

Netcentric Computing

Patrick van der Pal – 6223931,
Mike Trieu – 6366295 / 10105093
&
Floris Turkenburg – 10419667

Labboek

Week 1 / 02-06-2014 – 06-06-2014

Chapter 1: De opdracht en de theorie.

De opdracht voor het vak Netcentric Computing was een servo-systeem door middel van een mobiel Android apparaat te besturen. Een servo-systeem bestaat uit een elektromotor met op de as een potentiometer gemonteerd, waarmee de as-positie kan worden gemeten. Om dit servo-systeem aan te sturen, wordt het aangesloten op een MBED microcontroller en een mobiel Android apparaat. Samen vormen ze het regelsysteem. Dit regelsysteem, een node, wordt opgenomen in een sensor netwerk van nodes. Binnen dit netwerk kunnen de nodes onderling communiceren en elkaars functie overnemen. Hiervoor wordt Bluetooth gebruikt van de Android mobile apparaten.

De MBED microcontroller bestaat uit een ARM processor, geheugen en verschillende programmeerbare input/output randapparaten. Het is speciaal ontworpen voor rapid prototyping. Rapid prototyping wil zeggen dat er snel werkende programma's die delen van een systeem aansturen geprogrammeerd kunnen worden. Dit is handig om de werking van een systeem sneller te begrijpen.

De potentiometer van het servo-systeem wordt opgemeten in spanning tussen de nul en één, terwijl de as-positie van nul tot tien loopt. Om door middel van een mobiel Android apparaat de as-positie van een servo-systeem te verplaatsen, moet de as-positie tegen de spanning opgemeten worden. Uit deze gemeten waarden moet een functie berekend worden. Aan de hand van polynomiale functies kan iedere functie benaderd worden. Dit is de reden dat wij polynomiale regressie hebben gebruikt om een functie op te stellen die behoort bij de gemeten punten. Het idee achter polynomiale regressie is dat een polynomiale functie

$$y_i = a_0 + a_1x_i + a_2x_i^2 + \dots + a_mx_i^m + \varepsilon_i \quad (i = 1, 2, \dots, n)$$

herschreven kan worden tot een matrix vergelijking

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Deze vergelijking in vector vorm:

$$\vec{y} = \mathbf{X}\vec{a} + \vec{\varepsilon}.$$

Vervolgens kan dit met behulp van de kleinste kwadraten methode weer herschreven worden tot

$$\hat{\vec{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y}.$$

Hiermee zijn alle coëfficiënten uit de originele polynomiale functie berekend en is deze volledig bekend. Met deze functie kan de voltage berekend worden uit de as-positie.

Om de as-positie uit het voltage te berekenen moet er een inverse functie berekend worden aan de hand van de gemeten waarden. Om de inverse functie te berekenen dienen simpelweg de x en y waardes omgewisseld te worden, waarna dezelfde aanpak als die hierboven beschreven is, gebruikt kan worden.

Chapter 2: Het plan van aanpak.

