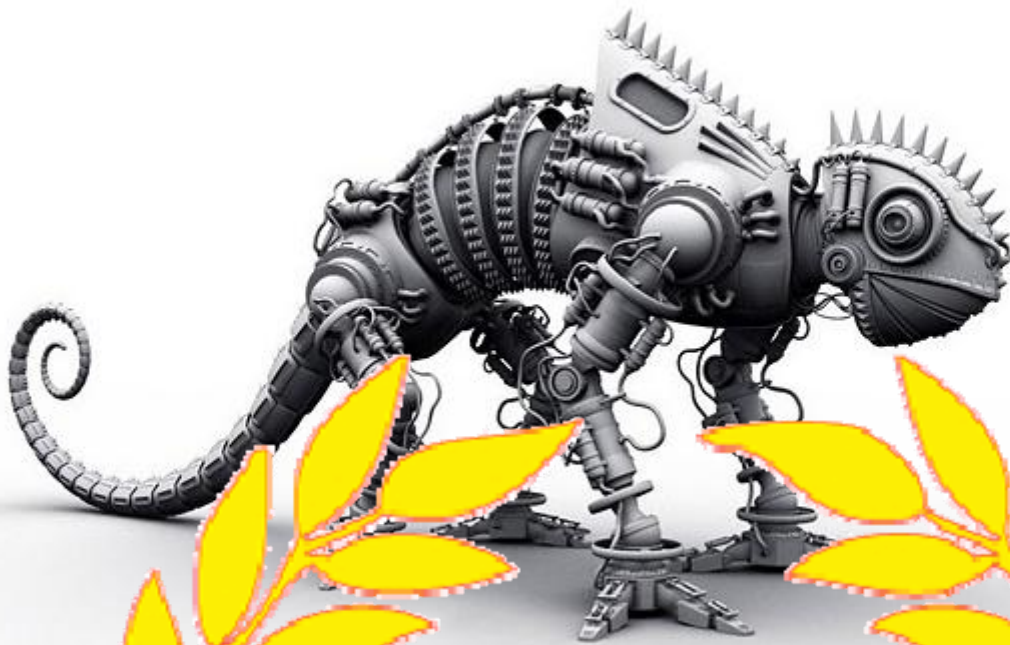


ROBO-

BITS-50

Jaargang 13, nummer 3, september 2010



50

hcc [!] **robotica**

De Robobits is een uitgave van de hcc!robotica gebruikers groep, en wordt vier keer per jaar als PDF beschikbaar gesteld aan de leden. hcc!robotica is een onderdeel van de hcc! (hobby computer club), een vereniging van bijna 150.000 leden.

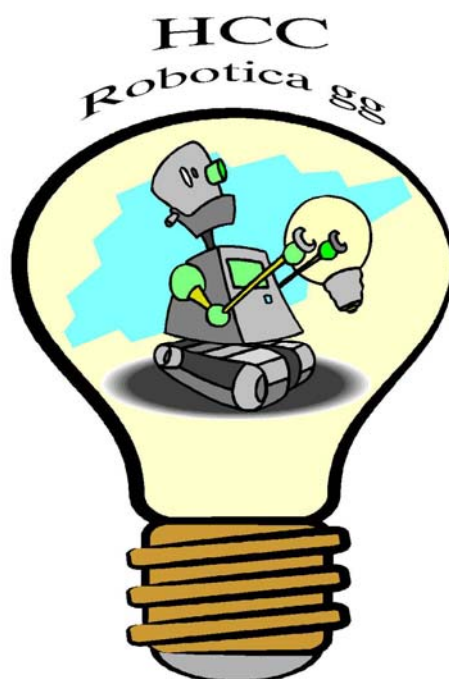
=====
Redactie adres: H.J. de Gans, Koelmanhof 2, 3816GG Nijkerk. hj.de.gans@gmail.com
Tekst aanleveren in WORD of platte tekst in ASCII. Afbeeldingen los er bij in JPG, GIF of BMP formaat.

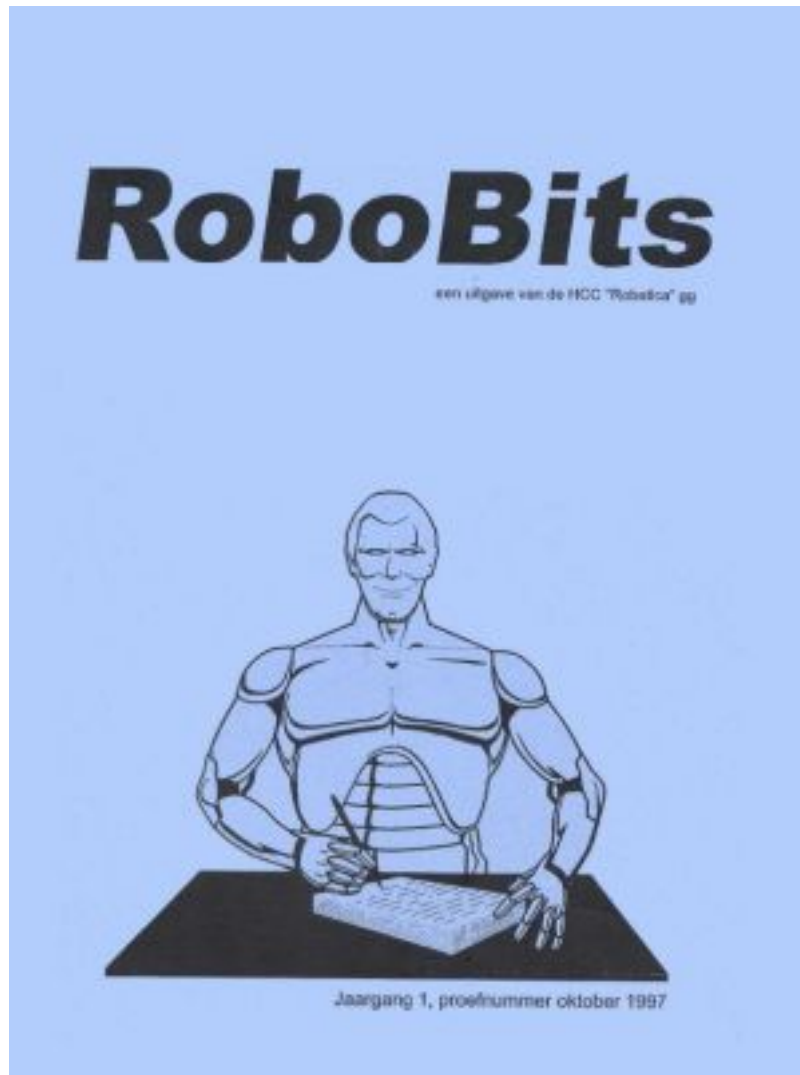
=====
Dagelijks bestuur:

Voorzitter: E.F.O.Buzzi(Ed), Ed.Buzzi@net.hcc.nl
Technisch adviseur: Ing.H.M.A.van Bodegom(Henny) ing.h.m.a.van.bodegom@hccnet.nl
Technisch adviseur: H.M.P. van Sint Annaland (Hinnie) h.vansintannaland@xs4all.nl
Secretaris: M.W.J. van Harmelen (Rien) r.van.harmelen@hetnet.nl
Penningmeester: H.J. de Gans(Henk) hj.de.gans@gmail.com
Lid/webmaster: W.C.de Boer (Wim) wim.deboer@nl.thalesgroup.com

inhouds opgave:

- Bladz. 3 Redactie.
- Bladz. 5 Van de secretaris!
- Bladz. 6 Balancerende Robots, echt niet moeilijk!
- Bladz. 12 Xplainboard met Bascom
- Bladz. 14 Stappenmotoren meten en herkennen.
- Bladz. 17 Verslag RoboRama wedstrijd te Hooglanderveen.
- Bladz. 21 Voetbal met Robotics Studio en Mindstorms NXT





En hier begon het dan allemaal mee, de eerste RoboBits! Over de geschiedenis hiervan kunt u verderop lezen in een verslag van Rien. En nu dan de 50 ste RoboBits! Een heugelijk feit waar ik niet aan voorbij wil gaan! Maar ook een moment waar ik met enige weemoet tegenaan kijk. Verschillende redacteuren gingen mij voor, en ik nam het stokje over van Abraham Vreugdenhil met ingang van RoboBits 28. Ik maakte gelijk een blunder door het versienummer te vergeten te vermelden op het kaft (wat ik gelijk te horen kreeg van Jack, die de RoboBits, (net als velen overigens) verzamelde). Eerst liet ik ze drukken bij de firma Wevers in Hoevelaken, en plakte ik er zo'n 500 dicht met de adresstickers, om ze vervolgens af te leveren op het postkantoor. Later nam een sociale werkplaats, de firma Grafifors uit Amersfoort het drukken voor zijn rekening. Deze stopte de RoboBits netjes in een plastic hoesje, en zorgde ook voor de verzending. Maar toen plotseling begin dit jaar kregen we te horen dat de hcc geen geld meer ter beschikking stelde voor een papieren periodiek. Enorm jammer, we hadden graag de papieren versie willen blijven verspreiden! We wilden eigenlijk ook graag RoboBits 50 in een speciale papieren versie kleurendruk uitgeven, maar helaas ook hier was geen geld voor. Maar iedereen is nu toch weer enthousiast om de RoboBits dan maar digitaal te verspreiden, dus hopelijk blijft ons blad dan toch nog bestaan.

Nu kan een redacteur echt helemaal niks zonder dat leden kopij aanleveren. En ik moet zeggen dat dit vrijwel altijd goed verliep. Gelukkig zijn er toch een hoop hobbyisten die

kennis en ervaring willen delen met anderen. Mijn hartelijke dank naar al diegene, is hier dan ook helemaal op zijn plaats! Ik hoop dat jullie dit allemaal volhouden, en ik roep diegenen die nog nooit iets geschreven heeft, ook zijn hobby, kennis en ervaring met ons te delen.

