

Lab - DC Motor Control

Gruppe 50

Medlemmer: Simon Vu, Christoffer Broberg, Lars Holm

1. Øvelse 1: PWM-styring af en DC-motor

1.1 Indledning

Denne opgave handler om at styre rotationshastigheden af en DC motor vha. pulsbreddemodulering (PWM). PWM-signalet skal styres af en PSoC. Der skal laves et program, hvor et input til PSoC'en via UART bruges til at styre PWM-signalet, herunder tænde/slukke og styre hastigheden på signalet.

Der opstilles følgende forventninger til kredsløbet, der skal opnås, for at øvelsen er lykkedes:

- 1) DC-Motoren tænder, når der gives et tænd-input til PSoC via UART
- 2) DC-Motoren slukkes, når der gives et sluk-input til PSoC via UART
- 3) DC-Motoren øger hastighed, når der gives et input herom til PSoC via UART
- 4) DC-Motoren mindsker hastighed, når der gives et input herom til PSoC via UART

1.2 Implementering

Motoren styres ved hjælp af PWM. PWM gør at gennemsnitsspændingen bliver mindre på motorens input alt afhængig af hvor høj duty cycle der er på PWM signalet. Vi opretter et PWM signal ved hjælp af PWM komponenten i PSoC Creator.

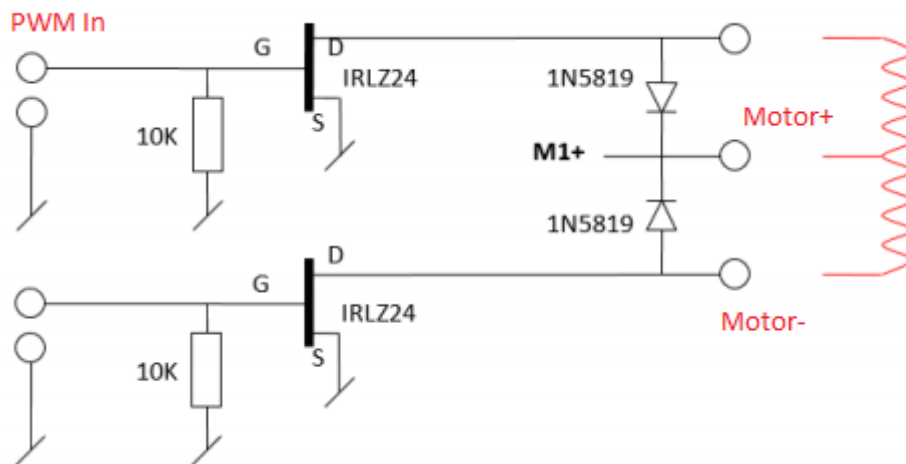
1.2.1 Komponenter

Der bruges følgende komponenter:

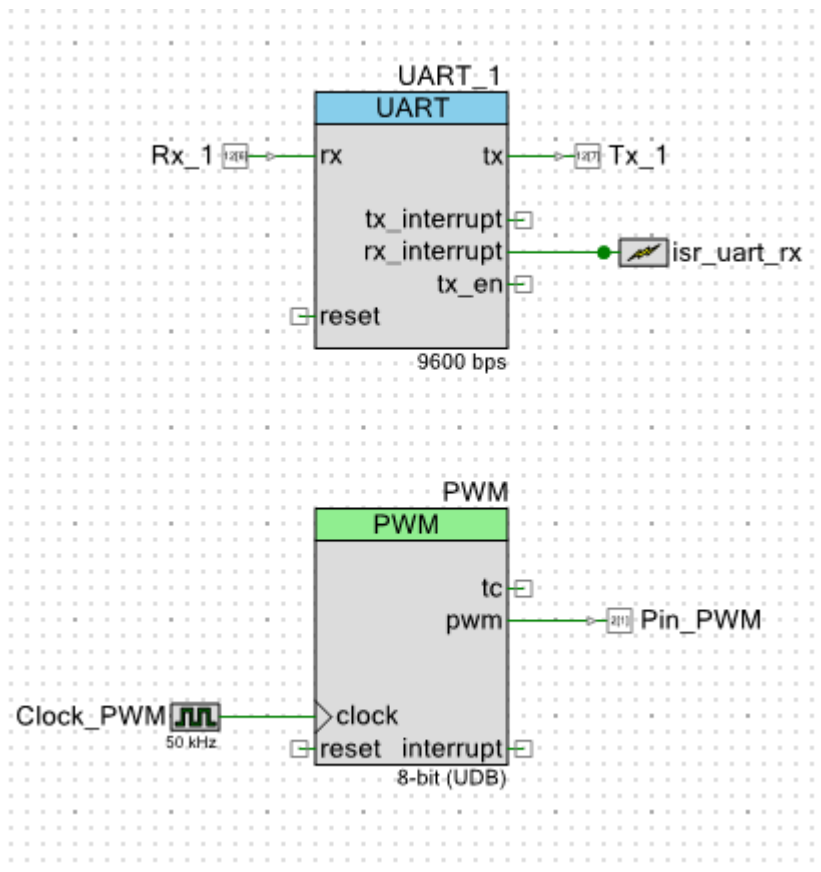
- PSoC
- DC Motor
- MOSFET IRLZ44

1.2.2 Kredsløbsopbygning

Pin	Pin
PSoC PWM (P2_1)	Q1 (Gate på IRLZ44)
GND PSoC	GND
Motor+	M1+
Motor-	M1



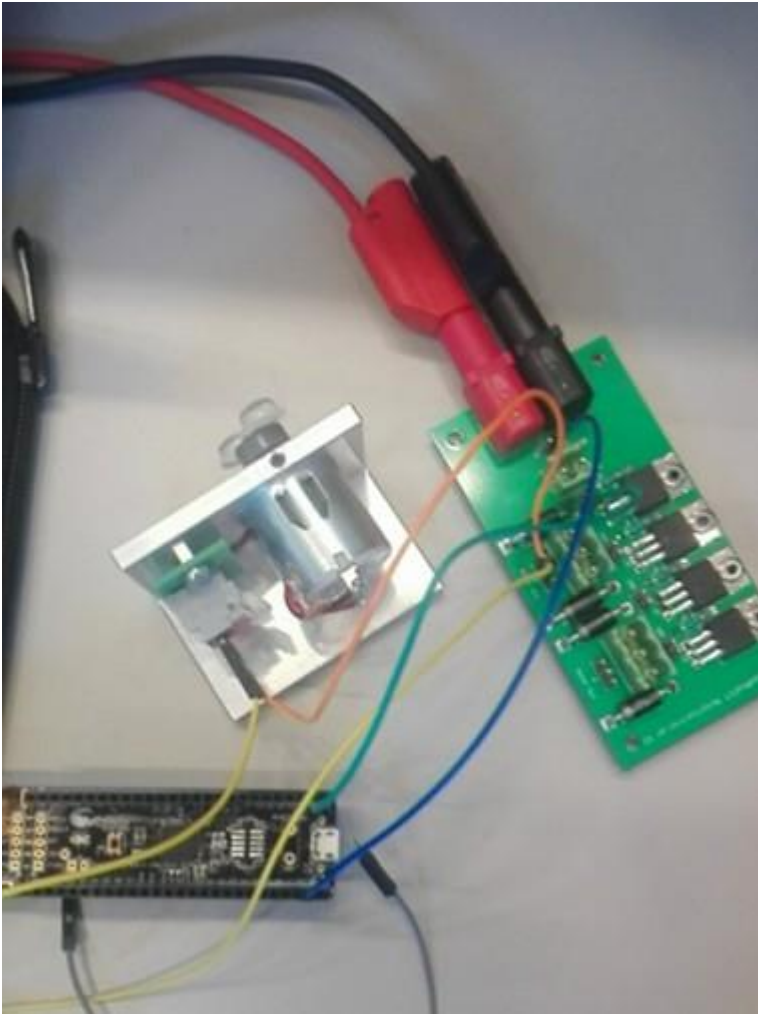
Figur 1 - Kredsløbsdiagram med forbindelser



Figur 2 - Top level design fra PSoC Creator

<input checked="" type="checkbox"/>	Pin_PWM	P2[1]	▼	63	▼	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Rx_1	P12[6]	▼	20	▼	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Tx_1	P12[7]	▼	21	▼	<input checked="" type="checkbox"/>

Figur 3 - I/O fra PSoC Creator



Figur 4 - Realisering af kredsløb

1.2.3 Kode

Koden fungerer således:

1. PSoC starter PWM signal ved 50% duty cycle. Ved compare match bliver PWM signalet skiftet
2. PSoC modtager en karakter fra UART og reagerer derfra via en ISR