1. Het installeren van de opstelling, namelijk het servo-systeem, de MBED microcontroller en het mobiele Android apparaat.
 - a. Sluit de MBED microcontroller via de stekker aan op het stroomnet.
 - b. Sluit de MBED microcontroller via de USB-kabel aan op een computer.
 - c. Sluit poort 17 van de MBED microcontroller via een stroomdraadje aan op de uitgang van de potentiometer van het servo-systeem.
 - d. Sluit poort 27 van de MBED microcontroller via een stroomdraadje aan op de elektromotor van het servo-systeem.
 - e. Sluit poort 28 van de MBED microcontroller via een stroomdraadje aan op de elektromotor van het servo-systeem.
 - f. Sluit de poort met 3.3 volt van de MBED microcontroller via een stroomdraadje aan op de 5 volt uitgang van het servo-systeem.
 - g. Sluit de grondpoort van de MBED microcontroller via een stroomdraadje aan op de grond uitgang van het servo-systeem.
2. Het installeren van de benodigde software, waaronder de HTC Sync Manager, MBED compiler, Tera Term en Android Studio.
3. Het mogelijk maken van het uitlezen van de gemeten spanning.
 - a. Om de gemeten spanning uit te lezen via poort 17 van de MBED microcontroller is de volgende code essentieel:

```
AnalogIn voltage(p17);  
float value = voltage.read() of float value = (float) voltage
```
 - b. Meet handmatig de waarde van de punten een tot en met tien van de potentiometer en noteer de gemeten waardes.
4. Het opstellen van een continue functie aan de hand van de gemeten spanning.
 - a. Gebruik de eerder beschreven methode om de coëfficiënten van de polynomiale functie te bepalen.
5. Het mogelijk maken van het aansturen van de potentiometer via de terminal van Tera Term, aan de hand van de in stap 4 opgestelde functie.
 - a. Eerst moet via de terminal een waarde verstuurd worden.
 - b. Deze verstuurde waarde wordt door de MBED microcontroller door de continue functie gehaald worden om deze om te zetten in een waarde congruent met de spanningswaardes die zijn gemeten.
 - c. Vervolgens wordt aan de hand van het verschil tussen de gewenste positie en de huidige positie bepaald welke kant de elektromotor moet gaan draaien, totdat de gewenste positie bereikt is.
6. Het opstellen van de inverse functie van de continue functie, opgesteld in stap 4.

- a. Gebruik de eerder beschreven methode om de coëfficiënten van de polynomiale functie te bepalen.
7. Het mogelijk maken van het uitlezen van de positie van de potentiometer aan de hand van de inverse functie, opgesteld in de vorige stap.
 - a. Net zoals eerder in stap 3a moet de gemeten spanning uitgelezen worden.
 - b. Vervolgens wordt de waarde opgenomen in een variabele, waarna deze variabele door de inverse functie wordt gehaald zodat de positie van de potentiometer bekend is.
8. Het verbinden van het mobiele Android apparaat met de rest van het systeem.
 - a. Sluit de HTC Desire C via de USB-kabel aan op de MBED microcontroller.
 - b. Als gevolg van het framework kan er direct via de USB-kabel verbinding gemaakt worden tussen het Android apparaat en de MBED microcontroller.
9. Het uitvoeren van de opdrachten uit stap 5 en stap 7 op het mobiele Android apparaat
 - a. Het ontwerpen van de applicatie.
 - i. Voor het ontwerp van de applicatie laten we veel van het framework onveranderd. Het Bluetooth gedeelte blijft volledig gelijk, terwijl het MBED gedeelte zodanig moet worden aangepast dat er ruimte is voor een invulscherm om een positieve waarde in te vullen en ruimte voor twee knoppen, waarvan er één gebruikt wordt om de ingevulde waarde te versturen en één om de huidige positie van de potentiometer op te vragen.
 - b. Het omzetten van de invoer en de uitvoer van de eerder opgezette functies.
 - i. Voor de nieuwe wijze van invoer maken we gebruik van de volgende code:


```
getMbed().manager.write( new MbedRequest( COMMAND_GET, args ));
```

Terwijl de uitvoer simpelweg wordt ontvangen in een Serializable object:

```
Serializable object = intent.getSerializableExtra( MasterManager.EXTRA_OBJECT );
```

Voor het geval er nog meer getypt moet worden in dit lettertype.
10. Het verbinden van het mobiele Android apparaat met een ander Android apparaat.
 - a. Als gevolg van het framework kan er direct via Bluetooth verbinding gemaakt worden tussen twee Android apparaten.
11. Het uitvoeren van de opdrachten uit stap 5 en stap 7 via een ander Android apparaat.
 - a. Nadat deze functies eerder al correct geïmplementeerd zijn, is alleen nog nodig dat de juiste gegevens worden verstuurd naar het juiste apparaat.

