

Broomba

3. SEMESTERPROJEKT

Aarhus University School of Engineering

Modul & Integrations test

EE3PRJ3 - GRUPPE 16

<i>Christoffer Broberg</i>	<i>201605997</i>
<i>Mathias Fredslund Jensen</i>	<i>201502302</i>
<i>Erik Kato Ipsen</i>	<i>201406553</i>
<i>Parweiz Haqshenas</i>	<i>201408675</i>
<i>Lars Holm</i>	<i>201304971</i>
<i>Rasmus Aabo Jørgensen</i>	<i>201303676</i>
<i>Minik Nathanielsen Olsen</i>	<i>201600341</i>

VEJLEDER

Søren Nielsen

Aarhus University School of Engineering

DATO

22-12-2017

Versionshistorie

Version	Dato	Navn	Beskrivelse
0.0.1	18.12.2017	Mathias Fredslund Jensen	Dokument oprettet - dokument opsætning
0.1.0	19.12.2017	Christoffer Broberg Mathias Fredslund Jensen	Integrations test
1.0.0	20.12.2017	Mathias Fredslund Jensen Rasmus Aabo Jørgensen	Modul test
1.1.0	20.12.2017	Hele gruppen	Gennemgang og rettelser

Ordforklaring

Ord/Forkortelse	Beskrivelse
PWM	Pulse width modulation
PSoC	Programmable System-on-Chip
UART	Universal Asynchronous receiver-transmitter
Waveform	Digilent Analog måle enhed
ADC	Analog to digital converter
IHA	Ingeniør Højskolen Aarhus
NO	Normaly open
SPI	Seriell Peripheral Interface
RPI	Raspberry Pi Zero Wifi
UI	User Interface

Indhold

1 Modul test	4
1.1 Motor	4
1.1.1 Pumpe til vand	4
1.1.2 H-bro	5
1.1.3 DC motor skrubber	7
1.1.4 Motordriver	7
1.2 Batteri	9
1.2.1 Spændingsregulator	9
1.2.2 Batterisensor	11
1.2.3 Strømmåler	12
1.3 Front sensor system	14
1.3.1 Front push	14
1.3.2 Frontsensor	15
1.4 Væskesensor	16
1.4.1 LiquidSensor	16
1.4.2 LiquidControl	18
1.5 SPI-kommunikation	19
1.5.1 SPI-program	19
1.5.2 SPI-driver	20
2 Integrations test	21
2.0.1 Test 1	22
2.0.2 Test 2	22
2.0.3 Test 3	22
2.0.4 Test 4	22
2.0.5 Test 5	22
2.0.6 Test 6	22
2.0.7 Test 7	22
2.0.8 Metode	23

1 Modul test

1.1 Motor

1.1.1 Pumpe til vand

Pumpen til vand blev testet ved at give pumpeprintet et PWM signal fra Analog Discovery, og en 6V forsyning fra spændingsforsyningen. Det kan ses på Figur 1, at når pumpen kører kontinuerligt, bruger den 0.08A.



Figur 1: Test af pumpe med 40% pwm

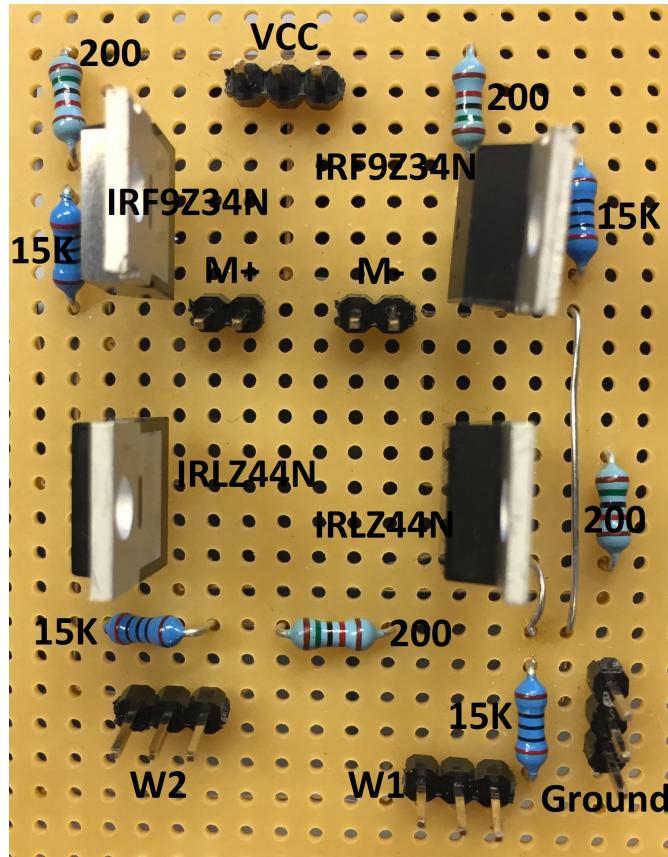
Der var dog et lille problem, da startstrømmen var lidt højere på grund af motorens stilstand. Så skal den bruge et større moment for at trække aksen rundt på slangen, end hvis den kører kontinuerligt. Det passede dog ved 40% PWM kunne den lige præcis starte med en strøm, der ses på Figur 2, og herefter stabilisere sig på de 0.08A fra Figur 1



