

ROBOBITS₋₆₃

VAN DE BESTUURSTAFEL

Beste lezer,

Met het einde van 2013 en 16 jaar HCC!Robotica in zicht is het een goed moment om even om te kijken ..

Voor HCC!Robotica was het een interessant jaar met veel activiteit. Het begon in januari met de brainstormsessie waarover al diverse keren in de nieuwsbrieven is geschreven. De sessie heeft niet alleen een redelijk aantal onderwerpen opgeleverd, maar verheugend was ook dat er voor die onderwerpen genoeg leden waren die klaarstonden de leiding te nemen om er iets moois van te maken. Resultaat:

- Vernieuwd kwartaalblad: Robobits
- Aangepaste website
- Mannen met de pet tijdens de bijeenkomsten (aanspreekpunt voor bezoekers)
- Houden van presentaties (BASCOS, Raspberry Pi, 3D printer)
- Samenstellen van een Starterkit

Een mooi resultaat een jaar na dato! Ook de bijeenkomsten worden goed bezocht, gemiddeld zo'n 25 enthousiaste leden.

Onder aanvoering van organisator en scheidsrechter Bert werd Roborama 2013 weer een onvergetelijke dag.

In tegenstelling tot het ledental van de meeste andere Interesse Groepen blijft het bij HCC!Robotica constant op 290. In de begin van het jaar hebben weliswaar 16 leden afgezegd maar in de loop van het jaar zijn er 16 nieuwe leden bijgekomen en dat is een goed teken. Welkom.

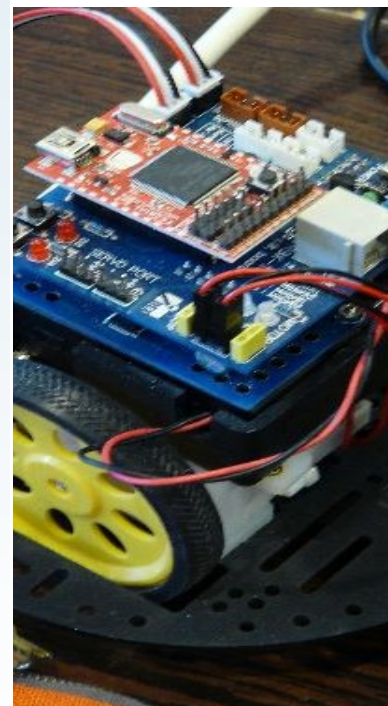
De activiteiten van het eerder zo succesvolle KDP staan op een laag pitje omdat er nog geen opvolger(s) is/zijn voor Bert die het wat rustiger aan moest gaan doen. En wat brengt 2014?

In de eerste plaats is er vanaf januari de Arduino / C workshop. Met meer dan 30 deelnemers wordt het een uitdaging voor Joep en Karel om de workshops in goede banen te leiden.

Ook zijn we zeker niet klaar met alle ideeën die tijdens de brainstormsessie naar voor werden gebracht. Werk aan de winkel! Het allerbelangrijkste is en blijft dat wij met z'n allen en iedereen op zijn eigen wijze de hobby met plezier beleven. Laten we dat elkaar toewensen voor 2014.

Het bestuur wenst alle leden **een gelukkig en gezond nieuwjaar**.

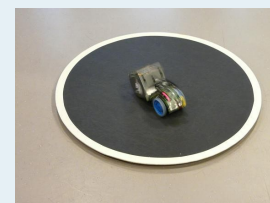
Ed Buzzi



IN DIT NUMMER

Van de bestuurstafel.....	1
Nieuws.....	2
Ombouw radiografisch bestuurbare auto.....	2
Lego workshop.....	3
Raspberry Pi.....	4
Mechatronica.....	6
Roborama 2013.....	8
Gedragafhankelijk navigeren met je robot.....	9
Agenda HCC ROBOTICA.....	12

Roborama 2013



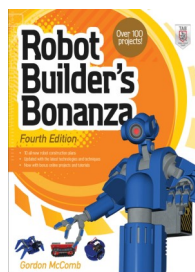
Een verslag van Roborama 2013 is te vinden op pagina 8.

Nieuws



Lego Mindstorms EV3 is een feit. De prijs van de opvolger van NXT 2.0 varieert tussen de 330 en 380 euro ! Let dus goed op waar je hem koopt!

Dit keer is de Mindstorms uitgerust met twee grote en een kleinere motor, kleursensor, aanraaksensor en een infrarood afstandbediening. De brick bevat een 300MHz ARM9 processor, 64 Mb RAM en draait op Linux. Via bluetooth zijn de robotcreaties te besturen via de Android- of iOS smartphone. Ook een wifi dongle wordt ondersteund zodat ook besturing kan plaatsvinden via het internet.

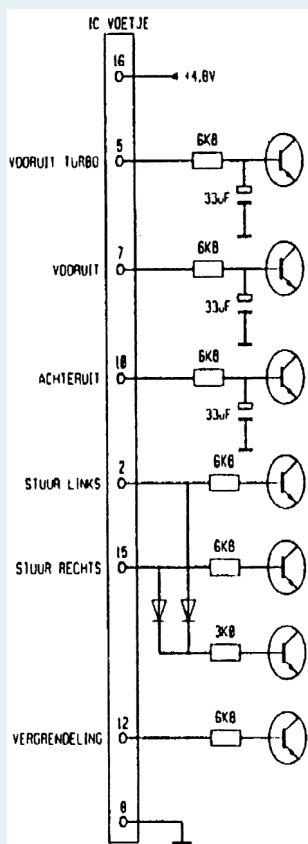


Van het boek RobotBuilder'sBonanza is de vierde versie nu gratis te downloaden (meer dan 700 blz). Even zoeken en dan vindt je meerdere links hiervoor, alleen de kaft heb ik als bijlage toegevoegd omdat het boek zelf 10 Mb groot is. Wellicht is dit boek ook bruikbaar als naslag werk bij de Cursus Arduino C..

Hinnie

(Redactie: <http://robochamp.ru/files/PDF/Robot%20Builders%20Bonanza%20-%20Gordon%20Mccomb.pdf>)

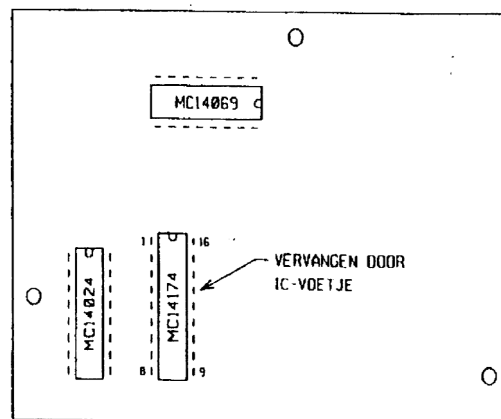
.... Bij het rijden met deze auto bleek de snelheid voor mijn doel veel te hoog te liggen. Daarom de aandrijf motor vervangen



RETRO ARTIKEL: Ombouw radiografisch bestuurd (RC) auto.

