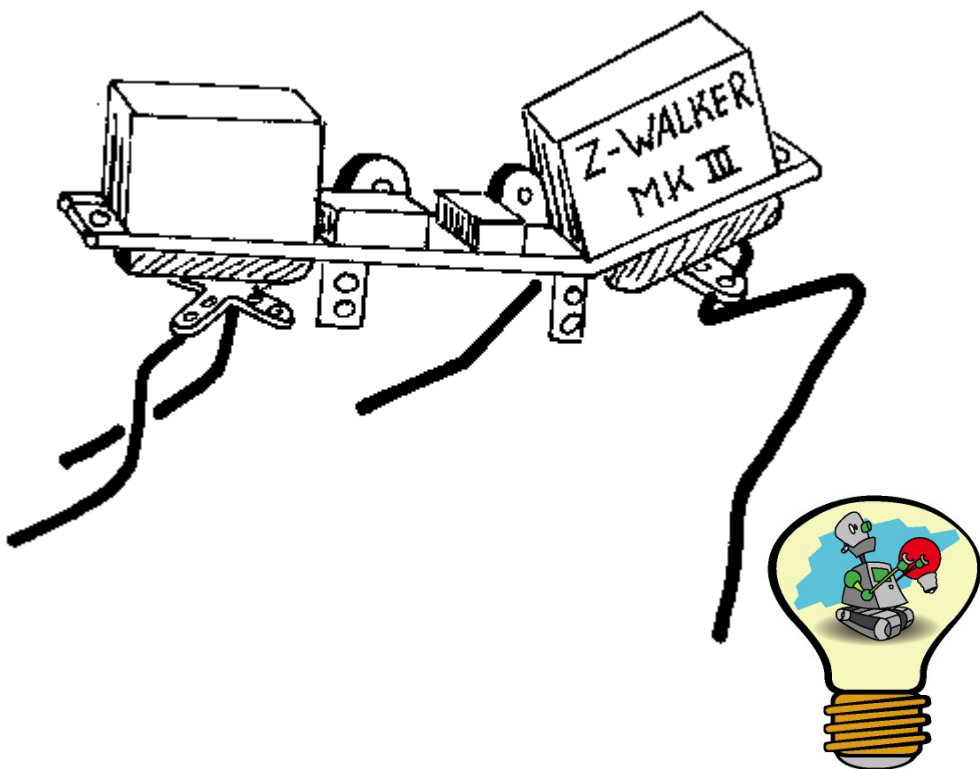


PTT Post
Port betaald
Port payé
Pays-Bas

ROBO- BITS-30

Jaargang 8, nummer 3, september 2005



HCC Robotica gebruikersgroep

Colofon

De Robobits is een uitgave van de Robotica-GG en wordt vier keer per jaar aan alle leden van de gebruikers groep toegezonden.

De oplage is 600 exemplaren.

Tevens is de Robobit voor leden gratis te downloaden als PDF bestand van de website van de gebruikers groep. De Robotica –GG is een onderdeel van de Hobby Computer Club.

Redactie adres:

H.J. de Gans,

Anjerlaan 3,

3871 EV Hoevelaken.

e-mail:

hj.de.gans@hccnet.nl

Tekst aanleveren in Word of platte tekst in ASCII. Afbeeldingen los erbij in TIF, BMP of JPG formaat.

Bestuur

Voorzitter:

B.T.J.A. Buiskool

Pilotenlaan 11

7943 CH Meppel

0522-241444

robot@buiskool.net

Technisch adviseur:

Ing. H.M.A. van Bodegom

Stadionlaan 180

7552 VE Hengelo 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

penningmeester:

H.J. de Gans

Anjerlaan 3

3871 EV Hoevelaken

033-2535479

hj.de.gans@hccnet.nl

Lid:

P. Smits

Lijtweg 302

2341 HB Oegstgeest

071-5156090

psmits.1@hccnet.nl

inhouds opgave:

- Bladz. 3 Redactie.
- Bladz. 4 Z-Walker MK3
- Bladz. 11 Eenvoudige.Z-Bridge Motordriver
- Bladz. 12 HCC TNO ROVER
- Bladz. 19 BOTIX open course..
- Bladz. 22 ROBORAMA.
- Bladz. 25 Servo controler besturing.
- Bladz. 28 Agenda.

REDACTIE

Ik merk duidelijk dat mijn oproepen om kopij voor de ROBOBITS resultaat beginnen af te werpen! Ik denk dat ook nu weer, een zeer gevarieerde ROBOBITS voor u ligt. Natuurlijk kunt u nu reeds uw bijdrage voor het december nummer insturen, zodat wij allen de kerstdagen door kunnen brengen met ons periodiek op schoot! Maar voor die tijd hebben we eerst nog de HCC dagen, waar we u hopen te ontmoeten en te spreken over onze gezamenlijke hobby. Voor de HCC dagen zeg ik vast : TOT DAN!, en voor nu zeg ik: VEEL LEESPLEZIER!

Henk de Gans

REGELS voor onze bijeenkomsten!!!!!!

DE GRABBELDOOS:

Het komt regelmatig voor dat iemand "zijn schuur" opruimt, en een aantal goederen meeneemt naar de bijeenkomsten. Dit onder het mom van grabbelen maar, voor ieder wat wils! Dit is prima en iedereen is hier blij mee, ALLEEN NEEM DAN WEL DE RESTANTEN MEE, DIE NIEMAND WIL HEBBEN, als je naar huis toe gaat! Wij als bestuur kunnen namelijk deze restanten niet achterlaten, en hebben ook geen zin die zelf mee naar huis toe te slepen!

KOFFIE:

Voor een ieder die onze bijeenkomsten bezoekt staat een kopje gratis koffie klaar! Het tweede en daarop volgende bakje koffie kost 50 eurocent. Dit wordt niet gecontroleerd, en deze regel berust dus op "goed vertrouwen". Deze opbrengsten zijn tot nu toe nooit kosten dekkend geweest, en dat is ook niet noodzakelijk! Wij verzoeken u allen echter vriendelijk deze regel in acht te nemen en het geld in het daartoe bestemde mandje te gooien. Voor bier dan wel frisdrank geldt dat bij gebruik van deze goederen ook de "verbruikslijst" van t'Dok "afgevinkt" dient te worden!

Alvast onze dank.

Het bestuur

-
deadline deadline deadline deadline deadline deadline deadline deadline

-

De deadline voor kopie voor Robobits 31 die eind december 2005 zal uitkomen, is gesteld op vrijdag avond 16 december 2005!!!! Maar u hoeft natuurlijk niet zo lang te wachten, nu reeds insturen kan natuurlijk ook! En iedereen wil heel graag eens wat lezen over dat waar u mee bezig bent, en natuurlijk met onze hobby te maken heeft! Vooral van u die niet in de gelegenheid bent onze bijeenkomsten te bezoeken! Zoals al eerder gezegd, hoeven het geen hoogdravende journalistieke meester werken te zijn (mag natuurlijk wel; -)). Gewoon een leuk stukje over uw creatie of iets dergelijks, of wat informatie over iets waarvan u gehoord of gelezen hebt enz. Ook zoek ik nog een boekbespreking (of tijdschrift).

-
deadline deadline deadline deadline deadline deadline deadline deadline

-

Z-Walker Mk 3

Handleiding voor een 4-potige loper.

(Een uitgebreidere uitleg in het Engels is op het internet te vinden op deze url: <http://vsim.zoelen.net/walk-kit.html>)

Omschrijving

De robot is gebouwd met het volgende in gedachte; gemakkelijk te bouwen, robuust, betrouwbaar, goedkoop en gebouwd van makkelijk te verkrijgen onderdelen. Et wordt een simpel, 4 insectachtige loper zonder sensors. Deze kunnen later eenvoudig aangebracht worden. De loper maakt gebruik van een MicroCore, een uitvinding van Mark W. Tilden (ook bekend van de RoboSapiens), een pulse neutraliseer circuit (PNC) voor het opstarten en 2 H-bridgen.