Voor de berichten van de Master naar de Slave wordt één constante gebruikt, namelijk 20, om aan te geven dat de Master de 'Current Position' aanvraagt. Als de Master de potentiometer wilt instellen, wordt simpelweg de gegeven waarde meegestuurd, uiteraard binnen de range van 0 tot 10.

```
bluetooth.slave.sendToMaster( Bericht );
```

Voor de berichten van de Slave naar de Master geldt enkel dat de berichten met de opgevraagde positie van de potentiometer wordt verstuurd.

```
bluetooth.master.sendToAll( Bericht );
```

Chapter 3: De apparatuur en software

De gebruikte apparatuur voor de opdracht:

- HTC Desire C
- Servo-systeem
- MBED NXP LPC1768 microcontroller
- Animatronic Laboratory Board
- Verscheidene kabels, waaronder:
 - 2 USB-kabels, 1 Stekkerkabel en 5 Stroomdraadjes
- Computer

De gebruikte software voor de opdracht:

- Android Studio 0.5.2
- Online MBED compiler
- Tera Term

Chapter 4: De bezigheden.

02-06-2014 / Maandag

- Een kort hoorcollege van Edwin.
- De volgende punten van het experiment zijn uitgevoerd:
 - Het aansluiten van de as-opnemer op de ADC van de MBED microcontroller.
 - Het aansturen van de elektromotor met de stroomversterker op Animatronic board.
 - Het regelen van de as-positie van het servo-systeem door de MBED.
 - Sluit de opnemer aan op de MBED en print de gemeten waarden naar het scherm.
 - Maak een opstelling waarin de motor door middel van een terminal programma het servo-systeem in beide richtingen kan laten draaien.
 - Breidt de toepassing uit zodat via een terminal programma de positie kan worden geregeld en de huidige positie kan worden opgevraagd.

03-06-2014 / Dinsdag

- Een kort hoorcollege van Floris den Heijer.
- Mike Trieu is ons groepje komen versterken.
- De volgende punten van het experiment zijn uitgevoerd:
 - Ontwerpen van een Grafische User Interface (GUI) op de Android om de gewenste as-positie van het systeem te kunnen instellen en huidige as-positie te tonen.
 - In grote lijnen is deze stap afgerond, enkel een aantal kleine aanpassingen blijven nog staan, voornamelijk in de gebruiksvriendelijkheid.
 - De voorbeeld knoppen en functies average, sum en led party weggehaald. Deze dienden als voorbeeld om de communicatie tussen de MBED microcontroller en een Android mobile apparaat te laten zien.
- De volgende punten van de verslaglegging zijn uitgevoerd:
 - Een grafiek van de positie sensor uitgezet tegen de gemeten spanning. Schema's van de opstellingen en een korte omschrijving over de opstelling.

04-06-2014 / Woensdag

- De volgende punten van het experiment zijn uitgevoerd:
 - Ontwerpen van een Grafische User Interface (GUI) op de Android om de gewenste as-positie van het systeem te kunnen instellen en huidige as-positie te tonen.

- Het implementeren van functies om het servo-systeem met andere nodes te laten communiceren.

05-06-2014 / Donderdag

- De volgende punten van het experiment zijn uitgevoerd:
 - De geschreven waarde van GetPosition uit het ontwerp van de applicatie verwijderen. Deze zijn vervangen door de toastShort functie. Dit maakte het mogelijk voor de Master node de posities van alle Slave nodes achterelkaar te laten zien.
 - De knoppen en functies behorend bij het ping systeem zijn verwijderd. Deze dienden als voorbeeld om de werking van de communicatie door middel van Bluetooth te laten zien.

Chapter 5: Het eindresultaat.