Ved hastighedsændring

1. En global variabel *speed* bliver justeret med +/- 10 efter om hastighed skal falde eller stige.
2. Hvis speed er udenfor intervallet [0;100] forbliver den i dette interval.
3. *speed* bliver placeret i PWM komponentens compare register

Ved stopning:

1. Funktionen slukker for PWM signalet ved at sætte compare-værdien til 0

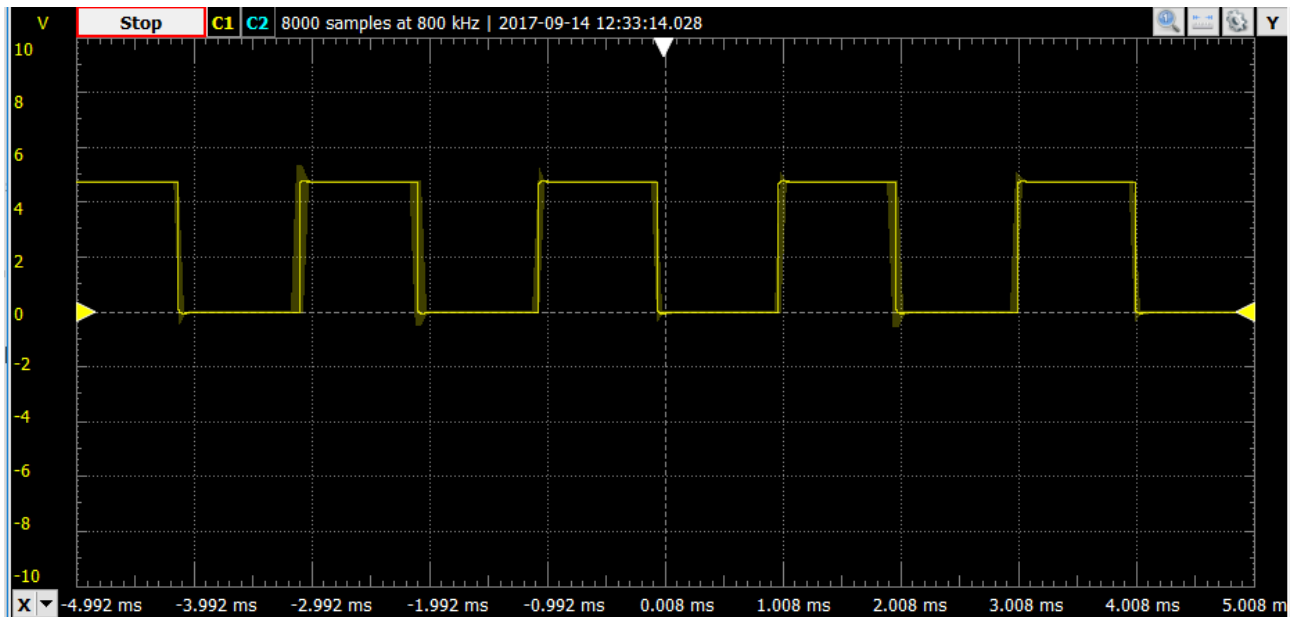
Ved start efter stop:

1. Retning er valgt efter stop
2. Compare-værdien sættes til 50 og PWM startes igen

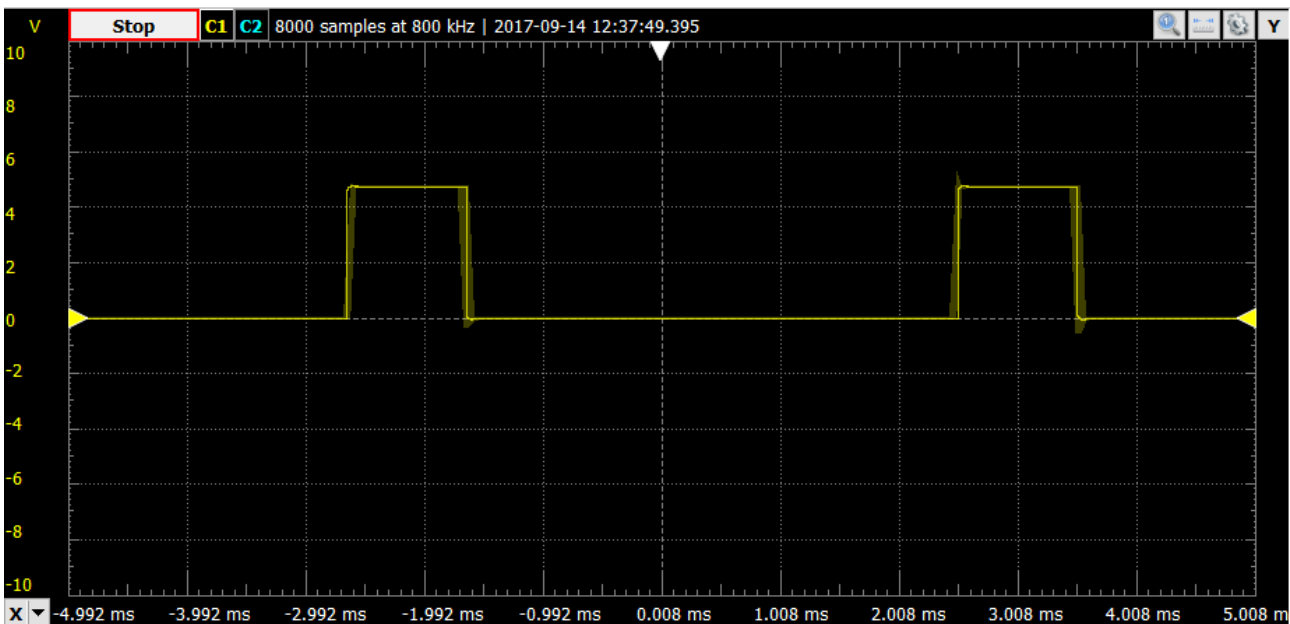
Koden er vedhæftet som bilag.

1.3 Resultater

Fejl! Henvisningskilde ikke fundet. og **Fejl! Henvisningskilde ikke fundet.** viser billeder af PWM-signal, når motoren ikke er sat til. Dvs. der er målt direkte på PWM-outputtet fra PSoC'en. Det ses, at PWM-signalet leveres som forventet. Programmet sættes op, således at inputs fra UART til PSoC'en kan hhv. stoppe, starte og ændre duty cycle på signalet. Dette lykkes også men kan selvfølgelig ikke dokumenteres.

