



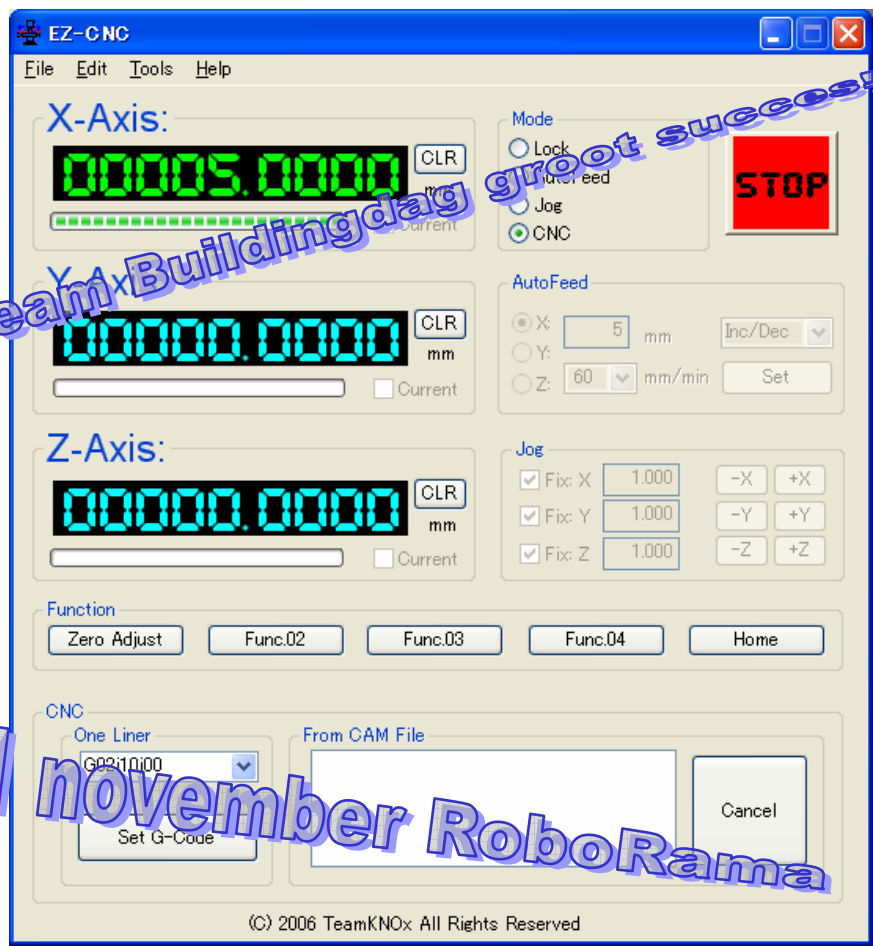
TNT Post
Port betaald
Port Payé
Pays-Bas

ROBO-

BITS-42

Jaargang 11, nummer 3, september 2008

Afz.hcc Robotica gg, p.a. Henk de Gans, Anjerlaan 3, 3871 ev Hoevelaken.



hcc[!]robotica

Robobits is een uitgave van de hcc!robotica gebruikers groep, en wordt vier keer per jaar toegezonden aan de leden. De oplage is ongeveer 490 exemplaren. hcc!robotica is een onderdeel van de hcc! (hobby computer club), een vereniging van bijna 180.000 leden.

=====
===

Redactie adres: H.J. de Gans, Anjerlaan 3, 3871EV Hoevelaken.
henkdegans@kpnplanet.nl Tekst aanleveren in WORD of platte tekst in ASCII.
Afbeeldingen los er bij in JPG, GIF of BMP formaat.

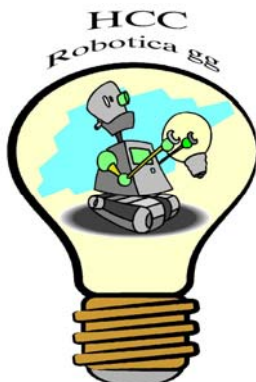
=====
===

Dagelijks bestuur:

Voorzitter: B.T.J.A.Buiskool(Bert), robot@buiskool.net
Technisch adviseur: Ing.H.M.A.van Bodegom(Henny)
ing.h.m.a.van.bodegom@hccnet.nl
Secretaris: A.J.Janssen (Lex) lex.janssen@hccnet.nl
Penningmeester H.J. de Gans(Henk) henkdegans@kpnplanet.nl
Lid: P.Smits(Paul) psmits.1@hccnet.nl
Lid: W.C.de Boer (Wim) wim.deboer@nl.thalesgroup.com

inhouds opgave:

- Bladz. 3 Redactie.
- Bladz 4 Sensor pakket mobiele robot, deel twee!
- Bladz. 9 Verslag RoboRama mei 2008!
- Bladz. 15 4 stappenmotors op PIC 16F877
- Bladz. 20 Robots Industrion!
- Bladz 21 Boek verslag!
- Bladz 22 Team Building Day, verslag Aloys en Rien!
- Bladz 27 Inschrijving RoboRama November 2008!
- Bladz 28 Agenda



REDACTIE

Met een nieuw seizoen voor de deur, ligt hier wederom een nieuwe "4 pagina's dikkere als normaal" Robobits voor u klaar! U vindt hier allereerst het vervolg van het artikel van Ad van der Weiden aan, over sensorpakket mobiele robot. Verder een verslag van de in mei gehouden RoboRama wedstrijd in België. Ook van twee leden een verslag van de voor het eerst gehouden team buildings dag, georganiseerd door RobotMC. Bas Boeteeke is ook weer bezig met de bouw van een nieuwe robot, en wil zijn bevindingen graag met u delen! En Abraham Vreugdenhil heeft een boekverslag geschreven en wil u opmerkzaam maken op een expositie in het Industrion. Geweldig dat weer zovelen hiermee hun bijdrage geleverd hebben, want zonder deze inbreng zou er geen Robobits zijn! Gelukkig is het tot nu toe elke keer gelukt de Robobits gevuld te krijgen met leuke copy, ik hoop ook in de toekomst op jullie inbreng te mogen rekenen.

Ik wens u veel lees en hobby plezier!

Henk de Gans

deadline deadline deadline deadline deadline deadline deadline deadline

De deadline voor kopie voor Robobits 43 die eind december 2008 zal uitkomen, is gesteld op woensdag avond 10 december 2008!!!! Maar u hoeft natuurlijk niet zo lang te wachten, nu al insturen kan natuurlijk ook!

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 boek bespreking (of tijdschrift).

deadline deadline deadline deadline deadline deadline deadline deadline

Kijk ook eens op onze website:

