werd is of er een wapenwedloop ontstaat in de experimenten. Bij een dergelijke wapenwedloop zouden de individuen steeds geavanceerder moeten worden om tegen elkaar op te kunnen. De wedloop bestaat uit 2 fasen die herhaald worden. Eerst wordt er een defensieve eigenschap gevormd bij de prooi waardoor de jager minder effectief wordt. In de tweede stap zal de evolutie er voor zorgen dat de jager op zijn beurt een eigenschap krijgt die het afweermechanisme van de prooi neutraliseert (3). Bij het onderzoek bleek dat de robots in een cyclus kwamen waarin steeds dezelfde resultaten elkaar bleven afwisselen. Hierdoor werd er geen generieke strategie ontdekt en uitgewerkt, maar wisselden de strategieën elkaar af. Om wel tot een verfijnde strategie te komen moet dit fenomeen, dat limiet cyclus wordt genoemd, vermeden worden.

Het artikel bespreekt een aantal mogelijke manieren om de kans op een limiet cyclus te verkleinen. Zo zou er mogelijk in een complexere omgeving minder snel een limiet cyclus optreden. Deze stelling wordt bevestigd door een onderzoek dat uitgevoerd is (1). Een andere mogelijke oplossing is het introduceren van plasticiteit van een individu tijdens zijn levensduur. Er wordt onder andere gesteld dat als het niet mogelijk is om een generieke strategie te vinden er een grote druk zal zijn om adaptief gedrag te ontwikkelen die afhankelijk van de situatie een keuze maakt voor een bepaalde strategie.

Uiteraard is er ook gekeken of het mogelijk is het probleem te verhelpen door de expliciete fitness functie te veranderen. Maar dit hangt samen met een tweede probleem, en dat is de vraag hoe de fitness van een individu bepaald moet worden. Want de fitness is namelijk afhankelijk van de tegenstander. Zo is het bijvoorbeeld mogelijk dat een predator op een bepaald moment een hoge fitness heeft, maar dat dezelfde predator vervolgens een lagere fitness krijgt doordat de prooi evolueert. Dit fenomeen, dat de individuen moeten verbeteren om dezelfde fitness te behouden, wordt het Red Queen effect genoemd (1) (naar het gelijknamige personage uit "Through the Looking Glass" (7)).

De oplossing wordt gezocht in het behouden van oudere individuen en die te ge-

Co-Evolutie

bruiken bij het bepalen van de fitness, in het artikel de Hall of Fame methode genoemd. Bij de gebruikte meet techniek, die door Nol_ en Flore-ano de naam Master Tournament is gegeven, worden alle beste individuen van alle generaties getest tegen elke beste tegenstander van alle generaties (1).

3 Predatie in de natuur

Een voorbeeld van parasitisme dat ingedeeld wordt bij predatie is ongedierte bij planten. In Nature is hierover een artikel verschenen van Rausher dat in gaat op de co-evolutie tussen de afweer van planten en de natuurlijke vijanden van de plant (3). In het artikel komt duidelijk naar voren dat een wapenwedloop in de natuur frequent voorkomt.

Sinds de perfectionering van genetische transformatie is het mogelijk om genen die verantwoordelijk zijn voor resistentie over te zetten tussen ongerelateerde planten soorten. De vooruitgang wordt echter tegengewerkt doordat de natuurlijke vijanden mee evolueren.

De High-Dose/Refuge(HDR) strategie - hoge dosis/toevluchtsoord strategie - behelst het ontwikkelen van een gewas dat een hoge dosis gif produceert en het planten van zowel resistente als vatbare varianten van het gewas. De vatbare varianten vormen hierbij een toevluchtsoord voor de natuurlijke vijanden van de plant die niet bestand zijn tegen het gif.

HDR is gebaseerd op een aantal basis principes. Om te beginnen is de snelheid waarmee het aantal voordelige allele's in een populatie toe neemt voor een groot deel afhankelijk van de (mate van) dominantie. Het hoge-dosis deel zorgt er voor dat de allele's die contra-resistentie overbrengen recessief zijn.

Het tweede principe waar HDR gebruik van maakt is dat de snelheid waarmee het aantal voordelige allele's proportioneel is aan het verschil in fitness tussen de genotypes. Dit principe wordt toegepast door het refuge deel, door een niet resistente variant van het gewas te planten hebben de vatbare individuen de mogelijkheid om zich voort te planten. Het verschil tussen contra-resistente en de vatbare individuen wordt daardoor verkleind. Daarnaast moet de plaatsing van deze schuilplaatsen er voor zorgen dat de contra-resistente individuen zich alleen voortplanten met individuen die de eigenschap niet hebben.

