

ROBO-

BITS-52

Jaargang 14, nummer 1, maart 2011



Afz. hcc Robotica gg, p.a. Henk de Gans, Koelmanhof 2 3861GG Nijkerk..

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.

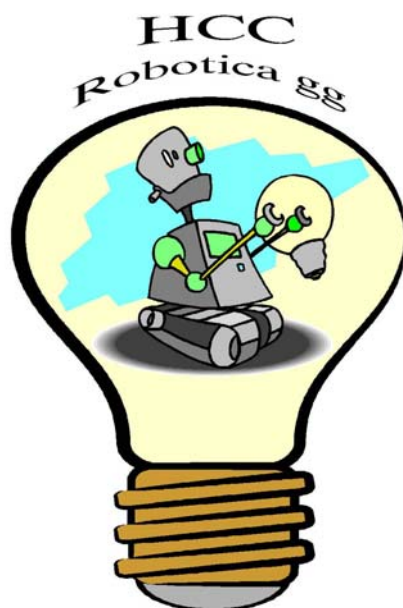
=====
Redactie adres: H.J. de Gans, Koelmanhof 2, 3816GG Nijkerk. hj.de.gans@gmail.com
Tekst aanleveren in WORD of platte tekst in ASCII. Afbeeldingen los er bij in JPG, GIF of BMP formaat.

=====
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. 4 Micromuis Kaasjager door Coen Roos!
- Bladz. 6 PWM of PBM door Jan Blok!
- Bladz. 9 Bouw je eigen comp.Joep Suijs/Karel Dupain
- Bladz. 14 LCD/I2C/BASCOM AVR door Rien v Harmelen.
- Bladz. 16 Onze Roborama sponsor!!!!
- Bladz. 17 Microsoft Robotics Development Studio
Introductie door Iwan Tolboom
- Bladz. 26 Microsoft Robotics Development Studio Joystick
gestuurde NXT door Iwan Tolboom



Ja, en dan schijnt de zon weer na een lange winter, en denk je na over al die lange winteravonden! Zou er nog flink wat hobby plezier zijn geweest deze winter? Ik kan u zeggen: dat was er en is er nog steeds!! En dat kunt u allen zien en lezen in deze robobits! Ik kreeg namelijk weer flink wat copy binnen, daarvoor allen hartelijk dank!!

Ik wens u allen veel leesplezier toe, en een mooie lente!

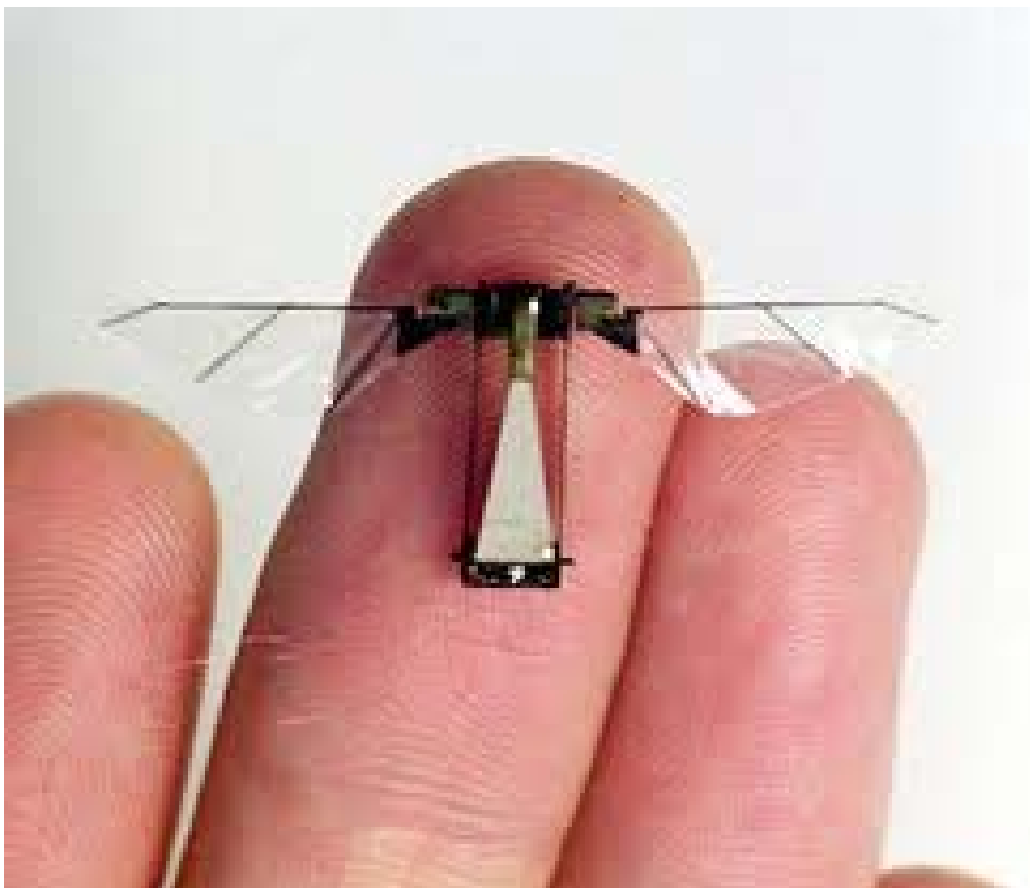
PS de artikelen van Iwan Tolboon staan ook als PDF apart op de website onder projecten!

Henk de Gans

deadline deadline deadline deadline deadline deadline deadline deadline

De deadline voor kopie voor Robobits 53 die eind juni 2011 zal uitkomen, is gesteld op zaterdag 18 juni 2011!!! 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



Mikromuis Kaasjager

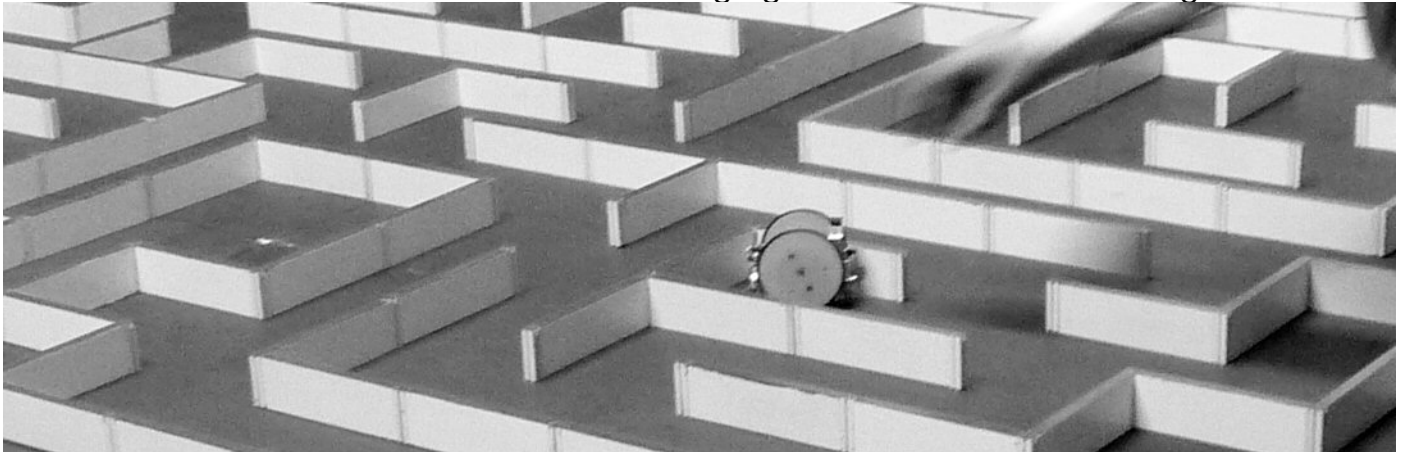
Door Coen Roos

Bij mikromuis of in gewoon nederlands 'micromouse' gaat het om het oplossen van een doolhof door een autonoom werkende robotmuis. De historie van micromouse gaat terug naar 1977 toen in New York een wedstrijd werd georganiseerd die 6000 inschrijvingen opleverde waarvan er uiteindelijk 15 van aan de start verschenen. Micromouse heeft daarna goed wortel geschoten want sindsdien worden over de hele wereld wedstrijden gehouden waaronder Japan en Engeland.

In een race tegen de klok start de muis in een hoek van het doolhof op zoek naar het midden. Het doolhof wordt opgebouwd uit 5 cm hoge muren op een 18 cm raster. Dit matrix meet 16 x 16 en is dus ongeveer 3 meter in het vierkant. Een muur is 12 mm dik waardoor een padbreedte van 16,8 mm ontstaat.

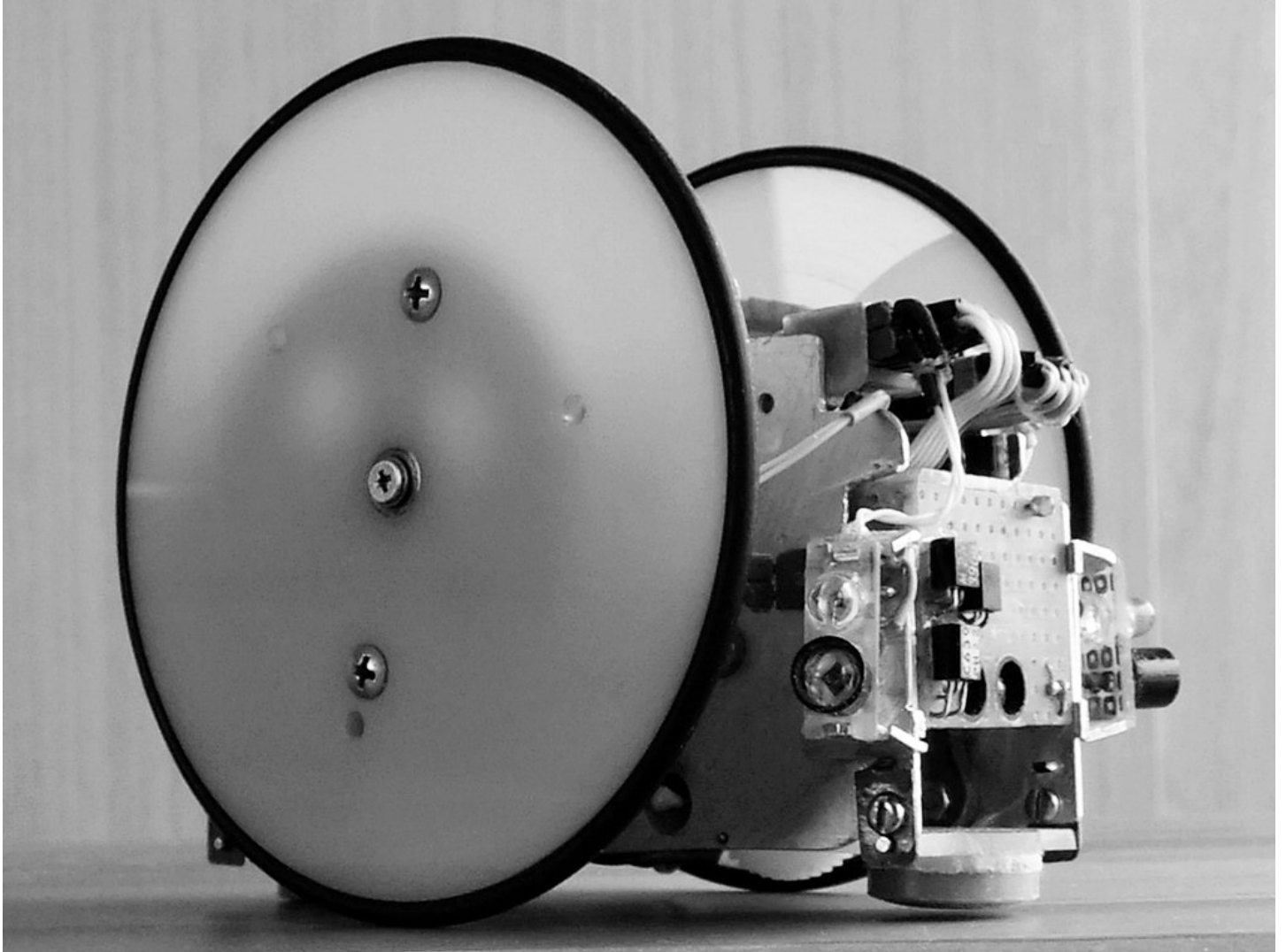
Het midden bestaat uit de centrale vier vakken van het matrix. De muis start in de hoek linksonder. In tien minuten moet het midden zijn gevonden, de snelste weg zijn bepaald en deze in de kortste tijd zijn afgelegd. Als de tijd na het vinden van het midden toereikend is kan de muis besluiten het doolhof verder in kaart te brengen op zoek naar een kortere- dan wel snellere route.

De snelste muis heeft gewonnen. Bepalend hierbij is de scoretijd en deze wordt berekend uit de runtijd, zijnde de tijd van start naar het midden na het vinden van de kortste route, met daarbij opgeteld 1/30 van de mazitijd. Deze mazitijd gaat in op het moment dat de muis voor het eerst in beweging komt tot aan de aanvang van de 'run'.



Met mijn muis Kaasjager heb ik een aantal keren meegedaan aan 'UK micromouse' dat plaatsvindt in het Technical Innovation Center in Birmingham. Door telkens wat te verbeteren is Kaasjager inmiddels wel zover vooruit gegaan dat in een redelijke tijd met redelijke kans op slagen een score wordt bereikt die in de buurt van de middenmoot van het klassement uitkomt. Veel beter zal door gebrekkigheden aan dit platform niet gaan lukken. De laatste en nog niet beproefde uitbreiding is het diagonaal kruisen van een zigzagpad of in jargon een 'staircase'. Een staircase is al sowieso een uitdaging waar menige muis op strandt want bij ondermaatse precisie van verplaatsen en draaien komt de muis al snel op ramkoers met een muur te liggen en kan hooguit met een herstart nog geprobeerd worden er iets van te maken. De prijs hiervan bedraagt een 3 seconden 'touchpenalty' op te tellen bij de runtijd.