WWW.HCCROBOTICA.NL

september 2008 3

Vervolg artikel RoboBits 41

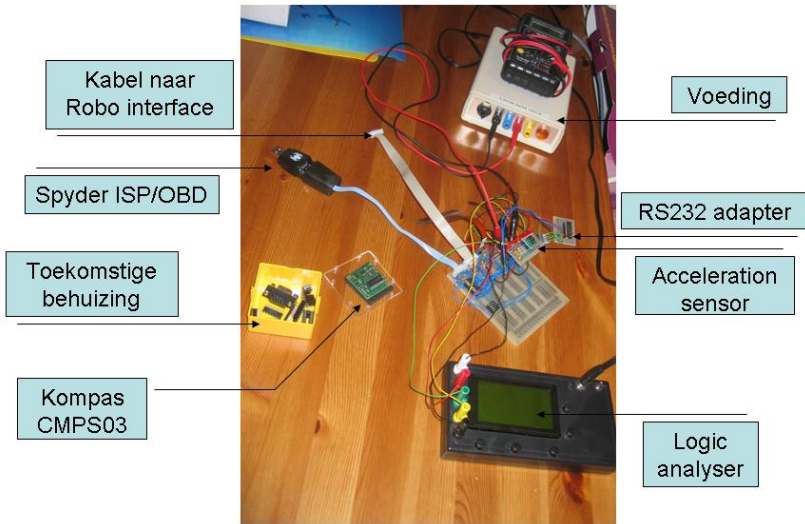
Door: Ad van der Weiden

Kompas

Het gebruik van het kompas is eigenlijk vrij recht-toe-recht-aan. Het enige hardware probleem is dat het kompas op een 5V I2C bus aangesloten moet worden en de rest van de sensoren op een 3,3V bus, gelukkig bracht een Application Note van Philips [8] uitkomst. Een andere Application Note van Philips [5] geeft inzicht in de processing die nodig is om tot zinvolle waarden te komen. Gelukkig voor ons is het meeste hiervan al in de processor op de CMPS03 module gestopt. Wel is het een voorwaarde dat het kompas absoluut horizontaal wordt gehouden omdat anders de compensatie voor de inclinatie (of dip) niet meer klopt. In de AN staat een methode beschreven om als men de tilt van het kompas kent, de aanwijzing te compenseren. Dankzij de versnellingsensor kennen we de roll en de pitch dus het begin is er. Volgens de AN moeten we echter over alle drie de componenten van het magnetisch veld beschikken en dat is wel een probleem. Ten eerste heeft de CMPS03 maar twee sensoren en de meetwaarden daarvan zijn beschikbaar in registers 4-7 maar het is niet duidelijk wat die waarden precies betekenen. Het tweede probleem is het gebrek aan de z-as magneet sensor. Intuïtief zou je zeggen dat er toch ruim voldoende informatie is. Van de magnetisch veld vector is voor een bepaalde plaats op aarde alles bekend, de stand van het kompas t.o.v. de vertikaal is bekend, alleen de orientatie (azimuth) ontbreekt dus nog, dat zou met 1 of 2 sensoren toch moeten kunnen.

Ik ben gelukkig niet de eerste die dit bedenkt en na enig zoekwerk vond ik een aantal oplossingen:

- Het kompas calibreren voor een aantal tilt hoeken en met behulp van de calibratietabel de azimuth compenseren.
- Filteren van de azimuth gebaseerd op het gegeven dat de tilt een ruis karakter heeft
- Het schatten van de ontbekende veldcomponent gebaseerd op het feit dat totale veldsterkte constant is en de andere twee componenten worden gemeten.



Proefopstelling

Ik heb gekozen voor de laatste oplossing omdat die vrij nauwkeurig is omschreven in patent US6836971 en in het artikel van Cho en Park [7]. Of de gebruikte formules helemaal equivalent zijn heb ik niet gecontroleerd. Ik heb [7] gebruikt omdat daar duidelijker instaat hoe ze tot de formules zijn gekomen en ook de locale inclinatie expliciet wordt gebruikt.

We beginnen met het schatten van de magnetische z-component uit de overige beschikbare gegevens (inclinatie λ , roll ϕ , pitch θ en de magnetisch veldcomponenten X en Y).

$$\hat{Z} = \frac{\sin \lambda + X \sin \theta - Y \cos \theta \sin \phi}{\cos \theta \cos \phi}$$

Met behulp van deze geschatte waarde kunnen nu gecompenseerde waarden voor de azimuth componenten worden berekend

Azimuth = $\arctan(b/a)$ met:

$$a = X \cos \theta + Y \sin \theta \cos \phi + \hat{Z} \sin \theta \cos \phi$$

$$b = -Y \cos \phi + \hat{Z} \sin \phi$$

Net als bij de berekening van de tilt hoeken doemen weer schrikbeelden op van `<math.h>` en IEEE floating point berekeningen. Gelukkig brengt ook hier CORDIC uitkomst. Na wat manipulaties met de formules blijkt dat we de hele berekening in slechts 7 CORDIC operaties kunnen uitvoeren, 5 voor sinus en cosinus berekeningen, 1 voor compensatie van de CORDIC gain en 1 voor de arctangens. Voor het berekenen van de sinus en cosinus gebruiken we de CORDIC in rotation mode (zie [4]). In rotation mode wordt de ingangsvector gedraaid over de opgegeven hoek, deze hoek wordt in het proces naar 0 geregeld.

```
void rotation(int &x, int &y, int &z)
{ for (int i = 0; i < ITERATIONS; i++)
  if (z < 0)
    { int t = x;
      x += y>>i;
      y -= t>>i;
      z += term[i];
    }
  else
    { int t = x;
      x -= y>>i;
      y += t>>i;
      z -= term[i];
    }
}
```

De hoek z moet wel tussen plus en min 90 graden liggen. Als je de rotation met de vectoring vergelijkt zie je dat het verschil eigenlijk alleen de conditie van het if statement is. Om de code leesbaar te houden heb ik functies gemaakt voor elk specifiek gebruik van CORDIC.

```
int cordic_cos(int x, int arg) // x*sin(arg)
{ rotation(x, 0, arg);
  return x;
}
```

```
int cordic_sin(int x, int arg) // x*cos(arg)
{ int t = 0;
  rotation(x, t, arg);
  return t;
}
```

```
void cordic_gain(int *x, int *y) // x*=1,647 y*=1,647
{ rotation(*x, *y, 0);
}
```

```
int cordic_arctan(int x, int y) // arctan(y/x)
```

6 **ROBOBITS**

```

{ int phase = 0;
  vectoring(x, y, phase);
  return phase;
}

```

De uiteindelijke berekening voor de tilt compensatie wordt dan:

```

int tiltcomp(int x, int y, int p, int r, int d)
{ //unity scaled to 10000
  int sl = 10000*sin(d*M_PI/180.0); //sin lambda (tilt or dip angle)
  y = cordic_cos(y, p);
  int xt = x;
  cordic_gain(&xt, &y);
  int a = cordic_cos(xt + cordic_sin(sl, p), r);
  int b = cordic_sin(cordic_sin(x, p) + sl*fact, r) - y;
  return cordic_arctan(a, b);
}

```

In de bovenstaande functie komen nog twee berekeningen voor die door constanten moeten worden vervangen, dat zijn de berekening van $\sin \lambda$ en de vermenigvuldiging met 'fact', fact is de CORDIC gain (1,647). Na die wijziging bevat de gehele procedure alleen nog maar optellen en aftrekken en schuifoperaties.

Conclusies

Hoewel nog lang niet af, zijn de meeste componenten voor het sensorpakket beschikbaar, zowel in hardware als in software. Op een experimenteerprint heb ik de aansturing van de I2C bus naar LCD en EEPROM, de SPI bus naar de RoboInterface en de RS232 aansluiting voor elkaar. Ook de versnellingsensor is aangesloten via een verloopprintje. De software voor de aansturing van de bussen en de LCD is ook klaar en getest. Momenteel ben ik bezig met de commando afhandeling (commando's kunnen komen van FischerTechnik (via SPI of RS232) of van de computer (via RS232). Naar aanleiding van de commando's worden dan gegevens verstuurd naar de SPI of RS232 bus of naar het LCD. De data acquisitie is klaar (sommige parameters als sample frequentie, middeling en discriminatievenster moeten nog definitief bepaald worden). Zoals hierboven te zien is heb ik ook de code voor de data verwerking al gereed. Daarnaast ben ik nog bezig met een printontwerp in Eagle. Zodra alles gereed is zal ik alles op mijn website publiceren of eventueel een aanvullend artikel insturen.

Het berekenen van snelheid en positie uit de versnelling blijkt geen sinecure. Een zoektocht op internet levert vooral op dat anderen er ook veel problemen mee hebben. Het probleem is dat zelfs vrij kleine fouten enorm cumuleren en dat de robot daardoor ogenschijnlijk binnen enkele seconden al meters is weggereden. Ik werd dus gedwongen te kijken naar de oorzaken van deze fouten.

1. Ruis; deze is afhankelijk van de bandbreedte, in mijn geval ca. 100Hz; de RMS ruis is in dat geval ca. 3,5 mg.
2. Temperatuur drift; deze verschilt per as en is in het slechtste geval 6mg/°C.
3. Tilt; de bijdrage van de zwaartekracht aan de horizontale versnelling bedraagt 17mg/°.
4. ADC fouten; deze zijn gespecificeerd als 2LSB, omgerekend is dat ca. 8mg.