Specificaties

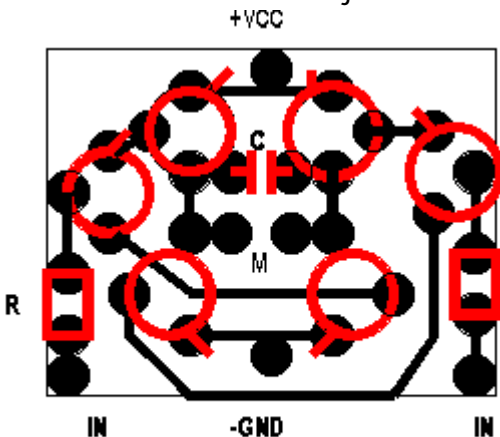
Hoogte	: 115 mm
Breedte	: 160 mm
Lengte	: 180 mm
Gewicht excl. Batterij	: 142 Gram
Snelheid	: 18 sec/meter
Voeding	: 3-15V
Verbruik	: 100 mA

Motor en motor driver

Je hebt 2 motortjes nodig. Ik heb 2 servo's gebruikt omdat hieraan de tandwielkasjes al vastzitten en die kunnen gelijk dienst doen als ophangpunt voor de rest van het frame. De gebruikte motoren zullen de afmetingen van je loper bepalen. De motoren moeten ongeveer 25 rpm draaien en zoals altijd, hoe lager het verbruik hoe beter.

Een standaard servo is niet te gebruiken als DC motor. Voor ons doel moet je de draaibare weerstand en het printplaatje eruit halen. De eind stop hoeft je niet te verwijderen. Deze stop kan gebruikt worden om te voorkomen dat de poten later helemaal door draaien. Je houdt nu een motortje met tandwielkastje over en een prima

ruimte om de motor drive in te stoppen. De motor driver kan je zelf maken. Een H-Bridge is zeer geschikt hiervoor.



H-Bridge

copper side

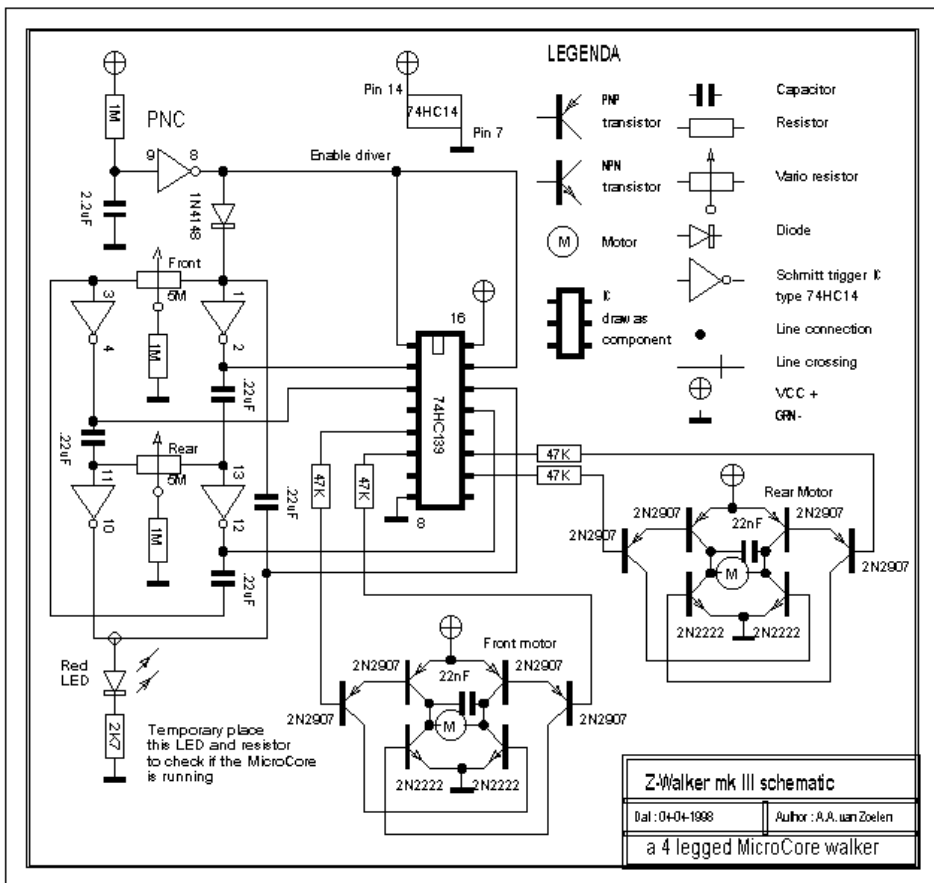
Onderdelen lijst

Transistors	: 4x PNP [2N2907], 2x NPN [2N2222]
Condensators	: 1x 22nF
Weerstanden	: 2x 47K

Deze onderdelen kunnen ingebouwd worden in de lege ruimte in de servo. Leid de 4 draden naar buiten. 1 draad voor +VCC, 1 voor GND en twee voor de draairichting. Controleer of alles werkt voordat je de servo weer dicht schroef. Sluit de plus en min op een batterij aan. Houd nu één van de twee draden voor de richting tegen de plus en de motor moet gaan draaien. Gebruik je de andere draad dan moet de motor de andere kant op draaien. Sluit de servo en doe hetzelfde met de andere servo.

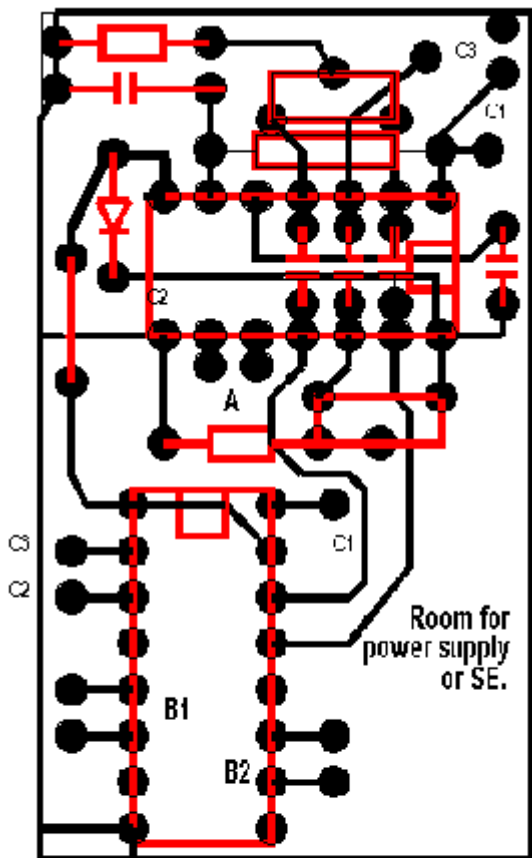
Frame

De breedte van de servo's bepaalt de minimum breedte van de body van de loper. Je moet wat ruimte vrij houden tussen de 2 servo's om de MicroCore en de overige componenten te kunnen plaatsen. Ik gebruik nog een extra driver om de H-bruggen aan te sturen. Deze driver dient op de motors te ontkoppelen tijdens het opstarten. Je kan alles in de lucht solderen volgens het onderstaande schema of een printje gebruiken.



Onderdelen lijst

- IC's : 1x 74HC14, 1x 74HC139
- Condensatoren : 4x 0.22uF, 1x 2.2uF
- Weerstanden : 3x 1M, 1x 2k7
- Diodes : 1x 1N4148
- Overig : 2 IC sockets, 2x 5M vario-resistor
- LED red, batterij houder, Schakelaartje



PCB not checked or tested

Remarks :

A - Spare inverter

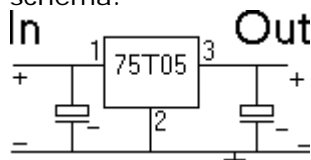
B1-2 to motor 1-2

C - Wire bridge

Power regulator parts
5V - 78T05 or 3V -
LD1117
Condensators : 2x
10uF/16V

Het gebruik van een voltage regelaar is optioneel. Maar wel aan te raden. Hierdoor kan je zonder problemen allerlei formaten batterijen gebruiken zonder je zorgen te hoeven maken over de juiste spanning.