In januari heb ik in 'de Dissel' in Hooglanderveen een show met Kaasjager gegeven. Met losstaande muurtjes is in de roboramabak een subdoolhof uitgezet, 'het midden' zat daardoor niet bepaald in het midden. Het grote voordeel was dat ik zelf het gangenstelsel kon inrichten en zo de gebreken van de muis buiten beeld kon houden wat niet helemaal lukte. De muis probeerde als een dolle stier een muur te slechten wat nog aardig lukte op de gladde ondergrond. Door de muurtjes iets meer precies te plaatsen kon het midden worden gevonden en werd de race vanaf de startpositie ingezet.



PWM of PBM!!

Door Jan Blok

Vooraf duidelijk stellen: dit verhaal geldt voor kleine DC-motoren dus max 24 volt, hooguit 1 amp en permanente magneet met commutator.

En we willen dat de motor zich ongeveer net zo gedraagt als wanneer gevoed met gelijkspanning. Zowel voor regeling toerental als voor gedrag bij belasting.

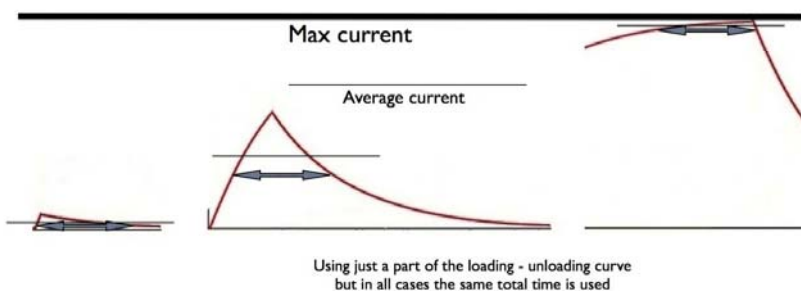
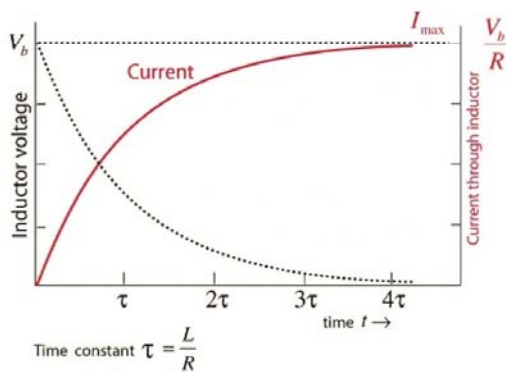
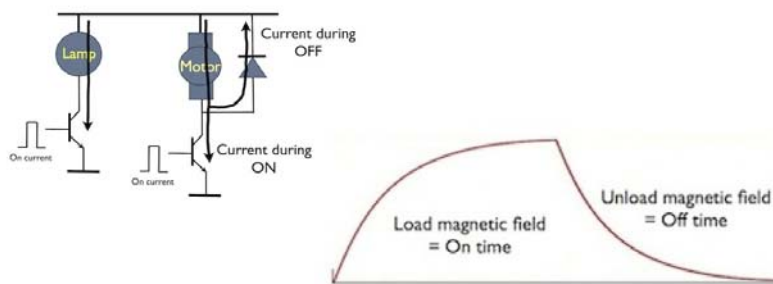
De gedachte van velen achter PWM is dat, als er slechts een deel van de tijd voedingspanning wordt aangesloten op een motor, deze ook slechts een deel van zijn toerental en vermogen zal afgeven.

Voor een lamp geldt inderdaad dat de hoeveelheid afgegeven licht een duidelijke relatie heeft met het toegevoerde vermogen.

De vraag is of dit ook zo simpel is voor een motor.

Er is al direct het feit dat de lamp niet door een motor vervangen kan worden en andersom ook al niet. Zie eerste plaatje.

De diode is bijgevoegd omdat, als de stroom door de motor onderbroken wordt, er een grote positieve spanningpiek op de collector van de transistor ontstaat maar:



Wat er gebeurt is dit: als de transistor geleidt, loopt er stroom de motor in die een magnetisch veld opbouwt.

Een zelfinductor verzet zich tegen een verandering van stroom hetgeen betekent dat bij abrupt afschakelen van de stroom er een positieve spanning op de collector zal optreden maar dat gebeurt niet want de diode gaat in geleiding en de stroom door deze diode loopt in dezelfde richting bij het afbreken van het veld als toen het veld werd opgebouwd.

De stroom is de belangrijke faktor want die levert het koppel. Het gaat er dus helemaal niet om wat de gemiddelde spanning is die over de motor staat maar wel over de gemiddelde stroom die altijd in dezelfde richting door de rotor loopt.

En die gemiddelde stroom moet bij voorkeur continu zijn dwz geen al te grote variatie vertonen.

Als het goed is, is het wel zo dat de gemiddelde spanning over de motor een goede maat is voor de stroom die door de motor loopt.

Nu de schakelfrequentie.

Eerst wordt een hoge frequentie gekozen.

Wat hoog is laten we nog even in het midden, we kijken naar wat er in de motor gebeurt.

De motortijdconstante zegt alles over welke tijd nodig is om het maximale veld in de motor op te kunnen bouwen.

Wat je kunt zien is dat bij het toenemende percentage van de laadtijd de gemiddelde stroom door de motor toeneemt.

En nog iets belangrijks.

Een draaiende motor is tevens een generator. Die generatorspanning wordt afgetrokken van de voedingspanning. De spanning die de stroom opbouwt is altijd kleiner dan de voedingspanning. Maar bij een PWM van tien procent wordt wel de volle voedingspanning gebruikt maar de tegen EMK van de

motor is nog klein. Ofwel, als de motor langzaam zou moeten lopen is hij moeilijk in de hand te houden want er gaat al heel gauw een grote stroom lopen.

En dat is nu juist mijn stelling: langzaam lopen van de motor lukt minder gemakkelijk want er gaat gauw een grote laadstroom lopen.

Als je in de buurt van PWM honderd procent komt maakt het allemaal niet meer zoveel uit; de motor staat dan gewoon aan.

Stel nu eens dat we ons niets aantrekken van die motortijdconstante en de frequentie verlagen. Zeg 100 Hz.

Stel de dutycycle in op 50 procent en kijk wat er gebeurt.

Er staat nu een wisselspanning van 100 Hz over de motor. En wat doet een DC motor op wisselspanning? Juist, niets.

Dit is ook weer niet helemaal waar want er gaat een flinke wisselspanning door de motor lopen omdat het inductieve deel van de totale impedantie weinig of niets doet bij deze frequentie.

De stroom wordt begrensd door het ohmse deel van de totale impedantie en die is heel klein. Er gaat dus een grote stroom lopen en die warmt te motor op en maakt vaak ook geluid. Pas als de dutycycle in de buurt van de 80 procent komt worden deze effecten minder en zal de motor gaan draaien.

Echter gelijk op bijna volle snelheid en dat is ook niet de bedoeling. En waar denk je dat die wisselstroom vandaan komt? Ja, uit de accu en die is op die manier gauw leeg.

Dan nu de conclusie: Is PWM bruikbaar? Het antwoord is : JA, als je het goed doet wel. Hooguit is de regeling aan de lage kant niet soepel.

Nog een opmerking: De PWM schakeling zoals die meestal gepresenteerd wordt is prima geschikt om van draairichting te wisselen.

Referenties:

Lezenswaard is:

<http://www.4qdtype.com/pwm-01.html>

Het plaatje met stroomcurve komt van:

<http://hyperphysics.phy-astr.gsu.edu/Hbase/electric/indra.html#c1>

Gebruik de Time Processing Module van de HCS08 voor PWM timing

http://www.freescale.com/webapp/sps/site/overview.jsp?code=784_LPBB_HCS08TPM

Koppel, snelheid en vermogen van motor:

<http://lancet.mit.edu/motors/motors3.html>

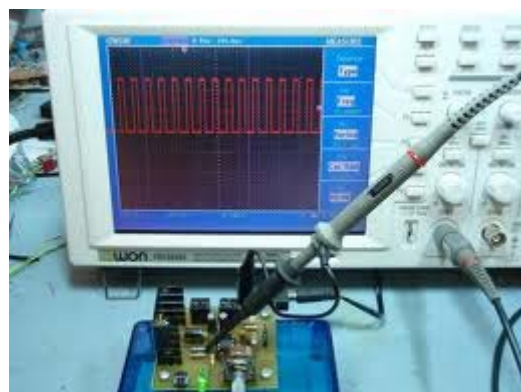
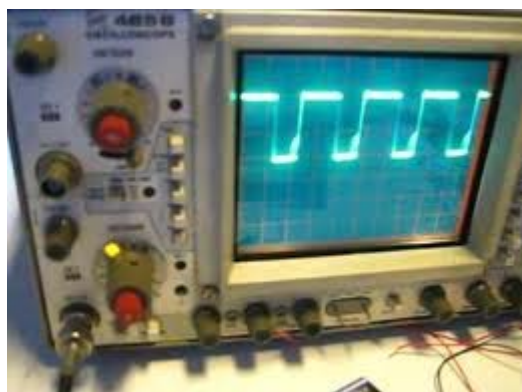
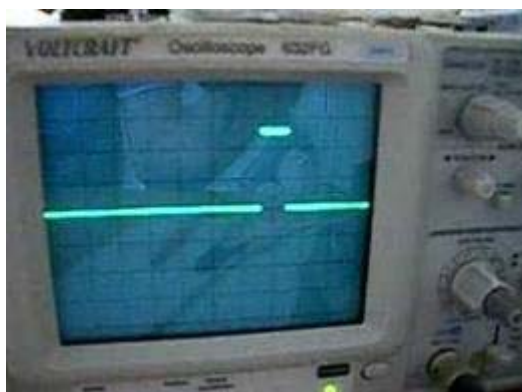
Een voorbeeld van soepel sturen met PWM:

<http://emse.nl/blikvanger.html>

Filmpje met groter beeld:

<http://vimeo.com/14898934>

red.leuke plaatjes om blad te vullen, maar past wel bij verhaal ;-)



Bouw je eigen computer - deel 1

Beste Roboteer,

Als dit artikel zou beginnen over een moederbord met Intel processor en de waar je op moet letten bij een voeding in een ATX computerbehuizing, zou je teleurgesteld zijn. Op deze manier modules samenvoegen verschilt niet wezenlijk van het kopen van een complete PC en hierop de monitor aansluiten. Op deze plaats verwacht je een verhaal over een Atmel microcontroller, of eentje van Microchip, zoals de 16F690.

Maar... is dit wel zo verschillend? Ook een 16F690 is een complete module, waarop alleen voedingsspanning aangesloten hoeft te worden om werkend te maken. Okay, je moet wat moeite doen om een programmatje te maken en in de chip te zetten om bijvoorbeeld een ledje te laten knipperen. Bill Gates en Linus Torvalds hebben met hun software de PC eenvoudiger toegankelijk gemaakt. Maar als we naar de hardware kijken is er niet zoveel verschil: je koopt een complete module en hoeft je niet te verdiepen in de details van de werking.

Toch zijn die details interessant, leerzaam en minder magisch dan we wellicht zou denken. Een microcontroller is eigenlijk een 'computer op een chip'. In vergelijking met een PC is dit een hele eenvoudige computer en de werking van een microcontroller lijkt dan nog veel op de werking van de 'Little Man Computer' (LMC), een 'educatief model' van de werking van computers, dat zo'n 45 jaar geleden is ontwikkeld.

De 'Little Man Computer' is opgebouwd uit de volgende onderdelen:

- 100 postvakjes, genummerd van 0 tot en met 99.
- Een programmateller.
- Een rekenmachine.
- Een input-stapel en een output stapel.
- Het mannetje (m/v) waaraan de computer zijn naam dankt en die het werk doet.

Dat 'werk' van het mannetje is het uitvoeren van de stapjes (statements) van het programma. Het mannetje begint hiervoor bij de programmateller (program counter). Bij de start (na reset van de computer) staat deze teller op 0 en dit is voor het mannetje een teken dat hij naar postvakje 0 moet gaan om de waarde af te lezen van het briefje wat daarin ligt. Deze waarde is de opdracht (instructie of opcode) en bestaat uit een getal van 3 cijfers, soms voorafgegaan door een '*' of een '#'. Straks moet hij deze opcode uitvoeren. Maar eerst onthoudt hij de opcode en drukt op de knop van de programmateller, zodat die met 1 wordt verhoogd.

Met het uitlezen van het opcode en ophogen van de programmateller is het ophalen van de instructie (fetch) afgerond en is het mannetje toe aan uitvoering van de instructie (execute)

Het meest linkse cijfer van de zojuist opgehaalde opcode geeft aan wat het mannetje moet doen. Zo betekent bijvoorbeeld '1' dat hij een getal moet invoeren op het rekenmachine. En de rechtse twee cijfers geven – samen met een eventueel '*' of '#' – aan welk getal bij de opdracht hoort en – in dit specifieke geval – ingevoerd moet worden op de rekenmachine.

Instructie	Mnemonic	Opcode	Omschrijving
LOAD	LDA xx	1xx	Lees het 3-cijferige nummer in postvakje xx en voer dit in op de rekenmachine

LOAD	LDA #xx	#1xx	Voer xx in op de rekenmachine
LOAD	LDA *xx	*1xx	Lees de laagste twee cijfers in postvakje xx, lees het 3-cijferige nummer in dat postvakje en voer dat laatste nummer in op de rekenmachine

Met andere woorden: als het mannetje opcode '123' leest in postvak 0, zal hij naar postvakje 23 gaan, leest daar het nummer uit, bijvoorbeeld 199, en zal hij dit nummer invoeren op de rekenmachine. De beide postvakjes zijn dus onveranderd en op het scherm van de rekenmachine staat 199. Had het mannetje echter #123 gelezen in postvak 0, dan was hij direct naar de rekenmachine gegaan en daar 23 ingevoerd. Opcodes met een * werken op een vergelijkbare manier, maar laten we verder buiten beschouwing.