De ruis en de ADC fouten zijn toevallige fouten met gemiddelde 0. De bijdrage aan de integratie zou daarom beperkt moeten zijn. De andere fouten zijn systematische fouten en zullen vermeden moeten worden (tilt) of er moet voor gecompenseerd worden (beide). De temperatuurdift is misschien nog wel het eenvoudigst, ik ben van plan om binnenkort een temperatuursensor toe te voegen. Verder zal verbetering toch moeten komen uit een slimmere signaalverwerking, als iemand een goed idee heeft dan hou ik me van harte aanbevelen.

Referenties

- [1] Accelerometer, Elektor april 2007
- [2] Freescale AN3461, Tilt Sensing Using Linear Accelerometers
- [3] Freescale AN3397, Implementing Positioning Algorithms Using Accelerometers
- [4] Ray Andraka, A survey of CORDIC algorithms for FPGA based computers
- [5] Philips AN00022, Electronic Compass Design using KMZ51 and KMZ52
- [6] Devantech, CMPS03 – Robot Compass Module, datasheet
- [7] Cho and Park, Electronic Letters 30-10-2003, Vol.39 No.22, Tilt compensation algorithm for 2-axis magnetic compass.
- [8] Philips AN97055, Bi-directional level shifter for I²C-bus and other systems

hcc[®]robotica



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

robot mc



<http://www.robotmc.org>



Zaterdag 17 mei 2008 - Na een mooie zonnige week begint de dag wat druilerig en koud. Om 10u hebben we afspraak in het De Nayer Instituut in Sint Katelijne Waver om alles in gereedheid te brengen. We hebben afspraak in de garage waar men proefbanken heeft voor auto's en vrachtwagens. Sinds september 2004 beschikt Campus De Nayer over een ultramodern autotechnologisch opleidingscentrum. Met een totale vloeroppervlakte van 3.000 m², die hoofdzakelijk wordt ingenomen door laboratoria en testruimten, is dit het grootste autotechnologisch opleidingscentrum in België. Deze wordt 364 dagen per jaar gebruikt door studenten om hun bachelor in autotechniek te ondersteunen. Ze doen er de analyses van verkeersongevallen en schade expertise, bestuderen er motortechiek en bedrijfsvoertuigen. Maar die ene dag staat alles in het teken van de Opendeurdag en wordt alles in gereedheid gebracht om het grote publiek te ontvangen.



De Deelnemers!

Wij hebben er een ruime plek gekregen om er onze jaarlijkse Roborama te organiseren. Tussen 10 en 12 wordt alles netjes en veilig opgebouwd en leggen de deelnemers de laatste hand aan hun hardware en software. Om 12u gaat de eerste ploeg naar de kantine voor een heerlijke lunch : frietjes met kippepastei of een koude schotel. Om 12u30 is het de beurt aan de tweede shift.



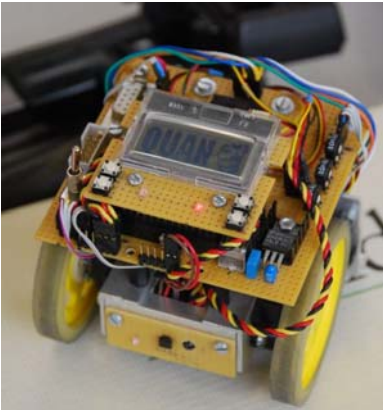
Om 13u30 is het dan zover en beginnen we met de wedstrijd. Het eerste onderdeel is "**Heen en Weer**". 16 Robots nemen deel : Cerberus, Dominator 2, JoJoBot, Occy, NXTruck, Charlie, NaXiTi, Rond, RP6, Multicram V2, Matchbox, Colafinder, Cram-WII, Ro_Bert_A3, T-Rex, Non-RC-bot. Het zijn de nieuwelingen die de prijzen

wegkopen. Alhoewel de Matchbox van Marc Huyghe de snelste tijd neerzet, kan hij de opdracht niet foutloos uitvoeren en is het Toon Peters die wint met zijn '**Non-RC-Bot**', een R/C auto die werd omgebouwd met een Arduino als controller. In 8,81sec legt hij het parcours foutloos af. Proficiat, Toon ! Opvallend is dat de Lego robots het zeer goed doen : van de eerste 5 zijn er 4 Lego Mindstorms NXT robots.

Daarna leggen we het iets hoger en wordt er "T-Tijd" gereden. 11 Robots nemen hieraan deel en alweer is het één van de nieuwelingen die met de eer gaat lopen. Ben Fondu wint de proef met een omgebouwde Lego Technics Truck, de **NXTruck**. Slechts 22,93 seconden heeft hij nodig om deze opdracht af te werken. Ook nu valt het op dat de Lego Mindstorms NXT robots het zeer goed doen. We vinden vier NXT's in de top 5.

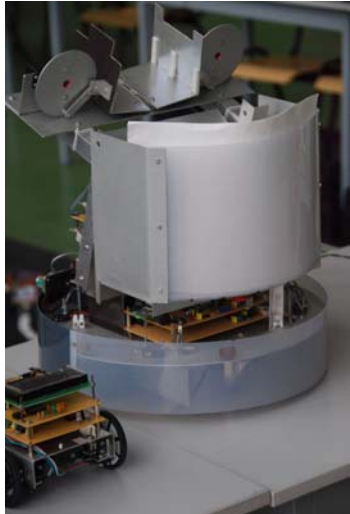
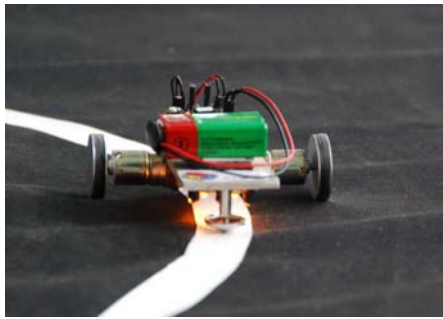


We zitten wat voor op schema en lassen een half uurtje pauze in zodat de nodige natjes en droogjes kunnen gehaald en verorberd worden. Ondertussen is het al gezellig druk en kan Roborama op heel wat interesse van de bezoekers rekenen. Als de motoren van de echte auto's echter op de testbank gezet worden, zorgt hun oorverdovend lawaai dat niemand nog wat kan verstaan. Op een bepaald moment vult de garage zich met een wit-blauwe wolk : jawel, men heeft de motor van de testwagen opgeblazen. Spectaculair, dat wel, maar misschien niet wat gehoopt was door de organisatoren ☺.



Om 15u gaan we verder met de wedstrijden. Nu is het tijd voor de "Mini-Sumo". Een beetje tegen de verwachting in zijn er 'slechts' 6 deelnemers aan deze wedstrijd. We hadden gedacht dat HCC!Robotica hier voltallig zou aan deelnemen gezien het feit dat er vorig jaar toch 14 mini-sumo's gebouwd werden. Niet getreurd. De robots werden gewogen en gemeten en de jury was onverbiddelijk : 500gr en 10x10cm. De 6 deelnemers voldeden allen en zorgden voor de nodige spanning. Ook dit jaar bleek Quan te sterk voor de competitie en dus werd de verdiende winnaar 'Quan' van Robert Moro, gevolgd door TinMan2, Parabola en BB.

Om kwart voor vier is het tijd voor het **Lijnvolgen**. 14 Robots nemen eraan deel. Je ziet dat er verschillende robots deelnemen : diegene die vooral goed zijn in Heen en Weer tot en met diegene die duidelijk gebouwd zijn om blikken te vangen, maar tegen de speciaal voor deze discipline gebouwde **Cramliner PVC** van Marc Huyghe is niemand opgewassen. In



een luttele 18,63 seconden heeft hij zich rond de lijn gewerkt. Nu zijn het duidelijk de zelfbouw robots die de sterkere zijn in deze discipline. De eerste Lego NXT vinden we op de vierde plaats.