Figur 2: Test af pumpens start strøm

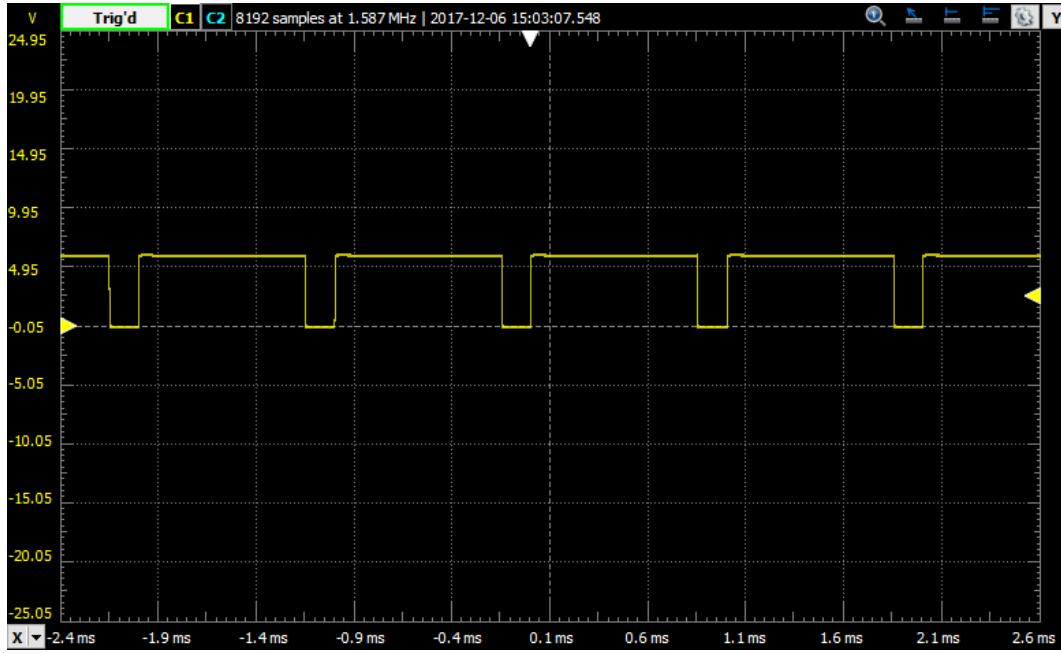
1.1.2 H-bro

Designet er loddet udfra en selvlavet kredsløbstegning Dette kan ses i figur 3. Her endes der med et kredsløb, der testes om der kommer det ønskede signal igennem eller ej.



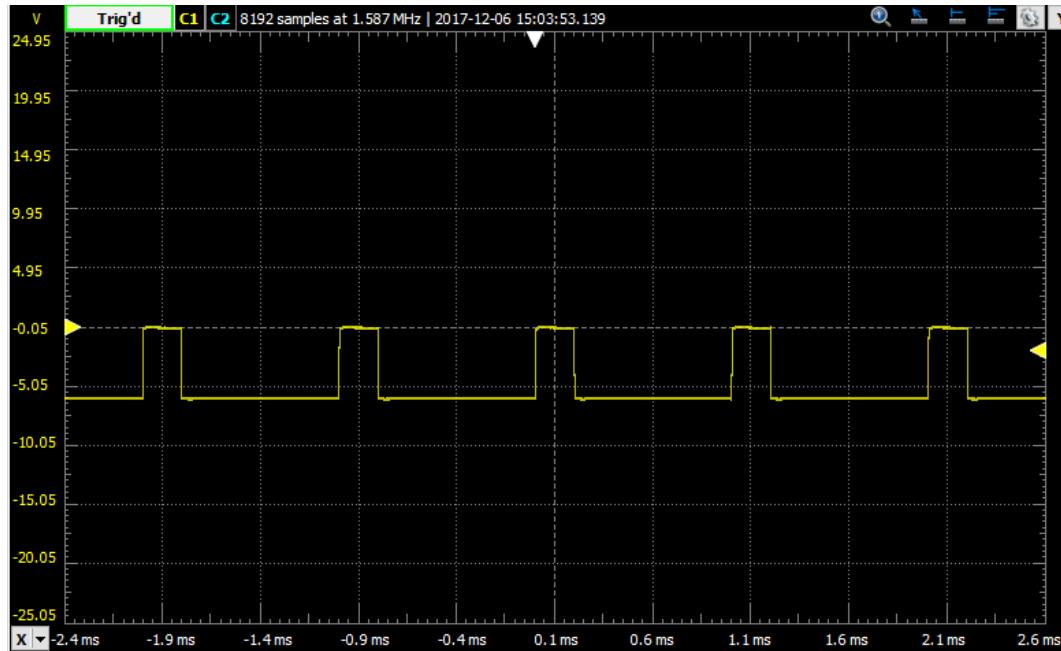
Figur 3: Her ses det loddede print, hvor man kan se på figuren hvad der er hvad.

Erstatter man et oskilloskop med motoren, så kan man måle at det PWM signal der sendes ind, stemmer overens med dét, der modtages. Derudover kan man se, at den modtager de 6V, der forventes modtaget.



Figur 4: Oscilloskopet viser et PWM signal med en duty cycle på 85, samt en spænding der svare til de 6 volt.

Ligeledes kan det samme ses når den kører i modsat retning:



Figur 5: Oscilloskopet viser et PWM signal med en duty cycle på 25, og en spænding der svare til de -6 volt.

1.1.3 DC motor skrubber

DC motor skrubber skulle blot bruge en 6V fast forsyning fra spændingsforsyningen. Her kan det ses på Figur 6 at når motoren bare står og kører kontinuerligt uden et skrubbe modul på, bruger den 0.04A



Figur 6: Test af DC motor skrubber uden modstand

For at give et mere realistisk syn på hvad motoren eventuelt ville bruge, blev der tilføjet modstand ved at sætte en finger på enden af den forlængede akse. Her kan det så ses på Figur 7, at forbruget steg meget i forhold til modstanden, der blev tilført. Ved hvad vi mente var en passende modstand, for at motoren skal trække en klud rundt på gulvet, brugte den da 0.09A istedet.

Aksen hvor der blev tilføjet modstand via en finger kan ses på Figur 8.