Een derde evolutionair principe, dat veranderingen kostbaar zijn, zou er voor kun-

Co-Evolutie

nen zorgen dat de HDR strategie de ontwikkeling van contra-resistentie oneindig lang uitstelt. Hierbij kan pleiotropie, het verschijnsel dat een gen meer dan een kenmerk beïnvloed, een rol spelen. Helaas kunnen er geen verdere concrete uitspraken over gedaan worden met betrekking tot de HDR strategie omdat niet bekend is hoe vaak de contra-resistentie kostbaar is.

De HDR strategie is echter een theoretisch model en is nog onvoldoende empirisch onderzocht. Het is gebaseerd op de aanname dat de toevluchtsoorden zodanig geplaatst kunnen worden dat panmixia, dat wil zeggen dat paringen willekeurig zijn, kan worden bereikt. Een andere aanname is dat de (opgroeïende) individuen niet op zowel giftige als niet giftige planten leven, omdat anders het gif verdund zou kunnen worden wat het hogedosis principe tegenwerkt.

Een tekortkoming van de besproken strategie doet zich voor bij insecten die zich asexueel voortplanten. Echter insecten die zich asexueel voortplanten zijn op hun beurt niet goed in staat om in te spelen op veranderingen van andere organismen. Een andere beperking doet zich voor bij een haploid organisme, deze organismen bezitten slechts een stel chromosomen per cel. Dit soort organismen kunnen alleen worden tegengewerkt met behulp van een toevluchtsoord, deze moet echter groter zijn dan dat economisch acceptabel is.

Naast een afweermechanisme dat een negatieve invloed op de predator heeft, wordt er nog een interessante oplossing aangedragen door het artikel. Er zijn wellicht mogelijkheden om de planten tolerant te maken voor bepaalde natuurlijke vijanden. Het voordeel hiervan is, in het kader van het ontwikkelen van gewassen, dat er juist geen wedloop ontstaat.

Ongeveer 90% van alle plantenetende insecten hebben een beperkt dieet, ze eten slechts van een bepaalde plantensoort (3). In het artikel worden twee mogelijke redenen aangedragen voor dit verschijnsel: (1) het verschil in de fitness van een individu op verschillende planten zorgt er voor dat de evolutie individuen die alleen van de beste planten eten prefereert en (2) het in stand houden van eigenschappen om afweermechanismen van verschillende planten te omzeilen is te kostbaar voor het ontstaan van generieke herbivoren. Voor de insecten die eenmaal een voorkeur voor een bepaalde voeding hebben ontwikkelt is het erg moeilijk om zich aan te passen op een ruimer dieet.

- Volgende Robobits het tweede deel van dit artikel -

Reparatie zwenkwiel Cybot

Ik had het CYBOT zwenkwieltje gebroken, dat heb ik gerepareerd met een hout-schroef met aan het andere einde schroefdraad. Dit is zo te koop, lengte ong. 30-35mm. In het afgebroken stuk had ik een gat geboord van 3,5 of 4 mm. De hout-schroef erin gedraaid en klaar is kees. M4 zit er aan de andere kant, een moer erop en klaar.

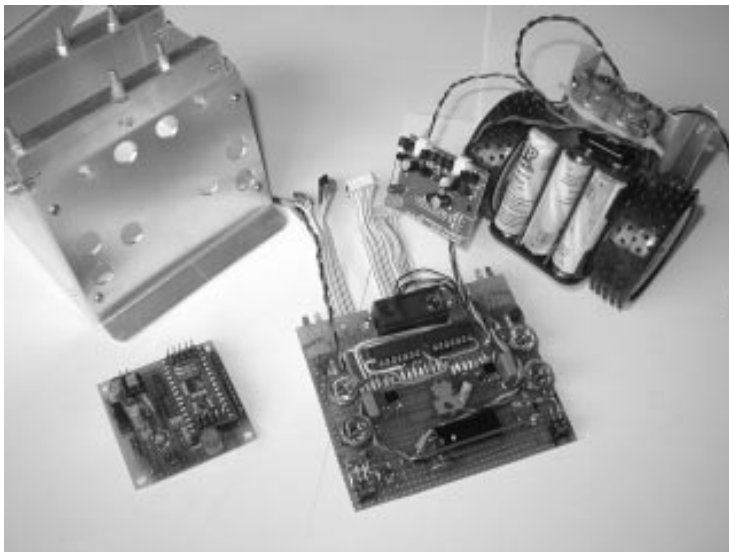
Richard.