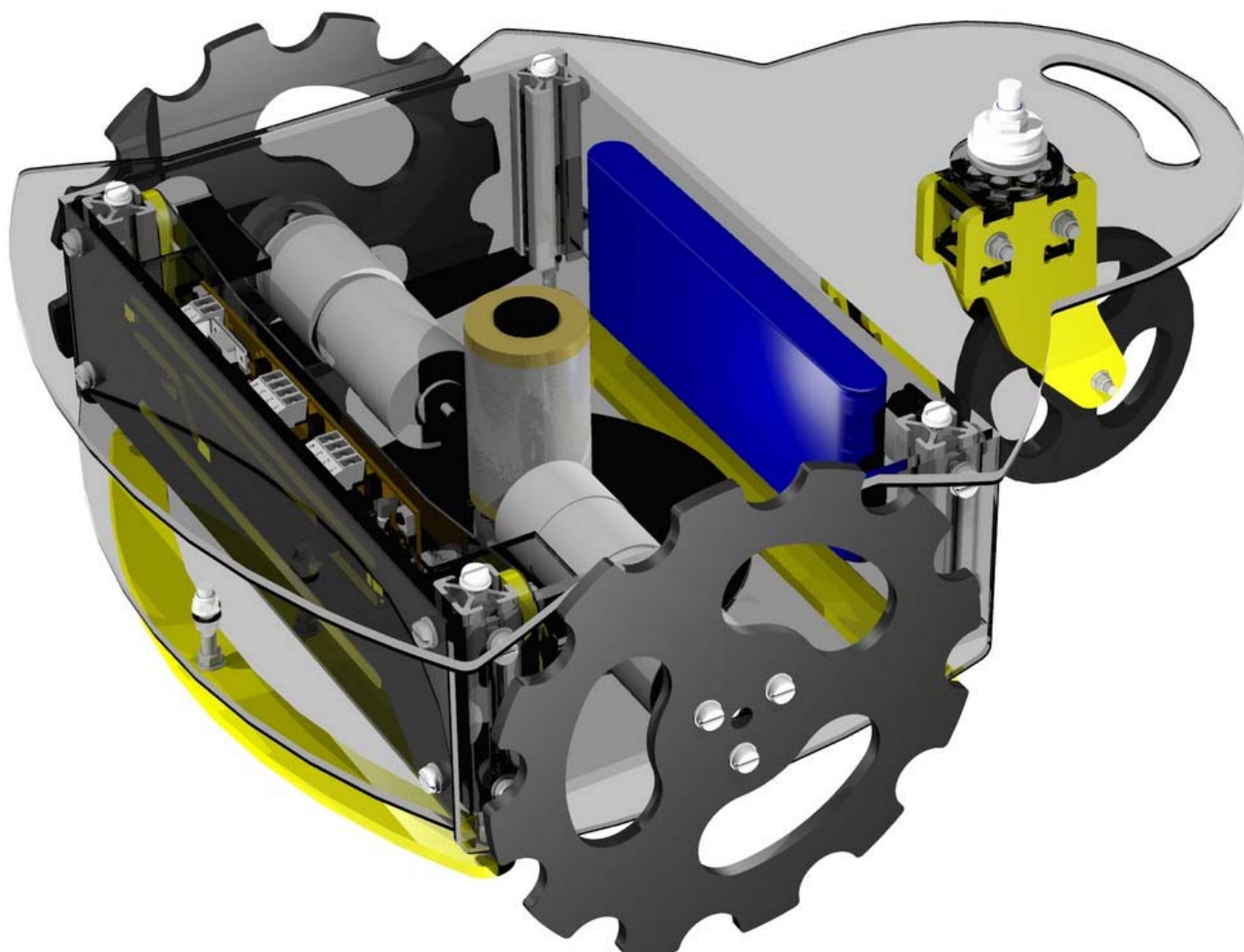
Ook in deze RoboBits weer een paar leuke artikelen, en ik wens u allen dan ook veel leesplezier toe!

Henk de Gans

deadline deadline deadline deadline deadline deadline deadline deadline

De deadline voor kopie voor Robobits 51 die eind december 2010 zal uitkomen, is gesteld op zaterdag 25 december 2010!!!! 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



van de secretaris!

Bij de 50^{ste} Robobits

De eerste Robobits verscheen als proefnummer in oktober 1997 onder bezielende leiding van Hans Ligthelm. De tweede Robobits verscheen ruim een half jaar later in juli 1998, daarbij werd als doel gesteld iedere twee maanden een Robobits te laten verschijnen. Bij het doorzoeken bleek dat 4 nummers per jaar het maximaal haarbare was/is.

De eerste 47 nummers konden per post verstuurd worden. Door de bezuinigen bij de HCC is vanaf no. 48 de Robobits verschenen als PDF file (te downloaden via onze website).

Ondanks dat het voor de (eenmans) redactie altijd moeite gekost heeft om kopie te verzamelen (het blad moet volgeschreven worden door de leden !!!) is het toch iedere keer weer gelukt om op tijd een "volle" Robobits te laten verschijnen. Hulde aan de redactie maar ook aan de schrijvers van de artikelen !!!!

Omdat ik pas actief lid geworden ben sinds 2003 ken ik vele mensen van het eerste uur niet . Toch komen in de eerste nummers enkele namen voorbij van mensen die ook nu nog actief zijn (Abraham, Dré, Henny, Jack). Enkele leden van het eerste uur zijn inmiddels overleden Hans Ligthelm † 2001 (voorzitter van het eerste uur tot - 2001), Jan Wubben †2002 (voorzitter 2001-2002). Beide leden hebben naar ik begrepen heb een grote invloed gehad op het reilen en zeilen van de HCC!Robotica.

Wat valt er te zeggen over de inhoud.

Ten eerste dat er zeer interessante artikeltjes bijzitten die zeker aan nieuws waarde niets ingeboet hebben, bv enkele artikeltjes over finit state/multitasking in Robobits no 7, 21 en 38.

Ten tweede dat je alle Robobits moet "doorlezen" om iets "speciaals" te vinden. Een tijdrovende klus, omdat er nog nooit een inhoudsopgave gepubliceerd is. Om het zoeken te vergemakkelijken is er nu van de eerste 50 Robobits een inhoudsopgave gemaakt, binnenkort te downloaden vanaf onze website. Een ieder kan (na de file te hebben gedownload) de inhoudsopgave naar eigen inzicht aanpassen/verfraaien.

Bij het doorlezen bespeur je toch ook een bepaalde ontwikkeling van eigen gemaakte bordjes naar kant en klare in de winkel/website gekochte bordjes. Tot pak weg Robobits no. 23 wordt en nog "reclame" gemaakt voor de door Henny ontwikkelde bordjes. Mogelijk (maar dat weet ik niet zeker) is deze service in de begintijd ontstaan door gebrek (of misschien te dure) in de handel verkrijgbare bordjes.

Ook bespeur je in de artikelen een tendens van ontwerp/beschrijving van een detail van een robot (uitlezen van een sensor of aansturing van een motor) naar ontwerp/beschrijving van een complete robot.

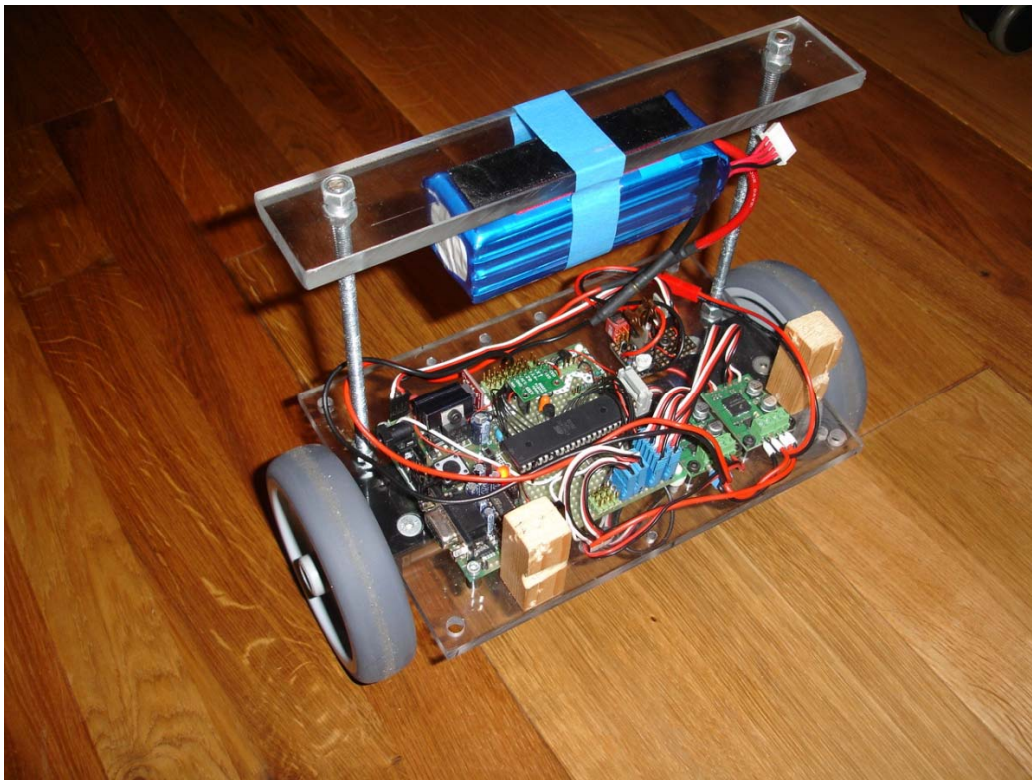
Wat ook opvalt is dat je denkt hé... dat is een interessant artikel..... maar het beloofde vervolg moet nog geschreven worden..... (bv Bully uit Robobits no 12..... Bert je krijgt nog een herkansing)

Al met al met zeer interessant om alle nummers nog een door te lezen en te moeten constateren dat van de vele vragen die je nog had/hebt de oplossing mogelijk "prachtig" beschreven wordt in een "oud" nummer van de Robobits.

Rien van Harmelen – secretaris

balancerende robots, echt niet moeilijk!!