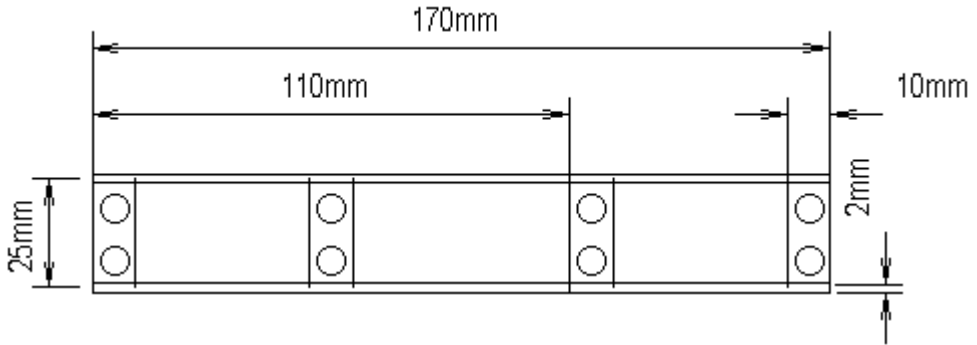
Hier als voorbeeld een schema.



Je kunt het frame van twee dikke koper draden maken. Van een paar stukjes printplaat heb ik de steuntjes voor

de servo's gemaakt en tussen de in de vorm gebogen koperen draden gesoldeerd. Als later de print ertussen geplaatst wordt dan vormt het geheel een stijf genoeg frame.

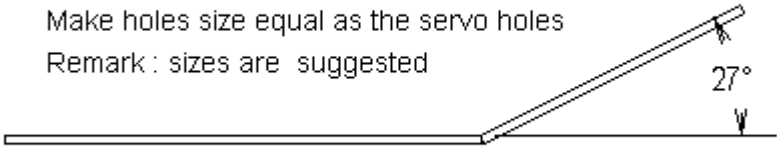
Hier zijn de afmetingen van het frame zoals ik dat gemaakt hebt.



Copper wires with PCB plates

Make holes size equal as the servo holes

Remark : sizes are suggested



Frame

MicroCore testing

Soldeer nu eerst de componenten voor de MicroCore op de zijn plaats. Meer details over de MicroCore kan je vinden op <http://vsim.zoelen.net/amiller/microcore.html>

De werking kan getest worden door de uitgangen van de MicroCore met een LED en een 2K7 weerstand te verbinden met GND (Pin 10 bijvoorbeeld). Als het goed is moet de LED nu knipperen. Als de MicroCore werkt kan je de overige componenten toevoegen. Als alles klaar is kan je de spanning erop zetten en kijken of de MicroCore nog steeds werkt. Bij het opstarten moeten alle pulsen gestopt worden en slechts 1 puls mag rond gaan. Na een korte pause moeten de servo's gaan werken. De goede volgorde van de bewegingen is:

+-----+-----+-----+-----+-----+

|Rear | CCW | off | CW | off |

+-----+-----+-----+-----+-----+

|Front | off | CCW | off | CW |

+-----+-----+-----+-----+-----+

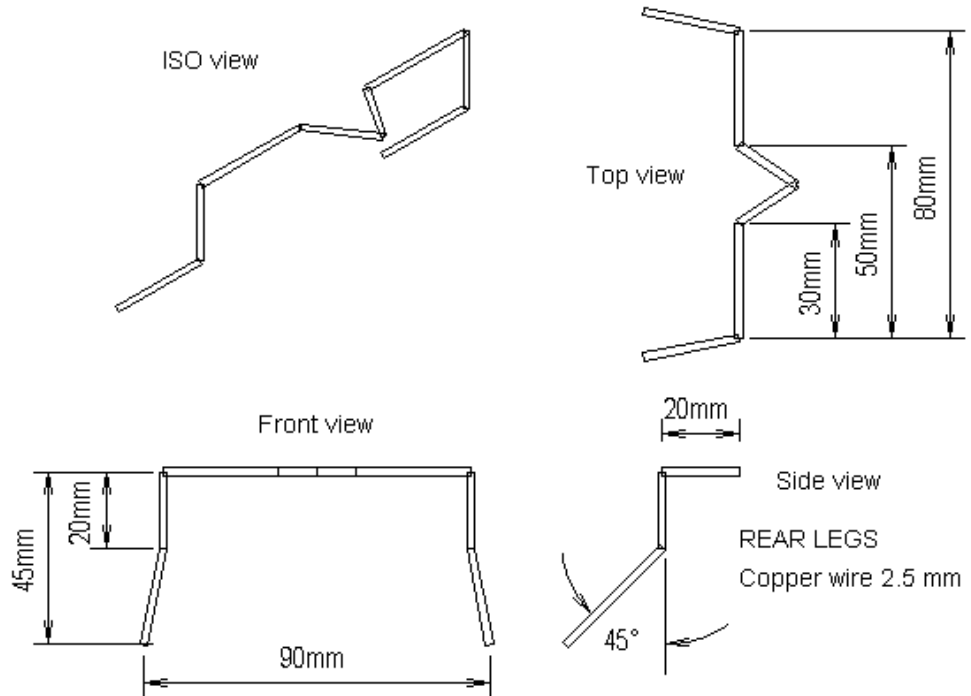
CC ClockWise CCW CounterClockWise

Als dit niet het geval is dan moet de twee draairichting bepalende draden van 1 servo omdraaien. Als alles goed werkt kan je de LED en de test weerstand verwijderen

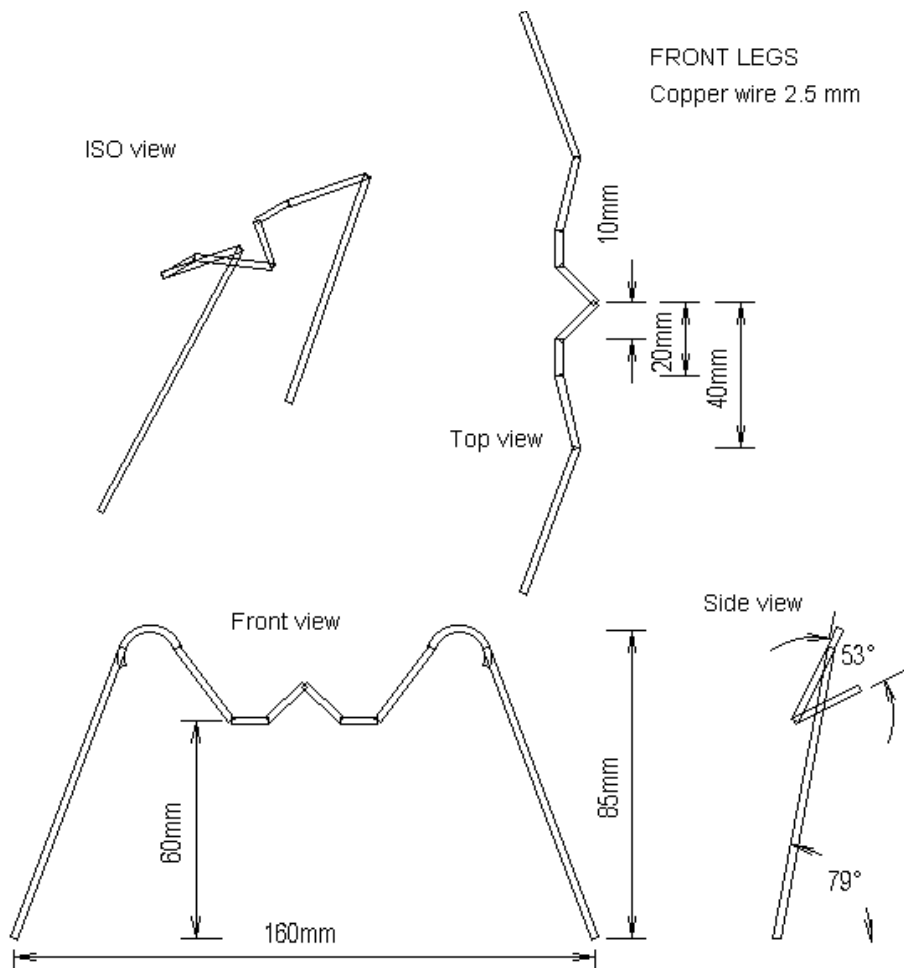
Opmerking.