SHARP GP2D02

Bij onze Belgische robotica vrienden zijn ze nogal actief bezig met Sumo robots tot 500 gram. Maar om wedstrijden te kunnen houden moeten er ook verliezers zijn, en de Olympische gedachte meedoen is belangrijker dan winnen, heeft mij er toe aangezet om ook maar zo'n ding te bouwen. De denkwijze is natuurlijk anders dan een traditionele robot. Normaal gesproken probeer je een obstakel te ontwijken, maar bij Sumo wedstrijden is het de bedoeling het obstakel op te zoeken om er dan flink tegenaan te rammen in de hoop daardoor de tegenstander uit het veld te krijgen. Om een en ander te kunnen realiseren was het noodzakelijk om mijn Tamiya Twingear aandrijfsetje te herstellen. Verder zijn er twee SNY70 infrarood sensoren die dienen als eindewereld schakelaars, en als nabijheidsschakelaars gebruik ik 2x de IS 471 F gemoduleerde IR-detectors in combinatie met het schema van Henny van Bodegom. Een van de grootste problemen bij Sumo's is om binnen het maximale gewicht te blijven, daarom was de Mega8 Mini Module een geschikte controller. Om de motoren te sturen heb ik even de Cybot-motorprint geleend, met al zijn voordelen zoals kant-en-klaar, klein en licht, en het schema is ook al getekend en het kost maar 7,90 euro. Om de „vijand" op grotere afstand op te sporen, had ik nog een Sharp GP2D02 infrarood afstand meetsensor liggen. In

een vorig project was die in combinatie met het aansturen van stappen motoren wegens te groot tijdsverlies niet bruikbaar. Voor de Sharp GP2D02 is er namelijk wat software nodig om de digitale meetwaarde te kunnen uitlezen. Als basis



SHARP GP2D02



gebruik ik het programma en het aansluitschema dat A b r a h a m Vreugdenhil heeft gemaakt voor zijn B+al.ans. Die was gemaakt met het bekende B+ bordje en geschreven in Bascom LT (8051). In die combinatie had ik het wel werkend, maar nu moest dat even wor-

den omgezet naar Bascom-AVR. Aangezien ik met de Atmel controllers nog niet zo veel ervaring heb was het toch even zoeken. Het blijkt dat het configureren en uitlezen van de poorten even op een iets andere wijze gebeurt dan bij de 8051. Omdat ik toch wat meer met de Atmel controllers wil gaan werken, had ik een geregistreerde Bascom-AVR versie op mijn (stok) oude portable PC geïnstalleerd, en dat werkte niet zo goed maar na een mailtje met Mark Alberts van MCS was dat al snel opgelost. Maar zolang het programma niet werkte bleef de twijfel natuurlijk bestaan. Eerst heb ik geprobeerd om te lokaliseren welke delen er werkten en waar de zaak op strandde. Na wat testen bleek dat het tweede deel van het programma, het in klokken van de bits, geen foutmelding gaf. Dus zat het probleem in het eerste gedeelte, het starten van de meting en de wachttijd van +/- 70 msec. Na wat zoeken in de uitgebreide helpfunctie van Bascom-AVR en het bekijken van een soortgelijk programma van een temperatuursmeting, heb ik de nodige aanpassingen kunnen aanbrengen. Met de wachttijd van 70 msec heb ik wat geëxperimenteerd, en het bleek dat in mijn opstelling 72 msec een betere uitlezing gaf bij de verste afstanden. Dit zal wel verschillen per controller en/of sensor. De listing van het meetgedeelte wordt hierbij afgedrukt, de rest van het programma

SHARP GP2D02

kan natuurlijk om strategische redenen niet worden vrij gegeven. Zaterdag 19 juni zal ik het mobieltje in het strijdperk gooien, dus als dit Robobitje ter perse gaat, zal ik mijn eerste nederlaag al te pakken hebben ?

Aloys Verstraeten.



```
' (c) 2004 Afstandmeetsensor SHARP GP2D02
' Dit is een copy van bascom 8051 herwerkt naar BASCOM AVR
' 09-06-2004 Aloys Verstraeten
' Toegepast op de MEGA8 Mini module
' (c) 2001 Afstandmeetsensor SHARP GP2D02
' file: GP2D02.BAS Sharp (Model abraham vreugdenhil)
'*****BASCOM AVR*****
```

```
$regfile = "m8def.dat"
```

```
'SHARP SENSOR GP2D02 (robobit)
```

```
'AANSLUITING:
```

```
'Pin 1 = GND ->
```

```
'Pin 2 = V-in=BC549 -> Col -> B -> 10K -> (PinD.3) Em = GND
```

```
'Pin 3 = Vcc +5 vDC
```

```
'Pin 4 = Vout -> (PinD.2) (GP2D02 12K internal pull up)
```

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

```
Config Portd = &B11111011 'Portd.2 as input
Reset Portd
```

18 **ROBOBITS**

SHARP GP2D02

Print "begin"

Begin:

Set Portd.3 'V-in P5.3 = 1
Waitms 10 '10 wacht even
Waitms 72 'Rekentijd

Inn = 0 ' = 0

For Dat = 1 To 7 'eerste 7 bits inklokken
Reset Portd.3 'P5.3 = 0
For Tt = 1 To 5 : Next Tt
Set Portd.3 'P5.3 = 1
For T = 1 To 5 : Next T ' < 200 uSec.
Buffer = Pind.2 : In1 = 0 'P5.2 'lees uitgang in