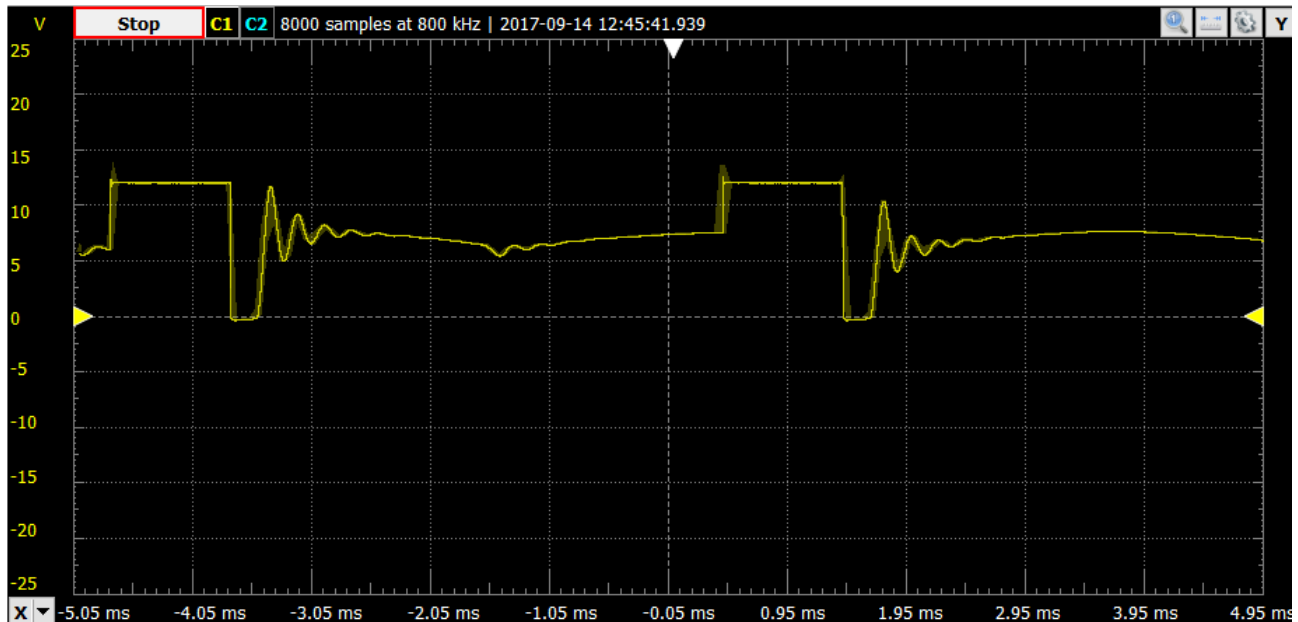


Figur 5 - Måling af PWM-signal med periodetid på 2.02 ms og duty cycle 50 %

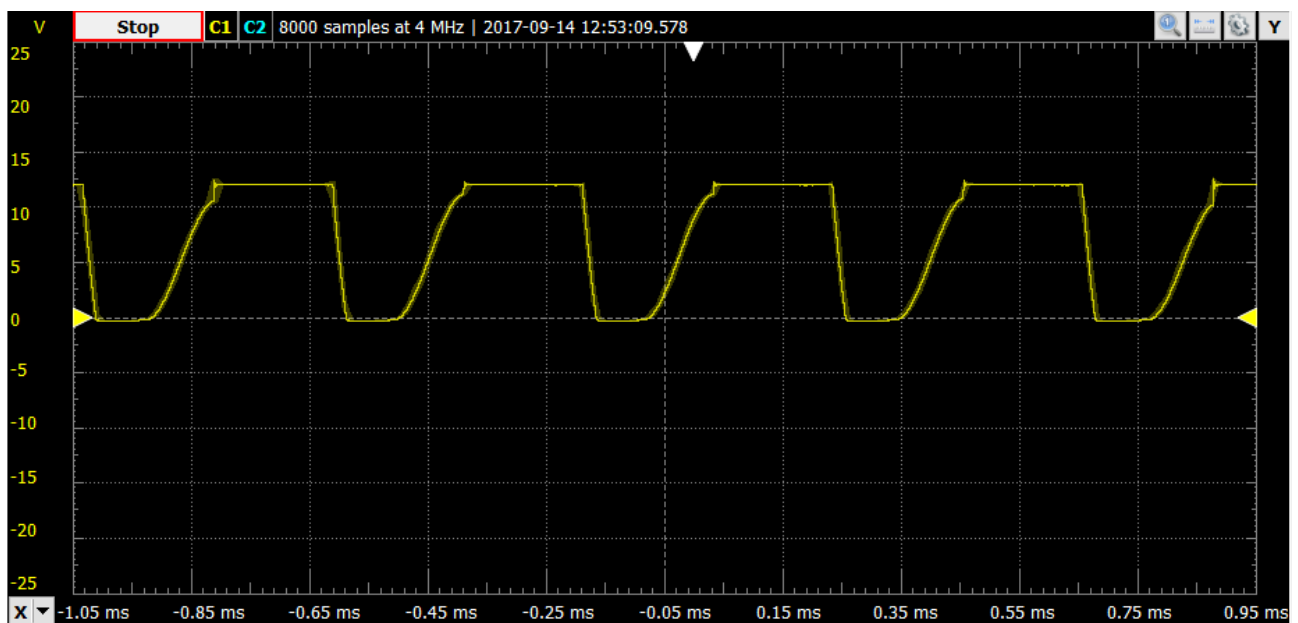


Figur 6 - Måling af PWM-signal med periodetid på 5.12 og duty cycle 20 %

Fejl! Henvisningskilde ikke fundet. og **Fejl! Henvisningskilde ikke fundet.** viser oscilloscopmålinger på PWM-signalet, når motoren er tilsluttet. På det første billede ses der, at når periodetiden er lang, forstyrres signalet af DC-motoren selv, der fungerer som spole. På det andet billede ses det, at når periodetiden sættes lidt ned, så kan spole-effekten fra motoren ikke nå at forstyrre PWM-signalet.