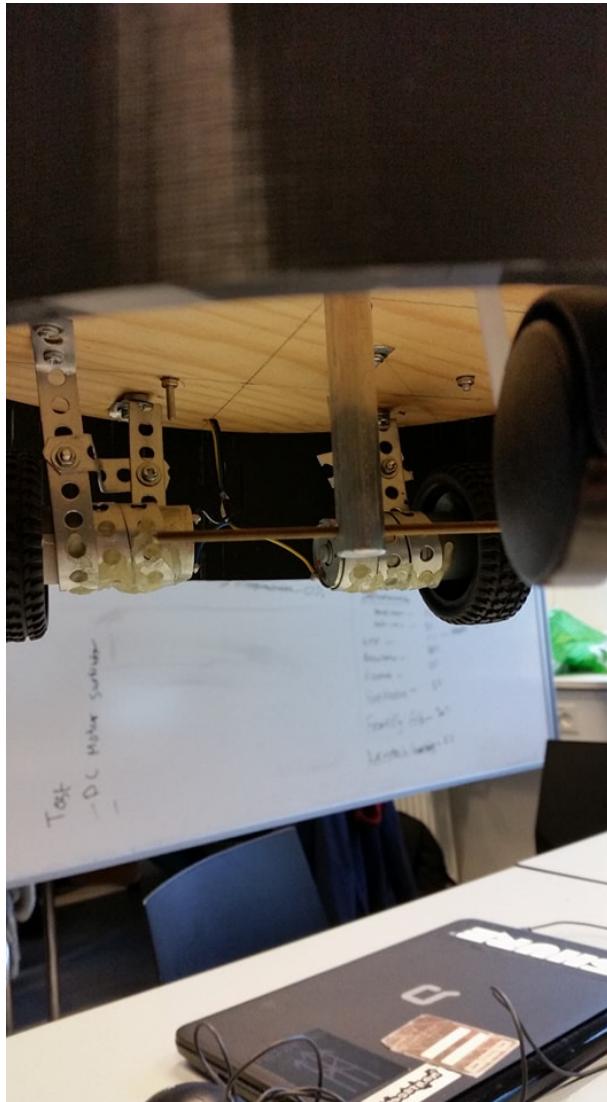


Figur 7: Test af DC motor skrubber med modstand

1.1.4 Motordriver

Motordriveren er placeret på PSoC'en. Formålet med denne driver er at styre motorerne. Jf. use case 1 skal PSoC'en styre PWM-signalerne til begge motorer, og bestemme retning samt hastighed. Motorerne styres vha. hver deres H-bro. Disse bliver så tilført PWM-signaler til enten den ene retning eller den anden retning, alt efter hvilken retning,

der skal køres. PWM 1 og 3 får begge hjul til at køre fremmad, hvor PWM 2 og 4 får dem til at køre baglæns. For at ændre hastigheden, skal vi blot ændre duty-cycle ved at skrive en ny compare value til pwm komponenten.



Figur 8: Unittest af Motor driver

For at kunne teste motordriveren, bruger vi UART til en PC. Dette lader os teste de forskellige funktioner blot ved at sende char værdier over UART. Nedenunder ses terminal vinduet med vores UART testprogram.

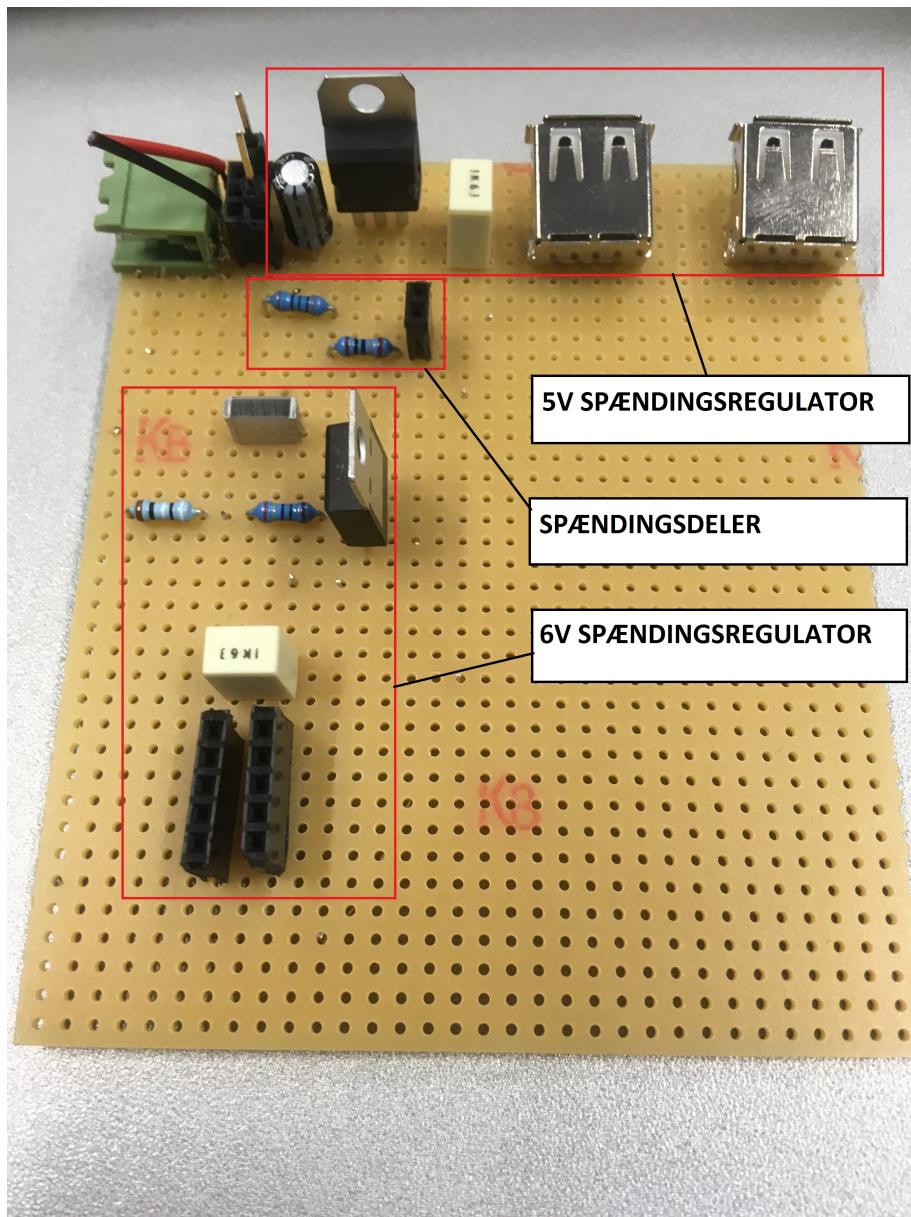
```
DC-Motor-PWM application started
0: Stop
1: Drive forwards
2: Drive backwards
q: Decrease speed
w: Increase speed
Broomba run
Broomba run
Broomba run
```