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

Next Dat

Reset Portd.3 'P5.3 = 0
For Tt = 1 To 5 : Next Tt
Set Portd.3 'P5.3 = 1
For T = 1 To 5 : Next T
Buffer = Pind.2 : In1 = 0 'P5.2

If Buffer = 1 Then
In1 = 1
End If

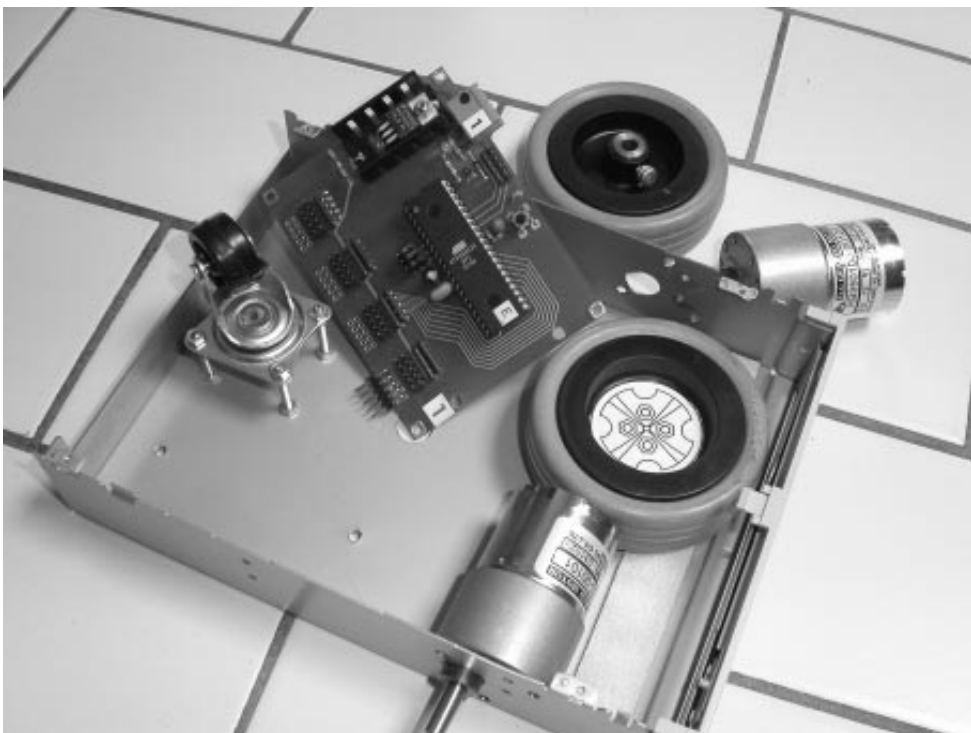
Inn = Inn + In1 'laatste optelling en dan niet meer doorschuiven

Waitms 1 : Reset Portd.3 : Waitms 2 : Print Inn : Waitms 2 : Goto Begin 'opnieuw

End

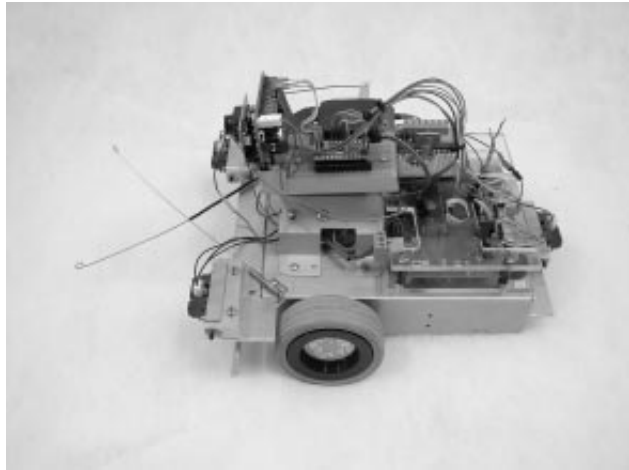
De robot van PEP

De robot van PEP. Een robotvoertuig kant en klaar kopen is gemakkelijk, maar er zelf van begin af aan een bouwen is veel leuker. Veel geduld moet je wel hebben en een beetje aansporing zo nu en dan heb je wel nodig. Probleem is bij bijna iedereen, hoe moet dat en hoe pak je zoiets aan. Daarom bezoekt Pep regelmatig de robotica club en ook diverse andere hobby afdelingen. De ideeën die Pep opdoet tijdens de bijeenkomsten, hebben hem doen besluiten om met een computer print te beginnen. Immers deze print wordt het hart van je robot, ook al weet je op dat moment niet hoe je robotvoertuig er uit zal zien of worden. Ook kocht Pep een print voor de motor sturing. Ik denk dat Pep onder de indruk was geweest van de veelzijdigheid van deze print, want je kan hem gebruiken voor het aansturen van verschillende typen motoren, normale motoren of stappen motoren. Alles kan worden verwezenlijkt door het kiezen van de juiste software. Er werd zelfs een