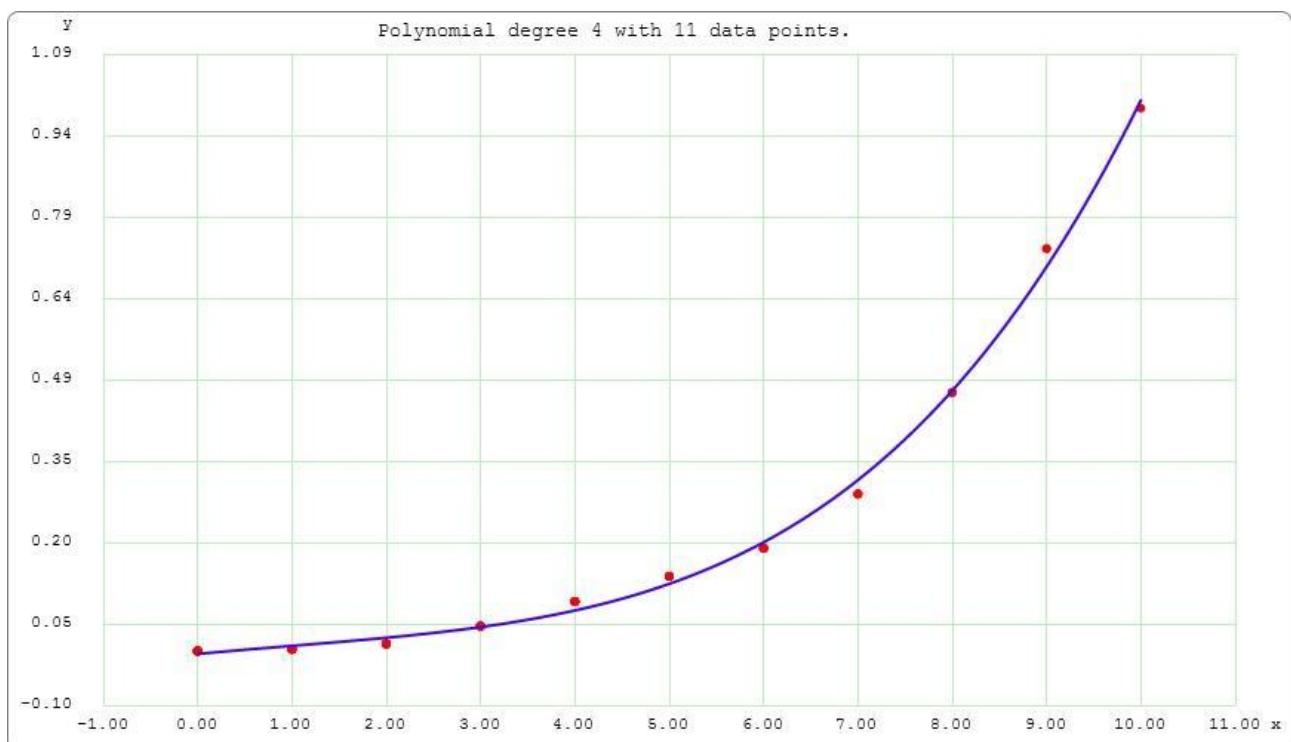
De uiteindelijke Android applicatie heeft de mogelijkheid om de potentiometer aan te sturen door middel van het invulscherm en de 'Send' knop. Ook is het mogelijk om de huidige positie van de potentiometer op te vragen.

Beide delen van de functionaliteit kunnen ook uitgevoerd worden op een ander Android apparaat via Bluetooth. Logischerwijs moet hiervoor eerst een Bluetooth verbinding gemaakt worden tussen de twee apparaten. Hiervoor zijn twee extra knoppen toegevoegd. Één knop om de Master binnen de verbinding te worden, waarmee gezocht kan worden naar Slaves voor de verbinding. De andere knop wordt gebruikt om een Slave te worden binnen de verbinding.