Figur 9: Unittest af motordriver vha. UART

1.2 Batteri

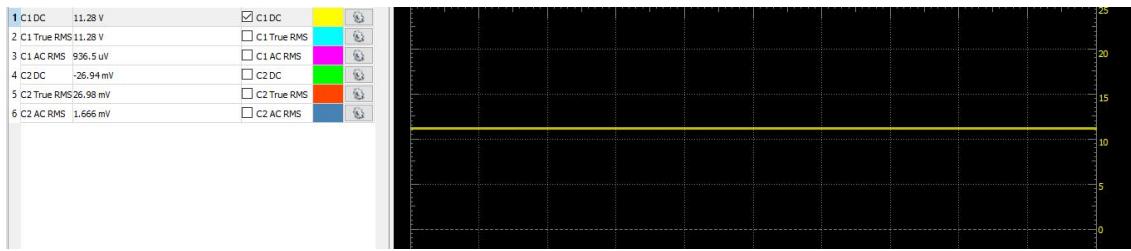
1.2.1 Spændingsregulator

Multisim simulationerne fortalte os, at implementation af spændingsregulatorerne og spændingsmåleren er mulig. Vi bygger Multisim-kredsløbet derfor disse på et veroboard, så vi kan teste print, inden vi implementerer disse på et endeligt veroboard. Testresultaterne vises i dette afsnitt. Vi foretager disse række målinger for at sikre os, at printet giver det ønskede spænding til henholdsvis DC motorerne og controllerne. Vi anvender Waveform 2015s logger funktion til dette formål.



Figur 10: Testprint af spændingsregulator og spændingsdeler

Testprintet, som Fig. 10 viser, er ikke det endelige. Hvis testprintet virker som forventet, bygger vi en mere kompakt udgave, da vi skal spare plads inde i Broomba'en.



Figur 11: Fuldt opladet batteri: 11.28V

Helt opladet batteri giver en spænding på 11.28V.



Figur 12: Vores testprint til DC Motor

Vi mäter over kredsløbets udgang. Spændingen er nu på 6.075V - En lille afgivelse fra simulationens 6.038V.



Figur 13: Vores testprint til microcontrollerne PSoC og RPI

Vi mäter microcontrollernes forsyning ved at mæle over USB-pins. Spændingen bliver 4.985V - En lille afvigelse fra simulationens 5.001V.

Vi kan nu konkludere, at vores testprint opfører sig som forventet, og vi kan derfor fortsætte med at bygge et færdigt print, der er mere kompakt.

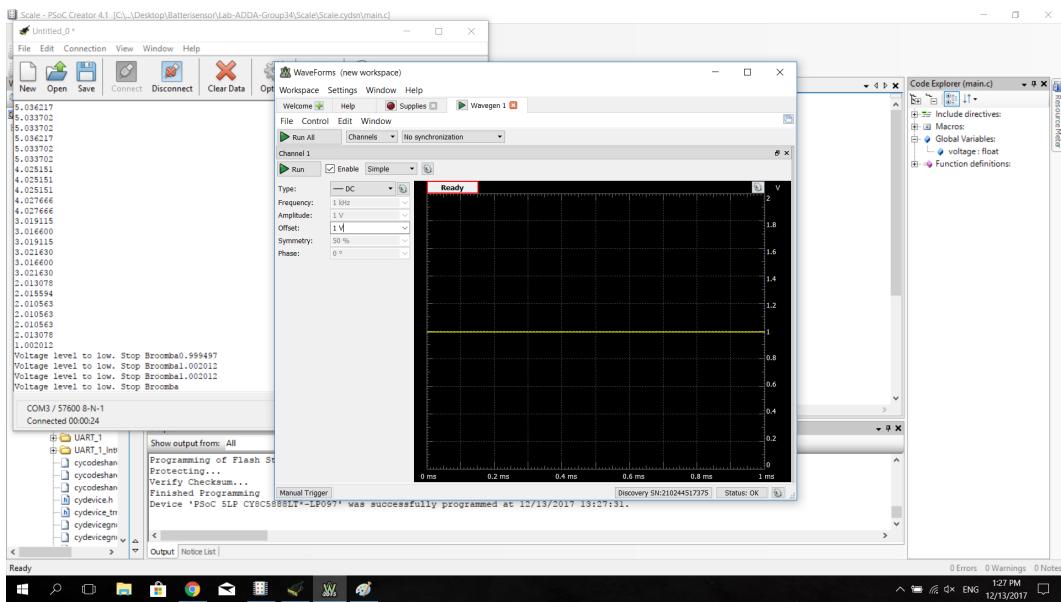
1.2.2 Batterisensor

Batterisensoren udarbejdes vha. en ADC på PSoC. ADC på PSoC er modificering et PSoC-projekt udleveret af Michael Loft, IHA, i kurset ”Grænseflader til den fysiske verden”. Batterisensorens funktion er at afbryde Broomba,

når spændingen på Broomba bliver for ”lav”. ADC'en mäter spændingen på batteriet, der er reguleret ned vha. en spændingsdeler, da PSoC max kan klare at blive påtrykket med 5V. Batteriets spænding er normalt på 9.6V, når den er fuldt opladt, men kan i nogle tilfælde overstige 11,28V, se bilag: Modul og Integrations 5.2, hvis den tages i brug lige efter den har opladet. Derfor reguleres batterispændningen ned til 1/3.