DOOR JAN HEYNEN



Figuur 1 : Balans robot in evenwicht

Balanceren

Het principe van balanceren is zeer eenvoudig : zorg ervoor dat het steunpunt zich loodrecht onder het zwaartepunt bevindt. Het steunpunt bevindt zich bij een tweewielige robot op de plaats waar de wielen de grond raken. Het zwaartepunt ligt uiteraard een stuk hoger. Om de balans te bewaren, moeten we dus zorgen dat de wielen zodanig aangedreven worden, dat het zwaartepunt zich "gemiddeld" loodrecht boven het steunpunt bevindt. Hiervoor moeten we de verticale positie van de robot kunnen meten. Dit gaat uitstekend met de volgende sensoren :

Gyro sensor

Een gyro sensor geeft de hoeksnelheid weer die hij ondervindt. Elk lichaam kan zich natuurlijk draaien over drie verschillende assen, die loodrecht tov mekaar staan (X,Y,Z). Voor een balansrobot volstaat een één-as sensor. Een meetbereik van $300^\circ/\text{sek}$ is ruim voldoende. Natuurlijk moet de sensor zodanig gemonteerd worden, dat de gevoelige as van de sensor evenwijdig loopt aan de as van de wielen. Dikwijls zijn het sensoren met een analoge uitgang. Deze uitgang is meestal niet "ratiometrisch". Dat betekent dat de analoge uitgang niet direct evenredig is met de voedingspanning. Daarom is het nuttig om als referentie van je AD-converter een precisie-referentie te gebruiken. Zo vermijd je dat schommelingen van de voeding een invloed hebben op de meetwaarde van de gyro. De gyroscoop meet dus de hoeksnelheid. Maar eigenlijk ben je vooral geïnteresseerd in de hoek. Deze kan je berekenen door de hoeksnelheid op vaste tijdstippen te gaan optellen (= integreren). Op deze manier kan je de hoekverandering meten tov de vorige meting. Helaas, er is steeds een kleine afwijking die ervoor zorgt dat sommatie van al je metingen langzaam gaat afwijken. Dit is de zogenaamde drift. Bij mijn gyro bedraagt deze drift ca $0.1^\circ/\text{sek}$. Na 1 minuut heb ik dus reeds een afwijking van 6° ! Drift is dikwijls ook nog eens afhankelijk van de temperatuur. Daarom dat er dikwijls een temperatuursensor is geïntegreerd in de gyro. Zo kan je de temperatuur van de gyro meten, en de temperatuurinvloed compenseren. De moderne gyros (MEMS) reageren ontzettend snel. Binnen enkele ms reageert de gyro op een verandering van de hoeksnelheid. Maar omdat de gyro drift heeft, kan je alleen hiermee de stand van je robot niet bepalen. Hiervoor heb je ook nog een versnellingsensor nodig.



Figuur 2 : Gyro Analog Devices ADXRS 610

Versnelling sensor

Deze sensor geeft de lineaire versnelling weer die hij ondervindt. Ook hier heb je weer drie mogelijke richtingen van de versnelling. Daar onze aarde gezegend is met de zwaartekracht, zal deze sensor de versnelling "g" kunnen meten.

Voor een balans robot is een 2-assige versnelling sensor voldoende. Een meetbereik van 2 tot 6 g is geschikt. Deze moet ook weer juist uitgericht worden tov de wielassen. Beide assen van de sensor moeten loodrecht staan op de wielas. Uit de combinatie van de beide versnellingen kan je de richting van de zwaartekracht afleiden. Dat is wat we nodig hebben !! Helaas meten deze sensoren ook de versnelling van onze robot zelf !! Bij stilstand heb je dus een perfecte meting, maar als de robot beweegt, is er een flinke afwijking !! En balancerende robots staan nooit stil...

De oplossing voor dit probleem is even eenvoudig als geniaal : over een langere tijd gezien, staat onze robot gemiddeld wel loodrecht. We moeten dus de gemiddelde richting van de versnelling meten over een langere periode.



Figuur 3 : 3 assige versnellingsensor Freescale MMA7260

Combinatie gyro en versnelling sensor

De gyro meet dus de hoeksnelheid. Hieruit berekenen we de hoek door te integreren. Helaas gaat door de drift deze hoek langzaam afwijken.

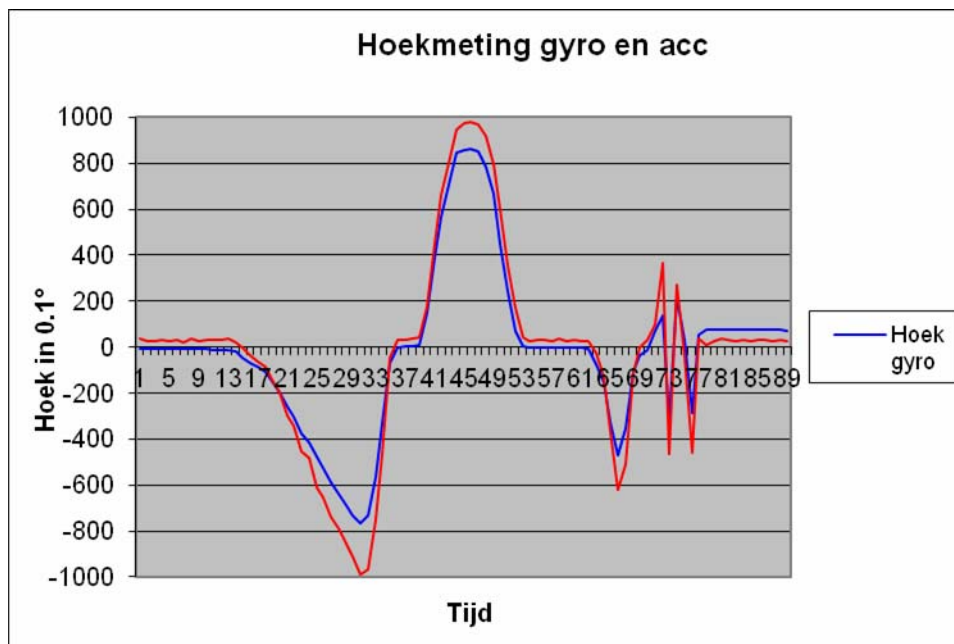
De versnelling sensor meet de hoek met de vertikaal, maar omwille van stoorfactoren is alleen de gemiddelde meting over een langere periode betrouwbaar.

Als we nu beide signalen samen tellen met bepaalde weegfactoren, heffen beide nadelen mekaar op. De som geeft ons dan een betrouwbare hoekmeting die zeer snel is (gyro-aandeel), maar die ook betrouwbaar blijft omdat de drift gecorrigeerd wordt (versnellings-aandeel). Softwarematig heb ik dat op de volgende manier gedaan :

- Ik bereken eerst de hoek volgens de versnellings-sensoren.
- Dan bereken ik de hoekversnelling van de gyro. Deze wordt elke 40 ms opgeteld (=integreren). De som is dan de hoek volgens de gyro.

- Het verschil van beide hoeken, vermenigvuldigd met een vaste weegfactor, wordt dan elke 40 ms weer bij de integraal bijgeteld. Dit is dan de zogenaamde driftcorrectie. Het resultaat wordt dan verder gebruikt in de regelkring.

In de onderstaande figuur zie je de 2 hoekmetingen zonder correctie. De print met sensoren werd langzaam gedraaid om de metingen te testen. Zoals je hier ziet waren er toen nog 2 fouten : De gemeten hoeken volgens de acc/gyro vertonen een duidelijk verschil. De calibratie was nog niet optimaal. De hoekmeting volgens de acc viel altijd groter uit. Het tweede foutje is de drift van de acc : op het einde van de meting heb ik een duidelijke afwijking van ca 8°.



Figuur 4 : Uitlezing Hoek gyro en hoek ACC

C-code Inertiaalsensoren

De gyro wordt elke 4 ms uitgelezen via adc0. Dit wordt 10 keer geïntegreerd (oversampling) . Bij stilstand geeft dit ca 6232.

Elke 40 ms wordt de hoek berekend.

Ook wordt elke 400 ms de temperatuur van de gyro gemeten via ADC1. Hiermee wordt de driftcorrectie aangepast. In principe kan bij deze sensor de drift kleiner zijn dan 0.1 °graad/sek.

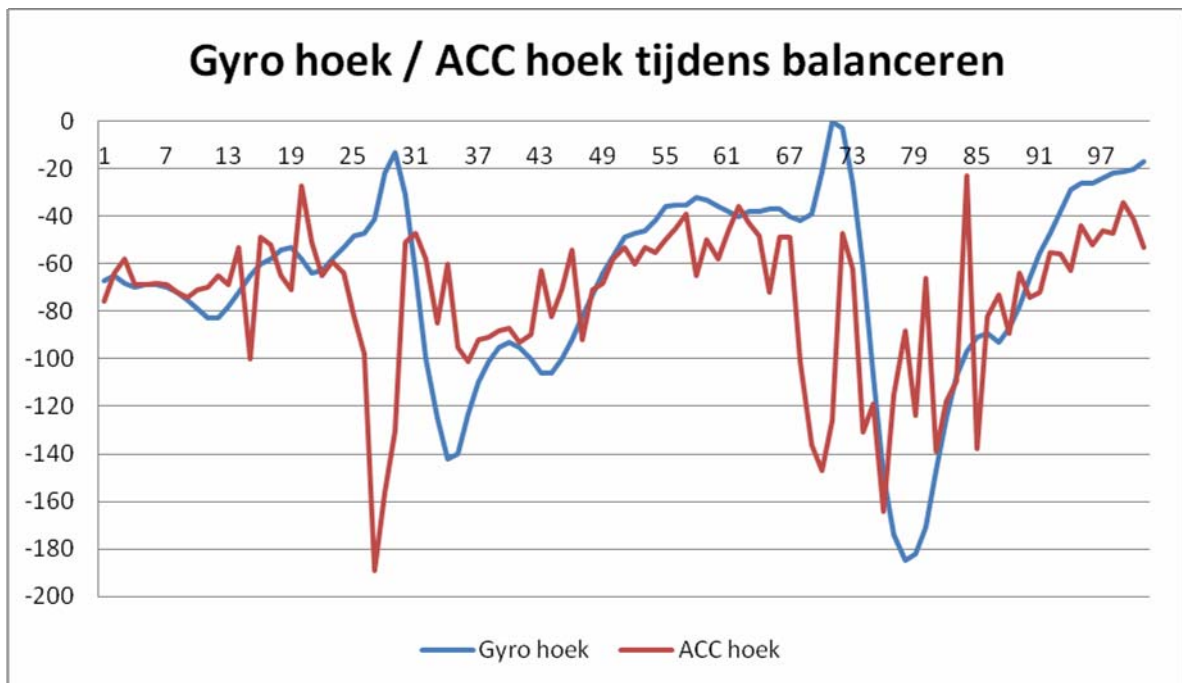
void task_initial(void)

```

{
    static uint8_t i=0;
    static uint8_t j=0;
    int16_t offset_temp=6140;
    //elke 4 ms worden de gyro + ADXL345 uitgelezen + opgeteld
    //getStopwatch is een variabele die elke ms verhoogd
    //na 40 ms worden de gemiddelde waarde gebruikt
    //De atan2 geeft dan de hoek van de versnellingsvector
    if(getStopwatch1() > 3) {
        setStopwatch1(0);
        adc0=adc0+readADC(ADC_0); //oversamplen gyro signaal
        adc5= adc5+readADC(ADC_5); //oversamplen acc X signaal
        adc6= adc6+readADC(ADC_6); //oversamplen acc Y signaal
        i++;
        if(i>9){
            aXX=(adc5-4200); adc5=0; // 0 g = 4200 voor de X-as
            aYY=(adc6-4300); adc6=0; // 0 g = 4300 voor de Y-as
            hoek=atan2(aYY, -aXX)/M_PI*1800; // hoek volgens acc
            gyro_d=(adc0-6232); adc0=0; // hoeksnelheid in °/sek=6232 (10 samples)
            gyro_hoek=gyro/35; //hoek volgens gyro in 0.1 graden
            gyro=gyro-gyro_d+0.25*(hoek-gyro_hoek); // gyrohoek+ driftcompensatie ACC
            i=0;
            if(program==1) task_balance(); else setMotorPower(0,0);
            j++;
            if(j>9){
                adc1=0.8*adc1+0.2*readADC(ADC_1); // temp gyro via sw filter
                offset_temp=(adc1-640); //temp comp. gyrodriфт
                gyro=gyro-offset_temp; //correctie elke 400 ms
                j=0;
            }
        }
    }
}