Tenslotte begint om kwart over vier het koninginnestuk : **“blikken vangen”**. Dat dit geen eenvoudige opdracht is bewijst het feit dat er zich maar vier robots aandienen : ‘Rond’ die de 6 blikjes netjes verzameld in z’n opvangbak om dan terug te keren naar de start, AIRobot die er wel 6 herkent, maar er slechts 5 kan opvangen en ze in z’n netje gooit

om dan naar de start terug te gaan. Colafinder die puur op beeldherkenning werkt. Deze vindt 5 blikken waarvan er 3 kunnen teruggebracht worden en tenslotte de nieuwkomer, Graayer, die iedereen verbaasde door z’n efficiëntie en originaliteit. Deze vond vier blikken, draaide zich dan om en er kwam een grijparm uit de robot die het blik vastnam en terugbracht. De winnaar van deze discipline werd verdiend ‘**Rond**’ die in 191 seconden de 6 blikken gevonden en teruggebracht had. Proficiat, Marc Robberecht!

Daarna ging de jury zich beraadslagen. Ondertussen werden er door een aantal robots een demonstratie gegeven over een *–mogelijk–* toekomstige discipline : slommen. Hier werden door een aantal mensen een demootje gegeven met wisselend succes. Wat eenvoudig lijkt (tussen drie blikken slalom rijden en terugkeren) blijkt toch niet zo triviaal als wel gedacht. Een leuk idee dat nog wat moet bijgeschaafd worden maar duidelijk mogelijkheden biedt ! Enfin, Uiteindelijk kon men een eindwinnaar aanduiden voor minisumo : **Quan van Robert Moro** en voor Roborama : **Alrobot**

van Aloys Verstraeten. Hierna volgde de prijs uitreiking. Dankzij onze sponsors (Hogeschool De Nayer, VOTI en Lego) hadden we voor elke deelnemer ook een kleine maar leuke prijs voorzien.

De winnaars :



vlnr : Aloys Verstraeten, Marc Robberecht, Ben Fondu, Robert Moro.
[Ontbreken op de foto : Toon Peters en Marc Huyghe]

Om 17u30 begonnen we alles op te ruimen en daarna gingen we nog een frisse pint pakken in de studentenkroeg. We stelden vast dat het ondertussen gestopt was met miezeren. Een mooi einde voor een mooie dag : we hadden ons goed geamuseerd en weeral wat bijgeleerd. Ik vermoed dat op de terugweg iedereen al plannen aan het maken was om volgende keer nog beter te scoren.

Mij rest niet anders dan de juryleden te bedanken voor hun inzet, de fotograaf / cameraman voor de mooie herinneringen, alle deelnemers voor hun enthousiasme en uiteraard onze sponsors om dit evenement mogelijk te maken.

HOGESCHOOL VOOR WETENSCHAP & KUNST | **DE NAYER INSTITUUT**

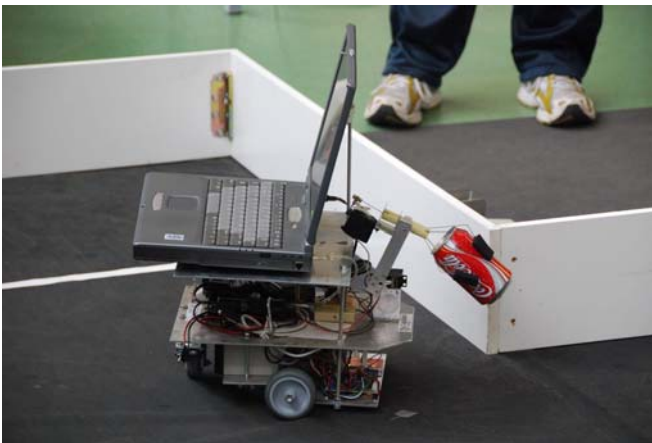


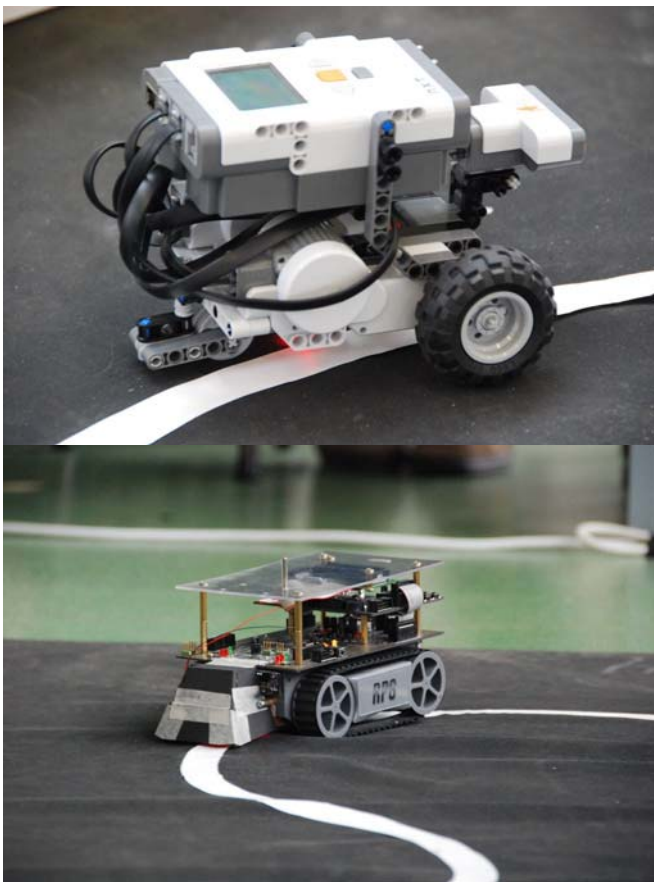
MINDSTORMS



Tot de volgende editie !

Patrick De Wachter
Voorzitter RobotMC

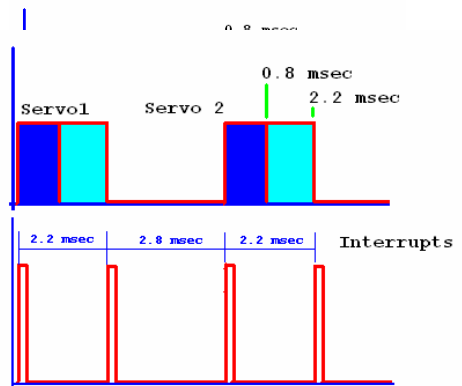




4 servo's op PIC16F877

Voor mijn nieuwe robot, die is uitgerust met twee servomotoren voor de aandrijving en twee servo's waarop een Sharp GP2D120 komt te staan om de voorkant af te scannen. Voor het aansturen van de servo's kan je twee technieken gebruiken:

- Je creëert voor iedere servo een wachtlus tussen de 0.8 en 2.2 msec en zorgt ervoor dat je na iedere 5 msec een andere servo aanstuurt. Om die interval van 5 msec te bereiken maak je meestal gebruik van een interrupt. Door het gebruik van



sep

wachtlussen (de computer kan dus niets anders doen) gebruik je tussen de 16 en 44 % van de CPU tijd.

- Je creëert een variabele interrupt die correspondeert met de tijd dat de poort hoog moet staan en een variabele interrupt die correspondeert met de tijd dat de poort laag moet staan. Wanneer we een variabele interrupt gebruiken kunnen we deze tijd reduceren tot minder dan 2%.

Hoe gaat dat in zijn werk op een PIC16F877?

Deze PIC heeft een 16 bits Timer (TMR1) die beschikt over een 16 bits preload-adres (TMR1HL). Op dit adres TMR1H (0Fh) en TMR1L (0Ehex) kunnen we een 16 bits getal zetten met een waarde die tussen 0000 en FFFFhex ligt. De clockpuls van de PIC verhoogt TMR1HL met de waarde 1. Als TMR1HL de waarde FFFFhex+1 bereikt dan wordt er een overflow vlag gezet en zal (indien deze is opgegeven) de routine op het interrupt adres worden uitgevoerd. Wanneer de frequentie van de clockpuls zeer hoog is kan het vullen van TMR1HL vertraagt worden door het instellen van een Prescale. (zie hiervoor blz:51 van PIC16F877) Voor een PIC16F877 met een cristal van 10MHz hebben we geen prescale nodig. We kunnen dus zeer nauwkeurig te werk gaan zonder frequentie deling van TMR1.