De functies

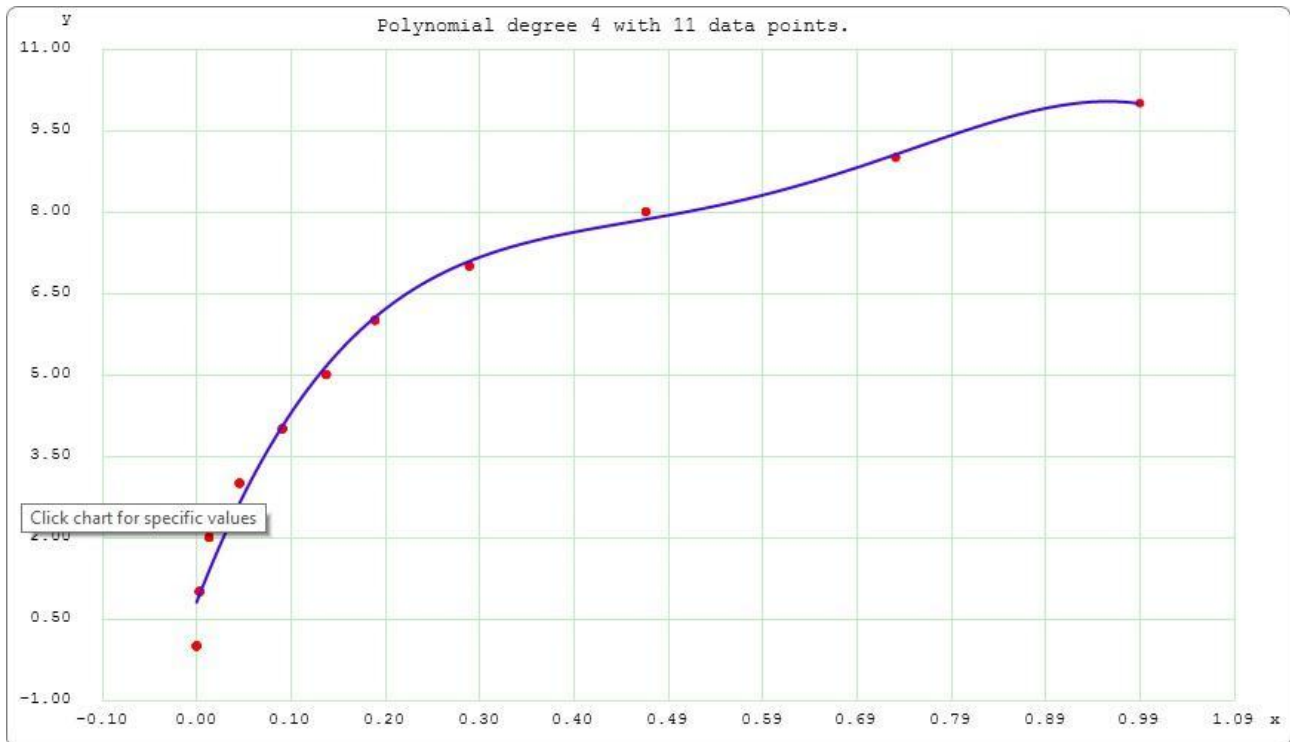
Tijdens het experiment was het nodig om twee functies op te stellen, één normale continue functie en diens inverse functie. Hiervoor hebben wij gebruik gemaakt van het PolySolve programma. Dit programma heeft alle datapunten als invoer en de polynomiale coëfficiënten als uitvoer. Ook toont het programma de uiteindelijke polynomiale functie.

$$f(x) = -5.594 \cdot 10^{-3} + 1.587 \cdot 10^{-2}x - 1.615 \cdot 10^{-3}x^2 + 4.085 \cdot 10^{-4}x^3 + 6.031 \cdot 10^{-5}x^4$$