Deze rubriek herhaalt een publicatie uit eerdere Robobits. Deze keer een artikel uit Robobits 8, december 1999 van J. Ruben. Het betreft het 'modden' van een robot-auto.

In een artikel van Ton v. Lankveld werd beschreven hoe een begin kon worden gemaakt (met een destijds bij Blokler gekochte Mercedes Container Truck van de firma "DICKIE speelzeug", art. nr. 19809) met het ombouwen van deze auto tot een Robot-auto. Deze Mercedes Truck is nog steeds te koop bij Conrad.



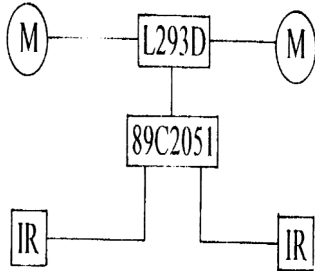
Aangezien veel mensen mij gevraagd hebben waar ik dit artikel gelezen had hier nogmaals het schema van de frontplaat ombouw en het te vervangen I.C. en de aansluitingen hiervan. (P.S. Mocht er iemand zijn die een schema van de ontvanger-print van de Mercedes Truck gemaakt heeft dan zou ik dat graag horen).

Zelf heb ik bij de Blokler destijds ook zo'n Mercedes Truck en een gele Peugeot 407 gekocht. Deze Peugeot heb ik eerst op de zelfde wijze onderhanden genomen als Ton.

Bij het rijden met deze auto bleek de snelheid voor mijn doel veel te hoog te liggen. Daarom de aandrijf motor vervangen door een omgebouwde servo type S21 (bestel nr. 232890-33 Conrad). Voor anti-botsing voor en achter een IR-sensor genomen. Deze is beschreven in Robobits nr. 2 door Henny van Bodegom.

Voor de besturing een 89C2051 gebruikt die voorzien is van een Forth programma dat te vinden is in het Egel-werkboek van de Forth-gg. Later de ontvangerprint verwijderd en voor de motorbesturing een LD293 D gebruikt. (Het schema is o.a. te vinden in Newbrain on line nr.22 door Dre Jansen).

Blokschema besturing Robot-auto:



**Vorbereitung RoboRama 2013
Hooglanderveen.**



De voorbereiding voor de wedstrijd is in volle gang. Nog even de baan controleren en de administratie op orde brengen... Zie het verslag van Ruben op pagina 8.

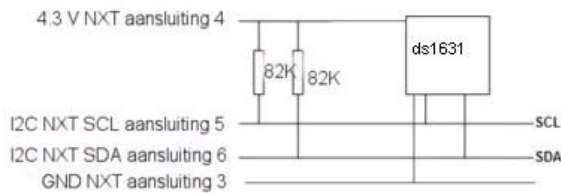
LEGO Workshop

Lego Mindstorms is natuurlijk bedoeld om robots te bouwen. Robots die zich verplaatsen en/of op de plaats bewegen.

Nu is een robot niet per definitie een bewegend apparaat. Volgens de wiki definitie is een robot een programmeerbare machine, die in staat is verschillende taken uit te voeren. Dus ook bijvoorbeeld temperatuur meten (en regelen).

Omdat de NXT module is uitgerust met een I2C bus kunnen we op eenvoudige wijze de NXT ook inzetten voor een temperatuurlogger! Zou de temperatuur in onze koelkast echt wel 7°C zijn ??

Voor de meting is de Dallas DS1621 chip toegepast. Deze chip is in staat de temperatuur met een nauwkeurigheid van een 0.5 °C te meten en via het I2C protocol te sturen. Bovendien kan de chip worden ingezet als thermostaat, zodat de temperatuur ook geregeld kan worden. De sensor wordt via een RJ12 plug verbonden met de NXT. Intern heeft de NXT geen pull-up weerstanden op de I2C bus. Daarom moeten deze extern worden aangebracht. Lego adviseert weerstanden van 82K op zowel de data (SDA)- als clocklijn (SCL):



Het programma, geschikt voor de Lego NXT brick, is geschreven in BricX Command Center. Deze programmeeromgeving is gratis te downloaden van de website <http://bricxcc.sourceforge.net/> en ondersteunt zelfs de nieuwste Lego ev3 brick.

Zie hiernaast het volledige programma.



Bij de firma Conrad (www.conrad.nl) is een complete print met de DS1631 erop, te verkrijgen. Op de foto is de print te zien met de aansluiting op de I2C bus SCL en SDA.

```

/*
This is driver/utility program to use
with ds1621 temperature sensor.

inclusive datalog option

Name: i2c_ds1621.nxc V1.05
Compiler: BricX Commander 3.3.7.17
written by Z.E.H. Otten to use with BricX NBC/NXC language
march 2007(c)

commando set DS1621
&HAA Lees temperatuur, 2 bytes
&HA1 Lees /schrijf high temperature waarde
&HA2 Lees /schrijf low temperature waarde
&HAC Schrijf naar configuratieregister
&HA8 read counter integer
&HA9 read slope
&HEE Start T Convert
&H22 Stop T Convert
*/

#include "NXCDefs.h"

#define u8 unsigned char
#define s8 char
#define u16 unsigned int
#define s16 int
#define u32 unsigned long
#define s32 long

long T;
int i;
u8 i2c_ds1621_access_config[];
u8 i2c_ds1621_start_convert[];
u8 i2c_ds1621_read_temp[];
u8 i2c_data[];

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Read the temperature data from ds1621 sensor //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

sub get_temp()
{
s8 result;
u8 bytes_ready;

u32 count;
count = 0;

asm {
arrbuild i2c_ds1621_access_config, 0x90, 0xAC
arrbuild i2c_ds1621_start_convert, 0x90, 0xAE
arrbuild i2c_ds1621_read_temp , 0x90, 0xAA
}

SetSensorType(IN_1, IN_TTYPE_LOWSPEED_9V);
SetSensorMode(IN_1, IN_MODE_RAW);
ResetSensor(IN_1);

do {
result = LowSpeedStatus(IN_1, bytes_ready);
} while (result > 0);
result = LowSpeedWrite(IN_1, 0, i2c_ds1621_start_convert);

while (true) {
count++;
NumOut(0,0,false,count);
TextOut(0,30,false,"Temp");
Wait(1000); // it takes about 750ms
TextOut(36,30,false," ");

do {
result = LowSpeedStatus(IN_1, bytes_ready);
} while (result > 0);
result = LowSpeedWrite(IN_1, 2, i2c_ds1621_read_temp);

do {
result = LowSpeedStatus(IN_1, bytes_ready);
} while (result > 0);
result = LowSpeedRead(IN_1, 2, i2c_data);

if (result == 0) {
if (i2c_data[0] < 128)
NumOut(36,30,false,i2c_data[0]);
else {
TextOut(36,30,false,"");
NumOut(42,30,false,256-i2c_data[0]);
}
TextOut(54,30,false,"");
NumOut(60,30,false,(i2c_data[1]*100)/255);
} else {
TextOut(36,30,false,"E");
NumOut(48,30,false,result);
Wait(10000);
return;
}
}

sub datalog()
{
byte handle;
short bytesWritten;

DeleteFile("dataNXT.txt");
if (CreateFile("dataNXT.txt", 8192, handle) == NO_ERR ) {
for (int i=0;i<1000; i++)
{
string tmp= NumToStr(i);
WriteLnString(handle, tmp,bytesWritten);
WriteString(handle,tmp,bytesWritten);
}
CloseFile(handle);
TextOut(0,30,false,"Data opgeslagen");
}

}

task main()
{
while(1)
{
get_temp();
/* druk op oranje lego knop en de data wordt opgeslagen in dataNXT.txt */
if (ButtonPressed(BTN3,0)) datalog();
}
}
  