```

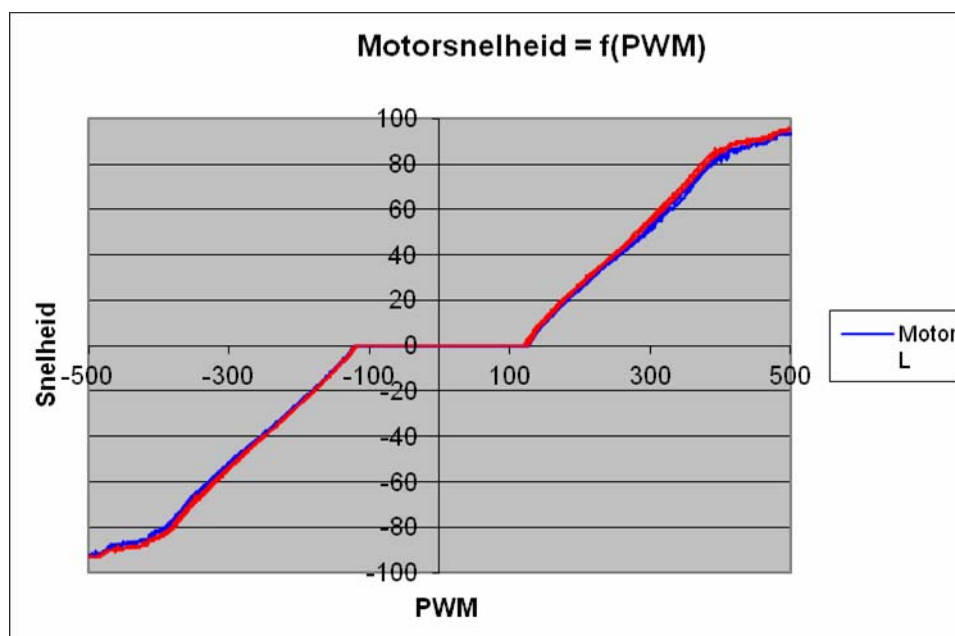
In de onderstaande figuur zie je metingen tijdens het balanceren . De ACC meting heeft zeer veel ruis en is rechtstreeks niet bruikbaar De geïntegreerde gyro meting daarentegen is zeer betrouwbaar.



Figuur 5 : Hoekmeting tijdens balanceren

Motor regeling

Als we de hoek van de robot betrouwbaar kunnen meten, komt de volgende stap : de motoren zodanig regelen dat de robot rechtop blijft. Hiertoe worden de motoren aangestuurd via een H-brug. Zo is het mogelijk dat we een traploze regeling hebben van de wielsnelheid/koppel, en dat in twee richtingen. De PWM waarde van mijn robot gaat van -500 tot 500. Pas bij een PWM waarde van -120/+120 starten de motoren. Juist deze zone is zeer belangrijk voor een balansrobot. Immers, er moet voortdurend geregeld worden rond het nulpunt. Hiermee moet zeker rekening gehouden worden met de aansturing van de PWM-waarde. In de onderstaande figuur zie je het verband tussen PWM en motorsnelheid voor de onbelaste motors van mijn robot. Het is zinvol om deze curve zelf op te meten voor je eigen balansrobot, om de start-waardes van de PWM te bepalen.



Figuur 6 : Motorsnelheid = f(PWM)

Regelkring

Op zich is de regeling vrij eenvoudig. De basis van de regeling bestaat uit de som van 2 factoren :

HOEKREGELING :

Het verschil tussen de werkelijke hoek en de hoek waarbij de robot perfect in evenwicht staat is de eerste factor. Deze hoek wordt vermenigvuldigt met een regelparameter en dat is dan een deel van de PWM aansturing. Het is duidelijk dat deze factor alleen nooit een stabiele robot kan geven : er wordt hier geen rekening gehouden met hoe snel de robot naar zijn evenwichtspositie terugkeert.

HOEKSNELHEID REGELING :

De werkelijke hoeksnelheid wordt weer met een regelparameter vermenigvuldigt en is dan een ander deel van de PWM aansturing. Deze factor gaat dus vooral de snelle bewegingen tegengaan.

Daarnaast heb ik ook nog 2 extra factoren, maar hiervoor zijn odometrie sensoren noodzakelijk :

SNELHEIDSREGELING :

De snelheid van de motoren wordt gemeten en dit wordt "meegekoppeld" met de PWM. Dit wil zeggen dat als de robot snelheid opbouwt vooruit, de hoek automatisch wordt teruggeregeld, de robot helt in de tegengestelde richting van de snelheid. Dit gebeurt door de snelheid van de motoren te vergroten !!! Het lijkt tegenstrijdig, maar hierdoor wordt effectief de gemiddelde snelheid naar omlaag geregeld ! Ook hier moet je weer een regelparameter instellen.

POSITIEREGELING :

De positie van de robot wordt gemeten met de kwadratuursensoren, en deze wordt tegengekoppeld met de PWM. Hierdoor zal de robot steeds terugkeren naar zijn startpositie. Hij pendelt heen en weer rondom de evenwichtspositie. Wederom hoort hier een regelparameter bij.

Hieronder zie je dan de C-code die ik gebruik. Het eerste regelkje is de som van de 4 factoren.

De "sollwert" is die hoek waarbij de robot ongeveer in balans is. Bij een afwijkend zwaartepunt heb ik zo de mogelijkheid om een "offset" te gebruiken.

"gyro_hoek" is de gecorrigeerde hoek van gyro + acc

"gyro_d" is de waarde van de gyro (hoeksnelheid)

"distance" is de waarde van de kwadratuursensoren, dus de positie van de robot

"speed" is de waarde van de actuele snelheid, ook van de kwadratuursensoren

Correctie aanloopkoppel wordt gebruikt om de "dode zone" van de PWM-waarde te overbruggen.

C-code Motorregeling

```
pwm=(sollwert-gyro_hoek)*P_balance/4+gyro_d*D_balance/64+distance*P_distance/8+speed*P_speed/4;
pwm_L=pwm-offset; //offset voor draaien
pwm_R=pwm+offset;

if (pwm_L>10) pwm_L=pwm_L+100; //correctie aanloopkoppel min PWM = 120
if (pwm_L<-10) pwm_L=pwm_L-100;
if (pwm_R>10) pwm_R=pwm_R+100; //correctie aanloopkoppel min PWM = 120
if (pwm_R<-10) pwm_R=pwm_R-100;
setMotorPower(pwm_L, pwm_R);
```

Draaien met de robot tijdens balanceren

Om de robot te laten draaien volstaat het om een tegengestelde offset op de PWM in te voeren bij de linker en rechter motor. De som van beide blijft dan hetzelfde en zorgt voor het balanceren. Het verschil tussen beide PWM waarde zal een draaien veroorzaken.

Vooruit / Achteruit rijden tijdens balanceren

Om de robot te laten rijden kan men de gemeten snelheid van de wielencoders mee als factor in de PWM waarde optellen ($speed * P_speed$). Speed is hier dan het verschil tussen gewenste snelheid en werkelijke snelheid. P_speed is de constante. Als de snelheid afwijkt van de gewenste snelheid zal de robot in de juiste richting overhellen en versnellen of vertragen. De positiefactor $distance * P_distance$ mag dan uiteraard niet meer meeregelen. Dit kan eenvoudig gedaan worden door de regelparameter P_distance op 0 te zetten.

Grenzen van het balanceren

Dergelijke balancerende robots hebben natuurlijk hun grenzen. Volgende parameters spelen een belangrijke rol :

- De maximale snelheid van de motor met de gegeven batterijspanning. Als deze snelheid wordt bereikt, kan de robot in één richting niet meer corrigeren. Hij zal omvallen

- De maximale fouthoek waarbij het vermogen van de motor/grip van de wielen onvoldoende is om de robot terug in balans te krijgen. Eens dat de fouthoek groter is, zal de robot omvallen. Om te vermijden dat de motoren volle kracht blijven draaien als de robot is omgevallen, worden deze uitgeschakeld als de fouthoek groter is dan deze waarde (ca +/- 45°).
- De regelparameters zijn afgesteld op een bepaalde positie van het zwaartepunt / traagheidsmoment van de robot . Deze mogen dus niet te veel wijzigen, of de regelkring kan onstabiel worden. Het aanpassen van de regelparameters kan dan terug een verbetering brengen
- Te hoge gladheid van de ondergrond is ook fataal voor een balansrobot. Immers, de krachten die nodig zijn om te balanceren kunnen niet meer gehaald worden omdat de wielen doorslippen. Je kan het vergelijken met fietsen op glad ijs.

links

[Video1](#) van de eerste balanceerpogingen

[Video 2](#) van draaien/rijden in de tuin

[Nbot](#), een mooie site over een balans robot

[Watterott](#), een goede webshop voor gyro's , acc en andere sensoren.