Figur 7 - PWM-måling med periodetid på 5.12 ms og duty cycle 50 %

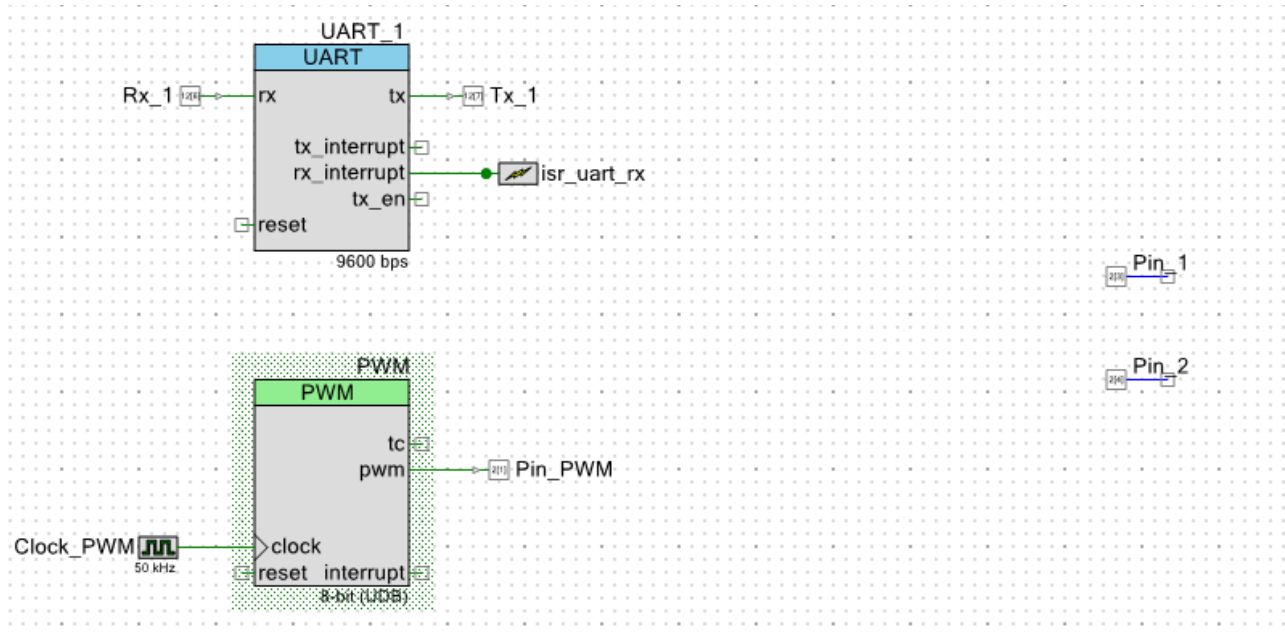


Figur 8 - PWM-måling med periodetid 420 mikrosekunder og 50 % duty cycle

1.4 Diskussion

Det lykkedes at lave noget kode til at styre en motor ved hjælp af et PWM signal. PSoC bliver med succes styret af en PC ved hjælp af en UART. Det er muligt at slukke PWM signalet og sætte gang i det igen ved kommando fra PC. Det er også muligt at ændre hastighed ved at ændre på PWM signalets compare værdi.

IN1 og IN2 vælger retning af motoren. Når IN1 er HIGH og IN2 er LOW vil motoren køre i en retning. Når IN1 er LOW og IN2 er HIGH vil Motoren køre den anden retning. ENABLE benet på H-broen tænder for motoren når benet er HIGH og slukker for motoren når benet er LOW. Bemærk at eneste forskel i PSoC-creator fra opgave 1 til denne opgave således er tilføjelsen af pins til IN1 og IN2, hvilket også fremgår af Figur 10.