```

Boeken:

Massimo Banzi heeft veel geschreven over de arduino. Een aardig overzicht over de verschillende Arduino's is te vinden op de wiki pagina:

<http://en.wikipedia.org/wiki/Arduino>

Arduino Startersgids

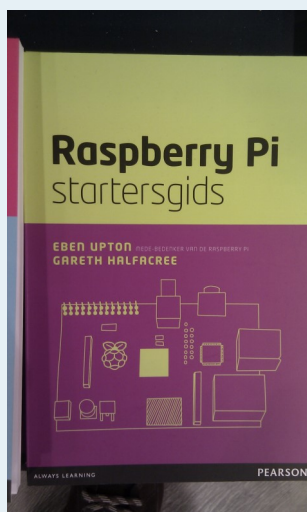
Massimo Banzi
Uitgever: perason



Een tweede boek van dezelfde uitgever betreft interessante projecten met de Raspberry Pi:

Raspberry Pi Startersgids

Eben Upton en Gareth Halfacree
Uitgever: perason



4

Raspberry Pi

Ik kon het niet laten: Een complete computer, bijna zo klein als een microcontroller, met HDMI, USB, ethernet aansluiting, opslagcapaciteit en een Linux OS. Voor ongeveer 40 euro heb ik hem aangeschaft: de Raspberry Pi.

Zeker een uitdaging om met deze configuratie wat te experimenteren. Mijn Linux ervaring/kennis is nihil. Hier mijn eerste bevindingen.



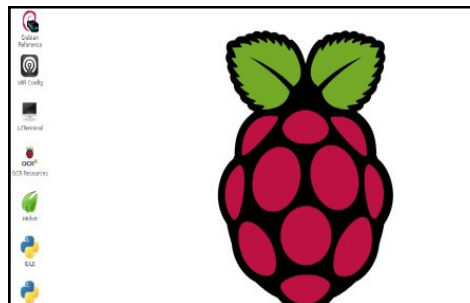
Installeren

Het betreft een Raspberry Pi met 4 Gb opslag SD card, met een usb toetsenbord en muis aansluiting. Via hdmi heb ik een 15" monitor aangesloten.

Via de website <http://www.raspberrypi.org/downloads> heb ik the New Out Of The Box Software (NOOBS) gedownload en vervolgens gekopieerd op de SD kaart. Nadat de SD card in de Pi is geplaatst en de voedingsspanning cq. adapter is aangesloten, start de Pi netjes op in een gekozen Linux variant (Raspbian "wheezy"). Ik heb dezelfde procedure gevolgd voor meerdere SD kaartjes waaronder ook een 32 Gb kaart. Je kunt ook heel eenvoudig de kaart kopiëren naar een andere kaart of naar de PC zodat je altijd back-ups kunt maken en kunt experimenteren met verschillende Linux varianten. Zo zou je ook voor elke toepassing een apart SD kaartje kunnen nemen. Het programma wat je hiervoor nodig hebt is Win32DiskImager

Eerste keer

Natuurlijk moet de netwerkaansluiting ook worden gebruikt. Via een patchkabel sluit ik de Pi ook aan op mijn thuisnetwerk. Zoals de foto laat zien wordt het nu al snel een onhandige kabelmassa die de kleine pi op eens heel groot maakt.



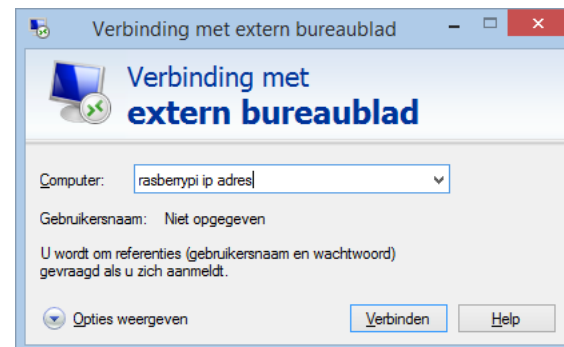
De stekker kan in het stopcontact. Na enkele seconden krijg ik dan resultaat op mijn beeldscherm. Omdat ik tijdens de installatie gekozen heb voor een grafische interface start de Pi op in de grafische userinterface (GUI). Of start de GUI met het commando startx.

Remote desktop

Al die aansluitingen, extra toetsenbord, muis en beeldscherm is niet echt handig. Gelukkig kunnen we de Pi op een PC of laptop overnemen wanneer we remote desktop gebruiken. Het commando om op de Pi de remote desktop aan te zetten is:

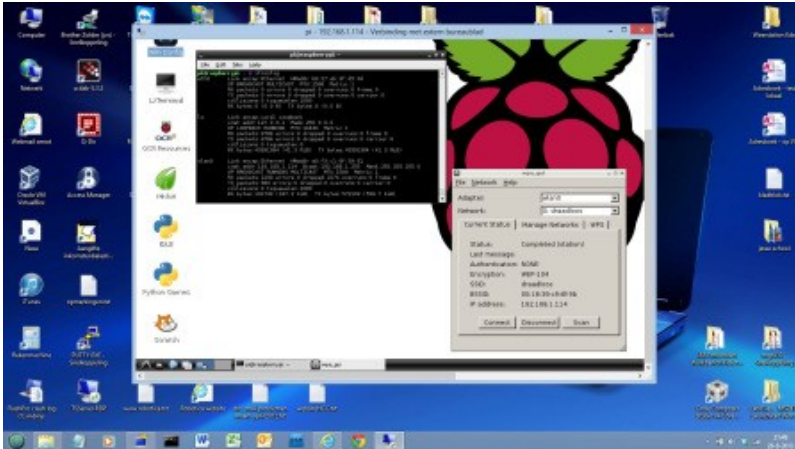
```
sudo apt-get install xrdp
```

Dit commando zorgt voor de installatie van de remote desktop. Met het commando *ifconfig* kunnen we achterhalen wat ons ip-adres is op het netwerk. Vervolgens kunnen we op de (windows) PC/laptop met een rdp sessie de Pi op het scherm krijgen.