In voorgaande tabel is naast de opcode en omschrijving ook een kolom 'Instructie' en een kolom 'Mnemonic' opgenomen. De instructie is de naam van de opdracht en de mnemonic is een hulpmiddel om het menselijk geheugen te ondersteunen, een symbolische naam die wij eenvoudiger kunnen onthouden dan de opcode. In dit geval is gekozen voor LDA en dat staat voor Load-Accumulator, het laden van de 'verzamelbak' – het scherm van de rekenmachine.

Dit mannetje kent de volgende instructies:

Instructie	Mnemonic	Opcode	Omschrijving
LOAD	LDA xx	1XX	Laad Accu met waarde uit postvak xx
STORE	STA xx	2XX	Sla de waarde van de Accu op in postvak xx
ADD	ADD xx	3XX	Tel de waarde uit postvak xx op bij de Accu
SUB	SUB xx	4XX	Trek de waarde uit postvak xx af van de Accu
INPUT	IN	500	Haal een waarde van de in-stapel af en stop deze in de Accu
OUTPUT	OUT	600	Leg een nieuw briefje op de uit-stapel met de waarde van de Accu
HALT	HLT	700	Stop (het mannetje is klaar met werken)
SKIP	SKN	800	Skip-Negative – sla de volgende instructie over als de waarde van de Accu negatief is
	SKZ	801	Skip-Zero – sla de volgende instructie over als de waarde van de Accu nul (Zero) is
	SKP	802	Skip-Positive – sla de volgende instructie over als de waarde van de Accu positief (of nul) is
JUMP	JMP xx	9XX	Stop de waarde xxx in de programmateller

STORE, ADD en SUB werken vergelijkbaar met LOAD. Bij INPUT gaat het mannetje naar het stapeltje 'IN' en haalt hier een vel papier met een nummer af. De volgende keer dat hij iets van 'IN' ophaalt, zal dat dus een andere waarde zijn! OUTPUT werkt vergelijkbaar met INPUT en HALT betekent dat het mannetje klaar is met zijn werk en kan stoppen.

De SKIP instructies zijn geen bewerkingen op getallen, maar beïnvloeden het verloop van het programma. Om precies te zijn: de SKIP instructies slaan de eerstvolgende instructie over als de waarde van de Accu (het scherm van de rekenmachine) aan de bijbehorende voorwaarde voldoet. Die volgende instructie kan iedere opcode zijn, bijvoorbeeld het laden van de Accu met 00 of een

JUMP. De taak van het mannetje is het uitvoeren van de controle (Negative, Zero of Positive) en als dit klopt, op de knop van de programmateller drukken.

De laatste instructie uit de serie is JUMP. Bij deze instructie voert het mannetje de laatste twee cijfers van de instructie in op de programmateller. De uitvoering van het programma wordt nu op een heel ander postvak voortgezet.

Van een aantal van deze instructies zijn ook varianten met een '#' of een '*' en verder worden door sommige LMC-modellen afwijkende instructies gebruikt. Het gaat te ver om al deze instructies hier in detail te beschrijven. Aan het einde van dit artikel staan wat links van sites met LMC informatie en meestal ook (online) simulators. Probeer bijvoorbeeld eens het volgende programma:

```
00 in;  
01 add #05;  
02 sta 80;  
03 out;  
04 hlt;
```

Het is opgebouwd uit de hiervoor beschreven mnemonics. Voor iedere mnemonic staat het adres waarop de betreffende opdracht opgeslagen wordt. Op de simulator van <http://www.acs.ilstu.edu/faculty/javila/lmc/> kun je met de knop 'Get Opcode' de broncode vertalen naar opcodes. Het resultaat is:

```
00 500;  
01 #305;  
02 280;  
03 600;  
04 700;
```

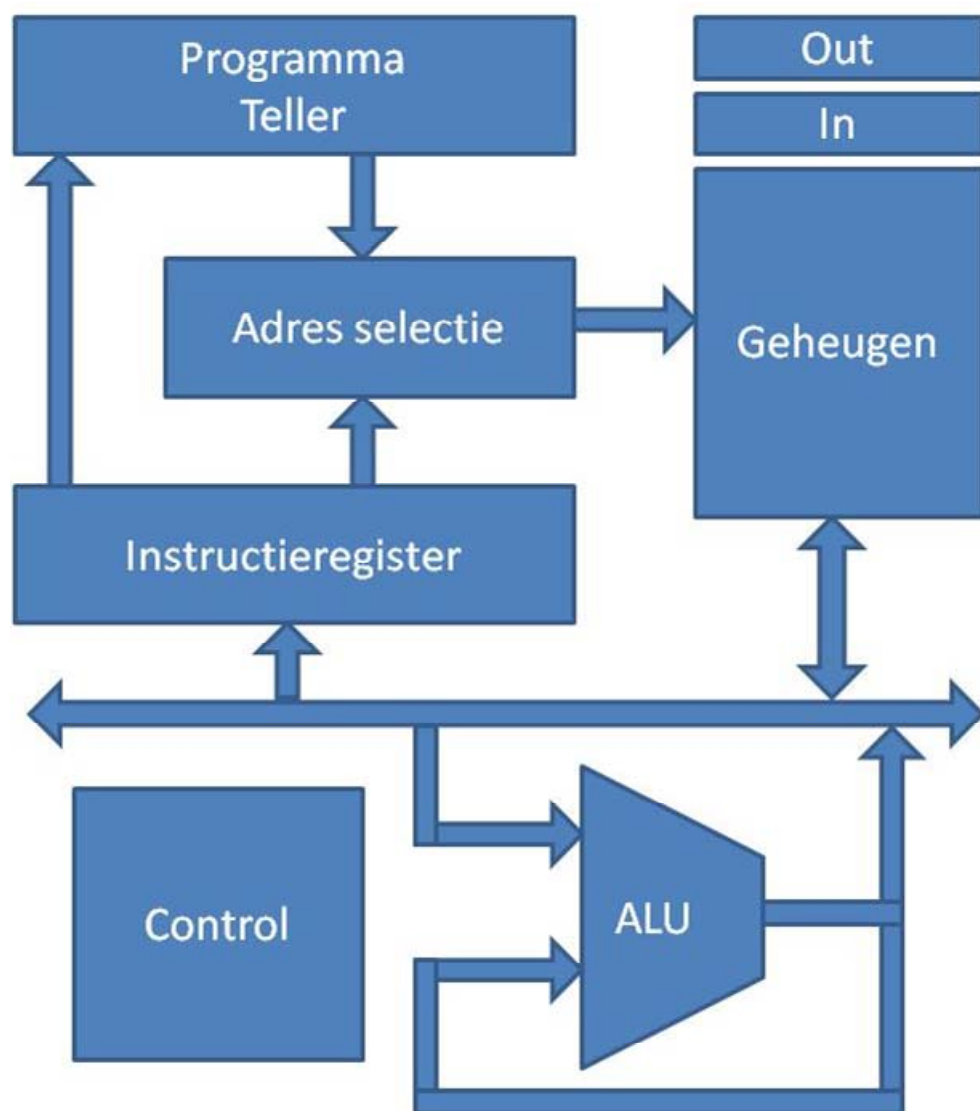
Met de knop 'Load' laad je de code in de postvakjes. Gebruik het veld 'Input' om iets op de invoer-stapel te leggen en druk op 'Execute' om het programma uit te voeren. Kijk daarna wat er op de output-stapel en in postvak 80 ligt...

Wat we nu gedaan hebben is de basis voor iedere moderne computer. We kunnen code schrijven in mnemonics en die met een assembler in target code (hex) vertalen. Met een programmer of een downloader ('load') zetten we deze getallen in het geheugen van de computer (in de postvakjes) en na een reset wordt deze uitgevoerd.

Natuurlijk zijn computers sinds de jaren 40 van de vorige eeuw veel geavanceerder: ze hebben meer en krachtigere instructies, meer geheugen en zijn steeds sneller geworden. Maar alle berekeningen die moderne computers uit kunnen voeren, kunnen ook door de Little Man Computer worden uitgevoerd als je hier de tijd voor neemt en deze meer geheugen zou hebben!

Laten we eens kijken naar de opbouw van deze computer. Het centrale component is de rekenmachine – in computerjargon een Arithmetic Logical Unit (ALU) – die bewerkingen uitvoert. Om resultaten op te slaan heeft de computer geheugen nodig – de postvakjes – en ditzelfde geheugen wordt ook gebruikt om de instructies op te slaan. IN en OUT zijn bij de LMC uitgevoerd als stapel, maar als we ons beperkten tot 1 waarde tegelijk, kunnen we hiervoor een geheugenplaats gebruiken.

Om instructies te verwerken moet het mannetje deze uit het geheugen lezen en even onthouden. Voor dit 'onthouden' wordt het 'instructieregister' gebruikt. De 'programmateller' wordt gebruikt om bij te houden waar de uitvoering van het programma is gebleven. Schematisch ziet dit er als volgt uit:



Het is opgebouwd uit de genoemde onderdelen, uitgebreid met 'adres selectie' en 'control'. 'Adres selectie' zorgt voor de keuze van het postvak: in eerste instantie gebruikt het mannetje hiervoor de waarde van de programmateller, maar bij een instructie zonder '#' gebruikt hij daarna de laatste twee cijfers van de instructie om een waarde uit een volgend postvak op te halen. De overige taken, zoals het intoetsen op de rekenmachine en weleggen van nieuwe briefjes in de postvakjes – de besturing van het mannetje – is ondergebracht in 'control'. Tot slot zijn er verschillende pijlen aangegeven. Dit zijn de 'paden' waarover het mannetje getallen en opcodes verplaatst.

Dit blokschema geeft de opbouw van een computer aan die gerealiseerd kan worden met logische componenten: NAND poortjes, D-flip-flops etc. Een dergelijke computer kun je bouwen met digitale poorten uit de jaren 60 van de vorige eeuw, of zelfs met vacuüm buizen of relais. We komen hiermee op het terrein van de computer-pioniers uit de jaren 40 van de vorige eeuw, als de [Atanasoff-Berry Computer](#) en de [Manchester Small-Scale Experimental Machine \(SSEM\)](#).

Zou het mogelijk zijn om een dergelijke computer te maken en bijvoorbeeld een variant van het [eerste programma van de SSEM](#) te draaien? Het lijkt ons een boeiende uitdaging en we zullen jullie in een vervolg op dit artikel laten weten hoe het ons is vergaan.

Karel Dupain & Joep Suijs

Little man computer links.

Dit artikel is gebaseerd op de LMC implementatie van

- <http://www.acs.ilstu.edu/faculty/javila/lmc/>.

De volgende links geven meer informatie over LMC, maar let op: niet alle LMC's zijn compatibel!

- http://teaching.idallen.org/dat2343/10f/notes/301_LMC.html
- <http://home.planet.nl/~leemb057/lmc/LittleManComputer.htm>
- <http://www.atkinson.yorku.ca/~sychen/research/LMC/LMCHome.html>



Aansturing LCD via I2C met Bascom AVR

Door Rien van Harmelen

Het gebeurt al snel dat het controllerbordje te weinig aansluitpoorten heeft, zeker bij gebruik van een LCD display (6 poorten in gebruik). Je kunt dit ondervangen door gebruik te maken van een I2C-LCD display met slechts 2 poorten in gebruik, waardoor er 4 poorten vrij komen. Je schrikt alleen een beetje van het prijsverschil tussen een "normale" LCD (1) en een I2C-LCD. Het blijkt ook mogelijk een "normale" LCD met I2C te kunnen aansturen met Bascom AVR via een portexpander PCF8574P (2). Toen ik op deze mogelijkheid gewezen werd, bleek deze methode (om wat voor reden dan ook) niet te werken met mijn LCD's. Er werkte bij mij slechts één LCD uit een serie van vier (goedkoop aangeschaft als restpost waarvan mogelijk de built-in controller niet helemaal compatible is met de HD44780 ??). Ook op het Bascom forum worden er soms vragen over gesteld.... zonder overigens duidelijke antwoorden. De methode liet mij niet los. Ik had echter geen idee wat voor LCD ik moest bestellen. Bij een beetje zoeken op het web kwam ik op de site www.seeedstudio.com met redelijke goedkope LCD's en..... waarin de methode met de portexpander PCF8574P wordt vermeld. Op mijn vraag of hun LCD's met de PCF8574P gebruikt kunnen worden kreeg ik een positief antwoord. Ik heb (toch voorzichtig als ik ben) slechts één LCD 4 x 20 characters besteld en..... deze LCD kan inderdaad met I2C aangestuurd worden via een PCF8574P.

Er zijn natuurlijk nog "tientallen" andere LCD's die ook op deze manier aangestuurd kunnen worden . Ik weet alleen niet waar je op letten moet.