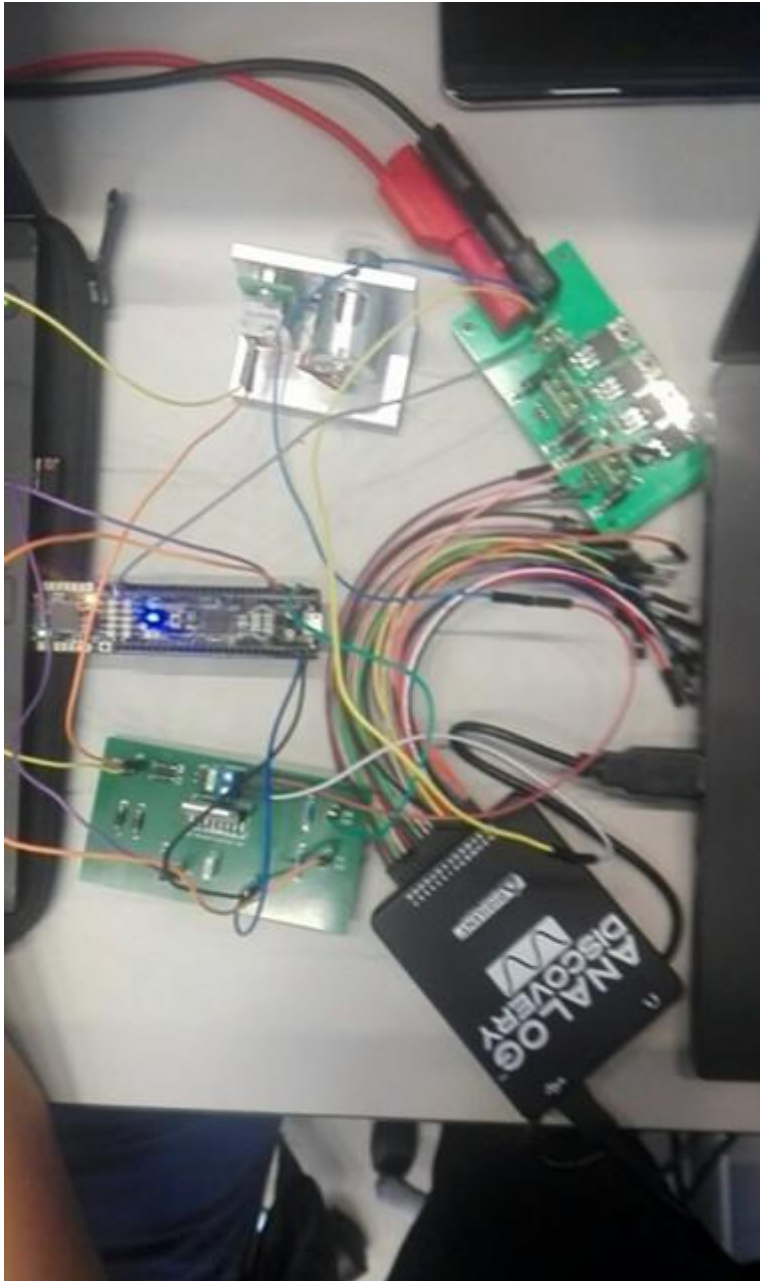
Figur 10 - Top level design for PSoC Creator

2.2.2 Kredsløbsopbygning

Til denne opsætning kobles DC-motoren til en H-bro af typen L298.

Pin	Pin
PSoC PWM (P2_1)	Q1 (Gate på IRLZ44)
GND PSoC	GND
Motor+	Motor+
Motor-	Motor-
<input checked="" type="checkbox"/> Pin_1	P2 [3] 65 <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Pin_2	P2 [4] 66 <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Pin_PWM	P2 [1] 63 <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Rx_1	P12 [6] 20 <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Tx_1	P12 [7] 21 <input checked="" type="checkbox"/>

Figur 11 - I/O view fra PSoC Creator



Figur 12 - Realisering af kredsløb

2.2.3 Kode

Koden fungerer således:

1. PSoC starter PWM signal ved 50% duty cycle. Ved compare match bliver PWM signalet skiftet
2. PSoC modtager en karakter fra UART og reagerer derfra via en ISR

Ved hastighedsændring

1. En global variabel *speed* bliver justeret med +/- 10 efter om hastighed skal falde eller stige.
2. Hvis speed er udenfor intervallet [0;100] forbliver den i dette interval.
3. *speed* bliver placeret i PWM komponentens compare register

Ved stop:

2. Funktionen slukker for PWM signalet ved at sætte compare-værdien til 0

Ved start efter stop:

1. Retning er valgt efter stop
2. Compare-værdien sættes til 50 og PWM startes igen

Ved retningsændring:

1. IN1 sættes til LOW og IN2 sættes til HIGH ved modsat kørsel
2. IN1 sættes til HIGH og IN2 sættes til LOW ved fremad kørsel

Koden er vedhæftet som bilag.

2.3 Resultater

Det lykkes at styre PWM-signalet, samt at ændre retning på motoren. Dette kan dog ikke dokumenteres vha. billeder.

2.4 Diskussion

DC-motoren blev styret af PWM-signalet. Vha. at styre strømmen i forskellige retninger gennem en H-bro, kunne DC-motorens retning styres. IN1 og IN2 bestemmer således, hvordan retningen skal gå, og softwaren sættes så til at styre, hvordan input på disse pins skal være.