'Ik ben nog niet helemaal tevreden. De stugge netwerkkabel zit behoorlijk in de weg.'

Zie hier: de Raspberry Pi op mijn windows 8 laptop:



De link <http://www.raspberrypiblog.com/2012/10/how-to-setup-remote-desktop-from.html> geeft nog wat extra informatie.

Wifi

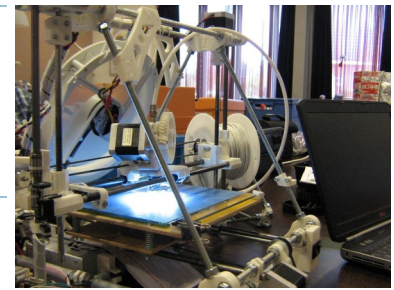
Ik ben nog niet helemaal tevreden. De stugge netwerkkabel zit behoorlijk in de weg. Gelukkig wordt door deze Linux versie wifi goed ondersteund. Ik plaats een wifi usb module in de Pi (TL-WN723N Mini Wireless N usb adapter). Deze wordt herkend en ik heb een configureerbare wlan0. In de GUI is een tooltje aanwezig waarin ik de ssid van mijn netwerk opgeef en mijn wachtwoord. Opnieuw opstarten zonder netwerkkabel.

Hieronder zie de de Pi ZONDER muis, toetsenbord en beeldscherm! Volledig beheerd via een remote desktop sessie op mijn laptop.



Een volgende keer hoop ik een toepassing te tonen. Ik heb al een io extensie aangeschaft (<http://learn.adafruit.com/adafruit-pi-cobbler-kit>). Hiermee gaat het interfacen met de Pi wat gemakkelijker:

Z. Otten



3D printer

In 2014 wordt er een doorbraak verwacht op de markt van de 3D printers. De prijs zal verder dalen. Voor een 'hobby' printer zal dan nog maar circa 500 euro betaald worden. Er kan dan geprint worden met smeltbaar kunststof zoals ABS of PLA. Maar ook vloeibaar hars kan worden gebruikt als printmateriaal.

Inmiddels is er ook al een Nederlandse fabrikant van 3D printers: Ultimaker. Een aardige site met veel 3D printer info is <http://www.makerpoint.nl/nl/>

In 2014 zullen we in de robots zeker meer aandacht gaan besteden aan het 3D printen.

Hieronder zie je enkele voorwerpen die zijn geprint tijdens de Roborama 2013 wedstrijden in Hooglanderveen:



ABS of PLA: ??



Wat doet de werkgroep

Mechatronica onder de vleugels van de HCC-Robotica groep?

De Mechatronica-groep is een voortvloeisel uit de 6500 Gebruikers Groep die zich bezighoudt met het aansturen van stappenmotoren.

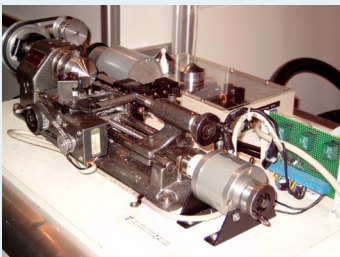
Deze stappenmotoren gekoppeld aan draadspindels, zorgen voor de bewegingen bij o.a. draai en freesbanken en overige mechanische bewegingen.

Dat deze stappenmotoren niet uit zich zelf gaan draaien spreekt voor zich.

Wat is daar allemaal voor nodig?

- Hardware
- Software
- Voedingsunit
- Motoren

Voor het draaien zijn op dit moment een horloge draaibank en een Emco 5 draaibank alsmede een zelf gebouwde Portaal freesbankje ugerust met stappenmotoren.



Mechatronica

Hoe krijgen we nu de spindels aan het draaien?

1. Interface tussen computer en de motoren

De interface die nu gebruikt wordt is de Stepper Motor Card SMC800-SMC1500 te koop bij Conrad. Deze is te plaatsen op de printerpoort van een PC en wordt geleverd met software voor rechtlijnige bewegingen. Dus niet geschikt om boogvormige bewegingen te maken.

2. Hardware

Hiervoor kan elke PC met een printerpoort gebruikt worden.

Dit geldt alleen tot en met besturingssysteem Windows 98. De printerpoort is niet meer exclusief te benaderen met de modernere besturingssystemen.

Dit probleem is nu ondervangen door de ontwikkeling van een dongel die van een USB_Bus een fictieve Printerpoort maakt.

3. Software

De software is natuurlijk wel te koop, maar is leuker om zelf te ontwikkelen. Tot en met Windows98 is een besturingssysteem geschreven met behulp van QB45, maar ook dat verouderd en moest dus vervangen worden door een modernere taal. De keuze is gevallen op Visual Basic, deze taal is een vervanger van QB45

Deze software is nu in gebruik voor o.a. bij het draaien van buffers voor de zelfbouw van wagons in de schaal HO.

4. Voedingsunit

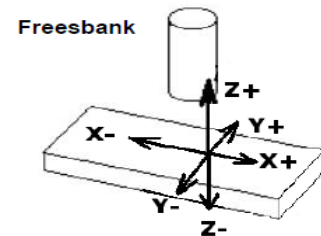
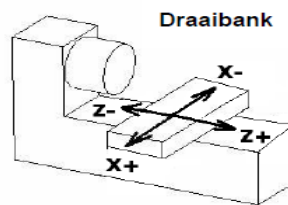
Voor de voeding van de SMC800/SMC1500 interface die op de stappenmotoren zijn aangesloten mag maximaal 30 volt bedragen bij een stroomsterkte van maximaal 1,5 Amp. Dit kan met een simpele trafo bereikt worden.