ADC modtager to input: Vref og Vin. Vref på 5V bestemmer, hvilken resolution ADC'en læser Vin med. Vin er spændingen fra batteriet reguleret ned til 1/3. Batterisensoren er kalibreret til at måle det antal volt den påtrykkes med, således at når ADC mäter 3.6, så modtager Vin 3.6V.

Grænsen for hvornår spænding på batteriet er for lav er sat til 20% af batteriets spænding, når det er fuldt opladt dvs. 1,9V. Det er dog ikke testet, hvordan Broomba rengører ved denne spænding på batteriet. Der er blot defineret en lav grænse på forhånd .



Figur 14: Unittest af batterisensor

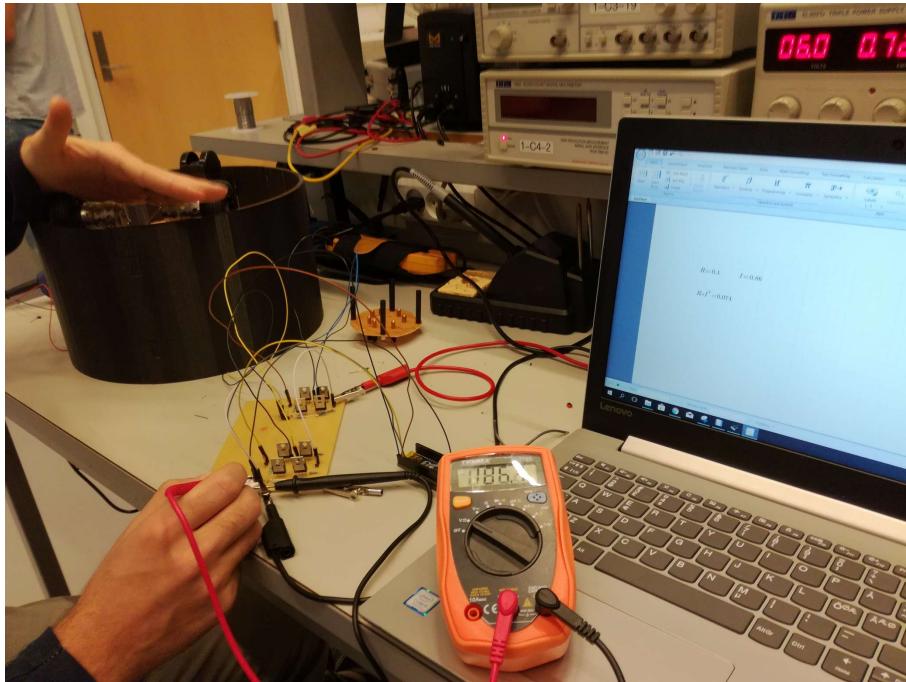
Batterisensoren på PSoC testes vha. et analog discovery. Vref modtager 5V fra source på analog discovery mens Vin modtager et signal, der reguleres ned fra 5V til under 1,9V.

1.2.3 Strømmåler

Strømmåleren er ligesom batterisensoren udarbejdet vha. en ADC på PSoC. ADC på PSoC er modificering et PSoC-projekt udleveret af Michael Loft, IHA, i kurset ”Grænseflader til den fysiske verden”. Strømmåleren fungerer som en sikkerhedsforanstaltning, der søger for at afbryde Broomba, hvis den kører fast og strømstyrken i H-broerne øges. For at måle, når strømstyrken øges, skal strømmen konverteres til en spænding ved at lade den gennemløbe en modstand på 0.1 ohm efter H-broerne.

Test af strømmåleren blev foretaget ved at nedbremse de to baghjul på Broomba med hånden, mens begge baghjul

blev påtrykket med 6V ved 100% duty cycle, og måle spændingen over modstanden på 0.1 ohm efter H-broerne. Testen viste at spændingen blev øget til 0.85V ved fuld nedbremsning mens strømstyrken blev øget fra 0.2A - 0.72A. På baggrund af undersøgelsen blev der sat en kritisk grænse på at spændingen over modstanden ikke måtte overstige 0.7V. Hvis strømmåleren målte en værdi over 0.7V skulle den afbryde Broomba.



Figur 15: Multimeteret mäter 0.86V ved 0.72A, när baghjulene bremses ved full hastighet

Modstanden kan klare at der afsættes op til 5W (henvisning til datasheet). Der tjekkes at den afsatte effekt i modstanden er under 5W ved belastning af baghjulene:

$$R := 0.1 \Omega \quad I := 0.86 A \quad \text{Under belastning.}$$

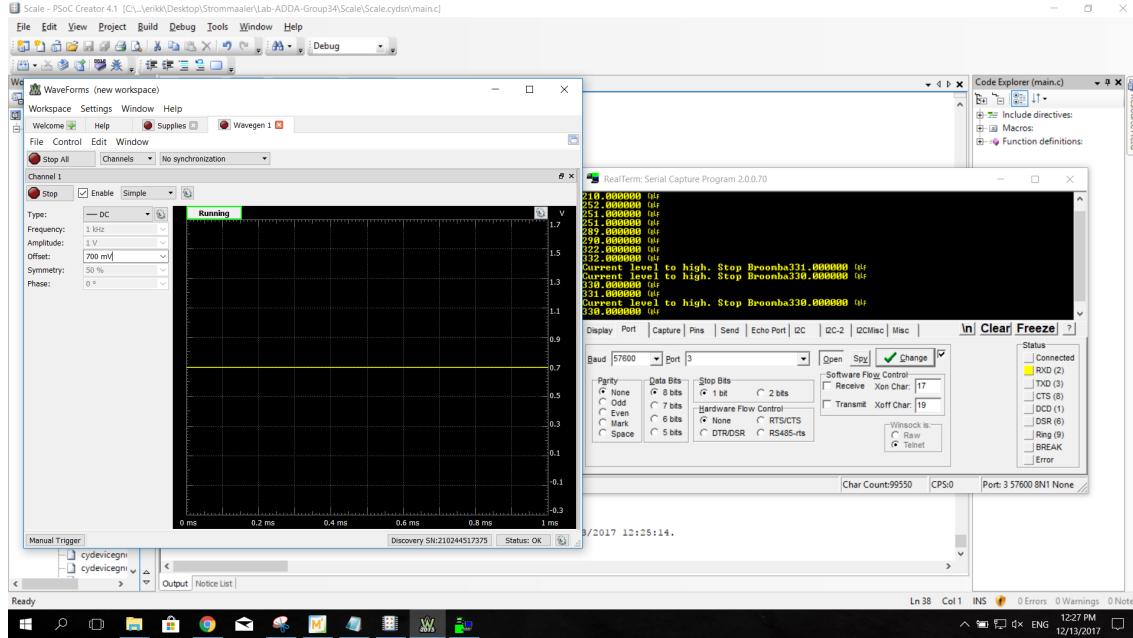
$$P := R \cdot I^2 = 0.074 W \quad P < 5W.$$