Een rekenkundig voorbeeld:

Stel: we zetten in TMR1HL de waarde 0000hex. Na het starten van de timer, TMR1ON=1, vindt er na 6.5 msec. een overflow plaatst. De CPU heeft dus 6.5 msec. nodig gehad om TMR1HL van 0000h te verhogen naar 10000h, dat komt neer op ong. 0.4 msec. per 100hex tikken. **Stel:** We moeten in dit voorbeeld een pulsduur, beetje hoog, instellen van 0.8 msec. **Vraag:** Wat is de waarde die we in TMR1HL moeten plaatsen?

Oplossing: Als we in TMR1 zetten 1000h - (2x100h) = FE00h dan zal er na ong. 0.8 msec een overflow plaatsvinden en onze interrupt routine worden aangeroepen.

Dit zelfde kunnen we doen voor de resterende tijd, beetje laag, (5 - 0.8 = 4.2 msec.) waarop de tweede servo bekrachtigd moet worden. Als we hetzelfde reken voorbeeld menen van hierboven moeten we 1000 - 1050 = EFB0h in TMR1HL plaatsen.

Hoe hebben we dit in mikroPascal opgelost?

```
Program Servo_4;  
Var ServoStanden : Array[1..4] of Byte; // Servo stand  
ServoActive : Array[1..4] of Boolean; // Gebruiken we deze  
ServoCorrect : Array[1..4] of Byte; // Correctie
```

```

ISRDummyW : Word; // variable binnen interrupt
i : Byte; // Algemene variable

Procedure interrupt;
Const HighConst : WORD = 63650; // Stnd. waarde HOGE bit ALLE servo's
LowConst : WORD = 55250; // idem LAGE bit
InActiveTime : WORD = 10600; // idem LAGE bit
Var Servonr : Byte; // Bereken Servonummer
TimerTijd : Word; // Tijd die in de Timer wordt gestopt.
Begin
If PIR1.TMR1IF = 1 // Timer1 overflow
then Begin
T1CON.TMR1ON := 0; // Turn Timer1 OFF
Servonr:=(ServoStage Div 2)+1; // 0 Div 2 = 0+1;
// 1 Div 2 = 0+1
// etc.
If ServoActive[Servonr] // Heeft Servo een PIN
then Begin // dan HOOG en laag zetten
If(ServoStage Mod 2)=0 // Als stage is "even"
then Begin // Bereken tijdsduur pin HOOG
ISRDummyW:=ServoStand[Servonr]+ServoCorrectie[ServoNr];
TimerTijd:=HighConst-ISRDummyW; // Verlaag constante
If Servonr=1 then PortC.1:=1; // Poort HOOG
If Servonr=2 then PortC.2:=1;
If Servonr=3 then PortC.3:=1;
If Servonr=4 then PortC.4:=1;
End
else Begin // Bereken tijdsduur en zet pin laag
TimerTijd:=LowConst+ISRDummyW; // Verhoog LowConstante
If Servonr=1 then PortC.1:=0;
If Servonr=2 then PortC.2:=0;
If Servonr=3 then PortC.3:=0;
If Servonr=4 then PortC.4:=0;
End;
End
else Begin
TimerTijd := HighConst-InActiveTime; // intervaltijd servo's
Inc(ServoStage); // Sla lage tijd over.
End;
// Zet de berekende interrupttijd in register.
TMR1H := HI(TimerTijd); // Hoge 8bits van 16bitswaarde
TMR1L := LO(TimerTijd); // Lage 8bits
// Verhoog teller voor volgende Stage.
Inc(ServoStage); // Volgende, maar niet groter dan 15.
If ServoStage>7 then ServoStage:=0;
// Reset interrupt vlaggen.
PIR1.TMR1IF := 0; //Reset Timer1 interrupt flag
T1CON.TMR1ON := 1; //Turn Timer1 ON
End;

Begin
TrisC := 0; // zet poort C (0=Output, 1=Input)
INTCON.PEIE :=1; // Peripheral interrupt enable (blz: 20)
INTCON.GIE :=1; // Global interrupt enable bit. (blz: 20)

```

```

PIR1.TMR1IF := 0; // TMR1 interrupt flag (blz: 22)
T1CON.TMR1ON := 1; // Turn Timer1 ON (blz: 51)

For i := 1 To 4 Do ServoStand[i] :=45; // De stand
For i := 1 To 4 Do ServoActive[i]:=True; // Alle gebruiken
While True Do // eindeloze lus maken
Begin End; // Doe helemaal niets ??????
End.

```

Ik heb hierboven een kort pascal programma weergegeven die alle instellingen laat zien. Waar het hier omgaat is wat we in de interrupt routine stoppen om de 4 servo's continu aan te sturen. (voor instellingen registers, zie bladzijde nummers in het PIC16F87x manual) We zullen hier alleen de interrupt routine doorlopen en uitleg geven waar nodig. In de interrupt routine worden 4 servo's aangestuurd met een interval van 20 msec. Dat betekent dat er iedere 5 msec. „n servo moet worden afgehandeld. Omdat we voor iedere servo twee handelingen moeten verrichten onderhoud de interrupt een teller (ServoStage) die binnen de interrupt verhoogt wordt. Als ServoStage de waarde 8 heeft bereikt dan wordt hij op 0 gezet. Aan de hand van ServoStage bepalen we welk servonummer, ServoNr, we behandelen en welke stand, vast gelegd in ServoStand[x], die servo moet aannemen. We slaan ServoStand[x] op in ISRDummyW, zodat de voor ServoStage en ServoStage+1 gebruikt wordt. In theorie zou de gebruiker als de pin hoog is de stand van de servo kunnen wijzigen, waardoor onze intervallen worden verstoord !!! Een grapje, ingebouwd in deze interrupt routine, is dat we een servo UIT kunnen zetten. Dat lijkt vreemd, maar voor mijn robots wel heel makkelijk. Een servo wordt uitgezet als de sensor in de goede richting staat of als de motorservo's stil moeten staan. We voorkomen zo het jutteren en dus onnodig stroomverbruik.

We zien dat de TMR1ON in het begin wordt uitgezet. Dit betekent dat TMR1HL niet meer verhoogd wordt. We moeten met de stilstand tijd rekening houden en er voorzorgen dat de tijdsduur tussen TMR1ON:=0 en TMR1ON:=1, ongeacht de doorlopen IF's gelijk is.



Zodra dat niet het geval is moeten we met NOP's dit zien te compenseren. In de praktijk bleek dat hier de lussen redelijk gelijk zijn en dat we er geen extra zorg aan hoefde te besteden.

Hoe de waarden HighConst, LowConst en InActiveTime bepalen? Je hebt daar eigenlijk een scope voor nodig omdat goed te kunnen afregelen. Als eerste beginnen we met de waarde

HighConst die er voor zorgt dat de Servo pin HOOG staat. In de variabele ServoStand[1] en ServoCorrectie[1] zetten we de waarde 0, hetgeen betekent dat onze computer het volgende uit moet rekenen:

```
ISRDummyW:=ServoStand[ServoNr]+ServoCorrectie[ServoNr];  
TimerTijd:=HighConst-ISRDummyW; // Verlaag constante
```

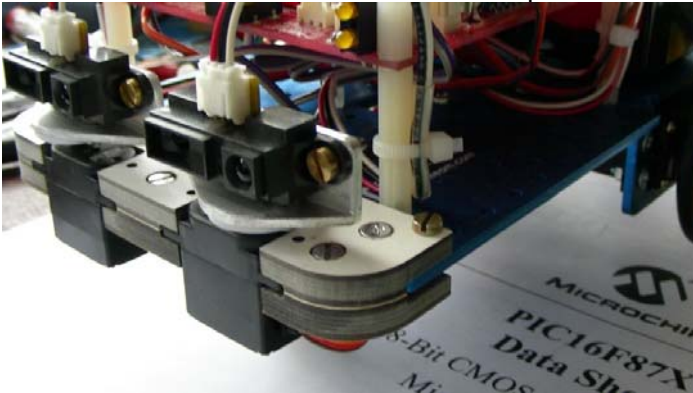
Dit kost een bepaalde tijd en die moet telkens gelijk zijn anders krijgen we tijdsafwijkingen. Nu we de waarde, TimerTijd, weten gaan we de pin HOOG zetten door alle IF's uit te voeren. Ook hier geldt dat we ervoor moeten zorgen dat het iedere keer evenveel tijd kost om alle vergelijkingen te doorlopen. Als het beetje HOOG staat zetten we de waarde TimerTijd in het register TMR1HL (lees dit als TMR1H en TMR1L). We verhogen daarna de Stage counter en zetten de TMR1 weer aan na het restten van de overflow vlag. We zijn nu klaar en de computer kan weer verder gaan met zijn activiteit voor de interrupt.