Aansluitingen Portexpander PCF8574P

1	ground		
2	ground		zie voor uitleg (2)
3	ground		
4(P0)	naar DB4-LCD		
5(P1)	naar DB5-LCD		
6(P2)	naar DB6-LCD		
7(P3)	naar DB7-LCD		
8	ground		
9(P4)	naar RS-LCD		
10(P5)	naar RW-LCD		
11			
12(P7)	naar E1-LCD		
13			
14	naar SCL-controller		
15	naar SDA-controller		
16	5 volt		

Overige aansluitingen LCD

1	ground
2	5 volt
3	middenaftakking 10 k var weerstand

- (1) *Onder een "normale" LCD versta ik een LCD met een built-in controller HD44780 or equivalent*
- (2) *Zie www.mcselect.com , Bascom AVR , Application notes, AN#118 (I2C LCD & Keyboard Library)*

Bascom programma

\$regfile = "m8def.dat"

\$crystal = 16000000

\$baud = 19200

\$hwstack = 75

\$swstack = 50

\$framesize = 75

\$lib "Lcd_i2c.lib"

Config I2cdelay = 1

Const Pcf8574_lcd = &H40

Config Scl = Portc.0

Config Sda = Portc.1

Dim _lcd_e As Byte

_lcd_e = 192

Config Lcd = 20 * 4

Cls

Cursor Off

Locate 1 , 1

Lcd "Eindelijk"

Wait 1

Locate 2 , 1

Wait 1

Lcd "123456789"

Wait 1

Locate 3 , 1

Lcd "!@#\$%^&*()-_="+

Wait 1

Locate 4 , 1

Lcd "Gelukt !"

Wait 1

Microsoft Robotics Development Studio

Introductie



Index

Inleiding	2
Introductie van MRDS	2
Installatie van Visual Studio	3
Installatie van MRDS	3
Systeemeisen	3
Installeren	4
Rondleiding door MRDS	5
Het Start Menu	5
De onderdelen van MRDS	6
Visual Simulation Environment	6
Codeplex samples	8

Inleiding

Microsoft Robotics Development Studio(MRDS) is een omgeving voor het component based programmeren, simuleren en besturen van robots.

In dit document wordt eerst MRDS kort beschreven, gevolgd door de installatie handleiding en een uitgebreide rondleiding.

Dit document heeft onderstaande opbouw

- **Introductie van MRDS**
In dit hoofdstuk wordt kort beschreven wat MRDS inhoudt, hoe het werkt en waarvoor het gebruikt kan worden. Na dit hoofdstuk gelezen te hebben moet je een idee hebben waarom je het zou gaan gebruiken.
- **Installatie van Visual Studio**
Voor het kunnen compileren van codeplex samples waaronder sumo bots, een robot voetbal simulator en een Fischer Technik Robot moet Visual Studio geïnstalleerd worden. De samples zijn hier te vinden <http://mrdssamples.codeplex.com/> Ben je niet geïnteresseerd in deze voorbeelden dan kun je ook zonder Visual Studio. Wil je eigen componenten toevoegen aan MRDS dan zul je toch Visual Studio moeten installeren.
- **Installatie van MRDS**
Waar kan je MRDS vandaan halen, wat zijn de systeemeisen en hoe moet je het installeren. Het installeren is vrij eenvoudig en ook deinstalleren is simpel. Wanneer je nog wat schijfruimte (400MB) over hebt, is het aan te raden MRDS meteen te installeren zodat je het hoofdstuk "
- **Rondleiding door MRDS"** kunt volgen.
- **Rondleiding door MRDS**
Hierin worden alle onderdelen van MRDS met voorbeelden beschreven. Omdat niet alle onderdelen even relevant zijn om een robot te besturen zal ook aangegeven worden wat de belangrijkste onderdelen zijn.

Introductie van MRDS

Microsoft Robotics Development Studio wordt op een PC of laptop geïnstalleerd. Met de ontwikkelomgeving kan visueel geprogrammeerd worden dankzij de VPL (Visual Programming Language) door het aan elkaar verbinden en instellen van functie blokken. Naast de reeds beschikbare blokken kan een programmeur ook in Visual Basic, C++ of C# zijn eigen blokken maken. Daarmee kan bijvoorbeeld interactie met Domotica of een spreadsheet plaatsvinden.

Voor het besturen van de robot hoeft er slechts éénmalig een programma op de robot geïnstalleerd te worden dat als doorgeefluik van signalen en commando's dient. Dit is bij Lego Mindstorms NXT al standaard aanwezig.

MRDS ontvangt waarden van de sensoren en stuurt de robot commando's voor de motoren of nieuwe instellings parameters voor de sensoren.

Wanneer een programma gerund wordt starten de CCR (Concurrency and Coordination Runtime) welke voor het parallel afhandelen van processen zorgt (sensors uitlezen en motoren aansturen) en de DSS (Decentralized Software Services) welke het verbinden van de programma blokken uitvoert en hun onderlinge communicatie verzorgt.

Deze laatste start tegelijkertijd een webserver die webpagina's biedt waarop de sensorwaarden of commando's gevisualiseerd worden. Dit ziet eruit als een dashboard van een auto dus niet erg gedetailleerd maar met de "DSS log analyser" of eigen programma's kunnen deze gegevens

Een programma kan in de VSE (Visual Simulation Environment) geprobeerd worden. Deze virtuele omgeving kan gebruikt worden om je robot te testen en volgt natuurwetten als zwaartekracht, kinetische energie en windkracht. Op een motorbasis kun je de positie voor je sensoren kiezen en in bijvoorbeeld een gesimuleerd appartement testen of de gekozen positie de meest geschikte is en niet de robot te hoog maakt om onder een bank door te rijden. Meer informatie is te vinden op <http://www.microsoft.com/robotics/>

Installatie van Visual Studio

Zoals vermeld op de website van codeplex

<http://mrdsamples.codeplex.com/documentation?referringTitle=Home> is C-sharp (c#) de meest ondersteunde taal voor MRDS.

Daarom wordt hier eerst beschreven hoe je c# kunt installeren maar hetzelfde verhaal geldt voor C++ en VB. Installeer VB in ieder geval ook alvast om VB samples te kunnen compileren.

Ga naar <http://www.microsoft.com/visualstudio/en-us/> en klik op de "Express" button. Op de Nederlandse site is "VS Express" niet te downloaden. Scroll naar beneden en klik op "Visual Studio C# 2010". Selecteer aan de rechter kant "English" en klik op "Install Now". Druk op "Run" in de dialoog van de installer. Eerst opslaan is weinig zinvol want daarmee sla je alleen de installer op. In mijn geval moet ik mijn firewall toestemming laten geven aan het "Setup" programma. Zet het vinkje of verwijder het bij "Yes, send information about my setup experiences to Microsoft Corporation." en druk de "Next" button. Klik de radiobutton naast "I have read and accept the license terms." en druk op "Next". Accepteer het standaard pad en druk "Install". Nu worden alle onderdelen gedownload en bij 2Mbit/sec duurt dat ongeveer een minuutje. Daarna begint de installatie welke ongeveer 15 minuten duurt. Gedurende de installatie vraagt Windows nog om toestemming om de Setup internet toegang te geven, dus het is niet alleen maar achteroverleunen. Kort hierna moet de computer herstart worden.

Als je Visual Studio langer dan 4 weken wil gebruiken zul je gratis moeten registreren met je Microsoft Live ID. Registreren vindt je in het "Help" menu.

Installatie van MRDS

Op de website van <http://www.microsoft.com/robotics/> staat rechts midden "Get Microsoft Robotics Developer Studio 2008 R3". Op het moment van dit schrijven dd 1-2-2010 is dat de laatste release welke slechts 6 maanden oud is. Bij Release 2 moest voor sommige mogelijkheden nog betaald worden. Hier zijn de releasenotes <http://go.microsoft.com/fwlink/?LinkID=133306> te vinden. Registreren is overigens niet nodig. Het hele installatie proces kost ongeveer een uur.

Systeemeisen

De systeemeisen voor de ontwikkelomgeving en runtime (CCR en DSS samen) zijn Windows XP Service Pack2, Windows Vista of Windows 7 inclusief de x64 edities. De CCR en DSS runtime draaien echter ook op Windows XP embedded.

Er is alleen een Engelstalige versie.

Voor het kunnen draaien van MRDS moet het .Net framework 3.5 SP 1 geïnstalleerd worden wat vanzelf tijdens de installatie meegenomen wordt.

Om de Visual Simulation Environment te kunnen gebruiken moet er een DirectX 9.0 videokaart in de computer zitten die Pixel Shader 3.0 ondersteunt. De meeste moderne videokaarten ondersteunen die maar als je problemen ondervindt kun je onderstaande gebruiken om een en ander te controleren.

Door "dxdiag" te runnen vanuit het startmenu kan je de DirectX versie vinden die geïnstalleerd is. Op de display tabs vind je welke adapter je gebruikt. Voor bepalen van support voor Intel OnBoard videokaarten kun je hier terecht: <http://www.intel.com/support/graphics/intel915g/sb/cs-011807.htm>

Je kunt de Pixel Shader (PS) versie vinden op deze website: <http://www.techpowerup.com/gpubd/> en die voor OnBoard Intels hier: <http://www.intel.com/support/graphics/sb/cs-014257.htm>

Installeren

Druk op de link [Get Microsoft Robotics Developer Studio 2008 R3](#) en druk vervolgens op "Get It Now". Stel even je download snelheid in zodat je weet hoe lang het downloaden gaat duren. Bij mij overigens zo'n 10 minuten inclusief lokaal copieren en verifiëren. Sla het bestand op als het lang lijkt te gaan duren of (en als de download compleet is) druk op "Run". Vervolgens wordt gevraagd of je "Microsoft Robotics Group" toestemming wilt geven om de installer te runnen. Druk "Continue". Wanneer het ".NET framework 3.5 SP 1" niet aanwezig is op je systeem wordt nu gevraagd of je dat wil installeren. Dit is al mee gedownload dus druk op "Install". Bij mij vraagt de firewall nog of deze installer mag starten, omdat de Windows system directory wordt gebruikt, wat uiteraard mag. Vervolgens gaat de InstallShieldWizard verder en verschijnt een kleine dialog met de titel "Installer". Daarna verschijnt de dialog "Microsoft Robotics Developer Studio 2008 R3". Druk na "calculating..." op "Next".

Accepteer de license agreement en druk weer op "Next".

Kies of je Microsoft op de hoogte wilt houden en druk "Next".

Kies "Automatically check for updates" en druk "Next".

Kies "Complete" en druk "Next".

Je kunt beter niet "Custom" kiezen zodat je een ander pad kan kiezen dan die welke door de installer wordt opgegeven. Dat is bij mij "C:\User\Iwan\ Microsoft Robotics Dev Studio 2008 R3\". Zou je het pad veranderen dan moet je voor diverse onderdelen van MRDS als Administrator runnen wat niet wenselijk is. Het betekent wel dat je voor iedere gebruiker op de computer MRDS opnieuw moet installeren.

Als je het zeker weet druk je op "Install".

De InstallShieldWizard gaat weer verder. DirectX 9.0 wordt nu geïnstalleerd ook al heb je reeds een hogere versie geïnstalleerd.

Hierna verschijnt de boodschap "Install Shield Wizard completed" en let op de kleine lettertjes: 'Before using samples, please run "Build All Samples" located in the Start Menu'. Druk eerst op "Finish".

Voor "Build All Samples" moet wel eerst het ".NET framework 3.5 SP 1" geïnstalleerd zijn. Heb je dat nog niet sluit dan de computer af terwijl gelijktijdig updates worden geïnstalleerd (Zie het uitroeptekentje bij de rode power knop in het Start menu) of installeer ze via "Windows Updates" te bereiken via het configuratie menu. Dit duurt ongeveer 5 minuten.

Kijk in het Startmenu in de folder met snelkoppelingen "Microsoft Robotics Developer Studio 2008 R3". Klik de snelkoppeling van "Build All Samples". Het bouwen duurt ongeveer 10 minuten. Mogelijk verschijnen er 1 of 2 error of warning meldingen maar die hebben te maken met bijvoorbeeld SQL server of een oude versie van Visual Studio, die niet op je systeem geïnstalleerd zijn. Dit is voor het verder kunnen volgen van dit document geen probleem.

Rondleiding door MRDS

Nu wordt het tijd een en ander uit te proberen.

Het Start Menu

Als je in het start menu naar "Microsoft Robotics Developer Studio 2008 R3" gaat zie je onderstaande items voorzien van een korte uitleg (Laat je firewall netwerk toestemming geven wanneer er om wordt gevraagd)

- Build All Samples
Dit werd al besproken in de paragraaf "[Installatie van MRDS](#)".
- CCR and DSS Runtime 2008 R3 Class Reference
Dit is een helpfile met uitleg over de Concurrency and Coordination Runtime en de Decentralized Software Services. Zij vormen de motor van MRDS maar begrip van hen is niet nodig om met MRDS te werken. Wanneer zelf services gebouwd gaan worden is enige kennis wel vereist.
- Documentation
Dit is de centrale helpfile voor heel MRDS welke ook online te vinden is maar de onderlinge verschillen zijn miniem. Hierin staan release notes, tutorials voor de ontwikkelomgeving, uitleg van de voorbeelden, programmeer voorbeelden in C# en uitleg hoe te ontwikkelomgeving en de simulatie omgeving te gebruiken. Programmeer voorbeelden in Visual Basic kun je downloaden van <http://mrdssamples.codeplex.com/> onder "download" waarover later meer.
- DSS Command Prompt
Hiermee start je een Command Prompt welke een aantal omgevingssettings effectueert waaronder het PATH. Zo kan je de DSS server met een gekozen service vanaf de prompt opstarten.
- DSS Log Analyzer
Dit gebruik je om door de Log files van de DSS te stappen. Dit vergt een aanpassing aan de dss config files.
- DSS Manifest Editor 2008 R3
Dit gebruik je om services te bewerken wanneer je deze zelf geschreven hebt. Daarna kunnen ze in de Visual Programming Language gebruikt worden.
- DSS Service Descriptions
Om services op je computer vanaf een andere computer te laten vinden is deze directory bedoeld, waarop die andere computer via RSS feeds geabonneerd kan worden.
- Packages
In deze folder sla je complete applicaties op die als packages verstuurd kunnen worden naar andere computers.
- Robotics Developer Studio
Dit opent verkenner in de directory waar Robotics Developer Studio is geïnstalleerd.
- Run DSS Node
Dit start DSS op met de bijbehorende webinterface
- Update and Feedback Options
Hiermee kan je wijzigen welke informatie Microsoft je naar Microsoft stuurt en hoe je updates wilt ontvangen.
- Visual Programming Language 2008 R3
Dit is de ontwikkelomgeving waar je verschillende services aan elkaar knoopt tot een programma.