5. Stappenmotoren

Stappen motoren van het type Bipolair zijn geschikt om met de SMC800/SMC1500 kaart te communiceren. Deze motoren zijn te koop maar natuurlijk ook uit oude printers te halen. De motoren van het type Unipolair zijn minder geschikt voor deze kaart omdat dan maar de helft van het koppel geleverd wordt.

Wat zijn de begrippen die wij moeten weten.

Principe van de As aanduidingen:



Bij CNC – machines, ook onze machines hoe klein ook, gaat de bewegingen met behulp van spindels, bij simpele machines met draadspindels en meer geavanceerd machines met kogelomloop spindels. Deze spindels bezitten een naamsaanduiding en in de hierboven getekende afbeeldingen worden de assen voor zowel een draaibank als freesbank aangegeven.

Speciaal voor het draaien is het programma DRMODUUL geschreven. In dit programma zijn alle aspecten verwerkt die nodig zijn om de baanbewegingen uit te voeren.

Dit houdt in dat rekening wordt gehouden met bv. de spoed van de spindels, het aantal pulsen wat een stappenmotor moet hebben voor een omwenteling van de as, materiaal afmetingen, snelheid van de beweging etc.

Deze aspecten zijn vertaald in zg. parameters en zijn ondergebracht in bestanden die vooraf gedefinieerde zijn. Deze bestanden worden geplaatst op een opgegeven locatie op de harde schijf.

Product definitie.

Nu moet ook nog een profielbeschrijving van het product gemaakt worden, dit doen wij dan vlg. de principes beschreven in de norm DIN 66025.

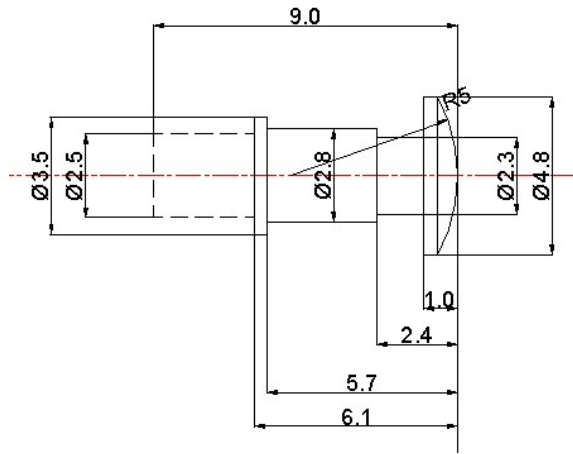
Uit deze definitie zijn de volgende codes gebruikt:

- G1 rechtlijnige beweging tussen 2 coördinaten
- G0 rechtlijnige beweging in ingang
- G2 circulaire bewegingen met de draairichting met klok mee CC
- G3 circulaire bewegingen tegen de draairichting van de klok CCW
- F Voeding van de beitel of frees

Werkstuk vertaling naar CNC programma.

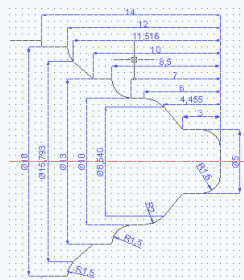
In dit voorbeeld (zie tekening), ziet een CNC programma, in principe beschreven vlg. DIN 66025 er als volgt uit (BUF02. Draaien van Buffer zonder beitelbreedte):

```
%BUF02.TXT
G1 X6 Z0 F0.05
G1 X0 Z0
G1 X4.8 Z0
G1 X4.8 Z-1
G1 X4.8 Z-0.69 F0.05
G3 X0 Z0 I-2.4 K-4.386
F0.03
G1 X4.8 Z0
G1 X4.8 Z-1
G1 X2.4 Z-1 F0.03
G1 X2.4 Z-2.4 F0.05
G1 X2.8 Z-2.4
G1 X2.8 Z-5.7
G1 X3.5 Z-5.7
G1 X3.5 Z-6.1
G1 X2.5 Z-6.1
G1 X2.5 Z-9.0
G1 X0 Z-9.0
%
```



Voorbeeld van een wat meer gecompliceerd draaimodel:

- 1 (100.TXT Draaistuk)
- 2 %100.txt
- 3 G1 X20 Z0
- 4 G1 X0 Z0
- 5 G1 X2 Z0
- 6 G2 X5 Z-1.5 I0 K-1.5
- 7 G1 X5 Z-3
- 8 G1 X8,540 Z-4,455
- 9 G2 X10 Z-6 I-1,27 K-1,1545
- 10 G1 X10 Z-7
- 11 G3 X13 Z-8.5 I-1.5 K0
- 12 G1 X13 Z-10
- 13 G1 X15,793 Z-11,516
- 14 G3 X18 Z-12 I,104 K1,016
- 15 G1 X18 Z-14
- 16 G1 X20 Z-14
- 17 G1 X20 Z0
- %



Wat zien we na het opstarten van de software!

In dit opstartscherm van de draaimodule zien wij de volgende gegevens als op de toets

“Tekenen Opspanning” is gedrukt:

- Werkprogramma: %buf02.txt
- Diameter materiaal 6 mm
- Uitsteeklengte 10 mm
- Schaal projectie 6
- Positie Verticale As 50
- USB compoort 4
- Vertraging X-as 30
- Vertraging Z-as 30

Nu staat dus het profiel getekend van het eerder vermelde product. Ook het werkprogramma kan vooraf worden gelezen voordat men de draibewerking met de Toets Draaien in werking zet. Hieronder de uitbeelding van het gedraaide product. De bewerkingstijd van deze buffer voor een wagon in HO uitvoering bedraagt ca. 2 minuten:

Dit project van de Mechatronica-werkgroep wordt uitgevoerd door:

- Wim Jansen: Motoren aansluiting
- Dre Jansen: algemene ondersteuning
- Abraham Vreugdenhil: in–output codes in VB6 en programmering Dongel
- Evert Verbeek: aanmaak voedingsunit en etsen printplaat
- Jaap Rademaker: Software aanmaak en bouwer freesbank

Deze uitgave is een korte samenvatting van het project.
Jaap Rademaker

Gedragsafhankelijk navigeren met je robot

Tot vorig jaar was ik bezig met het bestuderen en testen met het navigeren van een robot. De test resultaten hiervan waren aardig succesvol. Door problemen met mijn gezondheid heeft dit project daarna geen voortgang meer gehad, maar nu het met de gezondheid weer wat beter gaat heb ik de afgelopen twee maanden de draad weer opgepakt.