3. Øvelse 3: Stepmotor styring

3.1 Indledning

Denne øvelse omhandler styring af en stepmotor. Som for DC-motoren i den tidligere op, skal stepmotoren kunne startes, stoppes, ændre hastighed og ændre retning via en UART-forbindelse til en PSoC.

3.2 Implementering

Wavedrive: Har 4 positioner hvor den kører imellem dem i grove træk. Kun 1 magnet tændt af gangen.

Full-step: Har 4 positioner men 2 magneter der er tændt ad gangen. Full step giver en mere kraftig bevægelse men stadig grove hak. Bruger dobbelt så meget strøm.

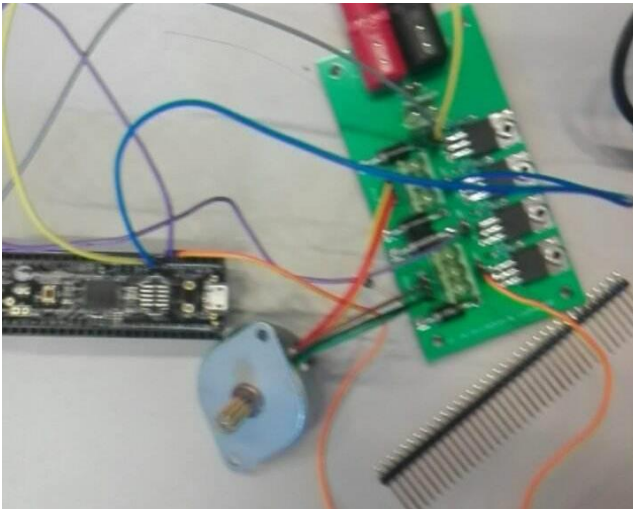
Half-step: Har 8 positioner, hvor den kører med pænere end nogen af de andre modes, da den har mange flere grader og mindre steps. Kører langsommere end full-step.

3.2.1 Komponenter

I denne opgave bruges følgende komponenter:

- PSoC
- Stepmotor PF35-48L4
- 4 MOSFET IRLZ44

3.2.2 Kredsløbsopbygning



Figur 13 - Opstilling af stepper motor samt MOSFET H-bro

3.2.3 Kode

Table 1. Excitation sequences for different drive modes

Phase	Wave Drive				Normal full step				Half-step drive							
	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8
A
B						
\bar{A}					
\bar{B}			

Figur 14 - Rækkefølge for steps

Generelt program:

1. Programmet initialiserer UART og Output pins
2. Programmet sidder fast i en uendelig løkke
3. I løkken er kode til at steppe motoren og et delay

Ændring af hastighed på motor:

1. Delay reduceres eller øges efter om motoren henholdsvis skal køre hurtigere og langsommere

Ændring af mode

1. Ved karakterinput fra UART ændres mode mellem FULLSTEP, HALFSTEP, WAVE, single rotation, single clockwise rotation og STOP

Ændring af retning

1. Ved ændring af retning køres sekvens for rotation i modsat retning

3.3 Resultater

Opnåede resultater stemmer overens med forventede resultater, derudover gav øvelsen en god forståelse for hvornår en stepper motor skal bruges samt stærke og svage sider af en stepper motor. Vi fik motoren til at udføre de forskellige steps fra øvelsesvejledningen, med få problemer.

3.4 Diskussion

Det lykkedes at lave software til et kredsløb, der indeholder en stepmotor. Stepmotoren kan køre i grader af præcision, således at der kunne vælge meget fin kørsel og lidt grovere kørsel. Det lykkedes ligeledes at få den til at ændre retning.

Der er forskellige fordele og ulemper ved at bruge hhv. en stepmotor og en DC, som man kan overveje, når man skal vælge motor. Mens stepmotoren bruger konstant strøm, så afhænger DC-motorens strømforbrug af hastigheden. En stepmotor er god, hvis man har brug for en langsom og præcis motor, mens en DC-motor kan åbne højere hastigheder. Stepmotoren er ligeledes meget fleksibel til at ændre retning, hvorimod en DC-motor først skal bremse, før den kan ændre retning. DC motoren er også meget mere simpel end en steppe motor.

Vi havde en del problemer med hvilke 4 pins der skulle forbindes til MOSFET'en, da de udleverede billede til steps (kan ses på figur 14) ikke stemmede overens med de reelle pins til vores MOSFET.