De robot van PEP

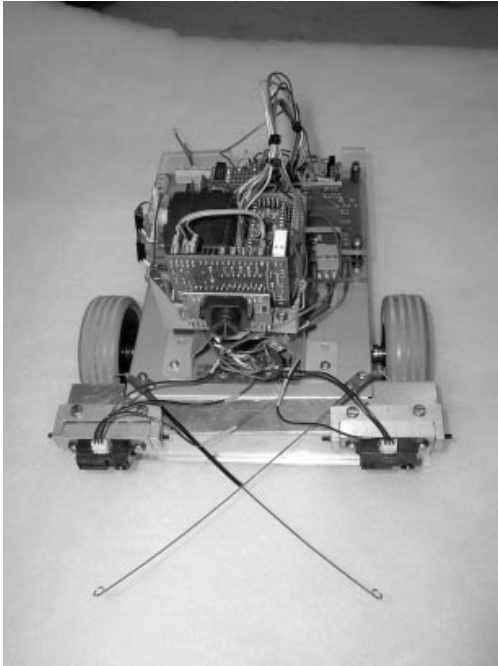
voorbeeld programma meegeleverd, in BASCOM geschreven. Ik denk dat BASCOM een heel toegankelijke taal is voor de begin programmeur, vooral als je een beetje engels kent. De computer opdrachten in zo een BASIC programma zijn herkenbaar. Vervolgens onderdelen voor de computer print gekocht ,



daarna solderen, de motorsturing erop aansluiten, de programma downloaden naar de chip, en dan heb je iets werkend. Motoren heb je niet direct nodig bij deze print want hij is voorzien van duo-leds. Door naar de leds te kijken zie je of de motoren stoppen, vooruit of achteruit draaien. Het is voor de beginnende robot liefhebber aan te raden een verkrijgbare print te gebruiken, liefst met alle onderdelen geleverd, om zo een robot computer te bouwen. Een goede print is overzichtelijk, de in- en uitgangen, de voeding en programmeer ingang hebben allemaal een eigen plaats en zijn voorzien van connectoren. Daardoor maak je minder fouten tijdens het experimenteren, wat je geld en ellende kan besparen. Hoe nu verder? In principe kan men er iets mee laten voortbewegen, maar verder ook niets. Er ontbreken nog sensoren, de oren en ogen naar de wereld toe, en behuizing met de nodige voortbewegingsmechanismen. Een paar jaren terug had Henny van Bodegom printen gemaakt voor verschillende typen Atmel AVR microprocessoren. Met Pep had ik toen besproken welk type AVR ik zou gaan gebruiken. Het werd de AT90S8535. Pep heeft er ook een aangeschaft.

Heel lange tijd werd het stil rond de AT90S8535 van Pep, totdat Aloys Verstraeten een overvolgig voertuig te koop aanbiedt. Dit aanbod komt als geroepen, want inmiddels had Pep zoveel elektronica verzameld, dat iets rijdbars gemaakt dient te worden. De basis van PEP's rijdende robot is een diskette drive behuizing. Het

De robot van PEP



woord is nu aan PEP die zelf uitvoerig op zijn robot ingaat.

Hallo, ik zal mezelf even voorstellen ik ben Jos Pepplinkhuizen en wordt meestal door iedereen PEP genoemd. Tijdens een bezoekje aan de HCC beurs ben ik geïnteresseerd geworden in de ROBOTICA en de mogelijkheden van deze bijzondere hobby. Op dit ogenblik ben ik bezig met mijn PEPBOT. PEPBOT is een rijdende robot en deze bestaat uit onderdelen van een cd kastje, een Microcontrollerbordje AVR AT 90S8535. Dat is een 40-pins AVR processor met 4 POORTEN, 8192 Flash, en 512 byte SRAM en 512 EEPROM. De twee motoren die ik

gebruikt zijn van CONRAD en zijn van het type RB 35, de 3 AFSTANDSENSOREN zijn van het type SHARP GP2D12. Tot slot nog een CMUCAM, dit is een camera die te verkrijgen is bij de www.roboter-teiler.de, deze CMU wordt op zijn beurt aangestuurd door een kleine kant-en-klare ATMEGA8 bordje, compleet met een RS 232 aansluiting en deze komt van de firma EMBEDIT.