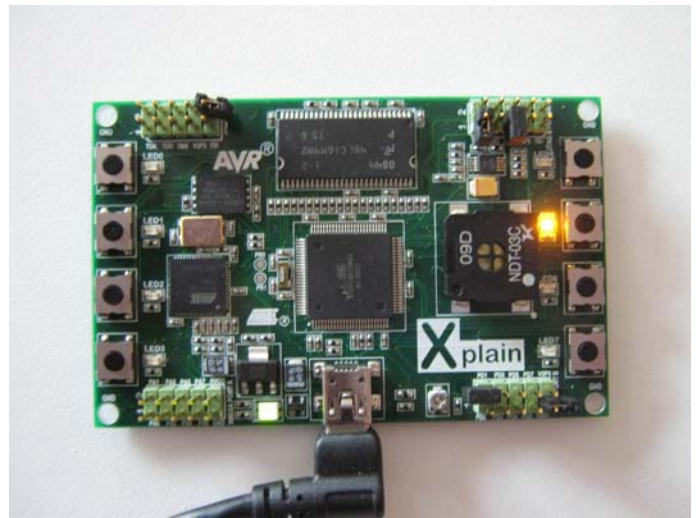
XPLAIN BOARD icm Bascom

XPLAIN BOARD icm Bascom.

Atmel is een hele bekende fabrikant als het gaat om processoren. In het verleden werden door Atmel de 8051-achtige processoren in allerlei soorten en maten uitgebracht. Wie kent niet bijvoorbeeld de 89C2051 procesor. Na de bekende 8051-achtige kwam Atmel met een eigen lijn, de AVR-processoren met een Risc-instructieset. De AVR-processoren begonnen klein en zijn nu ontwikkeld tot heel grote processoren met allerlei extra uitbreidingen, zoals extra Uart's, extra Eeprom, meer RAM en natuurlijk meer I/O pinnen.

In de loop van 2008/2009 werd er een nieuwe lijn processoren door Atmel aangekondigd. De XMega-processor. Atmel heeft een heel mooi demo bord gemaakt met hierop een

xmega processor, genaamd Explain-bord. Hierop zit de ATxmega128A1 processor. De kracht van deze processor zou liggen in de flexibiliteit. Elke I/O-pin is op heel veel verschillende manieren te gebruiken, Er zijn 8 8-bits poorten beschikbaar en elke poort heeft een eigen Uart. Er zijn 8 ADC kanalen die 12 bits zijn.



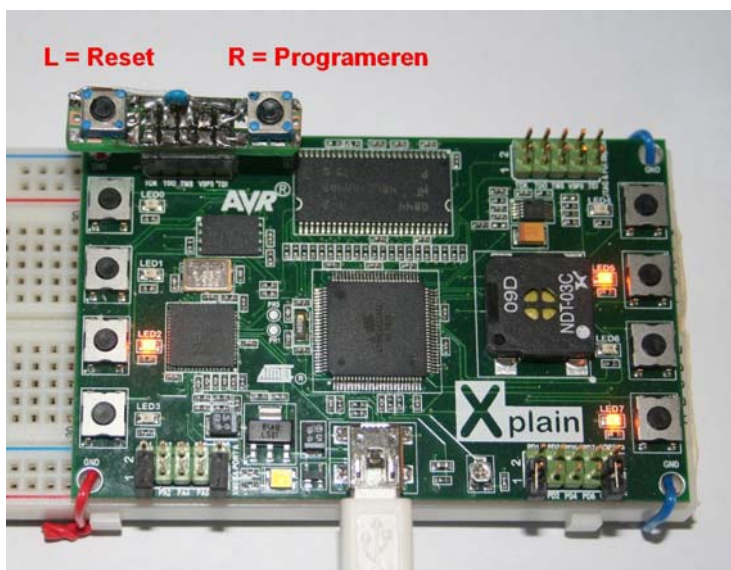
<i>Microcontroller</i>	<i>Flash KB</i>	<i>SRAM KB</i>	<i>I/O pins</i>	<i>Chassis pins</i>	<i>16-bit Timer</i>	<i>SPI/TWI /USART</i>	<i>12-bit ADC</i>	<i>12-bit DAC</i>	<i>Analogue comparators</i>
ATxmega128A1	128	8	78	100	8	4 / 4 / 8	2x8	2x2	4

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4506

Een nadeel van het Explain bord is dat er een JTAG-programmer nodig is om de xmega te programmeren. Veel hobbyisten beschikken niet over een JTAGprogrammer vanwege de kosten van deze programmer. Maar het Xplain bord beschikt ook over een USB interface. Nu is er een project op internet genaamd LUFA. Hiermee kan je via een USB interface een programma in een processor laden. Op internet is veel info te vinden hoe dit toe te passen ondermeer op het Xplain bord. Het lastige is dat deze technische beschrijvingen allen in het engels zijn en dat je verschillende documenten bij elkaar moet zoeken om een werkend geheel te krijgen. In het westland is een hobbyist (Jan Volkering) aan de slag gegaan met het Xplain bord en heeft hier met behulp van het LUFA project een methode, in het nederlands, uiteengezet hoe vanuit Bascom-AVR een programma in het Xplain bord te zetten. Ook zijn er bepaalde instructies in Bascom-AVR nog niet geïmplementeerd voor de

ATxmega128A1. Het gaat dan met name om de setting's hoe je de I/O-pinnen moetgebruiken. Ook heeft hij ook in de begeleidende documentatie beschreven. De beschrijving van dit hele verhaal is te vinden op:

<https://sites.google.com/site/bpluswestland/calendar/xmega-project>



Op deze site staat ook een schakeling die als reset button kan dienen om te schakelen tussen de programmeer en de communicatie modus van de USB interface. Ook staat er een klein voorbeeld programma om schakelaar's uit te lezen en Led's aan te sturen. Als voorbeeld wordt hier een doolhof gebruikt. Als je als hobbyist een aanvang wil maken met het gebruik van het Xplain bord is deze beschrijving een heel leuke start. Het Xplain bord biedt heel veel leuke mogelijkheden. Als je op internet naar

Xplain en Atmel zoekt kom je heel veel leuke schakelingen en toepassingen tegen. En dat alles voor een heel redelijke prijs, juist omdat je geen dure JTAG-programmer nodig hebt en het gewoon in Bascom te programmeren is. Veel plezier met het Xplain bord zou ik zeggen.

Abraham Vreugdenhil.



Stappenmotoren meten en herkennen

Redactie: ik kreeg dit artikel opgestuurd door Abraham. Het betreffende artikel heeft ook gestaan in de Elektor, en is geschreven door Wim Jansen. Voor meer informatie kunt u ook terecht op:

<https://sites.google.com/site/bpluswestland/home>

=====

Stappenmotoren meten en herkennen



Wim G. Jansen.

Het bepalen van de juiste aansluitingen van stappenmotoren is niet altijd duidelijk. Naar aanleiding van vragen uit een hobbywerkgroep, die besturing van mechaniek met stappenmotoren en electronica maakt, heb ik het volgende geschreven:

Stappenmotoren zijn er in vele verschillende uitvoeringen, meestal voorzien van een naamplaatje waarop fabrikant, stapgrootte, maximale stroomsterkte en soms de spelweerstand vermeld staan. Omdat bij motoren die zijn gedemonteerd uit oude afgedankte apparaten de documentatie ontbreekt, moet door meten de juiste aansluiting van de verschillende draden worden bepaald. Hiervoor gebruiken we een Ohm-meter, een wisselspanning-meter en een transformator die een spanning van 2 tot 6 V. levert.

De meeste stappenmotoren hebben twee of vier statorspoelen, die met 4 - 5 - 6 of 8 verschillend gekleurde draden naar buiten zijn uitgevoerd, zie fig.1.

Bij een motor met 4 draden zoeken we naar twee draden waartussen we een weerstand meten. De weerstandwaarde en de kleur van de draden schrijven we op. Op deze manier weten we de twee statorspoelen te onderscheiden, en dat het een bipolaire motor is.

Zie fig. 1

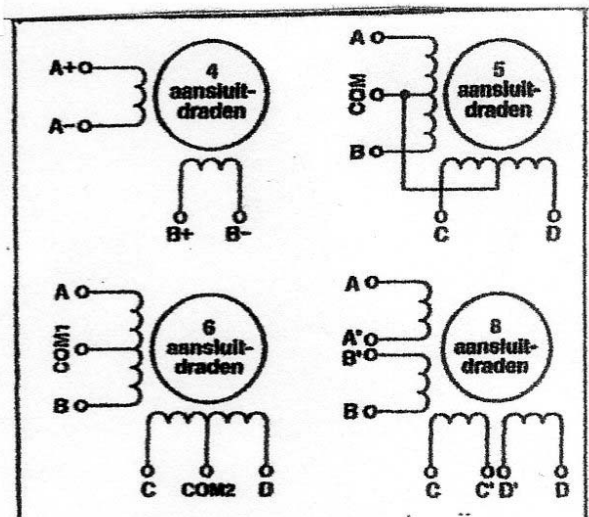


Fig 1

Aantal aansluitingen	Aansluitschema
4 draden	bipolair
5 draden	unipolair
6 draden	unipolair bipolair (in serie geschakeld)
8 draden	unipolair bipolair (in serie geschakeld) bipolair (parallel geschakeld)

Bij een motor met 5 draden (unipolair) is het moeilijk om de vier afzonderlijke spoelen te herkennen. We beginnen met de weerstand te meten tussen de verschillend gekleurde draden en hiervan een lijstje te maken.

Zie voorbeeld fig.2

We zoeken eerst de draadparen waartussen de laagste weerstand wordt gemeten en noemen die $R_x \dots \Omega$. De andere combinaties geven allen een weerstandwaarde $2R_x \dots \Omega$, die is voor ons niet van belang.

- Meetwaarden: geel / rood = $R_x \dots \Omega$
- blauw / rood = $R_x \dots \Omega$
- wit / rood = $R_x \dots \Omega$
- bruin / rood = $R_x \dots \Omega$

Hieruit blijkt dus, dat rood de gemeenschappelijke draad is.