Als alles correct is dan kan je het volgende doen. Als je de draden van 1 motor, meestal de achterste motor omdraaien. Die veroorzaakt dat de robot achteruit gaat. Dit gedrag kan je met een 'reverser' circuit bewerkstellen. Kijk om de site voor meer details hierover.

De poten



De poten bepalen in sterke mate het loopgedrag. Ik heb de voorpoten langer en wijder gemaakt dan de achterpoten. Hierdoor kunnen obstakels makkelijker genomen worden terwijl de achterpoten meer voorwaartse kracht leveren



Servo's hebben meestal een setje spaken en wielen. Deze kan je goed gebruiken om de poten met metaal draad hieraan vast te zetten. Om te voorkomen dat de poten te ver doordraaien kan je gebruik maken van allerlei systemen. Het simpelste zijn gewoon eind stops. Meer details over de poten is te vinden op de site

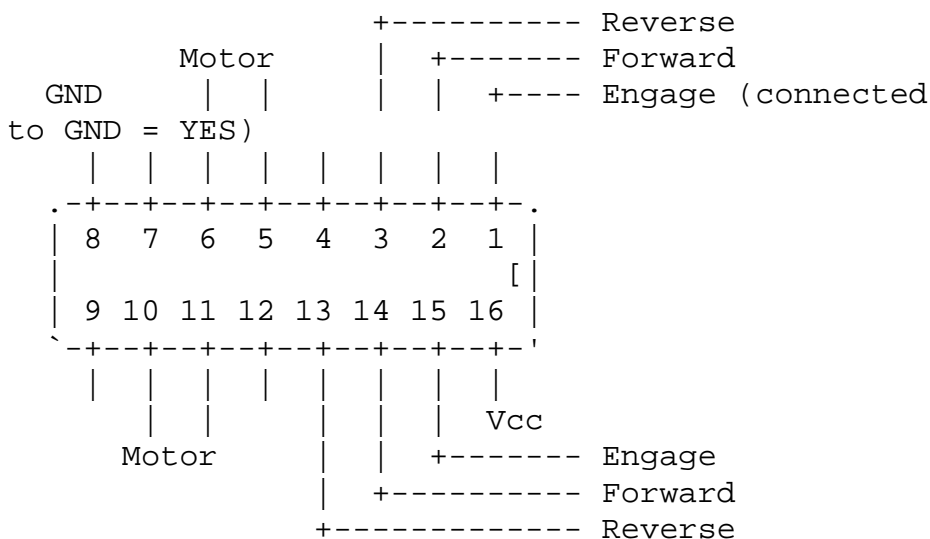
Met vriendelijke groet,

Bram van Zoelen.

Z-Bridge DC-Motordriver

Z-Bridge, DC Motor driver

Voor het aansturen van kleine DC motortjes wordt vaak een H-bridge gebruikt (teveel onderdelen) of een speciale driver chip (prijzig). Vaak ontbreekt er ook nog een draairichting controle mogelijkheid en een actief, non-actief functie. Ik heb een erg onconventionele manier voor een DC motorbesturing uitgedacht namelijk met 1 enkele makkelijk verkrijgbare chip, de 74ACT139. De besturing voldoet aan de volgende eisen namelijk; Enable/disable, draairichting controle, doorbrand bescherming, 2 motoren tegelijk aansturen en stapelbaar. Met het laatste wordt bedoeld dat je meer stroom kan trekken door simpelweg de chips te stapelen.



Meer details zijn te vinden op de volgende url:

<http://vsim.zoelen.net/z-bridge.html>

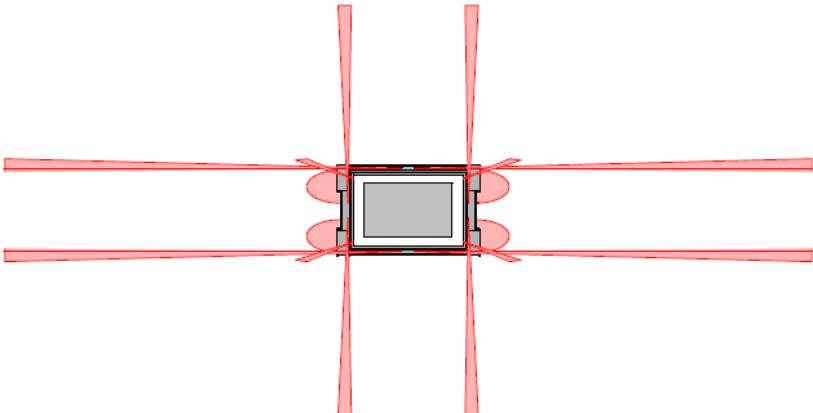
Met vriendelijke groet,

Bram van Zoelen.

De HCC Robotica gg robot voor deelname aan de TNO robot competitie 2005 heeft de volgende opzet:

- Tank onderstel Conrad Catwiesel (hiervoor wordt een bestaande Robotica gg robot omgebouwd).
- Stuurprint (TANK5 print) waarop een Power Monitoring and Control Unit zit, de P80C552 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 sensoren die de afstand meten tot een obstakel onder een hoek van 45 graden tot 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.
- Stuurprint (TANK6 print) met daarop 2 ATmega128 processoren
- De sensoren kunnen nog worden aangevuld of vervangen door bumpers met eenvoudige schakelaars.
- Intelligentie doormiddel van een laptop

De robot is zodanig opgebouwd dat simpel van rijrichting kan worden veranderd zonder dat een 180 graden turn nodig is.



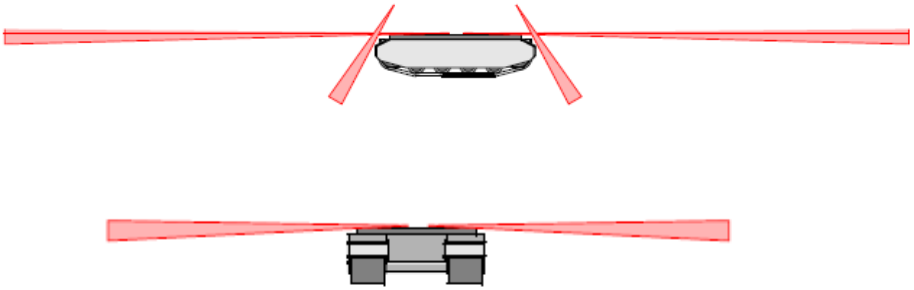


Figure 1. IR sensor opstelling.

Het TANK6 board welke ontwikkeld is kan meerdere sonar modulen aansturen om nauwkeurig de positie van meerdere obstakels te bepalen en te volgen. Dit wordt een opzet zoals in een phased array sonar. Echter hier bepaald iedere sensor afzonderlijk de afstand tot de aanwezige doelen (maximaal 17). In een phased array sonar worden alle ontvangers coherent gesampled en d.m.v. bundelvorming wordt de richting en afstand tot het doel bepaald. Dat gaat echter voor de door ons toegepaste microcontrollers te ver. De CMUCAM (of CMUCAM2) wordt gebruikt voor kleur herkenning van de bal, er wordt een kompas toegevoegd om de richtingsverandering te kunnen meten tijdens het draaien van de robot. Er wordt ook nog gewerkt aan een 2D LASER range finder. Deze was niet op tijd af. 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 een I2C bus. De meetwaarden van de diverse IR sensoren worden aan een neurale net toegevoerd. Het neurale net bepaalt de gewenste rijnsnelheid en de rotatie snelheid van de robot zodat contact met obstakels wordt vermeden. 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.