De print voor het aan sturen van de motoren is gemaakt door HCC PC gg PE werkgroep en is het printje met een L298N. De functie van deze print is het aansturen van maximaal 4 lampen of 4 Relais of 2 gelijkstroom motoren of 1 bipolaire stappen motor en het inlezen van 4 ingangen. Als er geen stappenmotor wordt gebruikt is zelf een combinatie mogelijk. De toestand van de uitgangen wordt zichtbaar gemaakt door rode en groen led's op de uitgang, van belang bij reverse van de draairichting van een motor.

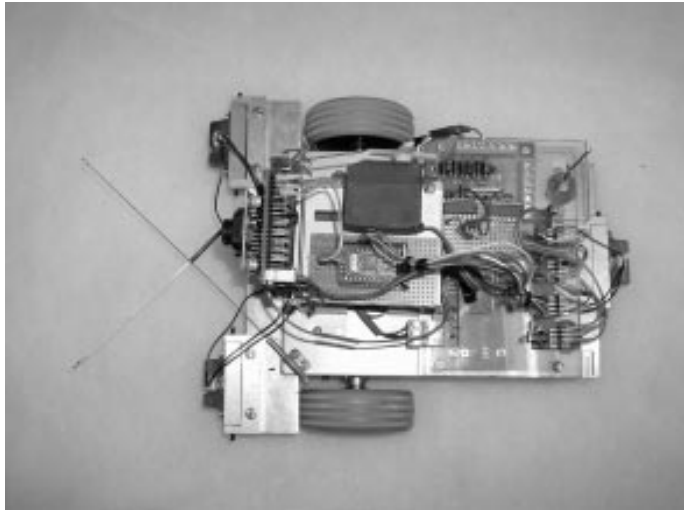
Sinds zaterdag 6-6-04 werkt mijn PEPBOT en kan ik naar hartelust met het ro-

De robot van PEP

botje experimenteren want daar leer ik het meeste van. Het programma is geschreven in C en is gemaakt door FRED EISEN die mij geweldig heeft geholpen met de PEPBOT. Ik heb helemaal geen programmeer ervaring en om met het programmeren meer inzicht te krijgen heb ik ook een experimenteerbordje gekocht en een boek. Hier wordt alles duidelijk en helder stap voor stap uitgelegd, en hiermee moet het lukken om dit onder de knie te krijgen. De software die ik gebruik is een Basic - Compiler voor de AVR- CONTROLE van de familie ATMEL.

Groeten PEP

Zoals jullie hebben gemerkt, werkt PEPBOT momenteel met een programma geschreven in C. De C-compiler (AVRGCC) waarmee het programma is vertaald is gratis te downloaden van internet. PEP heeft zelf te kennen gegeven dat hij verder wilt gaan met BASCOM.



Met het boek en het bordje ter ondersteuning zal het hem zeker lukken, zo niet zijn er altijd mensen in de club die hij kan raadplegen. Voorbeelden zijn er ook genoeg op de WEB, maar deze link:

<http://users.pandora.be/vandenheuvel/zwirbit/Zwirbitdesignnl.html>

van Koen van den Heuvel is zeer aan te bevelen.

Motto van het verhaal, samen komen we verder.

TNO-Rover van de Robotiva-GG

Zoals op onze laatste bijeenkomst aangegeven hebben wij, de HCC-robotica gg opgegeven als deelnemer aan de TNO robot competitie. Op dit moment bestaat het team uit Henny van Bodegom, Fred Eisen, Paul Wiegmans en Bert Buiskool. Het is de bedoeling dat wij met onze robot een aantal missies gaan uitvoeren.

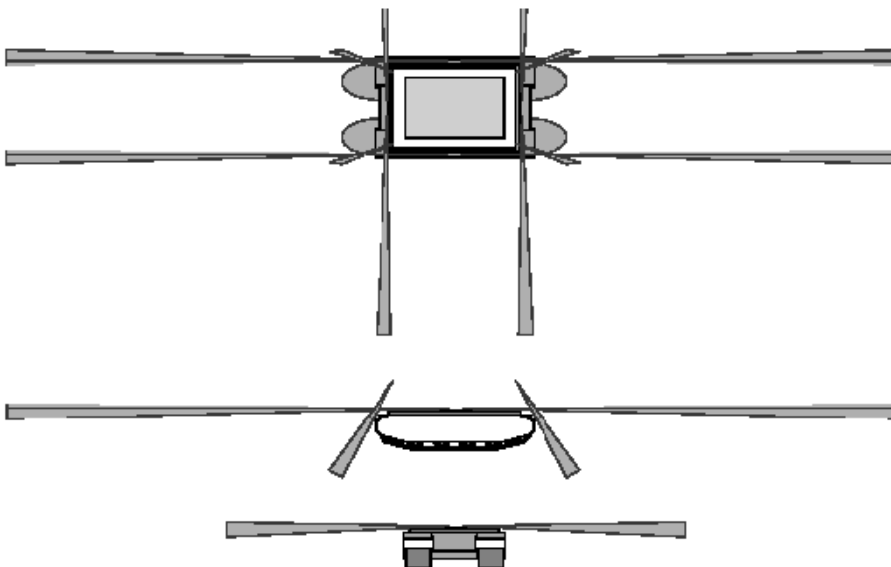
De missies bij de TNO competitie

Missie 1 heet "Doolhof".