- Visual Simulation Environment 2008 R3
Hierin staan een aantal voorbeelden die draaien in de simulatie omgeving. Je kunt deze samples ook runnen door de "DSS Command Prompt" te starten en dan het doel van snelkoppeling te kopiëren en te plakken in "DSS Command Prompt" gevolgd door Enter.

De onderdelen van MRDS

Visual Simulation Environment



De VSE vormt een van de belangrijkste onderdelen waarom je MRDS zou gaan gebruiken. Het is namelijk een 3 dimensionele virtuele omgeving waar je robots kunt testen zonder dat je een echte robot nodig hebt. Net zoals piloten in een flightsimulator leren hoe ze een vliegtuig van een paar miljoen moeten besturen zonder dat er gewonden vallen, kunnen wij onze robots testen zonder dat er mensen met blauwe schenen lopen of ons prototype verpletteren.

Het gebruiken van de VSE kan op 3 manieren.

Snelkoppelingen

De snelle manier is klikken op een snelkoppeling in de folder "Visual Simulation Environment 2008 R3" zoals beschreven in het hoofdstuk "**Fout! Verwijzingsbron niet gevonden.**". Klik bijvoorbeeld op "Apartment Environment". Verderop in paragraaf "[\(User\) Apartment Scene](#)" wordt beschreven hoe de VSE gebruikt kan worden. Je kunt ook de andere snelkoppelingen proberen.

Run DSS Node

De langzame manier is via de webservices. Klik op de snelkoppeling "Run DSS Node" in de "Microsoft Robotics Developer Studio 2008 R3" Start menu folder
Dit start DSShost32 op en opent de website <http://localhost:50000/>. Geef je gebruikersnaam en password van de gebruiker waaronder MRDS is geïnstalleerd en zet het vinkje bij wachtwoord

onthouden.

Klik links op "controlpanel" waarin een lijst van "contracten" is te zien. Een contract is de beschrijving van een samenstel van services die afhankelijk van de gemaakte toepassing met elkaar samenwerken. Een service kan bestaan uit meerdere andere services maar kan ook deel uitmaken van meerdere andere services.

Een voorbeeld zal een en ander verduidelijken:

Zorg dat het vinkje van "running services" weg is en type "apartment" in het "search" veld. Er verschijnen nu 2 "Apartment Scenes" en 1 "Apartment Simulation".

Start "(User) Apartment Scene" op door het manifest

"samples\Config\ApartmentScene.user.manifest.xml" te selecteren en op "Create" te drukken.

Verderop in paragraaf "[\(User\) Apartment Scene](#)" wordt beschreven hoe de VSE gebruikt kan worden.

Visual Programming Language

Deze manier van starten van de VSE zal in een later document besproken worden.

(User) Apartment Scene

Als je via de DSS Node gestart bent verschijnen er 3 windows, via de snelkoppeling maar 1.

"Microsoft Visual Simulation Environment 2008" laat het appartement zien.

Met de pijltjes toetsen beweeg je de "Main Camera" in de richting waarin je kijkt en die richting bepaal je door je muisknop in te drukken en de muis te bewegen. Je robot is een witte doos op de grond. Ter illustratie van de mogelijkheden van VSE kun je even het "camera" menu klikken en "Simulated Webcam" kiezen. Klik nu in de browser links op "Service Directory", vervolgens op 1 van de 2 "Simulated Webcam" onder "Service Instance" dan zie je hetzelfde beeld.

Zet in de VSE de camera weer terug naar "Main Camera".

Het tweede opgestarte window is het "Dashboard" waarmee de robot op afstand bediend kan worden. In de browser op de "Service Instance Directory" pagina en op de "home" pagina is dat de "simplifiedashboard/...UID..." service. Wanneer er een joystick aangesloten is op je computer zie je die ook staan als device op je "Dashboard". Om je robot te kunnen bedienen moet er eerst een verbinding tot stand gebracht worden met de robot door te drukken op "Connect" waardoor alle te verbinden services in een lijst verschijnen waaruit je kiest door te dubbelklikken op "(MotorBase With Drive) simulateddifferentialdrive". Druk op "Drive" en je kunt nu door met je muis in het rondje te bewegen terwijl je de muisknop ingedrukt houdt je robot bewegen. Doe dit wel voorzichtig want je kunt de robot door te hard rijden op zijn kant krijgen. Gebeurt dat ga dan in VSE naar menu "Mode" en klik "Edit". Selecteer links "MotorBaseWithDrive" en zet alle "Direction" waarden op 0. Zet dan de "Mode" weer in "Run".

Terwijl je door het appartement rijdt kun je op het derde window het beeld van de webcam van de robot zelf zien. Dit kan ook via de "SimulatedWebcamService" webpagina waar je wel even op "Start" moet drukken. Zo kan je bijvoorbeeld proberen een stoel omver te rijden.

Om de "Appartment Scene" af te sluiten moet je de VSE afsluiten maar dan wordt ook de DSS node gestopt en moet je deze de volgende keer helaas opnieuw opstarten. De website wordt ook iedere keer in een nieuwe tabpage van browser getoond. Je kunt nog meer Services opstarten via het "Control Panel" maar ze zijn niet allemaal even illustratief of werken niet standalone.

Later zal ik nader ingaan op de meer geavanceerde mogelijkheden van VSE.

Codeplex samples

Je kunt op <http://mrdssamples.codeplex.com/> nog andere samples downloaden waaronder Sumo, Soccer, Teleporter, Fischer Technik Robot, Kondo, NXT ShooterBot en VB samples.

Klik op de "Download" button aan de rechter kant en save de zip op je harddisk. Als de download klaar is pak dan de CodePlexSamples folder uit in de "Microsoft Robotics Dev Studio 2008 R3" folder waarin ook de "samples" folder te vinden is.

Start de "DSS Command Prompt" en type "cd CodePlexSamples". Type "Setup.cmd". Type vervolgens "BuildAll.cmd". Welke Visual Studio Express Edition talen je geïnstalleerd hebt bepalen hoeveel builds zullen slagen. Sommige C++ samples werken alleen onder Visual Studio 2008.

Wanneer je nu weer "Run DSS Node" start zal je in het Command venster de boodschap "Rebuilding contract directory cache" zien wat betekent dat de samples toegevoegd worden aan de eerdere services directory.

Kies weer "Control Panel" in het linker menu en type "soccer" in het "Search" veld.

Kies voor de "Simulated Soccer Referee" als manifest

"CodePlexSamples\Config\simulatedsoccer.legonxt.fourplayers.manifest" en druk start.

In het "Microsoft Visual Simulation Environment 2008" window is nu het voetbalveld zichtbaar. Met het "Sim Soccer Referee" window kan je de wedstrijd starten en omgevallen robots rechtzetten door hun letter te verslepen.

En om voetbal was het uiteindelijk allemaal te doen.

Volgende keer

In het volgende deel wordt beschreven hoe je met de Visual Programming Language omgeving een programma uit componenten opbouwt waarmee je je robot in het echt en in de VSE met een joystick bestuurt.

Colofon

Redacteur	Ed Buzzi
Redacteur	Zeno Otten
Technisch adviseur	Bert Berrevoets
Auteur	Iwan Tolboom

Copyright 2011, www.twintellect.com

Email: iwan.tolboom@chello.nl

Skype: iwantolboom

Microsoft Robotics Development Studio: JoyStick gestuurde NXT

Inhoud

Inleiding	2
Verbinden van de NXT Brick met de computer	2
Verbinden van NXT Brick via USB kabel	2
De firmware van de NXT upgraden.....	2
Bestanden op de NXT Brick zetten	3
Verbinden van de NXT Brick via BlueTooth met de PC.....	3
Configureren van BlueTooth in MRDS.....	3
Gebruiken van een Game Controller in MRDS.....	4
Configureren van de Game Controller	4
VPL JoyStick emulator	5
Besturen van de NXT met de Game Controller.....	5
Uitlezen van de Game Controller	5
Besturen van de NXT met een 2 joysticks gamepad	5
Configureren van Motoren in VPL	5
Aansturen van Motoren	5
Testen van het programma	6
De browser	6
Service Directory.....	6
Debuggen	7
Geavanceerde aansturing van de motoren.....	7
De formules	7
Voorbeeld	8
Implementatie in VPL	8
Gebruiken van de Differential Drive	10
Het hoofddiagram.....	10
Maken van een Activity	10
De Draaien Activity	11
De Rijden Activity.....	12
De Geluid Activity	12
Simulator	13
Volgende keer.....	13
Colofon	13

Inleiding

Om binnen MRDS een NXT te kunnen besturen met een joystick is een speciale aanpak nodig. Naast de keuze van de juiste componenten (services) moet ook het coördinaten stelsel van de joystick omgerekend worden naar aandrijving van een linker en een rechter wiel. Hier wordt uitgegaan van een Logitech Rumblepad maar iedere andere joystick is bruikbaar. Ook zonder JoyStick is dit verhaal te volgen omdat MRDS een Joystick emulator heeft die met de muis te besturen is.

Dit document is zowel van toepassing op Mindstorms NXT 1.0 als 2.0. Het verschil tussen beide versies zit hem in de sensoren en bouwstenen die Lego in de doos heeft gestopt.

Na de voorbereiding worden een aantal voorbeelden gegeven die langzaam moeilijker worden.



Verbinden van de NXT Brick met de computer

Om te beginnen moeten de Lego Mindstorms NXT robot en de Joystick bekend zijn in de Visual Programming Language (VPL). Die configuratie beschrijven we hier.

Verbinden van NXT Brick via USB kabel

In tegenstelling tot de software die door Lego bij de Mindstorms geleverd wordt (Lego Mindstorms NXT) kan MRDS geen contact leggen met de Brick via USB. Toch willen we USB gebruiken om onder andere de firmware van de NXT te upgraden. Wanneer je de USB kabel verbindt tussen PC en de NXT kunnen de drivers van Internet of van de door Lego geleverde CD worden geïnstalleerd.

Installeer ook meteen de "Lego Mindstorms NXT" software van de Lego CD.

Start het programma en verbind je NXT via de USB kabel en zet hem aan. Druk op het button panel, rechtsonder in het window, de button met de popup "NXT Window". Vul een naam in voor je robot, bijvoorbeeld "speler 1" en druk de button ernaast.

De firmware van de NXT upgraden

Vanuit "Lego Mindstorms NXT" kan ook de firmware van de NXT een upgrade krijgen.

Controleer de "Firmware version". Wanneer deze lager dan "1.29" is moet je upgraden. Ga naar het menu "Tools\Update NXT Firmware..." en druk op "Check" waarmee de meest recente versie van het Internet gehaald wordt. Selecteer deze versie en druk vervolgens op de "download" button. Het schermje van je NXT wordt leeg en de NXT begint te "klikken" totdat deze opstart op de normale manier.

Beginnt hij niet te klikken of denk je dat de NXT is vastgelopen, dan kun je hem ook resetten door met een schroevendraaier het zwarte reset buttontje in te drukken. Dit zit in het verbindingsgat aan de onderkant van je Brick, ter hoogte van de USB ingang. Kort drukken is gewoon resetten, lang drukken is resetten waarna de Brick gaat wachten op upgrade software. Druk als de NXT begint te klikken weer op bovengenoemde download button. Werkt genoemde versie niet met je NXT kies dan firmware met een lagere versie.

Bestanden op de NXT Brick zetten

We willen het "Lego Mindstorms NXT" programma ook nog gebruiken om bestanden (geluiden, plaatjes en programmatjes) vanuit het "Lego Mindstorms NXT" programma te downloaden naar de NXT. De plaatjes, geluidjes en programma's op de NXT kunnen we namelijk oproepen vanuit MRDS. Maak daartoe een nieuw programma aan met 4 "sound" componenten. Ken de voorgeïnstalleerde geluidsbestanden toe genaamd: "Left", "Right", "Forward" en "Backwards". Run vervolgens het programma zodat je de bestanden niet alleen naar je NXT upload maar ook kan controleren of alles gelukt is.

Verbinden van de NXT Brick via BlueTooth met de PC

Zet de bluetooth van de laptop of computer aan. De speciale BlueTooth Dongle van Lego is niet nodig wanneer BlueTooth reeds op een laptop aanwezig is. Laat je overigens niet afschrikken door de BlueTooth problemen die vanuit de "Lego Mindstorms NXT" ontwikkelomgeving ontstaan want die heb je vanuit MRDS niet.

Plaats een eventuele fysieke knop op je computer om BlueTooth (en vaak de WiFi) aan en uit te zetten in de goede stand.

Ga in Vista naar "Configuratie Scherm", "Hardware en Geluiden", "BlueTooth apparaten" en klik de "Opties" button. Zet het vinkje naast "Bluetooth apparaten mogen verbinding met deze computer maken".

Zet de NXT aan, ga naar het BlueTooth menu en zet de BlueTooth aan. Zet ook de "Visibility" van je NXT op "Visible". Verder hoeft je niets te doen in het "BlueTooth" menu, het is aan te raden om de andere opties met rust te laten om problemen te voorkomen.

Klik op de computer in "BlueTooth apparaten" op de "Draadloos apparaat toevoegen" button en vervolgens verschijnt een lijst met beschikbare BlueTooth apparaten waaronder de NXT.

Klik op de NXT en druk de "Volgende" button. Houd tegelijkertijd het scherm van de NXT goed in de gaten. Onthoud de code die op de NXT verschijnt (standaard '1234') en druk op OK. Op je computer verschijnt nu een dialoog waarin je deze code in moet vullen waarna je op "Volgende" drukt.