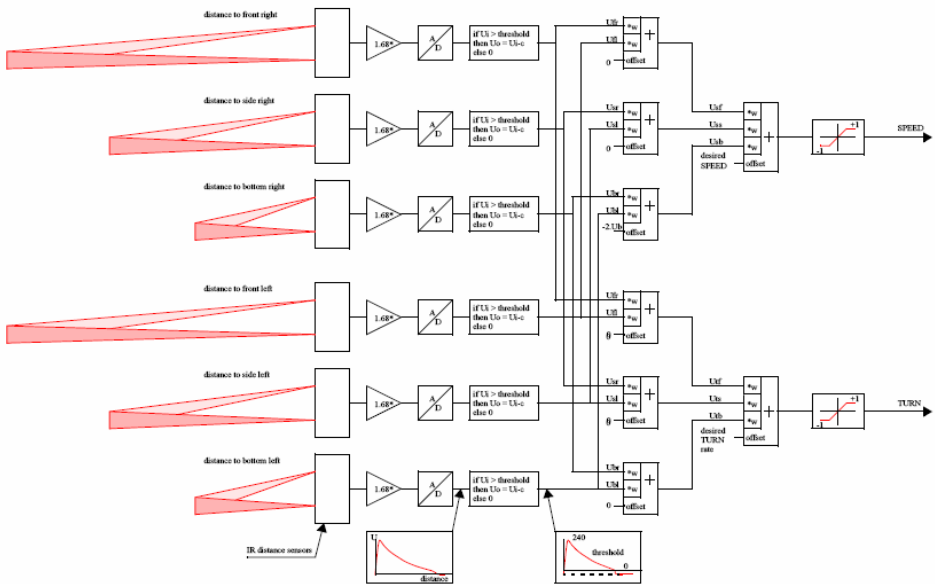


Figure 2. Neural Net.

2. TANK5 board beschrijving.

Het TANK5 board bestaat uit:

- Power Monitoring and Control Unit.
- Local power supply.
- MINI552 processor core.
- PWM DRiVe.

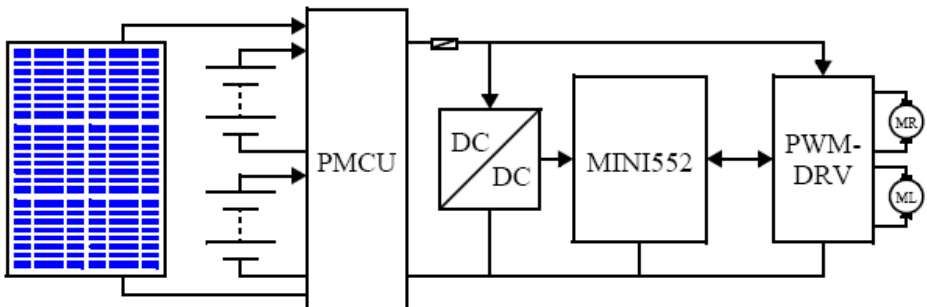


Figure 3. TANK5 blok schema.

2.1. Power Monitoring and Control Unit.

De PMCU heeft tot taak:

- de robot in en uit te schakelen.
- de accu met de grootste lading als eerste te kiezen voor het voeden van de robot en wanneer deze leeg is de andere accu te kiezen middels twee FET schakelaars.
- wanneer beide accu's bijna leeg zijn de robot uit te schakelen.
- wanneer voldoende spanning op het laadcontact (zonnepaneel) aanwezig is de minst geladen accu als eerste te laden totdat deze vol is en daarna de andere, middels twee FET schakelaars.
- het laden en ontladen van accu's onafhankelijk van elkaar te laten verlopen. (het kan dus voorkomen dat een accu wordt geladen die tevens wordt ontladen)

De PMCU processor kan middels een PCF8591 ADC op een I2C bus het volgende meten:

- klemspanning van het laadcontact.
- klemspanning van batterij 1.
- klemspanning van batterij 2.
- totale stroom geleverd aan de robot.

De uitgang van de PMCU naar de overige hardware van de robot is beveiligd door een resettable fuse van Bourns 3A nom. / 6A max.

2.2. Local power supply.

De TANK5 print heeft een operationeel spanningsbereik van 10V tot 25V. Om dissipatie te beperken en het rendement te maximaliseren wordt voor de voeding van alle digitale hardware een DC/DC converter gebruikt. Ook is er nog een dissiperende instelbare regelaar voorhanden voor het voeden van de IRPROXIM bumper sensoren, de CMUCAM(2) of een video camera.

2.3. MINI552 processor core.

De Mini552 processor core is opgebouwd rond een P80C552IBA processor van Philips. Deze processor heeft een kristal frequentie van 24MHz. Effectief kunnen maximaal 4M Instructies Per Seconde worden verwerkt. De processor beschikt over 32K*8 statische RAM. De processor heeft tevens 128K*8 Flash memory die wordt verdeeld in twee banken van 32K waarvan de een het monitor programma en zoektabellen bevat en de ander het applicatie

programma. Er zijn 16 parallelle I/O's waarvan er 4 worden gebruikt voor aansturing van de PWM drive, 4 voor de bumper schakelaars en nog 8 vrij voor uitbreidingen. Ook heeft de processor een 8 kanaals 10Bit A/D Converter (deze ingangen kunnen als CMOS compatible ingangen worden gebruikt) die in geval van de TNOROVER nog niet worden gebruikt. De RS232 poort is verbonden met een multidrop seriële bus welke met de PMCU processor is verbonden, met de terminal aansluiting en met een connector die naar de sonar processor print gaat. Zo zijn alle processoren in staat met elkaar om beurten te communiceren, zei het op een lage snelheid van 19.200 baud. Tot slot is er een I2C bus die is verbonden met 4 stuks 4 kanaals ADC's van het type PCF8591. Elke PCF8591 beschikt over een vierkanaals voorversterker die het signaal van 4 stuks GP2D12 (of soortgelijke typen) met 1,68 maal versterkt, om beter aan het maximum bereik van de ADC's te voldoen.

2.4. PWM drive.

De motoren worden middels Pulse Width Modulation aangestuurd. Hiervoor worden twee LMD18200T's van National gebruikt. Deze zijn tegen overbelasting beveiligd en schakelen de motoren ook uit wanneer de accuspanning te laag wordt (<10V). Deze IC's zijn echter niet kortsluitvast! De IC's hebben tevens een voorziening die het mogelijk maakt de motorstroomsterkte te meten. Deze meetpunten kunnen m.b.v. een patch kabel op de vrije ADC ingangen van de mini552 processor worden aangesloten.

3. TANK6 board beschrijving.

Het TANK6 board is nu ontworpen. We hebben gebruik gemaakt worden van een aantal CRUMB128 processor boardjes die elk een ATMEL ATmega128 bevatten. Deze sturen de sonar sensoren aan, het kompas, de CMUCAM en de zender voor data logging. De core voor dit board is geschreven in C. Het Tank6 board is opgebouwd rondom twee CRUMB128 (deze bevat elk 1 ATMEL ATmega128). Elke crumb heeft twee RS232 in/uitgangen. Elke crumb heeft 64 Mb RAM tot zijn beschikking. De CRUMB's zijn in-circuit te programmeren. In totaal zijn er twaalf I2C poorten naar buiten gebracht, 6 servo uitgangen en twee stripelasers.

3.1. Sonar.

Deze is nog in ontwikkeling. De 3 of 4 SRF08 sonar sensoren krijgen een dedicated CRUMB128 om deze sommetjes uit te rekenen. Het

algorithme is bekend en gesimuleerd. Een en ander staat of valt met de betrouwbaarheid en resolutie waarmee de afzonderlijke sensoren de afstand tot de verschillende obstakels kunnen meten. We zijn dit op dit moment aan het testen. Ernstige interferentie wordt verwacht van publiek dat op of over de rand van het missieveld hangt.! Aan dit project wordt nu nog gewerkt.

3.2. Kompas.

Het kompas wordt op een mast gemonteerd en wordt hoofdzakelijk gebruikt om de richtingsverandering te meten terwijl de robot stil staat en pas-op-de-plaats roteert. Getest en goed bevonden

3.3. CMUCAM(2)

Wordt primair gebruikt in missie 3 voor het onderscheiden van rode of blauwe ballen. Getest en is in de totale software geïmplementeerd.