Op de scope moeten we nu voor servo 1 een pin HOOG zien van 2.2 msec. Als die tijd korter is dan zal de waarde HighConst verlaagd moeten worden. Dat is een vervelend klusje, want we moeten weer compileren, uploaden, testen, wijzigen, uploaden etc. Wanneer we 2.2 msec hebben is de waarde van HighConst bekend. We gaan nu op dezelfde wijze LowConst doen, maar daarvoor moeten we eerst alle servo's op dezelfde pinnummer (bijv. PortC.1) zetten. De interval moet nu 5 mSec zijn. Wanneer deze lager dan 5 msec is dan moeten we LowConst weer verlagen. Is de interval groter dan moeten we LowHigh verhogen. Als we ook hiermee klaar zijn en de pinnen weer verwijzen naar verschillende servo's dan rest ons nog een afregel klusje voor de InActieveTime. We meten weer de eerste servo en schakelen ServoActive[2] uit. De interval moet 20msec. worden en daarvoor mogen we alleen InActiveTime variabele wijzigen, want High en Low waren al correct afgeregeld.

Laatste klusje.

Wat is de maximale waarde die in ServoStand[x] mag komen te staan? Om dat te bepalen gaan we gelijk aan de hierboven beschreven empirische werkwijze te werk. We sluiten de scope weer aan op Servo 1 en hebben, als ServoStand[1] en ServoCorrectie[1] 0 zijn een pulsduur van 2.2 msec staan. Wanneer we het getal in ServoStand[1] met 100 verhogen zal de pulsduur teruglopen. We

gaan net zolang door met verhogen tot dat de pulsduur 0.8 msec is. We hebben nu de maximale waarde bepaald.



Succes en mochten er vragen zijn dan zal ik die beantwoorden op onze bijeenkomsten.

Bas Boetekees.

Robots Industrion!

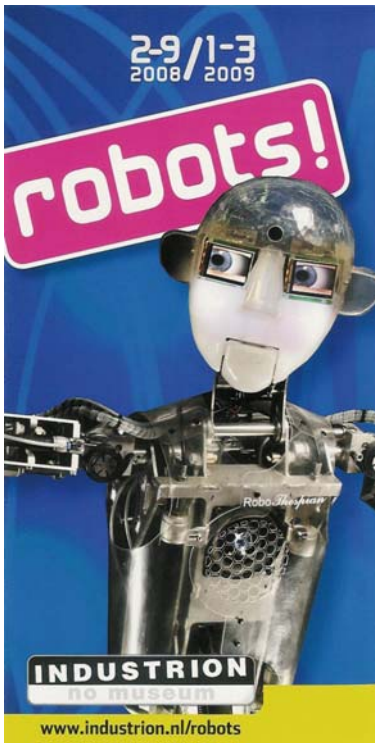
Robots!

Gigantische machines die de wereld zullen veroveren of handige helpers in het leven van alledag? Robots zweven in onze gedachtenwereld ergens tussen fantastische sprookjeswereld en technologische werkelijkheid in. Natuurlijk zijn robots allang onderdeel van ons dagelijks bestaan. Zo gebruiken we robots al tijden in de productielijnen van fabrieken en voeren zij taken uit die voor de mens te moeilijk of te gevaarlijk zijn. Eén ding is zeker: robots spreken al vele jaren tot de verbeelding.

Vanaf 2 september 2008 toont Industries aan de hand van talloze unieke objecten, welke rol robots in onze samenleving speelden en spelen. Schattige speelgoedrobotjes, futuristische blikken mannen uit de jaren '50 van de vorige eeuw en moderne industriële en huishoudrobots zijn te bewonderen.

Ga zelf aan de slag met robottechnologie tijdens de verschillende robotworkshops! En welke rol spelen robots in de toekomst? Ontdek het van 2 september 2008 t/m 1 maart 2009 bij Industrion!

Welkom in de blauwe zone



Kom binnen en neem een kijkje in de Robotfabriek! In déze spannende ruimte wachten robots in alle soorten en maten, badend in het blauwe licht, op jou! Ontdek de meest uiteenlopende robots uit het verleden, heden én toekomst; van industriële robotarmen tot speelgoedrobots en van de eerste 'stalen robot-acteurs' uit films uit de jaren '50 tot de robot uit een bekende videoclip van de wereldberoemde zangeres Björk. Laatje verwonderen door diverse robot- en kinetische kunstwerken, ontsproten uit het brein van kunstenaars die de roestwerend als inspiratiebron hebben gebruikt. In de robot-ontdekzone ga je zelf aan de slag met Robottechniek; creëer eigen 'scraprobot' uit allerlei onderdelen, brugje eigen Robot-game, en nog veel meer!

Ontdek ook de nieuwe 3D film van Industries '3D mania'; een reis door de derde dimensie. Volg de avonturen van de professor en zijn robot-assistent Max.

Robothespian zegt: hallo!

Alle robots in de expositie worden in de gaten gehouden door de vriendelijke, digitale ogen van Robothespian@. Deze menshoge robot is volledig interactief en een geboren entertainer. Sta er niet raar van te kijken als Robothespian@ opeens een show weggeeft. Deze robot praat en reageert op het publiek, doet een dans en speelt zelfs scènes uit bekende films na! Een robot-entertainer in hart en schroeven! Robothespian is onderdeel van Engineered Arts Limited.

Industrion
Museumplein 2
6461 MA Kerkrade
045-5670809
www.industrion.nl/robots

redactie: met dank aan Abraham Vreugdenhil.

Boek/Blad bespreking

Bas Haring
De ijzeren
wil
over bewustzijn, het brein
en denkende machines



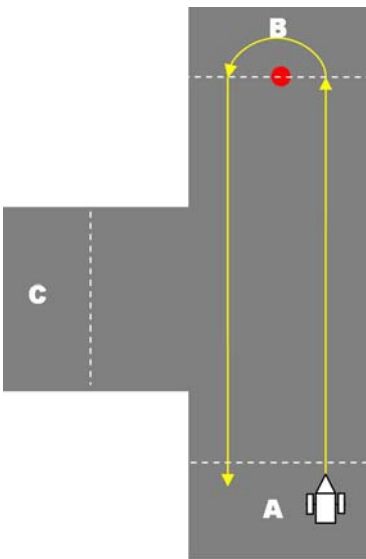
Bas Haring, De ijzeren wil,
ISBN 978 90 467 0341 0
Uitgeverij pandora, @2003, 2007

Dat computers intelligente machines zijn merken we dagelijks; en konijnen - of dieren in het algemeen - zijn meestal ook niet op hun achterhoofd gevallen. Toch lijkt het alsof er een speciaal plekje is voor de mens, de intelligentste van allemaal. Veel mensen hebben het gevoel dat wij iets extra's hebben ten opzichte van dieren en machines: een ziel, bewustzijn, of een wil. Maar is dit gevoel terecht? De ijzeren wil gaat over kunstmatige Intelligentie en over de filosofische vragen die daaraan verbonden zijn, De ijzeren wil geeft niet per se antwoorden, maar laat wel zien hoe je op een Intelligente manier met dit soort vragen om kan gaan. Speels, en met voorbeelden uit het dagelijks leven.