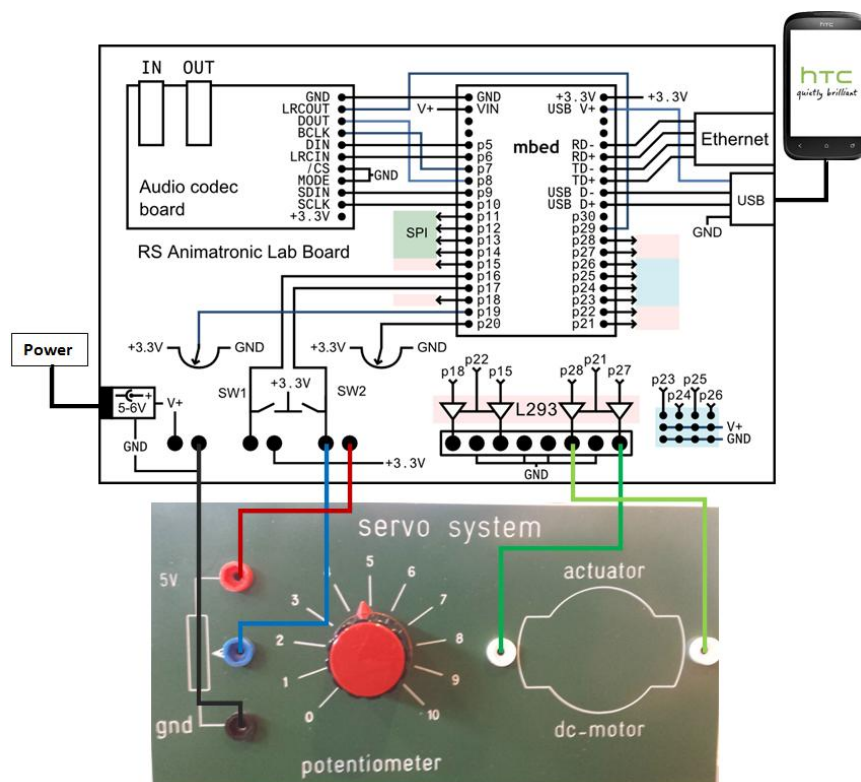
Voor de inverse functie heeft PolySolve enkel de 'Reverse' knop, hiermee wordt automatisch de inverse functie berekend.

$$f(y) = 8.044 \cdot 10^{-1} + 4.551 \cdot 10^1 y - 1.171 \cdot 10^2 y^2 + 1.378 \cdot 10^2 y^3 - 5.707 \cdot 10^1 y^4$$