Zie fig. 2

Twee spoelen vormen samen de A-A' en de B-B' fase. Om uit te zoeken welke bij elkaar horen, sluiten we op één spoel een kleine wisselspanning aan, eventueel met een serieweerstand. Bij voorbeeld op geel / rood. Nu meten we met een wisselspanningmeter de spanning op de andere spoelen. De spoel waarop we de hoogste spanning meten vormt samen met de stroomvoerende spoel één fase, of we die A-A' of B-B' noemen is niet belangrijk.

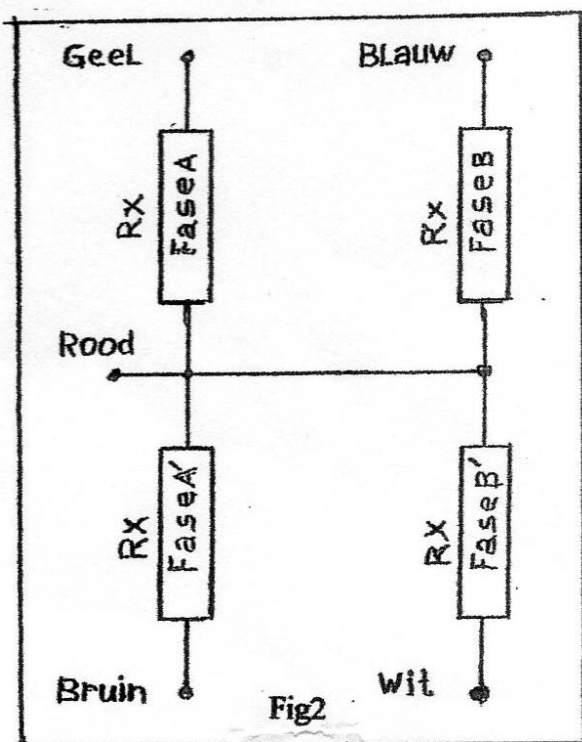


Fig2

Bij een motor met 6 draden (zowel bipolair als unipolair te gebruiken) is het makkelijk om de afzonderlijke spoelen te herkennen. We meten we de weerstandwaarden tussen de verschillend gekleurde draden en maken een lijstje.

Meetwaarden:

1. geel / rood = $R_x \dots \Omega$
2. rood / bruin = $R_x \dots \Omega$
3. blauw / zwart = $R_x \dots \Omega$
4. wit / zwart = $R_x \dots \Omega$
5. geel / bruin = $2R_x \dots \Omega$
6. blauw / wit = $2R_x \dots \Omega$

We vinden vier maal een lage weerstand $R_x \dots \Omega$ en twee maal een hogere weerstand $2R_x \dots \Omega$. Er is geen verbinding tussen de twee fasen.

Zie fig. 3

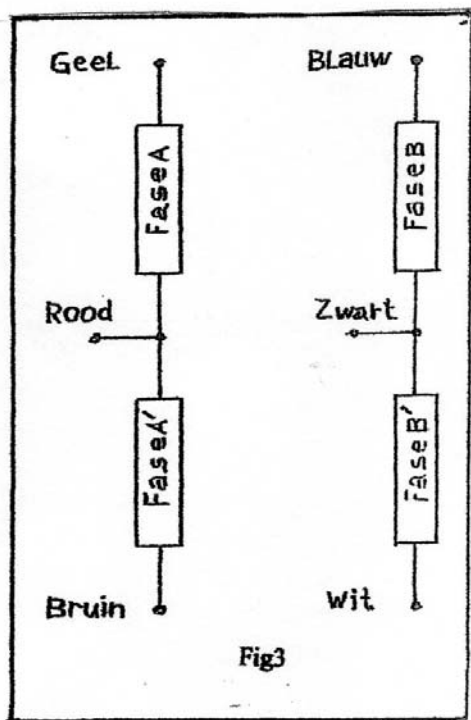


Fig3

Voor bipolaire toepassing gebruiken we aansluitingen $2R_x$. Bij bipolair gebruik wordt de midden aftakking wordt niet verbonden.

Bij een motor met 8 draden (zowel bipolair als unipolair te gebruiken) is het moeilijk om de vier spoelen van de twee fasen in de juiste volgorde aan te sluiten. We beginnen met weerstandmeting en een lijstje, dan zijn de afzonderlijke spoelen te herkennen als $R_x \dots \Omega$.

Zie fig. 4

Om de spoelen twee aan twee in de juiste fase aan te sluiten, moeten we de wikkelrichting per spoel bepalen. Hiervoor sluiten we de transformator op één spoel aan en meten met een wisselspanningmeter de spanning op de verschillende spoelen. De spoel met de hoogste spanning vormt met de stroomvoerende spoel één fase. Om ze in de juiste wikkelrichting aan te sluiten zetten we de bij elkaar behorende spoelen in serie, en sluiten de transformator op één spoel aan. Meet de spanning over de stroomvoerende spoel, en daarna over de twee spoelen in serie. Er zijn dan twee mogelijkheden ; of de spanning over de serieschakeling is ongeveer twee maal de spanning over de stroomvoerende spoel, of bijna nul. De juiste serieschakeling is die, waarbij de spanning het hoogste is. Bij bipolair gebruik schakelen we de twee spoelen per fase in serie of parallel, omdat dan het door de motor geleverde koppel het hoogst is.

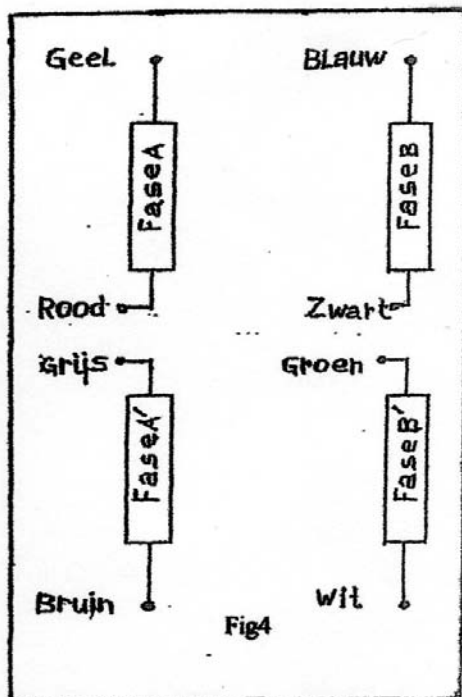


Fig4

verslag roborama oktober 2010

Roborama wedstrijd in Hooglanderveen

Zaterdag 2 oktober werd de 5e Roborama wedstrijd door HCC Robotica georganiseerd. Voorheen werden de wedstrijden steeds in november gehouden maar bij gebrek aan geschikte ruimte werd de datum verplaatst naar oktober. Nu blijkt oktober ook niet zo'n gunstige maand te zijn want de wedstrijd viel midden in een periode waarin er nogal wat Robot-activiteiten waren.

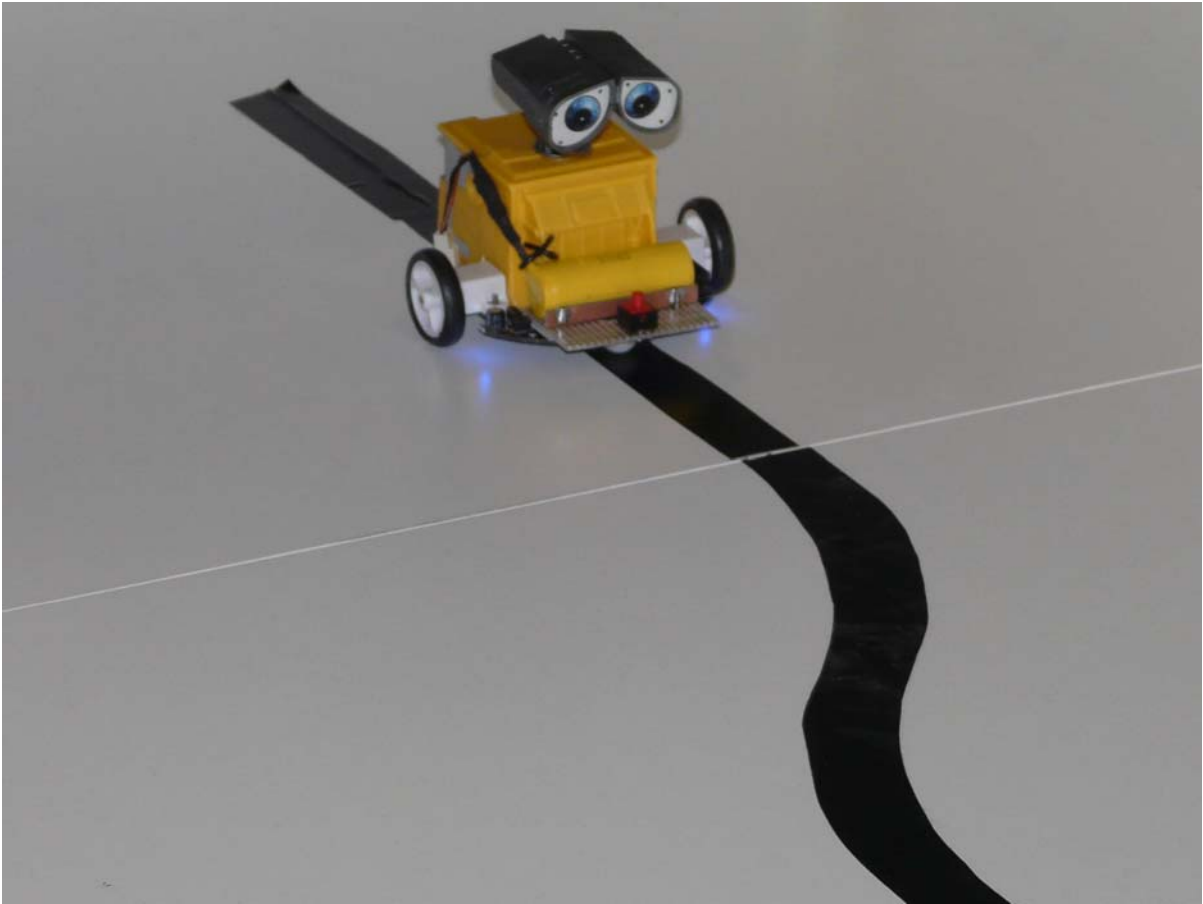


Het gevolg was wel dat er minder animo was, kortom minder inschrijvingen dan eerder verwacht.