Figur 16: Beregning af den afsatte effekt i modstanden

Den afsatte effekt i modstanden er under 5W. Modstanden er godkendt.

Softwareen for strømmåleren blev testet vha. et analog discovery. Strømmåleren er opbygget ligesom batterisensoren med en ADC på PSoC. ADC modtager to input-signaler: Vref og Vin. Vref på 5V bestemmer resolutionen, der læser Vin. Vin reguleres op fra 0V-0.7V indtil, der udskrives på consol-vinduet "Current level to high. Stop

Broomba”.



Figur 17: Test af strømmåleren

1.3 Front sensor system

1.3.1 Front push

Front push blev testet ved at sende et 5V signal fra Analog discovery ind på indgangs benet på DM1[1] og måle med osciloskopet på NO benet. Som det ses på Figur 18 ved aktivering af switchen bliver der sendt et 5V signal tilbage. Der er ikke meget prel i switchen ved et tryk fra finger, dette kan dog godt være der vil komme når broomba'en kører ind i væggen og får en hop tilbage og frem, men da det bliver styret af interrupt, skal PSoC'en blot modtage signalet og så starter den interrupt rutinen hvor den alligevel starter med at bakke.



Figur 18: Test af frontsensorerne, 5V.

1.3.2 Frontsensor

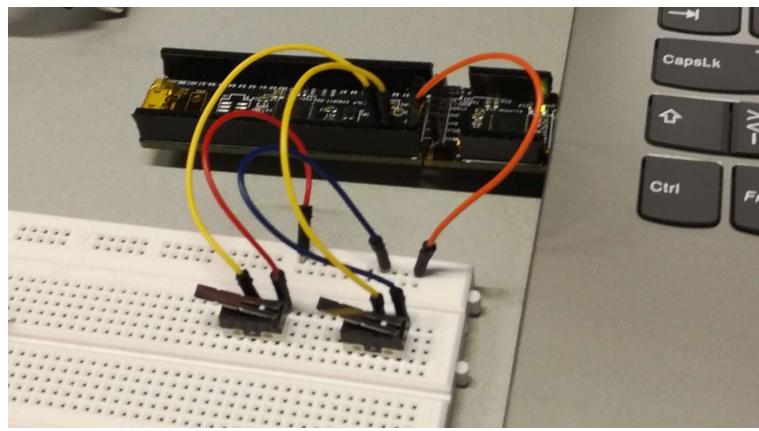
Frontsensorerne er testet ved at simulere de to tryk-sensorer på højre og venstre side af Broomba ved hjælp af to kontakter på fumlebræt (se Figur 20). Når kontakterne hver især påtrykkes, fremkaldes et interrupt på en pin på PSoC'en. Hver kontakt er forbundet til hver sin pin på PSoC, som simulerer hhv. højre eller venstre sensor. Afhængig af hvilken pin, der fremkalder interrupt udskrives en tekstbesked i consol-vinduet (se Figur 19, der simulerer hvilken sensor, der er aktiveret).

I consol-vinduet udskrives "Broomba run", når der ikke er fremkaldt interrupt. Når der fremkaldes interrupt, udskrives "Interrupt on Pin1" på consol-vinduet. Efterfølgende udskrives "Interrupt cleared", når interrupt-rutinen bliver nulstillet. Afhængig af om der fremkaldes interrupt fra højre eller venstre kontakt udskrives tekstdeskriptioner "Højre sensor" eller "Venstre sensor" for at simulere at hhv. højre- eller venstre sensor aktiveres. Efterfølgende går der 5 sekunder før der hhv. udskrives "Højre sensor gennemfoert" og "Venstre sensor gennemfoert" for at simulere processen, der går igang, når sensorerne aktiveres. Der kan i denne tid ikke fremkaldes interrupt fra kontakterne.

```
Untitled_0 *
File Edit Connection View Window Help
New Open Save Connect Disconnect Clear Data Options View Hex Help
Broomba run
Broomba run
Broomba run
Broomba run
Broomba run
Interrupt on Pin1
Interrupt on Pin1
Interrupt cleared
Interrupt cleared
Interrupt cleared
Broomba run
Højre sensor
Højre sensor gennemfoert
Broomba run
Broomba run
Broomba run
Broomba run
Interrupt on Pin1
Interrupt cleared
Interrupt cleared
Broomba run
Venstre sensor
Venstre sensor gennemfoert
Broomba run
Broomba run
Broomba run
COM3 / 57600 8-N-1
Connected 00:00:19
TX RTS DTR DCD
RX CTS DSR RI
```

Figur 19: Test af frontsensorerne. Consol output

Der kan indmellem opstå dobbelt-interrupt, når man påtrykker kontakterne. Dette opstår når man slipper kontakten igen, men dette har ikke indflydelsen på den del hver som sensorer skal udføre.