Windows zal nu de device drivers installeren en een melding geven "Het draadloze apparaat is gepaard met deze computer". Druk op "sluiten".

Onder "Bluetooth apparaten" is nu de NXT als apparaat toegevoegd.

Configureren van BlueTooth in MRDS

Maak in de folder waar je "Microsoft Robotics Development Studio" hebt geïnstalleerd, zoals beschreven in [Microsoft Robotics Development Studio Introductie](#), de folder "projects" aan. Om je eigen gemaakte services te delen met andere mensen op andere computers, is er binnen MRDS een programma (DssProjectMigration.exe) die geheel automatisch de projecten aanpast aan de huidige gebruiker. Maar dat werkt alleen goed als de map binnen de installatie map van MRDS ligt.

Start de Visual Programming Language (VPL) en maak een nieuw project aan dat je met de naam JoyStickNXT1 opslaat in bovengenoemde folder. Automatisch wordt er een nieuwe folder JoyStickNXT1 gemaakt waarin alle bestanden komen.

Voer links in "Find Service..." het woord "NXT" in en druk Enter. Druk vervolgens op het "+"-je naast "All Found". Op deze manier heb je een snelkoppeling gemaakt om sneller Mindstorms services te kunnen vinden. Zoek in de services "Lego NXT Brick (v2)"

Let op: Mogelijk ondervind je het volgende probleem: Er zijn 2 "Lego NXT Brick (v2)" services namelijk: "Lego NXT Brick (v2)" en "(User) Lego NXT Brick (v2)". Soms kun je "Lego NXT Brick (v2)" niet editen hoewel dit wel zo lijkt te kunnen. Ook na een herinstallatie hield ik dit probleem waardoor al mijn voorbeelden "(User)" versies van alle services gebruiken. Wees in ieder geval consequent in het gebruik van "(User)" of de normale versie.

Klik op "Lego NXT Brick (v2)" en sleep hem op het Diagram.

Klik op de service op het diagram en open de "Configuration" combobox op het "Properties" tabblad rechts. Kies "Set Initial configuration" en klik op het "+"-je van "Configuration".

Hier moet de "SerialPort" ingevuld worden voor de Bluetooth communicatie. Deze poort vind je bij de properties van het NXT apparaat, zoals dat in paragraaf "[Verbinden van NXT Brick via Bluetooth met de PC](#)" toegevoegd is. Dit staat op het tabblad "Services" van de NXT welke even tijd nodig heeft om te laden. Er staat dan bijvoorbeeld: "Serial port (SPP) 'Dev B' " " COM xx" Die xx moet ingevuld worden onder de "UserLegoNXTBrickv2" properties.

Controleer of de BaudRate is ingesteld op 115200.

Druk op het groene play buttontje waarna de applicatie gaat runnen. Er verschijnt nu een venster met allerlei logging informatie waarin je kunt zien welke services allemaal gestart zijn. Ook wordt een webpagina geopend waarin al deze services getoond worden. Dit kun je uitzetten door onderaan de "LegoNXTBrickv2" properties het vinkje naast "ShowInBrowser" te verwijderen. Je kunt namelijk altijd toegang tot de services krijgen door <http://localhost:50000/> in je browser te openen. Je moet wel de eerste keer je username en password invoeren en "onthoud mij" aanvinken. Als de NXT een korte piep laat horen dan is de BlueTooth verbinding gelukt. Je stopt het runnen door de "Stop" button te drukken op het logging venster.

Wanneer er tijdens het laden van de services een fout is opgetreden worden de logging regels rood. Zet de NXT dus even uit en start het programma opnieuw. Wacht ongeveer een minuutje en vervolgens verschijnen er een aantal meldingen waaronder: "Invalid Serial Port", "The semaphore timeout period has expired" en "Failed to connect to the Lego NXT hardware". Dit soort meldingen krijg je ook wanneer de Mindstorms buiten het bereik van de BlueTooth komt. Dus houdt daar rekening mee.

Soms blijft de NXT ongecontroleerd draaien omdat de verbinding weggevallen is. Er rest dan niets anders dan de NXT uit en weer aan te zetten.

Gebruiken van een Game Controller in MRDS

Configureren van de Game Controller

Sluit de GameController aan op de computer. De computer installeert eerst de software.

Zoek in VPL de "Game Controller" service en sleep hem op het diagram. Wanneer je één joystick hebt aangesloten wordt deze automatisch herkend nadat je het programma hebt opgestart.

Wanneer je meerdere Joysticks hebt aangesloten of als de Joystick niet werkt volg dan de nu volgende aanwijzingen:

Klik op de service op het diagram en open de "Configuration" combobox op het "Properties" tabblad.

Kies "Set Initial configuration" en klik op het "+"-je van "GameControllerState".

De velden "InstanceName" en "ProductName" kunnen gevonden worden via "Apparaten beheer". Open het "Configuratie Scherm", "Systeem Onderhoud", "Apparaten beheer".

Open "Human Interface Devices" en kies de het USB item van de Game Controller, bij mij "Logitech Rumblepad 2 USB".

Open de properties en copieer de "Location" waarde (bv Port_#0001.Hub_#0005) en plak deze in het "InstanceName" veld in VPL. Geef zelf een goede beschrijving (bv "Logitech RumblePad 2 USB") in het "ProductName" veld.

VPL JoyStick emulator

Onder de services is ook een "Desktop Joystick" te vinden die dezelfde messages verstuurt als een echte joystick en met de muis bedient wordt. Deze kan dus in plaats van "Game Controller" gebruikt worden.

Besturen van de NXT met de Game Controller

Uitlezen van de Game Controller

Omdat de waarden van de GameController Integer waarden tussen -1000 en 1000 zijn en de NXT Brick werkt met Double waarden tussen -1.0 en 1.0 moeten we deze omrekenen.

Daartoe worden 2 "Basic Activities" gebruikt. Deze staan in een apart vak onder de services.

Maak 2 "Calculate" componenten aan. Sleep vervolgens het cirkeltje van de "GameController" naar het ingaande pijltje van ieder van de "Calculates" en kies "UpdateAxes" in de "Connections" dialog.

Volg dezelfde stappen indien je de DesktopJoystick gebruikt. Je ziet dat beide joysticks dezelfde "Connections" mogelijkheden hebben.

Besturen van de NXT met een 2 joysticks gamepad

Omdat de RumblePad 2 joysticks heeft kan ieder aan een wiel of rupsband toegekend worden. Op die manier kun je de besturing van een "oude" tank nabootsen. "Moderne" tanks hebben tegenwoordig namelijk ook joysticks.

Voer "(double)Y / 1000" in bij de ene Calculate en "(double)Rz / 1000" bij de andere. *De Y staat voor verticale bewegingen van de linker en de Rz voor die van de rechter. Echter dit kan per joystick/gamepad verschillen. Voor Xbox gebruik je hier Y en Ry.* De casting naar "double" moet eerst gebeuren omdat er anders met Integers gewerkt wordt en dan wordt er na deling afgerond op gehele getallen. "double" moet met een kleine letter geschreven worden anders krijg je een rood uitroeptekentje bij de Calculate met de melding "No valid expression has been entered".

Bij een GameController met maar 1 joystick of the DeskTopJoystick voer je "(double)Y / 1000" in bij de ene en "(double)X / 1000" bij de andere Calculate. Daardoor wordt het linker wiel door de verticale beweging van de joystick bepaald en het rechter wiel door de horizontale beweging. Om beide wielen op deze manier vooruit te laten draaien moet de joystick in de rechterbenedenhoek gehouden worden en voor achteruit in de linkerbovenhoek. Dit is tegennatuurlijk vandaar dat in de paragraaf ["Geavanceerde aansturing van de motoren"](#) een betere besturing wordt uitgewerkt.

Configureren van Motoren in VPL

Zoek in VPL de "Lego NXT Drive (v2)" service en sleep hem op het diagram. Klik op de service op het diagram en open de "Configuration" combobox op het "Properties" tabblad. Kies "Set Initial configuration". Kies onder "Partners" voor "Brick" de "LegoNXTBrickv2". Onder "DriveState" kun je de "DistanceBetweenWheels" invullen in meters bijvoorbeeld "0.13". Meet dit zelf even op.

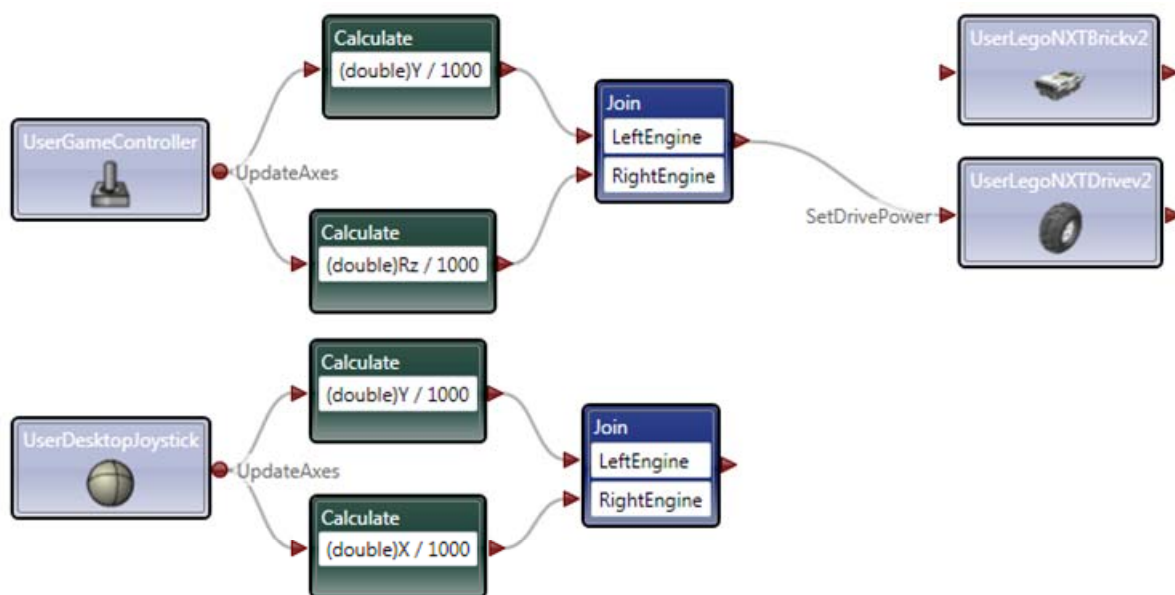
Kies voor "LeftWheel" en "Rightwheel" de poort die je gebruikt voor het linker en het rechter wiel.

Lego gebruikt meestal "Motor C" voor links en "Motor B" voor rechts. "Motor A" is dan de Action motor. Als de motoren de verkeerde kant opdraaien kan je "ReversePolarity" aanvinken maar je kan natuurlijk ook een "-" teken aan het begin van de berekening in de "Calculate"'s plaatsen. Vul ook de diameter van het rubber van de wielen in, bijvoorbeeld "0.035"

Aansturen van Motoren

Omdat er maar een enkel bericht tegelijkertijd naar de Mindstorm Drive gestuurd mag worden moeten de 2 berekende waarden samengevoegd worden door middel van nog een "Basic Activity" namelijk de "Join".

te klikken, zodat de nieuwe namen doorgevoerd zijn. Sleep vervolgens het uitgaande pijltje van de "Join" op de "LegoNXTRDrivev2" en kies in de "Connections" dialoog als "To" waarde "SetDrivePower" en kies in de daarop volgende "Data Connections" dialoog de juiste "Value"'s. Dit kan je ook aanpassen door op de gebogen lijn te klikken en de "Properties" aan de rechter kant te bewerken. Als alles gelukt is ziet het diagram er als onderstaande figuur uit:



Testen van het programma

Save het programma. Wanneer de NXT uit gegaan is zet je hem weer aan. Run nu het programma. Er verschijnt nu onder andere een venstertje met de "Desktop Joystick" welke je nu zou kunnen gebruiken wanneer je de onderste "Join" met de "LegoNXTRDrivev2" had verbonden.

De browser

Wanneer je het vinkje naast "ShowInBrowser" hebt gezet of hebt laten staan opent er een webpagina "NXT Brick" waarop de status van de NXT te zien is. Iedere keer wanneer het programma opgestart wordt, wordt er een nieuwe browser opgestart omdat het unieke ID (zie de URL) van de NXT service opnieuw door de server, waarop de services draaien, wordt toegekend. Ga je standaard naar <http://localhost:50000/> dan zie je op de pagina "Microsoft Robotics Studio Home" ook de "Lego NXT Brick (v2)" staan. Daar 2 maal achter elkaar op klikken brengt je naar dezelfde pagina als welke automatisch met de Brick werd opgestart en je hoeft dan niet na iedere run de tabpagina te sluiten. Als je programma gestopt is kom je daar altijd door op de home button te klikken. Verversen helpt niet door dat unieke ID.

Door op deze pagina op MotorB of MotorC te klikken en vervolgens op de url, kun je zien hoeveel power naar de motoren wordt gestuurd.

Service Directory

Door links op "Service Directory" te klikken worden alle gestarte services getoond en door op de links te klikken krijg je er nadere informatie over. Zo laat de gamecontroller link bijvoorbeeld alle waarden zien van de huidige stand van de controller. Als je de joystick in een stand houdt en de pagina met F5 verversst zie je onder <Axes> een waarde tussen -1000 en 1000 verschijnen. Onder <Buttons> zie je true of false afhankelijk of de button is ingedrukt of niet.

Bij de "lego/nxt/drive" service zijn de waarden die naar de motoren gestuurd worden te zien.

Debuggen

Wanneer het geheel nog steeds niet werkt kan het programma ook in Debug mode gestart worden. Stop eerst het lopende programma en herstart het met het groene driehoekje met de 2 blauwe voetstappen.

Er wordt nu automatisch een nieuwe pagina geopend voor de debugger.