Er is ondertussen over dit onderwerp wat meer documentatie beschikbaar gekomen doordat de firma RidgeSoft (vorig jaar gestopt met hun webshop) hun spullen in de uitverkoop hebben. Het boek "Exploring Robotics with the IntelliBrain™-Bot (Second Edition)" een inleiding tot Robotica en Java Programmering en is nu gratis bij hun te downloaden op <http://www.ridgesoft.com/articles/education/ExploringRoboticsEdition2.pdf>. Het boek geeft een goede inleiding van het gedragsafhankelijk navigeren, dat het boek de taal Java gebruikt bij de code voorbeelden moet je maar voor lief nemen. Het omzetten van de Java code naar C# of C++ is eenvoudig, maar naar niet object georiënteerde talen wat lastiger. Mijn huidige navigatie programma's heb ik de afgelopen maand aangepast naar de in dat boek gebruikte termen zodat ik het boek nu ook als mijn documentatie kan gebruiken.

De gedragsafhankelijk navigatie programma code voor een robot bestaat globaal uit drieën delen:

1. Robotframe (inclusief alle sensor en actuator componenten)
2. BehaviorController
3. Navigator inclusief localizer

1. Robot frame

Het robotframe is de starter module, het bevat informatie om de sensor en actuator componenten aan te maken. Deze componenten gebruikt het bij het aanmaken van de behaviormodules. Het robot heeft een gebruikers interface voor: drukknoppen, buzzer, leds, RC5, async communicatie en verzorgt de programma hartslag en logging faciliteiten.

Nadat er op de startknop is gedrukt wordt periodiek de BehaviorArbiter aangeroepen. De Navigator wordt door het robotframe gestart in een eigen thread waardoor dit proces verder zelfstandig doorloopt. Het robotframe bevat een lijst van behaviormodules en bij het aanmaken van het BehaviorArbiter component wordt deze lijst van modules mee gegeven, waardoor de BehaviorArbiter transparant is voor meerdere toepassingen. Bij de logging faciliteit kan er voor het vastleggen van de informatie gekozen worden voor een bluetooth verbinding naar een PC of deze informatie naar een SD-card te schrijven. Een klein voordeel van de bluetooth verbinding is dat de PC realtime de logging kan analyseren en deze direct kan weergeven op een beeldscherm.

2. BehaviorController

Het onderstaande schema is overgenomen uit het ExploringRobotics boek op blz. 178

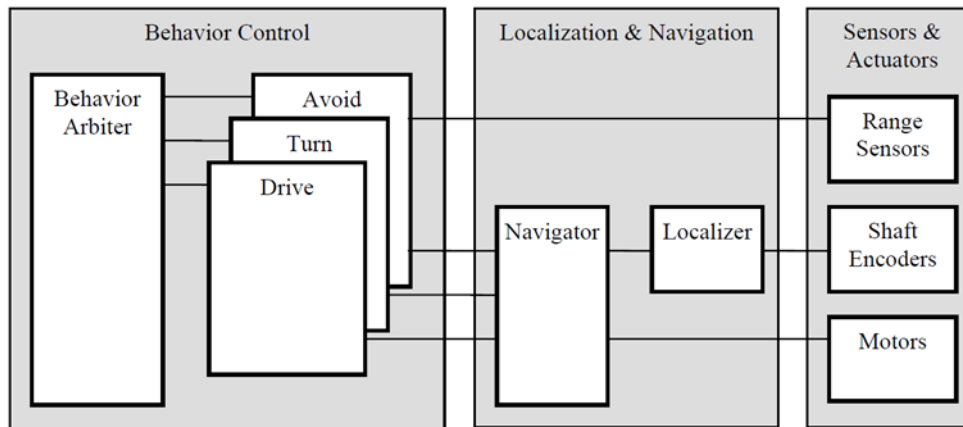


Figure 11-2 – Behavior-based Control Class Diagram

Voor de besturing van de Behaviormodules heeft de Arbiter de mogelijkheid om de modules aan te roepen met de functies "Poll" en "setActive". Van de modulelijst is er altijd maar één module actief. De modules worden opvolgorde aangeroepen met een "Poll" functie en als de "Active" schakelaar van de module aan staat wordt gelijk ook de moduletaak uitgevoerd. Bij de "Poll" functie wordt bepaald of de module afhankelijk van zijn sensor waarden de besturing wil hebben en geeft dit resultaat als return waarde van de poll aanroep terug. De arbiter zal dan afhankelijk van dit poll resultaat en de module prioriteit van actieve module wisselen. In deze versie heeft de arbiter een eenvoudig algoritme voor het bepalen van de module prioriteit, de volgorde in de modulelijst geeft namelijk de prioriteit volgorde aan.

In de Robobits 21 van juni 2003 geeft het artikel "Subsumption architecture" van Paul Wiegman een goede inleiding op het gedragsafhankelijk programmeren. Het gebruik van de "augmented finite state machines" wordt daarin aan de hand van voorbeelden toegelicht. In de behaviormodules worden deze state machines toegepast.

Zoals je in het Behavior-based diagram kan zien worden de motoren door de navigator bestuurd.

Dit wil zeggen dat de toepassingsprogramma's een ander soort rij opdrachten voor de robot hebben dan tot nu toe gebruikelijk was. Er worden nu geen PWM waarden meer meegegeven maar het zijn nu functionele bewegingsopdrachten aan de navigator zoals:

- `turn(int h)`. Roteer de robot met de opgegeven waarde in graden.
- `turn(double h)`. Roteer de robot met de opgegeven waarde in radialen.
- `turnToHeading(int h)`. Roteer de robot absolute naar de opgegeven richting in graden.
- `turnToHeading(double h)`. Roteer de robot absolute naar de opgegeven richting in radialen.
- `driveToPoint(int x, int y)`. Rij de robot naar de opgegeven locatie.
- `driveToPoint(int x, int y, int w)`. Met de parameter "w" kan er voor logging doeleinden een waypoint nummer meegeven.
- `driveHeading(int h)`. Rij de robot in de absolute richting in graden.
- `driveHeading(double h)`. Rij de robot in de absolute richting in radialen.
- `driveDistance(int d)`. Rij de robot de met opgegeven afstand in millimeters.

Voor controle van de beweegopdrachten is de functie `checkAction()`. Deze geeft de return code "true" terug als de laatst gegeven beweegopdracht is afgerond.

Toelichting: Bij een module "lijnvolgen" kan je nu niet meer sturen door de snelheid van de afzonderlijke motoren te regelen, maar je moet nu de `driveHeading` opdracht hiervoor gebruiken. Met de lijnsensors informatie kan de afwijking van de richting van de robot ten opzichte de lijn worden berekend om deze aan de functie `driveHeading` mee te geven.

3. Navigator en localizer

Voor het testen van de programmatuur gebruik ik een FEZ Mini robot welke is voorzien van GM9 motoren van Solarbotics en WW-12 wielsensors van Nubotics. Als ontwikkelomgeving wordt "Visual C# 2010 Express" gebruikt. De FEZ Mini robot heeft een ARM processor en is voorzien van het ".NET Micro Framework" waardoor er veel functionaliteit beschikbaar is.