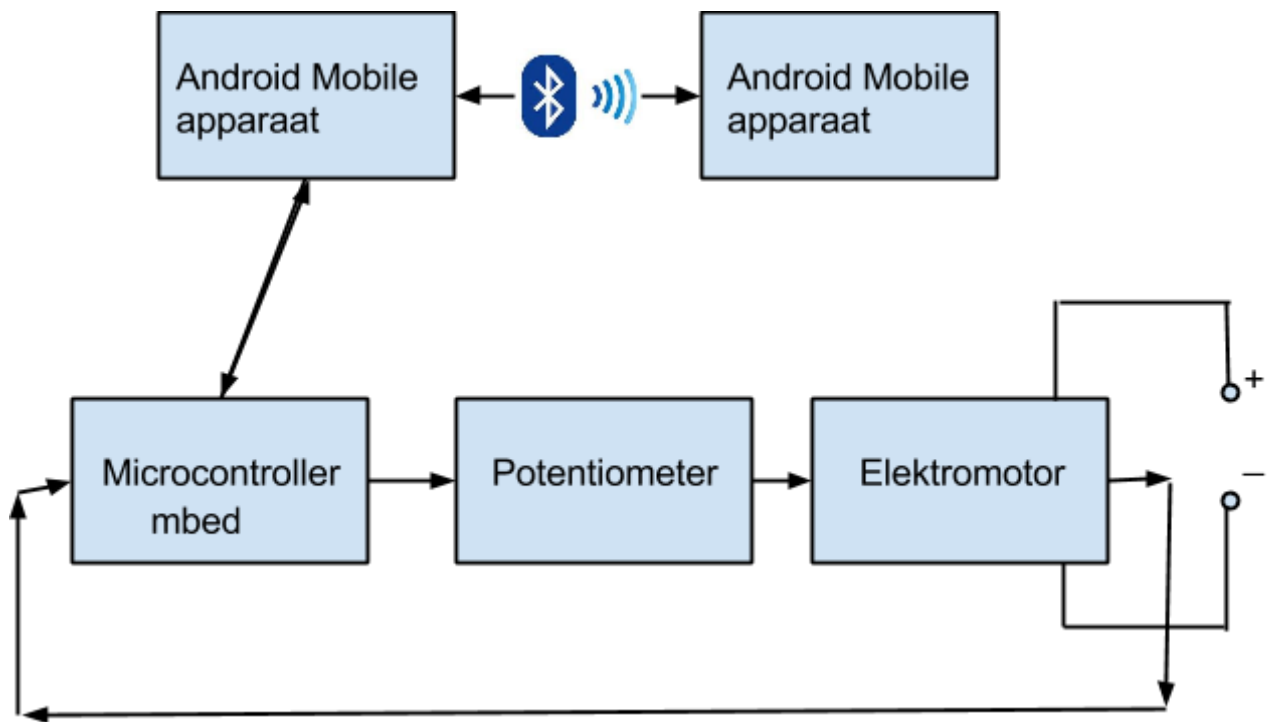
De opstelling

Zoals in het plan van aanpak is beschreven hebben we de opstelling op de volgende manier aangesloten.



Het Blokschema

Met daarbij het blokschema van de opstelling.



Chapter 6: De conclusie.

Uiteindelijk hebben we het experiment redelijk succesvol afgerond. We hebben vrijwel alle functionaliteit, die we aanvankelijk wilden implementeren, weten te implementeren.

Eén van de grootste struikelblokken, die we zijn tegengekomen tijdens de uitvoering van het experiment, was de gebrekkige kennis van het programmeren van Android. Een ander probleem was het juist installeren van Eclipse, hierdoor zijn we overgestapt op Android Studio.

Door een gebrek aan tijd, zijn we er niet aan toe gekomen om een extra controle, die controleert of de Slave binnen de verbinding daadwerkelijk is verbonden met de MBED microcontroller, te implementeren.

Appendix 1: De details van de apparatuur en de software.

De gebruikte apparatuur voor de opdracht:

- HTC Desire C
 - Android version: 4.0.3
 - Developer options:
 - USB debugging
 - Stay awake
- Servo-systeem
- MBED NXP LPC1768 microcontroller
 - High performance ARM® Cortex™-M3 Core
 - 96MHz, 32KB RAM, 512KB FLASH
 - Ethernet, USB Host/Device, 2xSPI, 2xI2C, 3xUART, CAN, 6xPWM, 6xADC, GPIO
- Animatronic Laboratory Board
- Verscheidene kabels, waaronder:
 - 2 USB-kabels, 1 Stekkerkabel en 5 Stroomdraadjes
- Computer met Windows 7/8

De gebruikte software voor de opdracht:

- Android Studio 0.5.2, SDK API19
 - <http://developer.android.com/sdk/installing/studio.html>
- HTC Sync Manager
 - <http://www.htc.com/us/software/htc-sync-manager/>
- Online MBED compiler
 - <https://mbed.org/compiler/>
- Tera Term
 - <http://tssh2.sourceforge.jp/index.html.en>
- PolySolve
 - <http://www.arachnoid.com/polysolve/>

Appendix 2: De broncode.

Zie bijlage ncc_week1.zip

Android App Framework

- <http://staff.science.uva.nl/~edwin/Downloads/NCC/NC%2028-05-2014.zip>

MBED binary Framework

- Import uva_nc in the online MBED compiler. Bij het importen kan er gezocht worden naar dit programma

Aangepaste files van het Android App Framework:

- MainActivity.java
- activity_main.xml

Aangepaste files van het MBED binary code:

- NetCentricApp.cpp
- NetCentricApp.h

Appendix 3: De Tussenresultaten.

Een tabel met de meetresultaten, gebruikt voor de calibratie.

	3 Volt	5 Volt
0	0.000	0.000
1	0.003	0.004
2	0.013	0.020
3	0.045	0.069
4	0.090	0.133
5	0.136	0.208
6	0.187	0.284
7	0.286	0.417
8	0.471	0.733
9	0.733	1.000
10	0.989	1.000