3.4. Telemetrie.

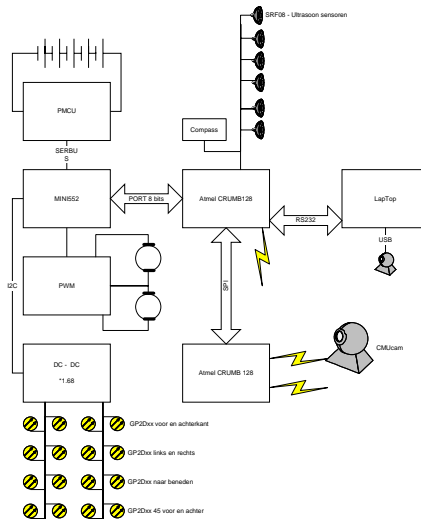
De robot zendt zijn gemeten gegevens real time uit om te kunnen worden gelogd voor latere analyse. Het is een eenrichtingsverbinding uitsluitend bedoeld om data te kunnen loggen en eventueel real time te kunnen zien wat de robot "denkt" waar te nemen. Werkt maar gaat voorlopig nog niet gebruikt worden. Al snel na de start van ons project werd ons duidelijk dat wij een logistiek probleem hadden. Hengelo – Meppel – Mijdrecht ligt niet naast elkaar. We konden dus niet soepel even de robot voor een avondje lenen en aan elkaar doorspelen. Al gaande het denk- en ontwerp proces kwamen we tot de conclusie dat het heel handig zou zijn als wij meer intelligentie aan de robot konden toevoegen. Wij hebben gekozen voor een laptop welke via een seriële poort met de tank6 praat.

Het was ook duidelijk dat wij de taken moesten gaan verdelen:

- Henny van Bodegom heeft het bouwen en opzetten van de hardware ter hand genomen. Ook het bedenken van diverse algoritmes is aan hem toebedeeld.
- Fred Eisen heeft een multiprocessor operating systeem ontworpen voor de tank6. Zodat we de communicatie tussen de diverse sensoren en tank6 konden doen. Evenals de communicatie van de tank6 met tank5 en tank6 met de laptop.
- Bert Buiskool heeft een simulator ontworpen en gemaakt om plaatsing van sensoren, missies en tactiek vooraf te kunnen

testen. Eveneens is er een protocol bedacht welke "universeel" toepasbaar naar andere robots moet kunnen zijn. Deze simulator is verder beschreven in het hoofdstuk glAre. (*hierover meer in ROBOBITS 31 red.*)

De testronde hebben wij niet met goed gevolg kunnen volbrengen. Op zich verwonderde ons dat niet. In week 11 kwam door allerlei oponthoudt eindelijk ons tank6 board gereed, maar moest toen nog gemonteerd gaan worden. Pas daarna was het moment daar om de diverse communicatie tussen de systemen te testen. Op papier was alles wel bedacht maar nu het geheel nog. Op tweede paasdag 2005 was het dan eindelijk zover dat wij bij elkaar konden komen en konden gaan testen. De communicatie Tank5 en Tank6 ging min of meer gelijk goed. De communicatie tussen Tank6 en de laptop wou niet vlotten. glAre(*hierover meer in ROBOBITS 31 red.*) is gebouwd in VB.Net en de RS232 communicatie ging niet soepel. De vrijdagavond voor de test zijn we weer naar Mijdrecht getogen en hebben eea werkend gemaakt. Op zaterdag hebben we een poging ondernomen om onze robot te laten rijden, Wij werden echter onaangenaam verrast door het tapijt. Dit bleek zoveel wrijving op te wekken (die wij niet hadden voorzien) waardoor het neurale net de robot niet om kon duwen als er een obstakel verscheen. Daardoor had onze poging geen goed resultaat. Gedurende de dag hebben we met een aantal keren proberen overigens de juiste waardes gevonden zodat onze robot in de toekomst wel in staat is om zich op een dergelijk tapijt te begeven.



Hulp gevraagd!

Botix

Begin september ben ik een open-source software project gestart om een *hardware-onafhankelijk framework* voor robot aansturing te schrijven in C. Het framework is geschikt voor 1 of meerdere processoren. Na enkele minuten hard nadenken heb ik het project de naam "Botix" meegegeven.

Waarom Open-Source?

Ik ben zelf een enthousiast gebruiker van open source software (Ubuntu linux, openoffice, firefox, thunderbird etcetera) en ben zeer gecharmeerd van het ontwikkelingsmodel ervan. Iedereen die is geïnteresseerd kan helpen een applicatie/project groot te maken en vaak lukt dit ook uitstekend. De ontwikkelaars kunnen zo ook de features implementeren die zij belangrijk vinden. Verder kunnen de niet-programmeurs kosteloos beschikken over software die ze zelf niet zo mooi hadden kunnen maken en helpen door de auteurs te wijzen op fouten (bugs) of op andere manieren een steentje bij te dragen.

Ontwerp

Nadat ik bij mijn informatica studie met de "Subsumption Architecture" van Brooks in aanraking ben gekomen heb ik deze theorie aan- en toegepast op mijn toenmalige robot-in-aanbouw, 'Mispunt'. Kort gezegd krijg je met deze architectuur voor elkaar dat je met eenvoudige regels intelligent gedrag kunt genereren op de robot. Deze implementatie van de Subsumption Architecture bleek heel erg goed te bevallen omdat de code overzichtelijk, flexibel en stabiel blijft, ondanks dat hij redelijk complex was.

In het framework worden met 3 'lagen' gewerkt, ruwweg te vertalen naar prioriteiten. Is er bijvoorbeeld een event in laag 0 (hoogste prioriteit, b.v. bumper botsing) dan wordt daar direct op gereageerd. Is er geen event in laag 0, dan wordt er gekeken of er een event in laag 1 is etcetera.

Voorbeelden van events:

Laag 1: Botsing, robot komt zeer dicht bij een wand of 'end of world'. *Oplossen van acute problemen.*

Laag 2: Robot is verder van wand, kan nog rustig bijsturen. Odometrie of kompas wijst uit dat de robot afdwaalt. Accuniveau laag. *Oplossen van niet-acute problemen.*

Laag 3: Uitvoering van opdrachten gekregen van processor 2 of wat je maar kan bedenken. *Overige acties.*

De rode draad in het ontwerpen van dit framework is dat de logica gescheiden wordt van de hardware middels een abstractielaag. Denk bijvoorbeeld aan functies als "Distance_To_Object_Left_Front()" of "Drive_Forward(throttle)".

Hardware

Alle ondersteunde hardware wordt in z.g. include files beschikbaar gemaakt en via gestandaardiseerde functies aangeroepen. Voor de programmalogica is het *in de regel* niet interessant welke hardware erachter zit, slechts wat de functie is. (Denk aan afstandsmeting, richting, odometrie/afstand, 'boem-is-ho bumpers' etcetc.) Om dit voor elkaar te krijgen, moet enige abstractie worden toegepast, die in de hardware-include wordt vertaald naar de pinnetjes, bitjes en timertjes. Het lastigste is volgens mij het op alle soorten processoren te laten werken die in C te programmeren zijn, als dat al mogelijk is.. :)

Status

Mja, op 't moment is 't nog niet veel... Er is een begin gemaakt met de structuur en dat is 't dan meteen zo'n beetje. Het project is ondergebracht bij Sourceforge.net (een van de grootste open-source ontwikkelsites die er zijn) en staat in CVS aldaar. CVS zorgt ervoor dat meerdere ontwikkelaars aan hetzelfde programma kunnen werken, zonder dat zij elkaars wijzigingen overschrijven. Verder is er de project-website, waar op dit moment helaas nog niet zoveel informatie te vinden is. Hier komt documentatie en is er gelegenheid met elkander te communiceren via een forum. Het verzenden van emails door de site is nog niet voorelkaar, hier heeft Sourceforge een nogal rigoreuze insteek in. (Hier is misbruik van gemaakt in 't verleden)

SF-site: <http://sourceforge.net/projects/botix>

Projectsite: <http://botix.sourceforge.net>

Beide sites zijn Engelstalig.