Hier zijn naast reeds genoemde waarden ook reken resultaten te zien. Vergeet niet de "Run" button te klikken op de webpagina waarmee het debuggen gecontroleerd wordt. De debugger vergt wel wat geduld en reageert wat traag. Houd daarom altijd de logging in de "Run" dialoog in de gaten.

Geavanceerde aansturing van de motoren

Tot nu toe werd gebruik gemaakt van 2 joysticks zodat een besturing als van een tank mogelijk was of van 1 joystick waarbij de horizontale beweging het ene en de verticale beweging het andere wiel aandreef wat resulteerde in een diagonale besturing.

Hier wordt een oplossing beschreven die meer intuïtief werkt.

In onderstaande tabel zijn 25 standen beschreven waarin de joystick zich kan bevinden en het vermogen dat naar beide wielen gestuurd moet worden.

Een "+" geeft aan dat de motor vooruit draait, een "-" achteruit en een "=" dat deze stilstaat.

Het aantal "+" en of "-" en geeft het vermogen aan. Per omrand vakje staan de linker en de rechter motor aangegeven.

		Vooruit											
Links	--	++	=	++	++	++	++	=	++	--	Rechts		
	-	+	=	+	+	+	=	+	-				
	=	=	=	=	=	=	=	=	=	=			
	+	-	=	-	-	-	=	-	+				
	++	--	=	--	--	--	--	=	--	++			
		Achteruit											

Wanneer de joystick in verticale richting bewogen wordt zonder horizontale component dan neemt het vermogen van de motoren toe in de ene en af in de andere richting.

Wanneer vervolgens de joystick horizontaal wordt bewogen verandert de verhouding van vermogen dat naar de wielen gestuurd wordt en daarmee de draairichting. Wanneer de linker motor stil staat zorgt alleen de rechter motor voor de draaiing. Door de joystick nog verder naar links te bewegen gaat de linker motor achteruit draaien waardoor de draai naar links versnelt.

Andere oplossingen waarbij de joystick naar links bewegen betekent dat er snel linksom gedraaid wordt en naar rechts bewegen snel naar rechts, vergelijkbaar met een vliegtuig, zijn te complex om hier te bespreken. Wie een eenvoudige oplossing hiervoor weet wordt vriendelijk verzocht deze aan het adres onder aan dit document te emailen.

De formules

Bij een enkele joystick wordt de verticale beweging door de "Y" waarde (-1000 / + 1000) gegeven en de horizontale door de "X" waarde (-1000 / +1000).

Uit bovenstaande tabel zijn de volgende regels te herleiden:

- 1) Omdat de "Y" waarde in alle situaties het toegevoerde vermogen representeert komt deze altijd voor in de formules.
- 2) Als de joystick in horizontale richting bewogen wordt krijgt altijd 1 van beide motoren het vermogen dat bepaald wordt door de "Y" waarde (naar rechts is dat de linker motor, naar links is dat de rechter motor). De andere is dan afhankelijk van de horizontale beweging en de "Y" waarde.

3) Voor zowel de beweging naar links als naar rechts van de joystick gelden de volgende vermenigvuldigings regels:

- a) In de neutrale stand wordt met 1 vermenigvuldigd. De motors krijgen het "Y" vermogen
- b) Halverwege 0 en 1 van de 2 uiterste standen (links of rechts) wordt met 0 vermenigvuldigd
- c) In de beide uiterste standen wordt met -1 vermenigvuldigd.

De waarden van "X" door horizontale bewegingen van de joystick lopen voor de linkerkant van -1000 tot 0 en voor de rechterkant van 0 tot 1000. Daarom wordt eerst de absolute waarde van "X" genomen. Daar wordt vervolgens 500 van afgetrokken, we willen namelijk van "-" via "=" naar "+" regelen aan de linkerkant en het omgekeerde aan de rechterkant. Het resultaat wordt gedeeld door 500, dus regelen we van -1.0 via 0 naar + 1.0. Vervolgens wordt dit vermenigvuldigd met vermogen "Y" en naar de motor gestuurd die afhankelijk is van de horizontale beweging. Zoals gezegd krijgt de andere motor het volle "Y" vermogen.

Voorbeeld

Zie onderstaande tabel voor enkele waarden. We stellen dat als "X" = 0 de rechter motor geregeld wordt. Dit stelt waarden voor van de geregelde motor in het kwadrant rechtsboven.

$$\text{Vermogen} = Y * (X - 500) / 500$$

	X = 0	X = 500	X = 1000
Y = 1000	$1000 * (0 - 500) / 500 =$ $1000 * -1 = -1000$	$1000 * (500 - 500) / 500 =$ $1000 * 0 = 0$	$1000 * (1000 - 500) / 500 =$ $1000 * 1 = 1000$
Y = 500	$500 * (0 - 500) / 500 =$ $500 * -1 = -500$	$500 * (500 - 500) / 500 =$ $500 * 0 = 0$	$500 * (1000 - 500) / 500 =$ $500 * 1 = 500$
Y = 0	$0 * (0 - 500) / 500 =$ $0 * -1 = 0$	$0 * (500 - 500) / 500 =$ $0 * 0 = 0$	$0 * (1000 - 500) / 500 =$ $0 * 1 = 0$

Uit de tabel blijkt dat in neutrale stand de resultante negatief is en in de uiterste stand positief. Dit is precies het tegenovergestelde van het gewenste en daarom wordt het resultaat nog met -1 vermenigvuldigd. Omdat de NXT motoren met waarden tussen -1 en +1 werken moet ook alles nog door 1000 gedeeld worden.

Implementatie in VPL

Sla het eerder gemaakte programma op als JoyStickNXT2.mvpl. Let op dat dit in de eerder genoemde parent folder "project" gebeurt zodat er een nieuwe folder "JoyStickNXT2" wordt aangemaakt.

Verbreek de verbinding tussen de "GameController" de twee "Calculate"s door met rechts op de verbinding te klikken en vervolgens op "Verwijder". Negeer voorlopig alle rode lampjes. Verwijder ook de verbinding van de "Join" naar de "LegoNXTDrivev2"

Maak even wat ruimte door de Shift-toets in te drukken en click vervolgens alle componenten van de "DesktopJoystick". Laat de Shift-toets los en click 1 van de componenten en sleep dat naar een lege ruimte waarmee tegelijkertijd de andere component meegaan.

Voeg uit de "Basic Activities" een "If" component toe. Verbindt het rondje van "GameController" met deze "If". Kies "UpdateAxes" in de "Connections" dialog. Voer "X>0" in bij het tekstvak van de "If".

Houdt nu de Shift-toets ingedrukt en klik vervolgens de twee "Calculate"s en de "Join". Houdt nu de Ctrl-toets ingedrukt, klik 1 van de 3 componenten en sleep zo een copy naar een lege ruimte op het diagram. Er zijn nu 2 groepen elk voor 1 van de paden van het "If" component. Sleep het bovenste pijltje van de "If" naar beide "Calculate"s van de ene Join. Sleep het "Else" pijltje naar de "Calculate"s van de andere join.

Beide "Calculate"s met "(double)Y / 1000" laten we onveranderd.

De andere "Calculate" van de "Then" tak wordt als volgt: $-\left(\frac{((\text{double})X - 500)}{500} * Y\right) / 1000$

De overeenkomstige "Calculate" in de "Else" tak wordt: $-\left(\frac{((\text{double})-X - 500)}{500} * Y\right) / 1000$

Het verschil zit hem in de "-" voor de "X" welke de range van -1000 tot 0 positief maakt.

De andere "-" is, zoals boven reeds vermeld, om de resultaat waarden die door de -500 negatief werden weer positief te krijgen.

Omdat als $X > 0$ de rechter motor de hele "Y" waarde krijgt en als $X \leq 0$ de linker motor moeten we in de "Join" behorende bij de "Else" de motor namen omkeren. Verwijder daartoe beide connections met de "Join" en verbind ze nu met het tegenovergesteld "Join" veld.

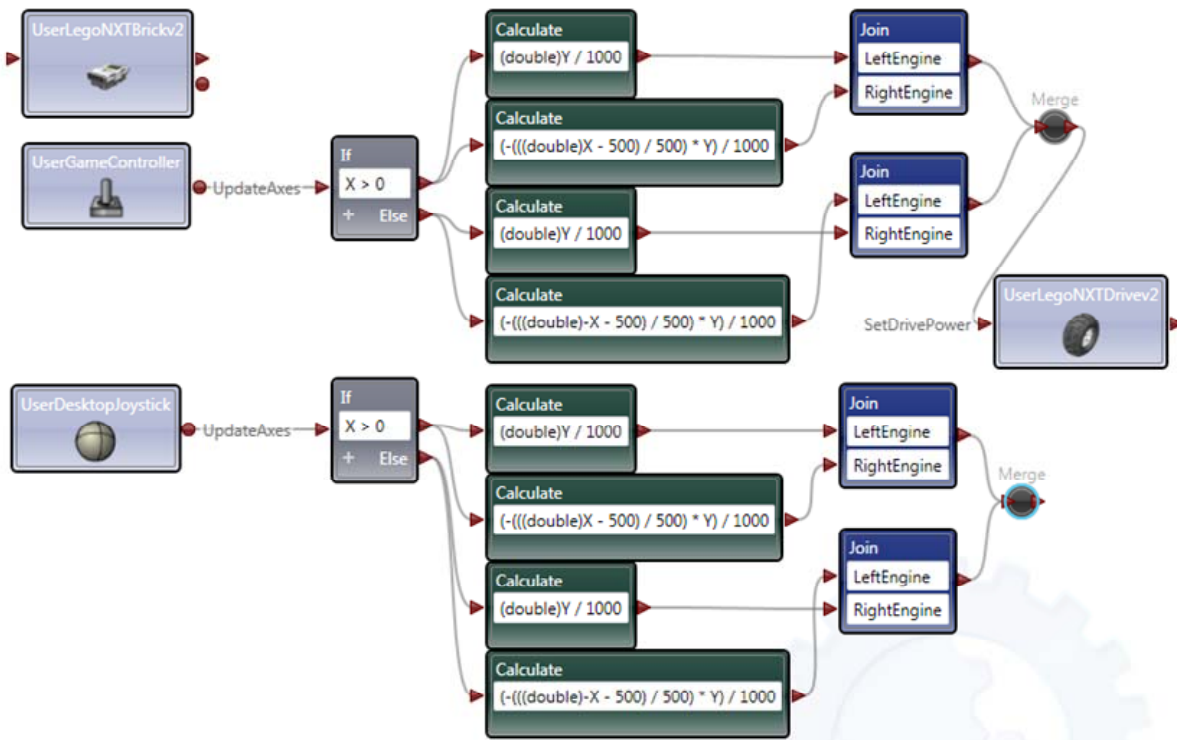
Van de beide "Join"s wordt er altijd maar 1 tegelijkertijd doorlopen. Deze twee "Join"s moeten nu verbonden worden met "LegoNXTDrivev2". Voeg hiertoe van de "Basic Activities" een "Merge" toe en verbind de uitgaande pijltjes van de "Join"s hiermee.

Verbind de uitgang van de "Merge" met de "LegoNXTDrivev2" waarbij in de "Connections" dialog "SetDrivePower" geselecteerd moet worden en in de "DataConnections" dialog "LeftEngine" aan "LeftWheelPower" en "RightEngine" aan "RightWheelPower" toegekend.

Let op: Het verschil tussen een "Join" en een "Merge" is dat hoewel beiden verbindingen samenvoegen de "Join" van 2 berichten 1 bericht maakt en de "Merge" alleen paden samenvoegd waarover berichten gaan. De "Join" is dus 2 mensen die 1 taxi delen en de "Merge" is 2 autowegen die samenkomen. Overigens zijn alleen waarden die in een Join zitten na de Merge toegankelijk in vervolg componenten.

Je kunt vervolgens alles copieren zoals boven beschreven en de "GameController" vervangen door de "DesktopJoystick".

Het eindresultaat is te zien in onderstaande figuur.



Gebruiken van de Differential Drive

Aangezien de "LegoNXTDrivev2" ook een differential drive is kunnen we de robot ook met alleen knoppen bedienen. Op de meeste joystick zitten wel 4 knoppen en deze kunnen gebruikt worden voor de volgende functies: 1) Draai 45 graden linksom, 2) Rijdt 10 centimeter vooruit, 3) Draai 45 graden rechtsom, 4) Rijdt 10 centimeter vooruit

Dit is een wat moeilijker uitwerking die uit 4 stukken bestaat namelijk het hoofddiagram en 3 Activity diagrammen.

Het hoofddiagram

Maak een nieuw programma JoyStickNXT3.mvpl wat je weer onder de project folder opslaat.

Plaats een nieuwe "Lego NXT Brick (v2)" aan volgens de instructies in de paragraaf [Configureren van BlueTooth in MRDS](#).

Plaats ook weer een "Game Controller", een "DesktopJoystick" en de Basic Activity "If" op het diagram en verbind het rondje van één van beide controllers met de "if".

Kies waar in het vervolg van dit artikel gesproken wordt over verbinden vanuit de Joystick, in de "Connections" dialoog in het "From" veld altijd "UpdateButtons"

Druk 3 maal op het plusje van de "if" en vul achtereenvolgens "Pressed[0]", "Pressed[1]", "Pressed[2]" en "Pressed[3]" in welke staan voor de buttons 0 tot en met 4 van de Rumblepad of buttons A, B, X en Y van de Xbox gamecontroller. Heb je een andere Joystick dan kan je zelf bepalen welke knop overeenkomt met welke boolean van Pressed zoals beschreven in de paragraaf [Service Directory](#).