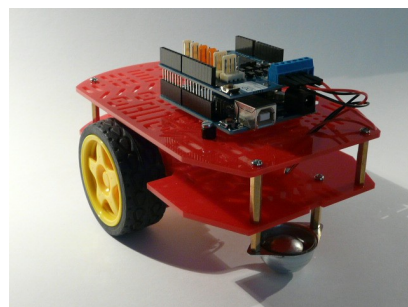
Bij het testen bleek dat de localizer vaak een onbetrouwbare positie aangaf. Bij het zoeken naar de oorzaak werd vastgesteld dat het veroorzaakt wordt doordat niet alle encoderpuls van de wielsensors werden geteld.

Dit probleem deed zich voornamelijk voor als de robot met een hogere snelheid rijdt. Deze fout is voor het navigeren desastreus want met behulp van de wielsensor pulsen wordt de stand van de robot berekend. De ARM processor voorzien van .NETMF van de FEZ Mini robot is soms door de "overhead" van de geboden functionaliteit niet snel genoeg is om alle pulsen te verwerken, dit ondanks de 72 MHz klok snelheid van de processor.

Een oplossing voor dit probleem is om een extra kleine processor te gebruiken voor de localizer functionaliteit, een eerste keuze was de ATtiny85 (deze kost slechts 1,95 euro). Deze processor heeft een I2C verbinding met de FEZ Mini processor. De ATtiny85 heeft net genoeg aansluit pinnen maar wat bleek dat als je alle aansluitpinnen (6 stuks, 4 voor de wielsensors en twee voor de i2c verbinding) gebruikt kan je de processor niet meer met ISP (In circuit Serial Programming) programmeren maar alleen nog met HVS (High Voltage Serial programming) waarvoor ik geen programmer bezit. Daarom verder zoeken en ik heb gekozen voor een Arduino NANO (deze kost in China slechts 7,50 euro). De Nano module heeft tevens het voordeel dat het een compleet klein printje is inclusief USB aansluiting voor het programmeren en debuggen. Dat er veel niet benodigde aansluitpinnen zijn is juist handig bij het testen omdat je die kan tijdelijk kan gebruiken voor extra debug informatie.

De localizer functionaliteit is getest en draait nu op de Arduino Nano, op de FEZ Mini is de oude localizer module gewijzigd en deze haalt zijn informatie nu via de i2c bus op bij de Arduino Nano. Voor de navigator maakt het niet uit waar de locatie berekeningen worden gedaan en dus waren er voor navigator geen aanpassingen nodig. De localizer op de Nano heeft voor zijn berekeningen wel een aantal robot eigenschappen nodig zoals wielbasis, wieldiameter en aantal encoder pulsen per wielomwenteling. Deze informatie kan in het Arduino sketch worden gedefinieerd maar ook bij het opstarten via de I2C bus worden verstuurd naar de Nano, na ontvangst wordt deze informatie in EEprom opgeslagen. Bij het opstarten van de Nano wordt deze informatie uit het EEprom gelezen en met deze waarden de berekeningen uitgevoerd. Door deze functionaliteit is het niet nodig om de Arduino sketch aan te passen als je de Arduino Nano wil gaan gebruiken voor een andere robot.

Het localizer programma op de Nano is volledig met het Arduino sketch geprogrammeerd, de benodigde sinus en cosinus functies en de interrupts voor de wielsensors worden standaard ondersteund door het [Arduino sketch](#).

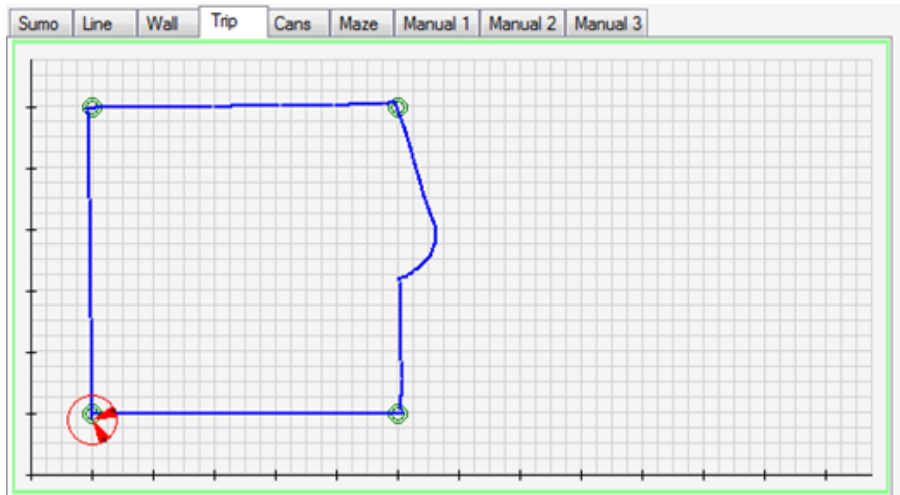


4. Praktijk

De Arbiter geeft de behavior modules de besturing en onderbreekt ze weer. De Navigator zou dit kunnen herkennen door dat een gevraagde opdracht nog niet is afgerond en toch er alweer een nieuwe opdracht van een andere module wordt gegeven. Toch kan de Navigator dit niet echt oplossen omdat de Arbiter de regie over de modules heeft. De gekozen oplossing voor dit probleem is dat de module onthoudt dat het wordt onderbroken (actief schakelaar wordt op uit gezet) en als de module weer actief wordt zal deze moeten bepalen hoe het de onderbroken opdracht verder moet afhandelen. Ingeval van de **driveToPoint** opdracht kan bijvoorbeeld de laatste opdracht zonder meer worden herhaald omdat in deze functie alleen informatie bevat waar het naar toe moet rijden.

Voorbeeld van een "Trip" programma met gedragsafhankelijk modules en navigator en logging.

De behavior lijst is gevuld met twee modules als eerste de AVOIDmodule en als laatste de Tripmodule. Als er geen obstakels worden gedetecteerd krijgt de Tripmodule de periodiek (bv 20Hz) besturing van de Arbiter, en deze zal de navigator opdracht **driveToPoint** geven waarna er vervolgens wordt getest met de **checkAction** of de gegeven opdracht is afgerond. Bij een negatief resultaat wordt de besturing terug gegeven. Bij de volgende cyclus van de Arbiter zal deze als eerste de AVOIDmodule met een Poll aanroepen om de sensors in de gaten te houden, in deze Poll wordt de besturing echter zo snel als mogelijk weer terug gegeven aan de Arbiter met de return waarde waarmee het aangeeft of het de besturing wil hebben. Als de AVOIDmodule geen actie vraagt dan wordt daarna de Tripmodule aanroepen waardoor het nogmaals met de **checkAction** kan afvragen of de robot reeds op het gevraagde punt is aangekomen. Als het wordt onderbroken doordat de Arbiter de zijn "actief" schakelaar uitzet dan onthoudt de module met een state machine waar het gebleven was.