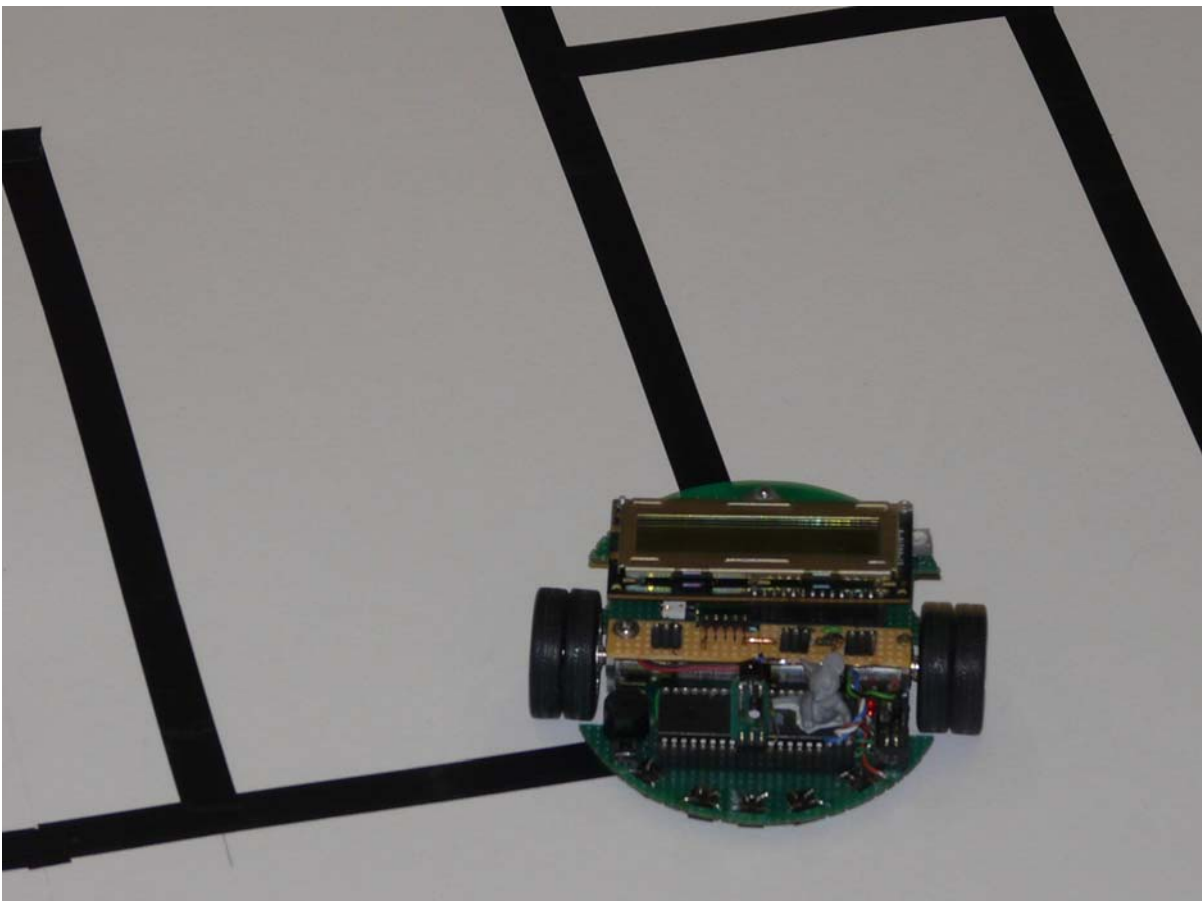
Dat neemt echter niet weg dat het een leuke en gezellige wedstrijd dag is geworden met deelnemers uit België (waarvoor onze speciale waardering) en ook wat bezoekers die de wedstrijden zeer geïnteresseerd hebben gevolgd. Mogelijk nieuwe leden voor de toekomst!

Voor de Roborama onderdelen waren er 11 deelnemers met 13 robots en voor het Sumo gevecht hadden zich 4 strijders met ieder een Sumo aangemeld.

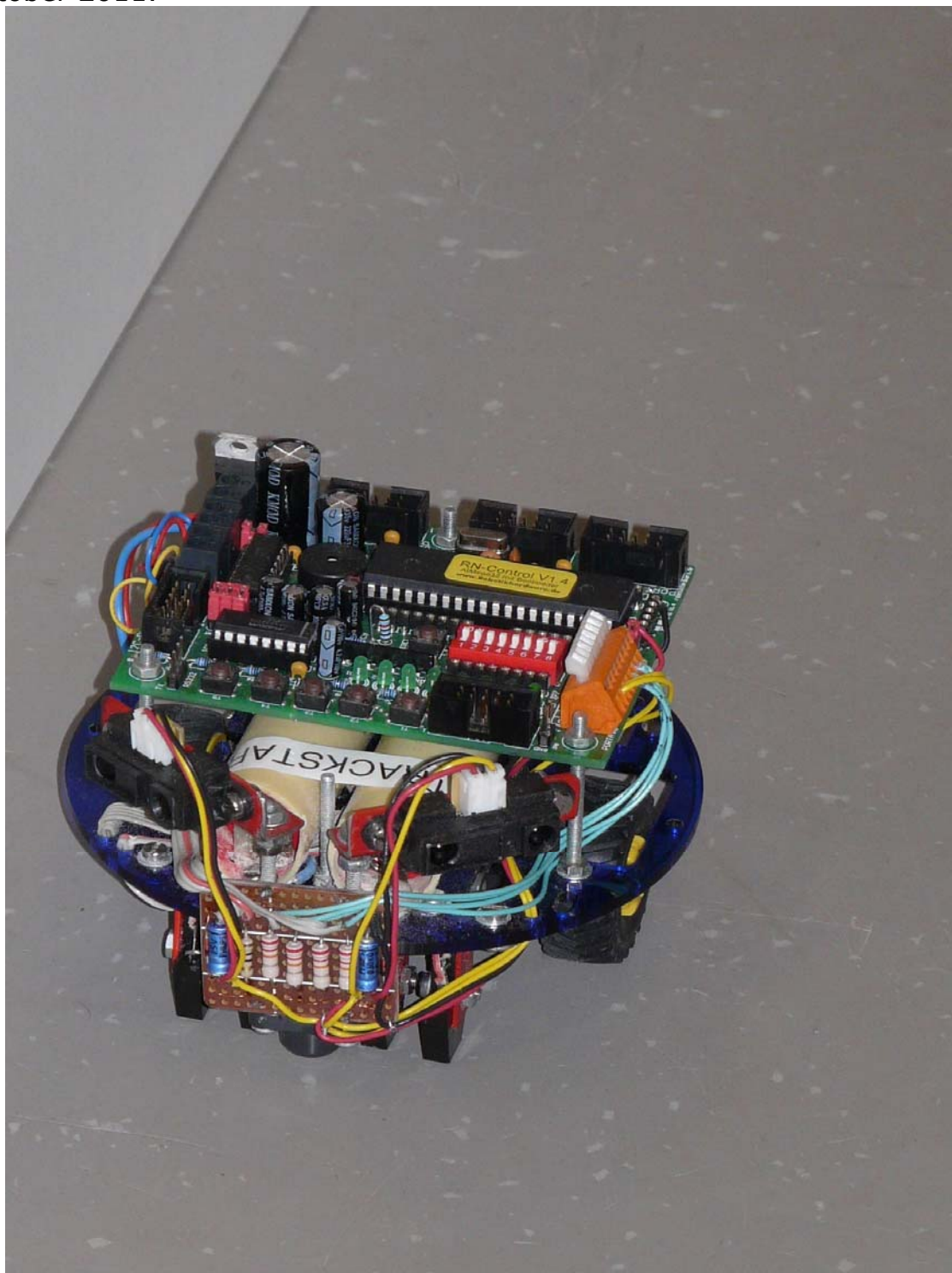
Om 10 uur ging de zaal open en werd deze ingericht. Net als vorig jaar was afgesproken om de bestaande ondergrond van de zaal te gebruiken. In principe is de grijze ondergrond met daarop de zwarte strepen (isolatieband) zeer geschikt. Toch was er een (!) robot die aanhoudend in de fout ging en het onderscheid niet kon maken.



Bij het testen in de gymzaal met goed schoongemaakte vloer werkte het perfect. In de wedstrijd ruimte met een wat minder goed schoongemaakte vloer .. ging het mis! Dat is een weetje voor volgend jaar!



Dit jaar waren er naast de bekende deelnemers en robots ook een paar nieuwelingen en wel enkele deelnemers aan het Roborama Robot Programma (RRP). Een extra proficiat aan deze doorzetters. Dat is zeker een stimulans voor de ander RRP-ers om alles op alles te zetten en hun machientjes gereed te maken voor de volgende wedstrijden in mei of oktober 2011.



De wedstrijden verliepen rustig zonder al te spectaculaire aanrijdingen. Wel hier en daar wat technische problemen of foutjes in de programmering waardoor wanden of blikjes over het hoofd werden gezien maar dat hoort nu eenmaal bij deze tak van sport. Ergenis voor de bouwer, spanning bij de toeschouwer. Als extra attractie werd er vlak voor de lunch een demonstratie gegeven met maze solvers. Mooi om te zien hoe deze machientjes hun weg zoeken, onthouden en deze vervolgens (soms) feilloos weer terugvinden.

Wouter (VOTI) was ook van de partij. Niet alleen om te kijken naar de prestaties maar ook om ons in zijn rol als sponsor weer te verrassen met de prijzen voor de winnaars. Voti, Wouter bedankt.

Bij deze bedanken wij ook de organisator (Wim) en de juryleden (Hinnie, Karel en Wim).