Plaats vervolgens 6 "Data" Basic Activities onder elkaar rechts van de if. Deze zijn allemaal type double en hebben achtereenvolgens de waarden: 45, 0.5, -0.5, 0.5, -0.5, 0.1. Verbind het ronde balletje van de Joystick met de bovenste "Data"(de hoek bij draaien) en de onderste "Data"(de afstand bij rijden). Verbind de 4 "If" uitgangen op volgorde met de andere "Data"'s. Verbind ieder van de twee groepen 0.5 en -0.5 met een "Merge". Dit zijn de vermogens voor draaien en rijden. De hoek en het bijbehorende vermogen van de "Merge" verbinden we met een "Join" en geven de twee messages de namen "Hoek" en "Vermogen".

Doe dit ook voor het rijden maar noem de twee messages "Vermogen" en "Hoek".

Maken van een Activity

Voeg uit de Basic Activities 2 "Activity"'s en verbind de uitgangen van de 2 "Join"'s elk met één.

Klik op de "Activity" om te draaien en vul in de Properties de volgende zaken in:

- Comment: ""
- Name: "Draaien"
- Friendly Name: "Draaien"
- Description: "Draait de robot het aantal graden met het opgegeven vermogen"

Klik op de "Activity" om te rijden en vul in de Properties de volgende zaken in:

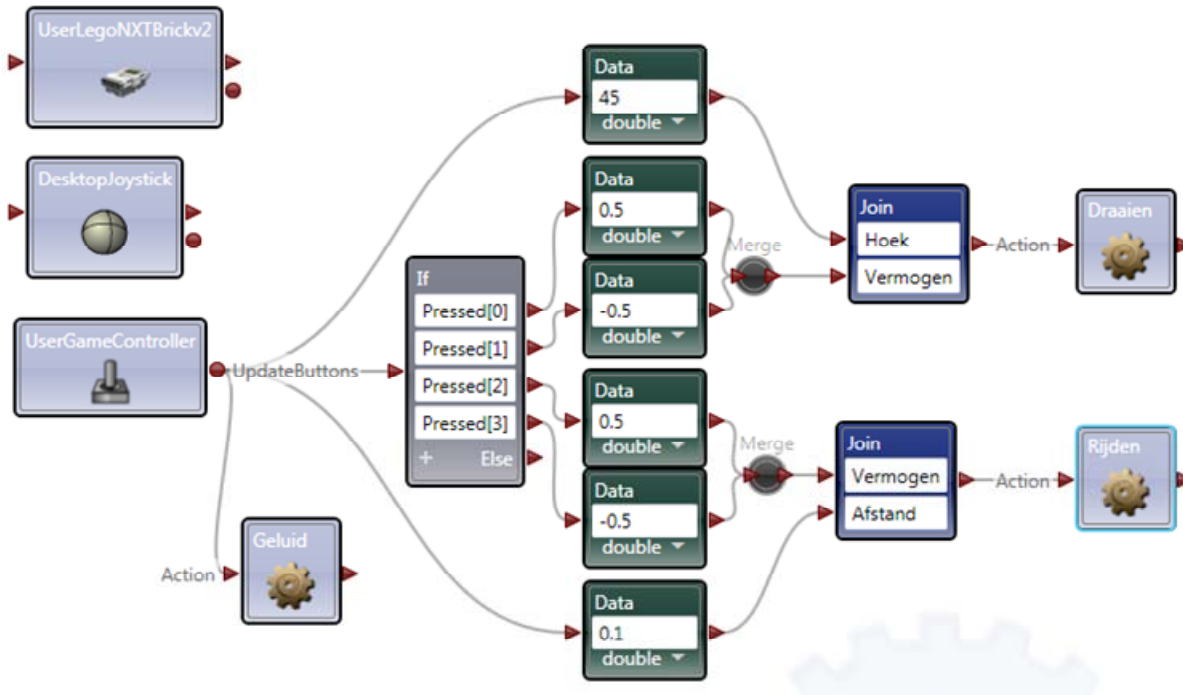
- Comment: ""
- Name: "Rijden"
- Friendly Name: "Rijden"
- Description: "Rijdt de aangegeven afstand met het opgegeven vermogen"

Maak nog een "Activity" aan voor het richting geluid en verbind die met het rondje van de Joystick.

Klik op deze "Activity" en vul in de Properties de volgende zaken in:

- Comment: ""
- Name: "Geluid"

Het diagram moet er nu ongeveer als volgt uit zien:



De Draaien Activity

Dubbelklik op de "Draaien" Activity om deze te kunnen bewerken.

Druk op de rode button naast "Action" onder het tabblad label. Druk in de "Actions and Notifications" dialoog onder "Input values:" op de "Add" button. Verander het woord "Field" in "Hoek" en selecteer "double" als het "Type". Voeg op dezelfde manier "Vermogen" toe. Druk op de "Ok" button.

Maak twee "Calculates" aan en type in de ene "Hoek" en in de andere "Vermogen". Dit is nodig om de "Input Values" uit te lezen. Maak een "Join" en verbind de "Calculates" ermee. De message bij "Hoek" noem je "Degrees" en die bij "Vermogen" noem je "Power".

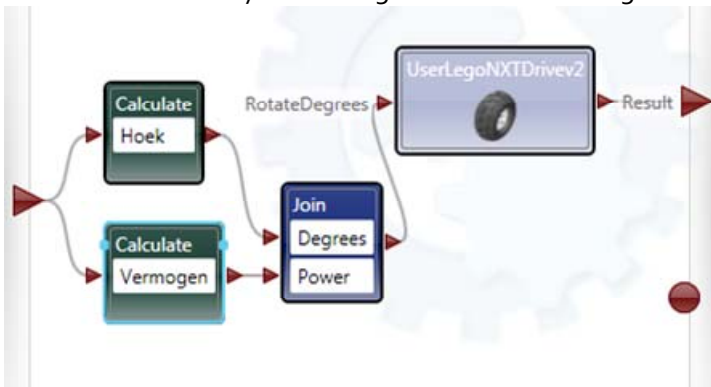
Maak een Service "Lego NXT Drive (v2)" aan en configureer deze zoals beschreven in paragraaf [Configureren van Motoren in VPL](#). Nu is het dus belangrijk dat alle waarden kloppen.

Verbind de uitgaande pijl van de Join met "LegoNXTDrivev2" en kies "RotateDegrees" in de "Connections" dialoog. In de "Data Connections" dialoog zijn "Degrees" en "Power" al gematched maar je moet nog even "DriveStage.InitialRequest" voor "RotateDegreesStage".

Verbind als laatste nog de uitgaande pijl van "LegoNXTDrivev2" met de "Result" pijl van de Activity en selecteer "RotateDegrees - Success"

Ga nu weer terug naar het "Diagram" tabblad en klik op de pijl tussen de Join en de "Draaien" Activity. Match nu zowel "Hoek" als "Vermogen".

De "Draaien" activity ziet er ongeveer uit als deze figuur:



De Rijden Activity

Dubbelklik op de "Rijden" Activity om deze te kunnen bewerken.

Druk op de rode button naast "Action" onder het tabblad label. Druk in de "Actions and Notifications" dialoog onder "Input values:" op de "Add" button. Verander het woord "Field" in "Afstand" en selecteer "double" als het "Type". Voeg op dezelfde manier "Vermogen" toe.

Druk op de "Ok" button.

Maak twee "Calculates" aan en type in de ene "Afstand" en in de andere "Vermogen". Dit is nodig om de "Input Values" uit te lezen. Maak een "Join" en verbind de "Calculates" ermee. De message bij "Afstand" noem je "Distance" en die bij "Vermogen" noem je "Power".

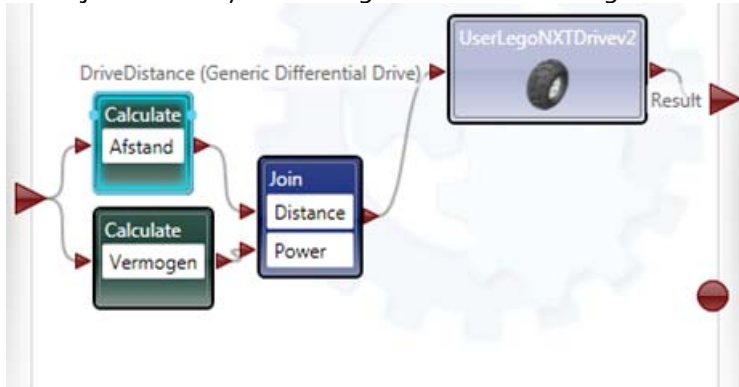
Maak een Service "Lego NXT Drive (v2)" aan waardoor je de dialog "Add Activity" krijgt, waarbij je "LegoNXTDrivev2" kiest zodat je die van "Draaien" hergebruikt met de daar al ingevoerde configuratie.

Verbind de uitgaande pijl van de Join met "LegoNXTDrivev2" en kies "DriveDistance (Generic Differential Drive)" in de "Connections" dialoog. In de "Data Connections" dialoog zijn "Distance" en "Power" al gematched maar kies nog even "DriveStage.InitialRequest" voor "DriveDistanceStage".

Verbind als laatste nog de uitgaande pijl van "LegoNXTDrivev2" met de "Result" pijl van de Activity en selecteer "DriveDistance (Generic Differential Drive) – Success"

Ga nu weer terug naar het "Diagram" tabblad en klik op de pijl tussen de Join en de "Rijden" Activity. Match nu zowel "Afstand" als "Vermogen".

De "Rijden" activity ziet er ongeveer uit als deze figuur:



De Geluid Activity

Dubbelklik op de "Geluid" activity om deze te bewerken.

Druk op de rode button naast "Action" onder het tabblad label. Druk in de "Actions and Notifications" dialoog onder "Input values:" op de "Add" button. Verander het woord "Field" in "Pressed" en selecteer "List of bool" als het "Type". Druk op de "Ok" button.

Maak een "If" met 4 keuzes te weten "Pressed[0]", "Pressed[1]", "Pressed[2]" en "Pressed[3]".

Verbind ieder met een "Data". *Wanneer je een pijltje sleept naar een lege plek op het diagram krijg je automatisch een PopUp met de Basic Activities en de Activities die je al aan het gebruiken bent.*

Geef de Data's achtereenvolgens de waarden, zonder quotejes: "Right.rso", "Left.rso",

"Forward.rso" en "Backwards.rso". Dit zijn de geluidjes die je in de paragraaf [Bestanden op de NXT Brick zetten](#) op de Brick hebt gezet. Verander het type van alle Data's in "string".

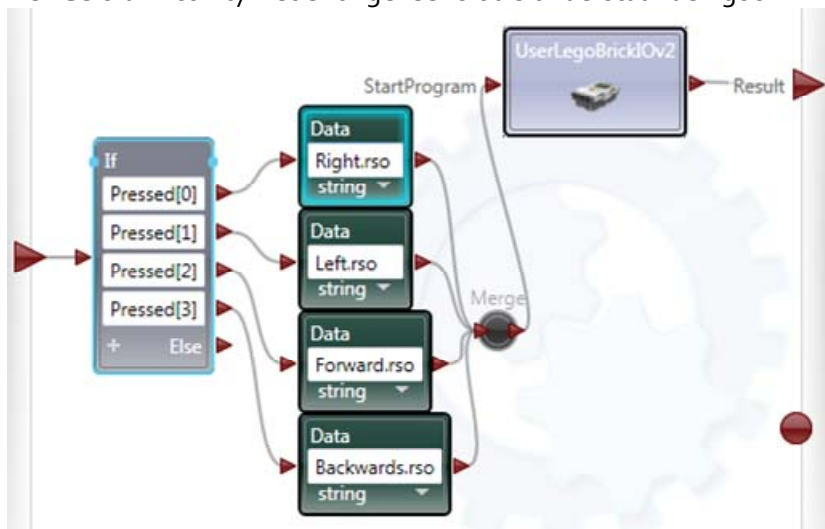
Maak een Merge en verbindt alle Data's ermee.

Maak een "Lego Brick I/O (v2)" aan. Deze hoeft je niet te configureren omdat "LegoNXTBrickv2" toch de enig geschikte partner is. Verbind de Join met "LegoBrickIOv2" en kies "StartProgram" in de "Connections" dialoog. Kies "value" in de "Data Connections" dialog voor "Value".

Verbind de uitgaande pijl van de "LegoBrickIOv2" met de uitgang van de Activity en kies "StartProgram – success" in het "From" veld van de "Connections" dialoog.

Ga nu weer terug naar het "Diagram" tabblad en klik op de pijl tussen de Join en de "Geluid" Activity en match "Pressed".

De "Geluid" Activity ziet er ongeveer uit als onderstaande figuur:



Simulator

Je kunt overigens ook met je Joystick in de simulator rijden. Maak een nieuw programma genaamd JoyStickNXT4.mvpl.

Plaats de service "Simulation Engine" op je diagram.

Selecteer hem en kies rechts onder "Properties" voor "Configuration": "Use a manifest".

Selecteer onder "Manifest" 1 van de 3 mogelijkheden en start de applicatie.

Er verschijnen nu 2 of 3 windows waarvan er 1 "Dashboard" heet welke je moet selecteren. Wanneer je een Joystick hebt aangesloten zie je deze linksboven naast het label "Device".

Klik nu op de "Connect" button van het Dashboard om een verbinding te leggen met de aanwezige robots in de simulator. Dubbelklik de robot die in het lijstje verschijnt.

Druk op de "Drive" button. Als deze van kleur verandert en die kleur houdt is de verbinding tot stand gekomen.

Nu kun je met je Joystick of de muis met je robot rijden.

Volgende keer

Volgende aflevering gaat over programmeren van loops en notification events om meerdere services elkaar te laten opvolgen en het gebruik van sensoren.

Colofon

Redacteur

Ed Buzzi

Redacteur

Zeno Otten

Technisch adviseur

Bert Berrevoets

Auteur

Iwan Tolboom

Copyright 2011, www.twintellect.com

Email: iwan.tolboom@chello.nl

Skype: iwantolboom