Figur 20: Test af frontsensorerne. Test med kontakter

1.4 Væskesensor

1.4.1 LiquidSensor

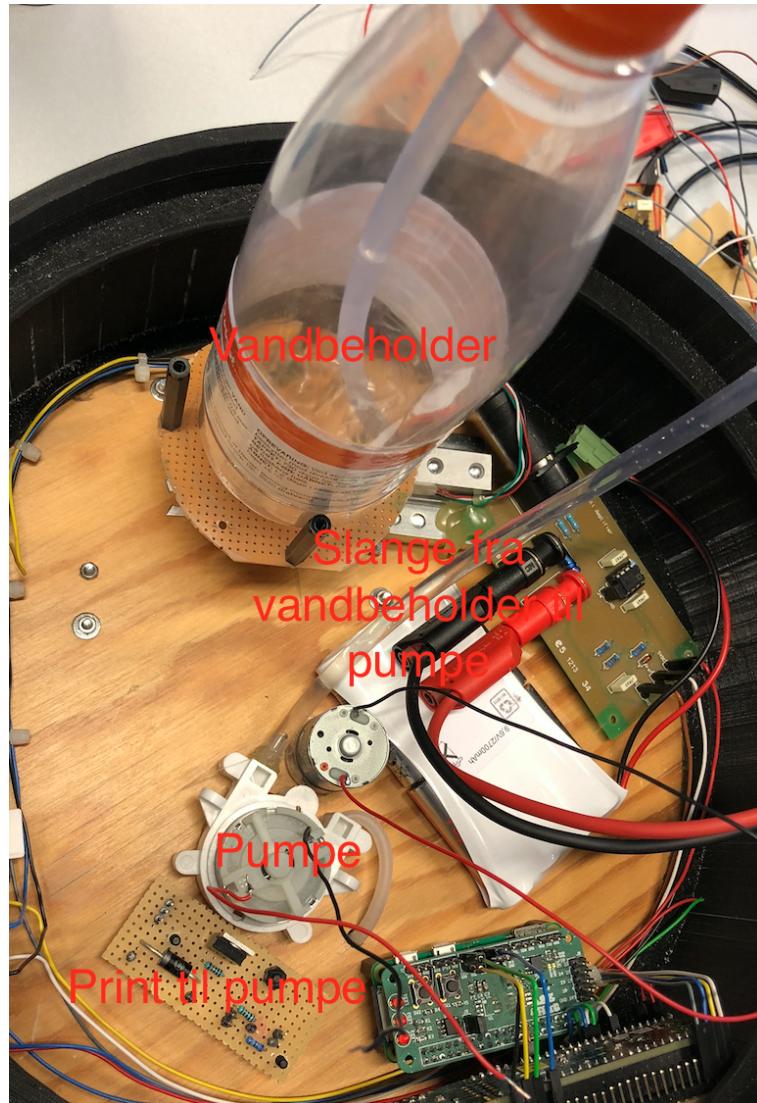
Loadcells formål er at måle hvor meget væske der er i beholderen, og videregive dette resultat til PSoC. Der er i den forbindelse lavet en driver, som kan bruges til at gøre arbejdet. Driveren har så vha. sine data udgivet en lineære funktion; $y = 1.4174x + 1471.8$, som er et udtryk for ADC-værdi pr. gram vægt. Dette resultat testes så for at se om det er pålidelig. Figur 21 og 22 viser henholdsvis målinger ved 50g og 100g vægte. Her kan man se, at loadcell kommer med et realistisk svar på, hvad vægten vejer, når man putter specifikke antal vægte op på loadcellen.

Figur 21: Måling ved 50g vægt

Figur 22: Måling ved 100g vægt

1.4.2 LiquidControl

Væskestyringen er styret igennem en peristaltisk pumpe. Der testes ved at lade pumpen være tændt i 1 minut ad gangen og derefter har en pause på 2 minutter. Pumpen kører på 40 procent duty cycle, og pumper 20 ml vand i minutet. Pumpen er forbundet til væskebeholderen, der er en 0.5L vandflaske. Der er boret et hul i låget af flasken, hvor slangen fra pumpen er limet fast til.



Figur 23: Pumpe og væskebeholder

1.5 SPI-kommunikation

1.5.1 SPI-program

SPI-programmet er placeret på RPI'en, og dens formål er at modtage beskeder via SPI-driveren. Jf. use case 2,3 og 4, så skal RPI'en skrive til brugeren, når den mangler vand, mangler strøm eller sidder fast. Metoden skal altså åbne den fil, der indeholder det, SPI-driveren modtager, samt åbne det dokument, der indeholder teksten, der lægges på webserveren, og ændre heri, hvis den får hhv. et 'B', et 'V' eller et 'O'. På billederne herunder ses det, at den modtager et 'V', hvorefter den skriver til webserveren.

```
root@raspberrypi0-wifi:~/# ./projekt
I loekken
Streng i bufferen: 86
Plads 0 i bufferen: 56
Plads 1 i bufferen: 54
Plads 2 i bufferen: 10
Plads 3 i bufferen: 0
Mail recieived. Received: V
Case V - Vandbeholder tom
```

Figur 24: Unittest af SPI-program - sender B

I figuren kan man se, at i den buffer, der modtages i, lægger der hhv. ascii-værdierne 56, 54 og 10, hvilken svarer til 8, 6 og newline. Dette er ascii-værdien for V, og der omdannes så fra ascii til én samlet char vha. metoden atoi(). På denne måde kan beskeden fortolkes og indholdet anvendes.

Paafyld vandbeholder
Paafyld vandbeholder
Paafyld vandbeholder
Paafyld vandbeholder
Paafyld vandbeholder

Figur 25: Unittest af SPI-program - skriver på webserver

1.5.2 SPI-driver

For at muliggøre SPI-kommunikation på RPI'en, er der udarbejdet en driver hertil. På figuren her kan det ses, at der i mappen ”dev” ligger en fil, spidrv0. Hvis man anvender cat til at læse på den, kan man aflæse, hvad RPI'en får ind via SPI. I dette tilfældes sendes ascii-værdien for 'V'. Det er denne fil, SPI-programmet tilgår.

```
root@raspberrypi0-wifi:~# cat /dev/spi_drv0
86
86
86
86
86
86
86
```

Figur 26: Unittest af SPI-driver - modtager V

2 Integrations test

Integrationstesten er opsat i en tabel, som kan ses herunder. Tabellens 6 tests er blevet nummereret, og efter testen kommer beskrivelse af disse tests. Derefter kommer et metode afsnit, om hvorfor der er valgt denne metode.