De opdracht bestaat uit het vinden van de uitgang (er is slechts een uitgang) door het missieveld te doorkruisen onder het ontwijken van een aantal obstakels. De ondergrond is grastapijt, de obstakels zijn gasbetonblokken, de uitgang bevindt zich aan een korte zijde van het veld. De startpositie ligt aan de andere korte zijde. De gasbetonblokken hebben de afmetingen 60 x 40 x 10 cm. Deze blokken zijn standaardblokken zoals ze in de bouw gebruikt worden. De posities van de obstakels zijn onbekend. De missie is volbracht als de robot in zijn geheel het missieveld door de uitgang heeft verlaten.

Missie 2 heet "Schatzoeken".

De opdracht bestaat uit het vinden van een ongemarkeerde (schat)positie in het missieveld aan de hand van een van tevoren gegeven kaart. De robot moet deze



TNO-Rover van de Robotiva-GG

schatpositie bereiken door het missieveld te doorkruisen onder het ontwijken van een aantal obstakels. De ondergrond is grastapijt, de obstakels zijn gasbetonblokken. De posities van de obstakels en de schatpositie worden een maand voor de testronde bekend gemaakt. Het gaat dus om een bekend parcours. De robot wordt neergezet op een onbekende positie in het parcours. De missie is volbracht als de robot stilstaat op de schatpositie en een signaal geeft. De missie is mislukt als de robot een signaal geeft zonder op de schatpositie te staan.

Missie 3 heet "Robosport".

De opdracht bestaat uit het zoeken en het fysiek contact maken met een blauwe bal (een standaard voetbal maat). De ondergrond is grastapijt. Er zijn geen obstakels in dit missieveld, wel zijn er drie rode ballen aanwezig. De missie is volbracht als de blauwe voetbal wordt aangeraakt en de robot tegelijkertijd een signaal geeft. De missie is mislukt als de robot een signaal geeft zonder dat de blauwe voetbal wordt aangeraakt.

Missie 4 heet "Scoren".

Het opdrachtveld lijkt op de vorige missie. Alleen de positie van de blauwe bal is bekend en er zijn geen rode ballen. De robot moet de bal verplaatsen naar een opgegeven positie (doel) in het missieveld. De posities van de bal, de robot en het doel worden een maand voor de testronde bekend gemaakt. De missie is volbracht als de bal zich op de doelpositie in het missieveld bevindt.

Missie 5 heet "Afgrond".

De opdracht is om een verhoogd plateau op te rijden en daar tegen een eindpunt aan te rijden. Het plateau is van ongeverfd underlayment dat zich op circa 10 cm boven het asfalt bevindt. Het eindpunt is een obstakel van 20 cm hoogte over de volledige breedte van het einde van het traject en bestaat uit ongeverfd underlayment. De helling is van hetzelfde materiaal en heeft een helling van ongeveer 20 graden. Het parcours wordt een maand voor de testronde bekend gemaakt. Na het oprijden van het plateau dient de robot op het verhoogde traject te blijven (tot aan het eindpunt). Het plateau bevindt zich op een ondergrond van grastapijt. De missie is volbracht als tegen het obstakel aan het eindpunt wordt aangereden. De ondergrond bij alle missies is grastapijt. De kleuren kunnen door diverse leveranciers iets afwijken. In alle gevallen gaat het om glad tapijt, niet om

TNO-Rover van de Robotiva-GG

kunstgras. Het parcours wordt zodanig opgesteld dat een robot met afmetingen van maximaal 60 x 60 cm de missies kan uitvoeren.

Tijdspad:

april 2004

31 mei 2004

juni 2004

30 november 2004

december 2004

2 april 2005

4 juni 2005

bekendmaking nieuwe spelregels

uiterste datum inschrijving voor eerste selectie

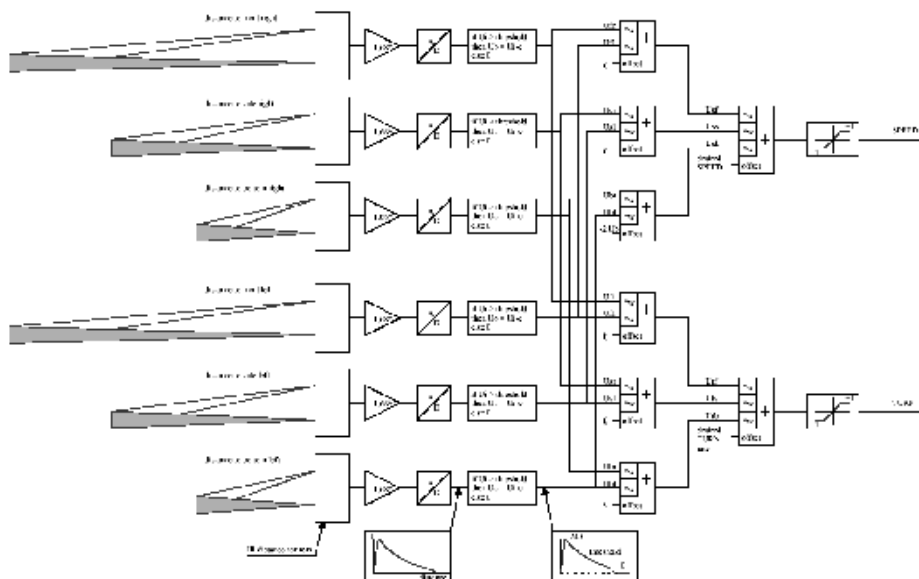
eerste selectie teams

uiterste datum inschrijving voor tweede selectie

tweede selectie teams

testronde in garage TNO-FEL. Uiterlijk twee weken na de test ronde informatie over ontwerp, bouw en financiële onderbouwing bij Jury.

wedstrijd te TNO-FEL uitwijken naar botshal bij regen.



TNO-Rover van de Robotiva-GG

De rover waarmee we eea willen gaan uitvoeren.

De HCC Robotica gg robot voor deelname aan de TNO robot competitie bestaat uit de volgende onderdelen:

- * Tank onderstel (hiervoor wordt een bestaande Robotica gg robot omgebouwd).
- * Stuurprint waarop een Power Monitoring and Control Unit zit, de processor core, de I2C bus om sensoren uit te lezen en de motor sturing.
- * Vier Infra rood sensoren die de afstand meten tot een obstakel voor de robot.
- * Vier Infra rood sensoren die de afstand meten tot een obstakel naast de robot.
- * Vier Infra rood sensoren die de afstand meten tot een obstakel onder de robot.
- * Vier Infra Rood proximity sensoren met een breed detectie veld om obstakels vlak voor en laag boven de grond waar te nemen indien deze niet door de primaire IR sensoren zijn waargenomen. Deze worden mogelijk nog aangevuld of vervangen door bumpers met eenvoudige schakelaars. De robot is zodanig opgebouwd dat simpel van rijrichting kan worden veranderd zonder dat een 180 graden turn nodig is. Zie plaatje. Bovenstaande onderdelen zijn reeds voorhanden en worden op dit moment getest. Er wordt nog een board ontwikkeld die meerdere sonar modules kan aansturen om nauwkeurig de positie van meerdere obstakels te bepalen en te volgen. Dit wordt een opzet zoals in een phased array sonar. De CMUCAM (of CMUCAM2) wordt gebruikt voor kleur herkenning van de bal. Mogelijk wordt er ook nog een kompas toegevoegd. Op dit board zal een (of meerdere) ATmega128 processor(en) worden gebruikt. Er wordt ook nog gewerkt aan een 2D LASER range finder. Het bal vang mechanisme wordt de veelvuldig gebruikte beweegbare hoepel. De software van de PMCU is geschreven in BASCOM (een basic compiler voor Atmel AVR processoren). De software van de Mini552 processor core is geschreven in C. De IR sensoren worden door de processor core met een frequentie van 32Hz bemonsterd via de I2C bus. De meetwaarden van de diverse IR sensoren worden aan een neurale net toegevoerd. (Zie plaatje) Het neurale net regelt de rijnsnelheid en de rotatie snelheid van de robot. De processor(en) op het sonar/video board zal obstakel en positie bepaling van de robot voor zijn rekening nemen. Dit board stuurt dan de Mini552 processor zo aan dat de gewenste snelheid en koers verandering wordt bereikt.

U kunt meer over de TNO competitie te weten komen op www.tno.nl/instit/fel/felnews/nl/robotcompetitie.html

Internet

Intressante site's op Internet.

<http://www.robotica.hccnet.nl/>

<http://www.ai.hccnet.nl/>

Agenda

Zaterdag 3 Juli	Open instuif	Nieuwegein.
Zaterdag 7 Augustus	Open instuif	Nieuwegein.
Zaterdag 4 September	Open instuif	Nieuwegein

Aanvang 10.00 uur tot 16.00 uur
Zie verdere info op de website.

**LET OP: extra bijeenkomst rond september in
HENGELO zie onze website.**



Adres bijeenkomsten:

t' Dok.

Hoornseschans 101,

3432TK Nieuwegein-Zuid

Wijk Fokkesteeg-Noord.

Tel: 030-6061.063.



_____oO_____Oo_____