Bas Haring is in het boek "de ijzeren wil" op een mooie manier erin geslaagd om de zaken rond intelligente machines duidelijk te maken. Wat is intelligentie en hoe kan je dat wel of niet bij robot's en machiene's voorstellen. Voor de robotbouwer is dit boek zeker een aanrader. Dit boek is ook al 2^e hands te koop bij ondermeer de Slegte of bol.com. Voor 2.50 euro ben ik het al tegen gekomen.

Abraham Vreugdenhil.

Team Building Day!



ROBOTMC ORGANIZES THE

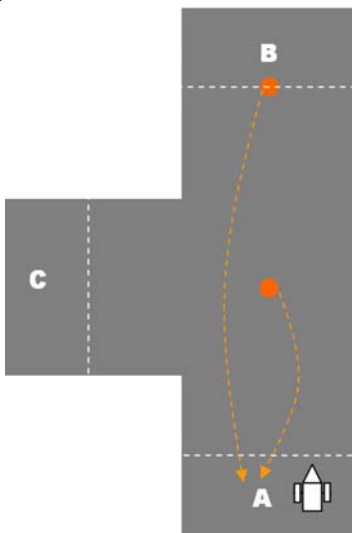
Team BUILD

SEPTEMBER 20TH

DE NAYER INSTITUUT, SINT K

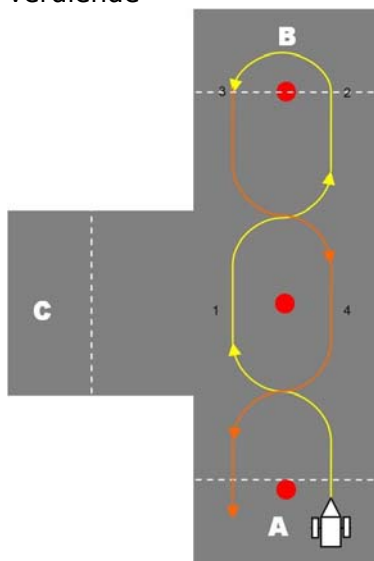
Afgelopen zaterdag was het dan zo ver, de eerste „Team Building Day“. Het idee kwam van Jildert De Wachter na de Roborama wedstrijd van Mei in Sint Katelijne Waver. Het idee om was om in

een dag een robot te bouwen en een of meerdere opdrachten uit te voeren. Dat betekent dat je de opdracht(en) s'morgens krijgt te horen en dat in de namiddag het zaakje moet werken. Dat het idee van een jong persoon komt kan je wel raden, die zijn opgegroeid met lego en snelle computerspellen. Het zal je ook niet verwonderen dat de al wat meer ervaren robotbouwer daar met gefronste wenkbrauwen op reageerden. Die hebben waarschijnlijk al meerdere meters op hun knieën achter hun robot gekropen om te voorkomen dat de robot tegen de muur of tafelpoot aanknalt. Dat het al moeilijk is om een van tevoren gekende opdrachten uit te voeren blijkt wel uit het feit dat na een paar jaar nog maar enkelen het aandurven om de vier Roborama opdrachten uit te voeren. Zelf was ik ook niet zinnens om mee te doen, niet om de eventuele constructieve aanpassingen aan de robot, maar wel omdat ik voor het schrijven van de code toch wel wat tijd nodig heb. Op een gegeven moment kreeg ik een mail van Joep Suijs of ik geen interesse had om samen met Karel Dupain mee te doen. Van Joep weet ik dat die zijn mannetje wel staat als het om software gaat. Marc Robberecht had zich aangeboden om de nodige opdrachten te verzinnen en de wedstrijd te begeleiden en te jureren. Dat Marc de opdrachten ging bedenken baarde mij wel wat zorgen. Marc is een „geduchte” Roborama tegenstander, en is een persoon die regelmatig allerhande robotwedstrijden in Nederland en België bezoekt. Hij kan perfect inschatten van wat een makkelijk dan wel een moeilijke opdracht is. Al snel bleek dat als er geen minimale aanwijzingen zouden worden gegeven, vooral voor zelfbouw robots het moeilijk zou zijn. Daarom zijn er in de aanloopperiode enkele aanwijzingen gegeven door Marc.



En is er van uit gegaan dat als je een basis robot hebt mits de nodige aanpassingen op die dag er mee kan deelnemen. Iedereen was er van overtuigd dat deelnemers met Lego of Fischertechnik in het voordeel waren. Maar uiteindelijk is deelnemen belangrijker dan winnen. In de uitslag komt het voordeel niet tot uiting, integendeel blijkt dat een flinke houtschroef in een plankje of een schroefje in een stuk aluminium wel veel degelijker dan de snelbouwmodulen. Om 9 uur werden de opdrachten bekend gemaakt. Het

Roborama parcours was het strijdperk echter zonder de zwarte ondergrond met de lijnen zone A-B-C. De zones A-B-C werden als denkbeeldige plaatsen wel gebruikt bij de opdrachten . Opdracht 1: Een blik geplaatst op het midden van lijn zone B. De robot start in zone A en moet heen en weer rijden om het blik heen. Opdracht 2: 2 ballen diam. 8 cm één geplaatst in het midden tussen zone A en B en één bal in het midden van denkbeeldige lijn B ophalen en in zone A deponeren. Opdracht 3: Slalom rijden rondom 3 blikken volgens het opgegeven patroon. En als bonus waren er extra punten te verdienen met het deponeren van een blikje in zone B = 10 punten en een blikje deponeren in zone C = 15 punten. De bonuspunten konden enkel verdient worden indien het blikje tijdens een geldige opdracht werd gedeponereerd. Nu de opdrachten gekend waren hebben we geprobeerd om een inschatting te maken wat voor ons haalbaar was. We veronderstelden dat opdracht 1 en 2 met eventueel een bonus opdracht wel haalbaar was. Opdracht 3 was voor ons lastiger ten eerste omdat de robot van Joep nieuw was en hij nog niet op alle onderdelen getest was. Ook wist ik dat er enkele deelnemers al eerder testen hadden uitgevoerd met een slalom rond 4 blikjes. Uiteindelijk hebben we onze doelstellingen bereikt een ongeveer een derde van opdracht 3 kunnen realiseren wat een tweede plaats opleverde. Met een beetje meer tijd hadden we waarschijnlijk de derde opdracht ook kunnen voltooien. De verdiende



eerste plaats was voor Jan Heynen met zijn RP6 robot. Hieruit blijkt dat het niet noodzakelijk is om met een bouw pakket te starten. Op het einde van de dag heeft elke deelnemer nog een mooi LCD kleuren klokje gekregen en werd er onder de deelnemer een **Propeller Education Kit** verloot. Deze is gewonnen door Rien Van Harmelen. Eindconclusie het was een leuke goed georganiseerde en lange dag die voor velen eigenlijk nog wat te kort was. Informatie over de deelnemers, info opdrachten, foto's en uitslagen kan je vinden op de site <http://wiki.robotmc.org>

Aloys.

=====

Eerste Team Building Day in België groot succes

Zaterdag 20 september vond bij onze Belgische vrienden de (eerste) Team Building Day (TBD plaats.

Op een TBD is het de bedoeling dat er in teams (kleine groepjes) gezamenlijk aan een of meerdere opdrachten wordt gewerkt. De opdrachten worden op de dag zelf bekend gemaakt.

Voor de eerste TBD waren er al van tevoren wat tips gegeven, waarmee door de deelnemers het nodige materiaal kon worden verzameld of eventueel de robot kon worden voorbereid. De lat zou volgens Marc (Ro) niet al te hoog worden gelegd om iedereen de kans te geven deel te kunnen nemen.

De dag begon al vroeg: verzamelen om 8u30 stipt !!