HELLUP!

Om tot een mooi resultaat te komen waar iedereen wat aan heeft is veel hulp nodig... Dit is de belangrijkste reden dat ik het project heb aangemeld op Sourceforge en voor de voertaal Engels heb gekozen. Ik hoop dat vele mensen het nut inzien van een dergelijk project en willen meehelpen!

Projectleden/ontwikkelaars

Vanzelfsprekend zijn er ontwikkelaars nodig, die aan het framework willen werken. Wellicht kan aan iedere ontwikkelaar een bepaald deel van 't framework worden toegekend, natuurlijk in overleg en naar voorkeur van het desbetreffende persoon. Als je wilt helpen, graag! Ook is er -als 't project groter wordt- wellicht een aparte webmaster nodig. Hier is geen programmeerkennis voor vereist, de website is voor 100% via de browser te beheren.

Code-snipjets donateurs

Als je code hebt om bepaalde hardware aan te sturen, ontvang ik 't graag om het in 't framework te hangen. Natuurlijk moet de code wel in C geschreven zijn. Ik kan alleen maar code voor de AVR testen, maar C-code voor de overige controllers is ook welkom en wordt bewaard in afwachting van projectleden/ontwikkelaars georiënteerd op overige hardware. Als er wat regels commentaar/uitleg inzitten is dat natuurlijk erg prettig!

Testers... maar pas als er een basis-framework ligt.

Iedereen die wil helpen kan contact opnemen met mij,
Ger Apeldoorn: botix@gerapeldoorn.nl

Op de projectsite <http://botix.sourceforge.net> is meer informatie te vinden en er wordt regelmatig info toegevoegd!

Met vriendelijke groet,

Ger Apeldoorn.



WAT IS ROBORAMA?

RoboRama is een wedstrijd waarin robots verschillende opdrachten uitvoeren. Zo wordt invulling gegeven aan het doel van RoboRama, het bieden van een forum waar robotbouwers de mogelijkheden van hun robots kunnen demonstreren. Door deze demonstraties zullen ook anderen aangemoedigd worden om robots te bouwen. Robotmc houdt één RoboRama wedstrijd per jaar. De eerste wedstrijd zal plaatsvinden op 6 mei 2006. Op elke Robotmc bijeenkomst zal een beperkt oefenterrein beschikbaar zijn.

ROBORAMA REGELS

De RoboRama regels zijn gebaseerd op de DPRG Roborama regels. (zie <http://www.dprg.org/competitions/>)

Uitgangspunten

Het moet mogelijk zijn met één robot aan alle opdrachten deel te nemen en om deze robot ook in andere omgevingen te gebruiken, bijvoorbeeld bij je thuis in de living. De wedstrijd moet toegankelijk zijn voor starters en ook een uitdaging bieden voor de meer ervaren robotbouwers.

Opdrachten

In een RoboRama wedstrijd voeren robots afzonderlijke opdrachten uit onder leiding van een scheidsrechter. Een robot kan aan één of aan meerdere opdrachten deelnemen. Bij aankondiging van een RoboRama wedstrijd op www.robotmc.org wordt aangegeven uit welke opdrachten de wedstrijd is opgebouwd. Op dit moment zijn de volgende opdrachten gedefinieerd:

Heen & Weer: kan uw robot van punt A naar B en terug gaan?

Lijnvolgen: kan uw robot een lijn op de vloer volgen?

T-tijd: kan uw door een eenvoudig T-vormig terrein navigeren?

Blikken: kan uw robot blikjes verzamelen?

In de toekomst worden nieuwe opdrachten toegevoegd.

Terrein

Het wedstrijdterrein bestaat, afhankelijk van de opdracht, uit een rechthoekige of T-vormige baan. De grenzen van het wedstrijdterrein worden gevormd door 20 cm hoge muren. De muren bestaan uit egaal wit geschilderd hout. De afmetingen van het terrein wijken in iedere richting maximaal 15 cm af van de opgegeven waarden. De vloer van het terrein zal bestaan uit de vloer zoals aanwezig op de plaats waar de wedstrijd gehouden wordt. In de meeste gevallen zal dit een linoleum vloer zijn met een lichte tot grijze kleur.

Lijnen

De wedstrijdcommissie zal waar nodig zorgen voor het aanbrengen van lijnen op de vloer. Middels deze lijnen kan een robot detecteren dat hij een bepaald vak binnengaat of verlaat. Op overwegend licht gekleurde vloeren zullen de lijnen zwart gekleurd zijn. Op donkere vloeren zal er gebruik gemaakt worden van witte lijnen. De kleur van de lijnen die voor een wedstrijd gebruikt gaan worden zal van tevoren bekend worden gemaakt.

HEEN & WEER

Primaire Doelstelling:

Een autonome robot van gebied A naar gebied B laten gaan en weer terug.

Secundaire Doelstelling:

Verwezenlijken van de primaire doelstelling in de kortste tijd.

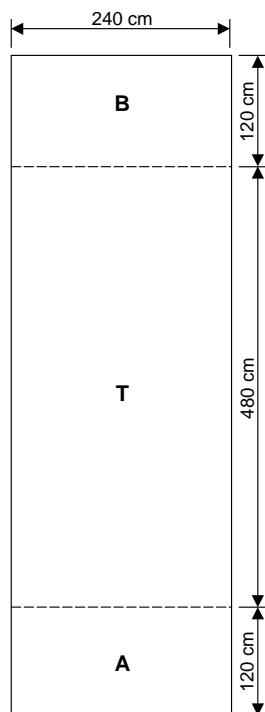
Terrein: Zie afbeelding.

Geteste vaardigheden: navigatie

Tijd om opdracht te voltooien: 5 minuten

Details: De robot moet in gebied A starten. Hij begeeft zich dan naar gebied B en daarna weer

Parcours Heen & Weer



terug naar gebied A. De deelnemer kan de robot overal in gebied A en in iedere richting laten beginnen.

De gebieden A, T en B zijn van elkaar gescheiden door middel van lijnen op de vloer. Deze lijnen worden gebruikt om te bepalen wanneer de robot een gebied is ingegaan.

Te behalen punten:

1ste kortste tijd om de opdracht af te ronden	40
2de kortste tijd om de opdracht af te ronden	20
3de kortste tijd om de opdracht af te ronden	10
Bereiken van gebied B	10
Terugkeren in gebied A	10
Stoppen in gebied A na terugkeer	10
Muren niet aanraken	15
Maximaal te verdienen	85

De benodigde tijd wordt gemeten vanaf het startsignaal tot het moment waarop de gehele robot de finishlijn heeft overschreden.

Het volledige reglement is te downloaden op

<http://www.robotmc.org>,

als voorbeeld is hier enkel de "Heen en Weer" getoond.

Wie gaat de uitdaging aan om zijn robot "Heen en Weer" te laten rijden?

Met vriendelijke groeten,
Koen Van den Heuvel

Naschrift bestuur:

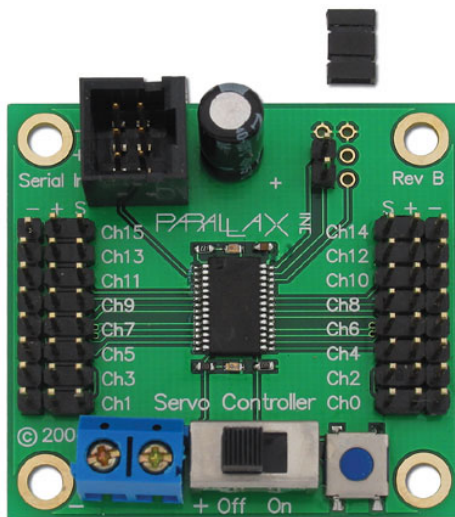
Een delegatie van het bestuur is op bezoek geweest bij onze Belgische mede hobbyisten, om eens met elkaar van gedachten te wisselen. Het zou namelijk een heel leuk idee zijn zowel in België als in Nederland een dergelijke competitie te houden. Met name het feit dat deze competitie laagdrempelig is, maar toch voldoende uitdaging biedt om ook de gevorderde roboticaan te inspireren, heeft ons doen besluiten ons in te gaan zetten om deze wedstrijd ook in Nederland te gaan organiseren. Wij beraden ons momenteel over hoe het een en ander vorm te gaan geven, en zijn voornemens op de hcc dagen van 2006 de eerste Nederlandse wedstrijd te gaan

houden. De wedstrijd in België vindt zoals Koen al schrijft in mei plaats. Dit is dus een mooie spreiding in het jaar. Wij houden u op de hoogte!