De trip opdracht: rij een parkoers van 4 coördinaten (een vierkantje met elke zijde 1 meter lang). De informatie op het plaatje is realtime gemaakt uit de logging informatie. Op het plaatje wordt er gestart bij het punt linksonder. De Tripmodule krijgt de besturing omdat de sensors geen obstakels detecteren. Er wordt eerst naar het punt rechtsonder gereden. Daarna naar het punt rechtsboven maar halverwege detecteerden de sensors een obstakel. De Tripmodule wordt dan onderbroken en de AVOIDmodule krijgt de besturing en deze zal dan uitwijken.

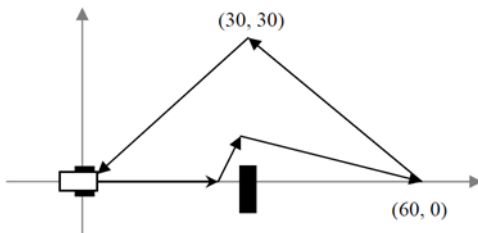


Figure 11-6 – Triangular Path with an Obstacle

Er wordt naar links of rechts gedraaid afhankelijk van de sensors waarden en daarna wordt er nog kleine afstand rechtuit gereden. Na deze uitwijking wordt de oorspronkelijk opdracht van de Tripmodule hervat, echter nu wel van af een andere start positie. Opmerking: door de navigator wordt er alleen op de plaats geroteerd als blijkt dat bij het rijden naar een punt de hoek naar dat punt groter is dan een bepaalde drempel waarde, anders wordt er gewoon met een bocht naar het gevraagde punt gereden.

De trip opdracht komt overeen met de triangel opdracht uit het ExploringRobotics boek op blz. 194.

Daar worden er hints gegeven voor het oplossen van deze opdracht.

Werkgroep

Aan de Robotica leden die geïnteresseerd zijn in het gedragsafhankelijk navigeren met een robot wil ik vragen om deel te nemen aan een werkgroepje die dit onderwerp verder wil gaan beproeven.

Je interesse hiervoor tonen kan door mij een mail te sturen via het contactformulier op de Robotica website.

Begin 2014 wil ik op de Robotica bijeenkomstdag de leden die gereageerd hebben uitnodigen voor een korte presentatie en om de doelstelling van het werkgroepje te bespreken. Aan het eind van die sessie kunnen de deelnemers definitief besluiten of ze mee willen doen met de werkgroep. Van de werkgroepleden wordt wel een actieve bijdrage verwacht.

Hinnie van Sint Annaland

HCC-Robotica ig

Voorzitter:	E.F.O. Buzzi
Secretaris:	M.W.J. van Harmelen
Penningmeester:	H.J. de Gans
Technisch adviseur:	H.M.P. van Sint Annaland
Technisch adviseur:	Z. Otten
Lid/webmaster:	W.C. de Boer
Roborama coordinator:	B. Ruben
Redactie Robotbits:	A. Vreugdenhil, Z. Otten

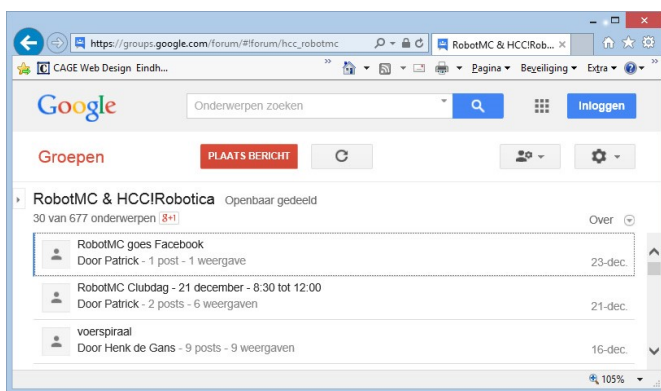
Website: <http://www.hccrobotica.nl>

Discussiegroepen

Maken en delen met groepen :

HCCROBOTICA:

http://groups.google.nl/group/hcc_robotmc

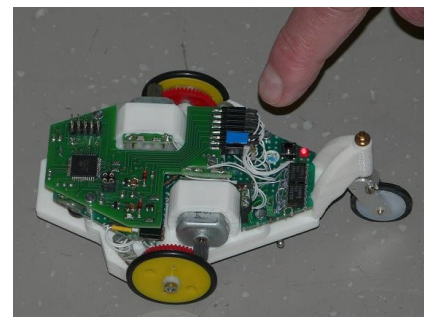


Arduino C cursus

Via een extra Nieuwsbrief, het forum of via Circuit online zijn zeer veel mensen op de hoogte gebracht van ons voornemen voor het organiseren van een cursus Arduino C. (start cursus januari 2014). Mocht je ondanks alles de uitnodiging toch gemist hebben, kijk dan op onze website. Daar zijn alle ins en outs van de cursus te vinden.

<http://www.hcc.nl/webzine/nieuws/hccrobotica-cursus-arduino>

Inmiddels hebben zich 30 mensen voor de cursus ingeschreven maar laat u zich dan niet afschrikken door dit aantal, er is een extra zaal gehuurd in de Dissel. U kunt er nog bij!



AGENDA hcc[®]robotica

Agenda HCC!Robotica tot en met maart 2014

- 4 jan: Maandelijkse bijeenkomst in de Dissel: Arduino C cursus
- 1 febr: Maandelijkse bijeenkomst in de Dissel: ALV & Arduino C cursus
- 1 maart: Ivm carnaval geen bijeenkomst in de Dissel en is verplaatst naar de tweede zaterdag in maart !!
- 8 mrt: Maandelijkse bijeenkomst in de Dissel: Arduino C cursus

Overige activiteiten:

- 21 dec: RobotMC: Maandelijkse bijeenkomst in het Nayer Instituut
- 18 jan: RobotMC: Maandelijkse bijeenkomst in het Nayer Instituut
- 15 feb: RobotMC: Maandelijkse bijeenkomst in het Nayer Instituut
- 15 mrt: RobotMC: Maandelijkse bijeenkomst in het Nayer Instituut

Zie voor bijzonderheden over RobotMC: www.robotmc.org