Tenslotte, de uitslagen en foto's staan op de HCC Robotica website maar ere wie ere toekomst, de ALLROUND winnaar Roborama werd Marc Robberecht met robot ROND en bij de Sumo werden Tinman II met Aloys de onovertroffen winnaars. Gefeliciteerd.



robot mc



<http://www.robotmc.org>



hcc[®]robotica



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

Voetbal met Robotics Studio en Mindstorms NXT

Inleiding	1
Doel van Voetbal met Robotics Studio en Mindstorms NXT	1
Introductie van Voetbal met Robotics Studio en Mindstorms NXT	2
Achterliggende regels	2
RoboCup competities.....	2
Small Size Robot League.....	3
Spelregels	4
Aanpak van Voetbal met Robotics Studio en Mindstorms NXT	4
RoboticaCup	4
Hoofdstukken	4
De Coach.....	4
De Speler en Rugnummer	4
De Camera en Lichtmasten	5
De Grasmatt	5
De Scheidsrechter	5
De Spelersbus	5
De Materiaalwagen	5
Mee doen?	5

Inleiding

Omdat dit jaar het Wereldkampioenschap voetbal 2010 plaatsvond waarin Nederland 2de werd na Spanje, besloten we dat dit het uitgelezen moment was om wat te gaan doen aan Robot voetbal. Dit document beschrijft in het kort de doelen van dit project, het reglement van Robocup, de planning van het project, opbouw van speelveld, constructie van de robot spelers, programmeren van de scheidsrechter en de robots, transport naar de lokaties, gebruik van materialen en gereedschappen.

Doel van Voetbal met Robotics Studio en Mindstorms NXT

Het doel van dit project is natuurlijk om te scoren en mogelijk mee te doen aan de Robocup wereldkampioenschappen zoals die in 2011 in Istanbul: <http://www.robocup2011.org/en/>
Hiervoor willen we ook sponsors aantrekken om de onkosten te dekken zoals bijvoorbeeld Lego, HCC en Microsoft.

Ons korte termijn (meer haalbare) doel is Robotvoetbal te demonstreren aan geïnteresseerde HCC-ers, scholen of bedrijven en hen te interesseren voor de Robotica GG.

Wij denken dat robotvoetbal een groter publiek aanspreekt dan Roborama omdat de regels van voetbal bij de meeste mensen bekend zijn en ze meteen begrijpen wat er gebeurt. Dankzij onze implementatie met Lego Mindstorms, wat men in elke goede speelgoedwinkel kopen kan, is de drempel ook voor jongeren laag. Ook Robotics Studio is gratis te downloaden en services daarvoor zijn zowel in C#, C als Visual Basic te programmeren.

Een implementatie met andere type robots zoals Sumobots uitgerust met Bluetooth is ook mogelijk maar die worden niet eenvoudig door Microsoft Robotics Studio ondersteund.

Naast plezier hebben met robots willen we ook dingen leren.

Bijvoorbeeld om uit te vinden hoe we meerdere robots met elkaar kunnen laten samenwerken zoals herkennen van andere robots van de eigen partij en die van de tegenpartij, opdrachten uitwisselen tussen robots van een team en rondspelen en schieten van de bal.

Daarvoor moeten we een aantal specifieke technieken toepassen: patroonherkenning voor waarnemen van bal en andere robots, odometrie voor bepalen van de eigen positie in het veld en berekenen van nieuwe posities en een regel handhavings programma welke de taak van scheidsrechter uitvoert.

Uiteindelijk kunnen we de WK 2010 finale Nederland-Spanje nog eens eerlijk overspelen.

Introductie van Voetbal met Robotics Studio en Mindstorms NXT

Achterliggende regels

Omdat we mee willen doen met RoboCup volgt hier eerst een beschrijving daarvan.

Een van de doelen van de RoboCup Federation is de volgende :

"By 2050, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official FIFA rules, against the winner of the most recent World Cup of Human Soccer."

Op de website <http://www.robocup.org/> zijn alle regels van RoboCup te vinden onder menu Robo Soccer, klik daarna op de competitie die je wilt zien en vervolgens op de rode titel. Hoewel er ook andere robotvoetbal competities zijn volgen we de regels van RoboCup omdat deze de grootste is.

RoboCup competities

Er zijn verschillende RoboCup competities, te weten:

- Simulation League
- 2D, 3D, 3D Development, Mixed Reality
- Small Size Robot League
- Middle Size Robot League
- Four-Legged Robot League (-2007) > Standard Platform (2008-)
- Humanoid League (from 2002)
- Kid-size, Teen-size

Simulation League is compleet in software met een 2D of 3D omgeving.

Small Size Robot League houdt in dat de robots niet autonoom zijn maar bestuurd worden via de laptop van de beide teams die hun gegevens ontvangen van een centraal systeem met camera boven het speelveld.

Middle Size Robot League betreft volledig autonoom rijdende robots.

Spelregels

In de laws worden de volgende zaken behandeld die voor het grootste deel voorzichzelf spreken:

The Field of Play, The Ball, The Number of Robots, The Robotic Equipment, The Referee, The Assistant Referee, The Duration of the Match, The Start and Restart of Play, The Ball In and Out of Play, The Method of Scoring, Offside, Fouls and Misconduct, Free Kicks, The Penalty Kick, The Throw-In, The Goal Kick, The Corner Kick.

Deze regels zijn de gewone voetbalregels maar aangepast voor robots.

Op 4 meter boven het veld moet een bevestigings buis zitten voor de camera's welke deel uitmaken van het Shared Vision System dat packets met localisatie gegevens via ethernet verstuurd naar de deelnemende partijen. Eigen camera's mogen niet bevestigd worden. Onderhoud hiervan gebeurt door Vision Experts (zie Appendix B op de website) en <http://code.google.com/p/ssl-vision/wiki/Manual> Ook zijn er regels opgesteld voor belichting, het verbod op reclame en de ondergrond.

Websites

Andere interessante websites zijn:

<http://www.robocup.nl>

<http://www.firstlegoleague.org/>

De regels die wij willen implementeren zijn nog eenvoudiger en worden in het volgende hoofdstuk uit de doeken gedaan.

Aanpak van Voetbal met Robotics Studio en Mindstorms NXT

RoboticaCup

Wegens de grote hoeveelheid regels van RoboCup gaan wij een veel eenvoudiger RoboticaCup organiseren.

Daartoe zijn zeer versimpelende randvoorwaarden gesteld met in het achterhoofd dat we beperkte middelen hebben. Om alles haalbaar te houden wordt ook de planning zo ruim mogelijk genomen maar toch precies. De onderdelen van het project worden hieronder kort vermeld maar de tijdschattingen, uitvoerenden, prioriteiten, statussen en de werkelijk gependeerde tijd worden bijgehouden in een nog te implementeren online spreadsheet.

Hoofdstukken

De randvoorwaarden die het mogelijk maken om het plan tot uitvoer te brengen worden in detail vermeld in de hoofdstukken die de komende tijd via de nieuwsbrief en het forum worden aangekondigt en op de website zullen verschijnen. Hieronder vast een overzicht.

De Coach

Het besturen van de Robots gebeurt door Microsoft Robotics Development Studio(RDS). In dit hoofdstuk gaan we beschrijven hoe je dat installeert en wat je er zoal mee kan doen. Verder wordt hier uitgelegd hoe je je Mindstorms via Bluetooth met een Joystick vanuit RDS kunt besturen. Uiteindelijk bepaalt De Coach de strategie volgens welke de Robots spelen.

De Speler en Rugnummer

De vorm van de robot moet zo zijn dat andere robots de bal kunnen afpakken maar ook dat de bal voor 80% gezien kan blijven worden door de camera's van het Shared Vision System. In dit

hoofdstuk wordt met bouwtekeningen hoe de Robot in elkaar moet zitten inclusief schietmechanisme en bevestiging van het plaatje met rugnummer.

Robots moeten een uniek rugnummer hebben ter identificatie door de scheidsrechter. Dit is een cirkel met 180 mm diameter. Ieder team heeft een eigen kleur namelijk geel of blauw. Deze kleur komt in de middelste cirkel op het dak van de robots en daaromheen nog 4 cirkels die uniek is voor de individuele robot. Deze kleuren mogen niet gedragen worden door hun eigenaars die op het veld komen om omgevallen robots recht te zetten om De Camera niet te verwarren.

De Camera en Lichtmasten

Om de spelers en de bal in het veld te kunnen herkennen gebruikt RDS een webcam die precies boven de middenstip hangt. In dit hoofdstuk wordt beschreven hoe die Camera bevestigd kan worden, welke resolutie hij moet hebben en hoe de lichtmasten geplaatst worden. De informatie over de posities worden aan De Coaches gegeven.

Ook geven we een programma om de Mindstorms te besturen door het losse plaatje met rugnummer voor de camera te bewegen.

De Grasmatt

Het speelveld is slecht 1/4 van het officiële speelveld en bovendien direct ommuurd door houten planken. Dit heeft als voordeel dat de bal niet bij uit of een overtreding door een mens ergens in het veld geplaatst hoeft te worden wat De Scheidsrechter zou verstoren. Ook kan na een goal de bal gewoon via een gat in de wand zoals bij tafelvoetbal (automatisch) het spel ingerold worden. Het veld is gemaakt van groene vilt of tapijt ommuurd met een 100 mm hoge muur terwijl de goal zijden 160 mm hoog zijn. De bouw ervan wordt in dit hoofdstuk beschreven en software om de plaats van je robot in het veld te bepalen. De bal is een standaard oranje golfbal.

De Scheidsrechter

De Scheidsrechter is het software component dat de minimale spelregels handhaaft.

Robots mogen met de bal lopen zolang ze maar om te scoren buiten het penalty gebied de bal loslaten. Overtredingen kunnen de robots niet maken omdat beide partijen via dezelfde scheidsrechter hun robots besturen. Daar gaat dit hoofdstuk over.

De Spelersbus

Om te zorgen dat de spelers goed blijven functioneren komen hier zaken aan de orde als transport van de Robots, testen voor het functioneren van de Robots, reserveonderdelen en batterijbeheer.

De Materiaalwagen

Wanneer we aan competities mee gaan doen moeten we ook ons veld, camera, verlichting en computers fatsoenlijk kunnen verplaatsen. Verder moeten er ook gereedschappen en dergelijke mee in de materiaalwagen. Die worden in dit hoofdstuk beschreven.

Mee doen?

Iedereen die mee wil doen aan dit project kan zich bij Iwan (iwan.tolboom@chello.nl) of Ed (ed.buzzi@hcc.net.nl) melden.

Als je ervaring hebt met een van de zaken genoemd in bovenstaande hoofdstukken en er onderzoek naar wil doen en/of over schrijven laat het ons dan weten.

Hou de nieuwsbrief in de gaten voor de aankondiging van de hoofdstukken.