Om 9u00 werden door Marc de opdrachten bekend gemaakt en uitgelegd. Daarna kregen de deelnemers tot 14u55 de tijd om de robot qua hardware en software klaar te stomen voor de demonstratie. Om 15u begon het uur van de waarheid.

De opdrachten moesten worden uitgevoerd in de Roborama "bak", dit keer zonder de zwarte rubberen mat.

1. De eerste opdracht was een variant op de Roborama-opdracht: "Heen en Weer". De variatie was dat er in vak B een blik geplaatst was, waar omheen moest worden gereden;
2. De tweede opdracht was het retourneren van 2 zachte pluchen ballen naar vak A. De ballen waren geplaatst op de grens van vak B/C en in het midden van vak C op 0m60 uit de lange wand;
3. De derde opdracht was een slalomproef met drie blikken als piketjes: de drie blikken waren geplaatst in een rechte lijn h.o.h. 1m20 en 0m60 uit de lange wand.

Er waren nog wat extra punten bij te sprokkelen door tijdens een van de opdrachten een blikje neer te zetten in vak B of vak C.

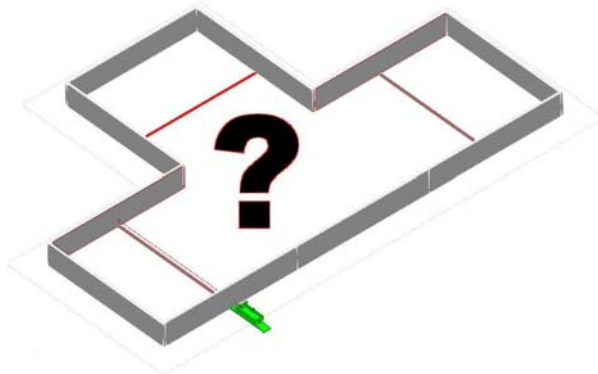
In totaal deden er 6 teams mee: 3 Lego-teams , 1 Conrad RPS-team, 1 Fischertechik-team en last but not least 1 Zelfbouw-team bestaande uit afgevaardigde uit "Holland" : Joep, Aloys en Karel.. Karel kon helaas door drukke werkzaamheden niet aanwezig zijn. Rien heeft de honours waargenomen.

Hoewel de meningen voor de wedstrijd verdeeld waren, blijkt dat je met een zelfbouw-robot waar de nodige "uitsteeksels" aan zitten om iets tegen aan te schroeven zeker niet onder hoeft te doen voor de kant en klare robots met zeer eenvoudige modulaire op/aanbouw. Je moet natuurlijk wel een Aloys in het team hebben. Hij had aardig wat gereedschap in zijn auto meegenomen om strippen aluminium te bewerken voor een feilloos werkende grijper en een blikkenafwerper.

Mogelijk dat de kant en klare robots wat voor liggen met de software (ik heb daar zelf niet zoveel kijk op) maar dat werd in "ons" team ruimschoots goedge maakt door Joep. Ook hij had zijn hele "hebben en houwen" (w.o zijn zelfbouw-robot met aluminium opbouw, extra servo's, extra sensors , etc.) meegenomen en thuis al wat voorbereidend programmeerwerk gedaan. Je moet in ieder geval "snel" kunnen programmeren, zeer vertrouwd zijn met de on/mogelijkheden van de je eigen robot en niet te snel afgeleid zijn door de drukte om je heen.

Uiteraard waren ook de andere teams met een grote hoeveelheid materiaal/materieel aanwezig. Zes lange rijen tafels waren in beslag genomen door computers, robots, stekkerdozen, batterijopladers, onderdelen, gereedschap en een grote hoeveelheid bedrading.

Het aardige van zo'n dag is (zeker voor mij) dat blijkt dat (bijna) iedereen tegen dezelfde problemen aanloopt als waarmee ik tijdens proefdraaien thuis/club te maken krijg (over het algemeen een zeer onwillige robot).



En dan breekt (toch nog veel te snel) het uur van de waarheid aan . De twee eerste opdrachten werden (bijna) door ieder team (min of meer) volledig uitgevoerd. De derde opdracht verliep wat minder succesvol: slechts een team (het Conrad RPS-team) slaagde erin de derde opdracht volledig uit te voeren. De overige teams lieten het (door tijdgebrek ? onwillige robot ?) een beetje afweten.

Drie teams slaagde erin om wat punten bij te sprokkelen door in een van de opdrachten een blikje neer te zetten in vak B (Conrad RPS-team , "ons" team en een van de Lego-teams). Geen van de teams heeft kans gezien een blikje neer te zetten in vak C.

"Ons" team eindigde uiteindelijk op een succesvolle tweede plaats. Goud was er voor het Conrad RPS-team.

Alle hulde voor onze Belgische vrienden (speciaal voor Marc) die erin geslaagd zijn er een voortreffelijk dag van te maken.

Rien van Harmelen

PS

Veel meer info (foto's , uitslagen) over deze dag is te vinden op de Belgische Website van ROBOTMC.

RoboRama November 2008

LET OP!! **De bijeenkomst van NOVEMBER 2008** IS NIET ZOALS GEBRUIKELIJK OP DE EERSTE ZATERDAG VAN DE MAAND, MAAR OP DE TWEEDE!!!!!!!!!!

Ook organiseren we dan voor de derde keer de RoboRama wedstrijd!!

En ook dit keer zullen de opdrachten zijn:

- Heen en Weer!
- T-Tijd!
- Lijnvolgen!
- Blikken!

En natuurlijk zullen ook de mini sumo robots weer de strijd met elkaar aangaan!

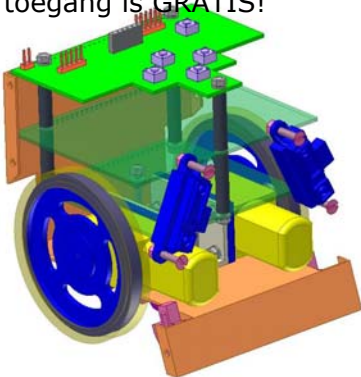
Voor inschrijven kunt u terecht op onze website:

www.hccrobotica.nl

Voor de reglementen en meer informatie, kunt u terecht op de website van RobotMC (in de toekomst ook op onze eigen site)

<http://wiki.robotmc.org>

Meer informatie kunt u ook krijgen bij Wim de Boer wim.deboer@nl.thalesgroup.com die ook deze keer de organisatie weer op zich wou nemen(waarvoor onze dank!). Bedenk dat ook deze keer weer, meedoen is veel belangrijker als winnen! De wedstrijd zal plaatsvinden op onze locatie in Hooglanderveen, en vangt aan om 10 uur.Vanzelfsprekend is ook publiek van harte welkom, de toegang is GRATIS!



Agenda

- Zaterdag 4 oktober Bijeenkomst Hooglanderveen.
- Zaterdag **8 november** Bijeenkomst Hooglanderveen.
- Zaterdag **8 november** RoboRama Hooglanderveen!!!!
- Zaterdag 6 december Bijeenkomst Hooglanderveen.
- Zaterdag 3 januari Bijeenkomst Hooglanderveen.

De bijeenkomsten te Hooglanderveen worden gehouden in Dorpshuis

"de Dissel" Disselplein 6 3829 MD te Hooglanderveen. De bijeenkomst in Hengelo wordt gehouden in de PV home van Thales(vroeger Holland Signaal), aan de Robijnweg in Hengelo. Het gebouw van de PV home vind u rechts naast de ingang.Route beschrijvingen op onze website of op die van onze voorzitter Bert Buiskool: www.robot.buiskool.net.

Voti
webshop
www.voti.nl



De zomer is voorbij ☹ Maar een klein motortje met een propeller er aan... Kant je het laten vliegen? Of langs een draad laten racen? Of maak een hoovercraft van 2 van die dingen en een PC ventilator? En misschien is die zwanehals met USB stekker ook nog te gebruiken?

Normaal €4.76, voor robobitlezers € 2.00.

<http://www.voti.nl/winkel/p/USB-FAN-01.html>
Vermeldt "RoboBits aanbieding 12"
Geldig tot de volgende RoboBits uit komt