Test ID	Test objekt	Beskrivelse	Forventet resultat
1	Test af Motormodul, med frontsensor	start Broomba, og test hver frontsensor	Broomba kører tilbage og drejer til en side
2	Test af Motor-, Frontsensor og Liquidsensor-modul	Tilkobl Liquidsensor, og test Frontsensor samt Motormodul med Liquidsensor	Broomba skriver værdien af Liquidsensor til UART, og udskriver til webserver når væsken er lav
3	Test af Motormodul, Frontsensor og batterysensor	Start broomba, og tjek om den udskriver ADC-værdien til UART	Broomba reagere på frontsensor tryk, og aflæser batterispænding
4	Test af Motormodul, Frontsensor og Liquidcontrol	Start broomba, og tjek om den pumper vand ud	Broomba reagere på frontsensor tryk, og pumper vand
5	Test af Motormodul, frontsensor, Liquidcontrol og brush	Broomba kører med pumpe, frontsensor og børste.	Broomba reagere på frontsensor tryk, pumper vand og børste motoren kører
6	Test af Motormodul, frontsensor, Liquidcontrol og brush	Broomba kører med pumpe, frontsensor, børste og Liquidsensor.	Broomba reagere på frontsensor tryk, pumper vand og børste motoren kører, og måler væskemængde
7	Test af Liquidsensor-modul, RPICom og UI	Start program med fyldt vandbeholder, fjern derefter vandbeholder	Beskeden ”Paafyld vandbeholder” vises på brugergrænsefladen

2.0.1 Test 1

Test 1 gik ud på at teste motormodulet med frontsensor-modulet. Testen blev godkendt og modulerne fungere sammen, både højre og venstre sensor virker med ubegrænsede tryk. Broomba kan både køre, og undgå forhindringer på samme tid.

2.0.2 Test 2

Test 2 gik ud på at teste motormodulet, frontsensor-modulet og Liquidsensor-modulet. Testen blev ikke- godkendt, der er problemer med at den bliver ved med at aflæse ADC-værdien. Programmet aflæser ADC-værdien 5 gange, og er derefter bliver den blokeret. Problemets er at når først den bliver blokeret, låser den så vi ikke kan få interrupt fra frontsensorene.

2.0.3 Test 3

Test 3 gik ud på at teste motormodulet, frontsensor-modulet og batterisensor-modulet. Testen blev ikke godkendt, da vi fik samme fejl som ved test af liquidsensor.

2.0.4 Test 4

Test 4 gik ud på at teste motormodulet, frontsensor-modulet og liquidcontrol-modulet. Testen blev godkendt, modulerne fungere sammen. Broomba kan både køre, undgå forhindringer og pumpe væske på samme tid.

2.0.5 Test 5

Test 5 gik ud på at teste motormodulet, frontsensor-modulet, liquidcontrol-modulet og brush-modulet. testen blev godkendt, modulerne fungere sammen. Broomba kan både køre, undgå forhindringer, pumpe væske og bevæge børsten på samme tid.

2.0.6 Test 6

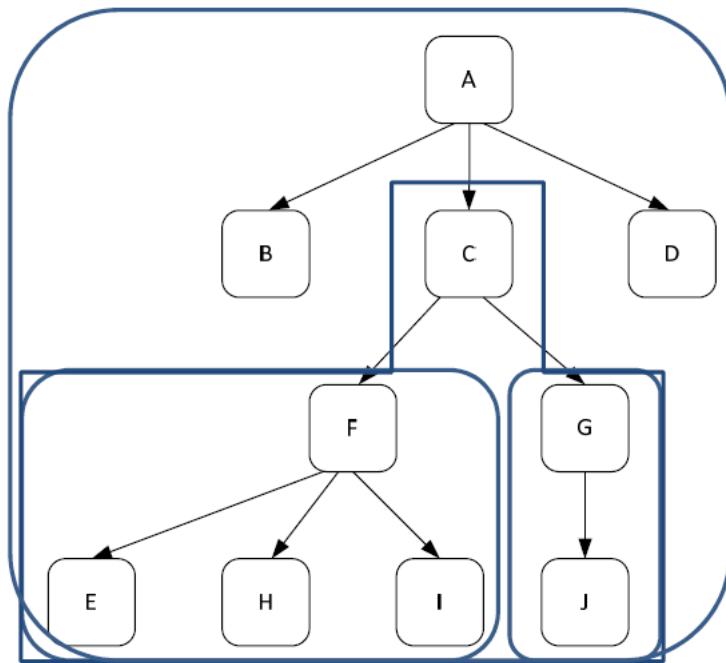
Test 6 gik ud på at teste alle modulerne fra test 5 samt liquidsensor-modulet og batterisensor-modulet. Testen blev ikke-godkendt, da det gav samme fejl som test 2 og 3.

2.0.7 Test 7

Testen går ud på at se, om der kan sendes besked til brugergrænsefladen, når en AD-converter fremtvinger en alarm. Denne test godkendes, da det kan aflæses på brugergrænsefladen, at når vandbeholderen tages af.

2.0.8 Metode

Vi har anvendt bottom-up metoden, hvor vi startede fra bunden, og tilføjede moduler en af gangen. Fordelen ved denne metode er at vi har fuld kontrol over hvilke moduler der virker. Der er dog en ulempe, hvilket er at vi ikke har en funktionel prototype. Vi har ved hjælp af metoden lokaliseret de fejl der er, og ved derfor hvad der skal arbejdes på.



Figur 27: Bottom-up integrationstestmetode