Het bestuur.

SERVO CONTROLLER BESTURING

Er zijn vele manieren om besturingen te doen. We denken dan aan de bekende printerpoort, de seriële poort en nu sinds kort de USB aansluiting die gebruikt wordt om binnen de hobbywereld metingen en besturingen te verrichten. Al deze poorten zitten op de ons bekende PC. Hiernaast zijn er een scala aan op microcontroller gebaseerde borden beschikbaar om metingen en besturingen te verrichten. Vaak zien we binnen de microcontroller wereld dat er weer bordje zijn met een specifieke taak. Hier komen we sensor- en motorstuurprinten tegen. Ook komen we de laatste tijd servocontrollers tegen. Vanuit onze microcontroller sturen we een "opdracht" naar deze controller en hij verzorgt deze opdracht. Het grootste voordeel hiervan is dat we veelal tot 16 servo's aan kunnen sluiten, zonder dat we ons zorgen hoeven te maken over de timing die voor de aansturing van de servo's komt kijken.



Van de ons wel bekende firma PARALLAX kwam ik onlangs zo'n servo controller tegen. Klein, grappig en redelijk geprijsd. Maar er werd vanuit gegaan dat je hem vanuit een BasicStamp aan zou spreken. Dit laatste was niet helemaal mijn bedoeling. Een kennis wilde hem vanuit de PC gaan besturen. Mbv van VB5.0 hebben we een klein programma geschreven om de servo controller aan te spreken. Vanuit VisualBasic5.0 gaan we als volgt te werk.

In de Form_Load zetten we:

```
MSComm1.CommPort = 1  
MSComm1.Settings = "2400,N,8,2"  
MSComm1.RThreshold = 1  
MSComm1.PortOpen = True
```

Hier staat dat we comppoot 1 gebruiken met de instellingen 2400 Baud, Geen pariteit, 8 databits en 2 stopbits. Daarna activeren we de comppoot en is hij klaar voor gebruik. Om nu een bepaalde servo te verstellen moeten we volgens een bepaalt protocol gegevens aan de Servo Controller aanbieden. We moeten een serie van 8 byte's versturen om een servo in een bepaalde stand te zetten. (De Parallax Servo Controller kent nog meer commando's, maar daar maken we hier geen gebruik van.) We doen dit op de volgende manier met het volgende VB5.0 commando.

```
x = "!SC" & Chr(ch) & Chr(ra) & Chr(low_byte) & Chr(high_byte) &  
Chr(13)  
MSComm1.Output = x
```

Wat doen we hier en wat betekent dat?

In de eerste regel maken we een string van 8 byte's, de eerste 3 zijn de characters ! S C, en daarna 4 variabelen en als laatste het character met de waarde 13. Op de tweede regel zetten we deze string in de outputbuffer van de seriële poort, en deze stuurt de string naar buiten toe.

De betekenis is als volgt:

<i>MSComm1.Output</i>	= Dit is het VB5 commando om gegevens naar de seriële poort te sturen.
<i>"!SC"</i>	Het voorloop kenmerk voor de controller
<i>ch</i>	Servo nummer, 0 - 15
<i>ra</i>	Servo verstellings snelheid, 0 - 63. 0 = snel, 63 = 1 minuut
<i>low_byte</i>	Lage byte van de servostand waarde
<i>High_byte</i>	Hoge byte van de servostand waarde. (servostand = 250 - 1250)
<i>13</i>	Afsluitende karakter

Door voor bovenstaande variabelen bepaalde getallen in te vullen kunnen we de verschillende servo's aansturen. Dit hebben we

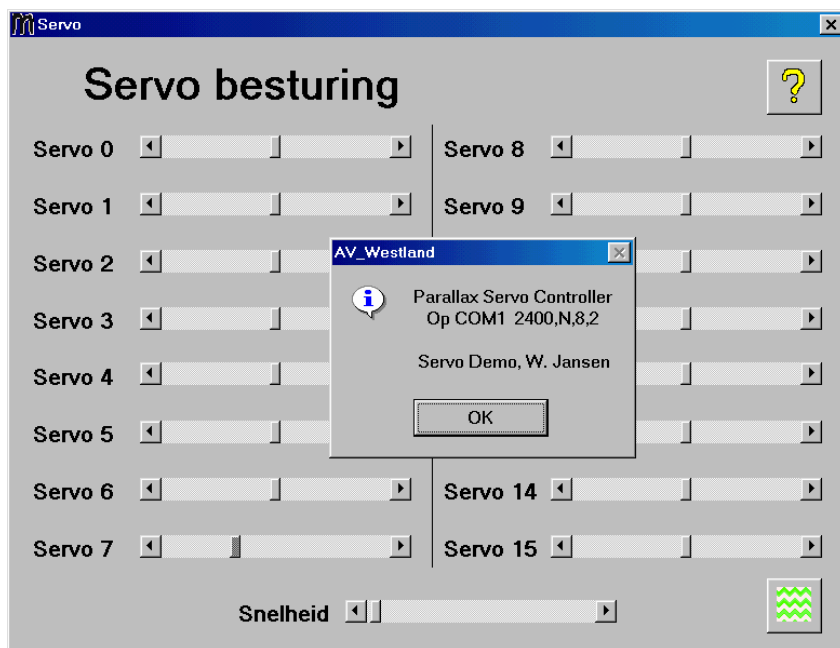
gedaan in het programma servo.exe Dit programma is onder Windhoos98 en XP getest. Als we de Mscomm32.ocx in de c:\windows\system directory zetten en we starten het programma op dan zien we 16 schuifbalken voor de bijbehorende servo's. Verschuiven we de balken dan zullen de bijbehorende servo's ook van stand veranderen. Het programma gaat ervanuit dat de servocontroller op compoort 1 zit.

Als hardware is het enige wat we nog moeten doen is een MAX232 tussen de compoort en de servocontroller zetten. Dit om de +/- 12 Volt van de RS232 netjes om te zetten naar 0 - 5 Volt van de controller logica.

Let er wel op dat de voeding voor de servocontroller via de 3-polige kabel gaat en 5 Volt moet zijn, en dat de voeding voor de servo's op de blauwe kroonsteen linksvoor op de Servo Controller zit. Deze mag tussen 4.8 en 6 Volt zijn, afhankelijk op welke spanning je de servo's wil laten werken.

Team "Westland":

Wim Jansen & Abraham Vreugdenhil.



Agenda

- Zaterdag 1 oktober Bijeenkomst Nieuwegein.
- Zaterdag 5 november Bijeenkomst Nieuwegein
- **25 t/m 27 november HCC dagen Utrecht.**
- Zaterdag 3 december Bijeenkomst Nieuwegein
- Zaterdag 7 januari 2006 Bijeenkomst Nieuwegein

De bijeenkomsten in Nieuwegein worden gehouden in Buurthuis 't Dok, Hoornscheschans 101 3432 TK te Nieuwegein. Voor route beschrijvingen kunt u terecht op onze website.



Microchip PIC microcontrollers
FTDI USB chips
TSOP IR receivers
pager (mini) motortjes
vele bouwpakketjes
LCDs
etc



RoboBits aanbieding: **€1.00** (normaal E 1.90)

NiCad pack 3.6 Volt 400mAh

zie <http://www.voti.nl/winkel/p/M-NICAD-3V6.html> - vermeldt "RoboBits aanbieding 2" - pas geld overmaken als u het verzoek tot betalen krijgt met het aangepaste bedrag - geldig tot de volgende RoboBits uitkomt