

Professional Drone, Hybrid Power Pack - Timebox 4

Team 2
Aarhus Universitet, Herning.

13. februar 2019

Stud. nr: 201602094	Navn: Søren Holm Korsgaard
Stud.nr.: 201607563	Navn: Jacob Gustafsson
Stud.nr.: 201704859	Navn: Jonas Buus
Stud.nr.: 20084327	Navn: Simon Rasmussen
Stud.nr.: 201704483	Navn: Thomas Dueholm Jensen

Indhold

1 Strategy and planning (Thomas)	2
2 Ensretter (Jonas)	2
2.1 Analyse	3
2.2 Simulering	4
2.3 Design og implementering	4
3 Spændingsregulator (Jacob)	5
3.1 Analyse	5
3.1.1 Structural Analysis	5
3.1.2 General Requirements	5
3.1.3 Behavioural Analysis	5
3.1.4 Interface Analysis and Design	5
3.1.5 Dimensioning	5
3.2 Designfasen	6
3.2.1 Structural Design	6
3.2.2 Behavioural Design	6
3.3 Implementering	8
4 Driver (servomotor)(Søren)	9
4.1 Analyse/design	9
5 Motorstyring (Simon)	12
5.1 Analyse	12
6 Mangler	13

1 Strategy and planning (Thomas)

Denne timebox omhandler videre teoretisk analyse af forskellige subsystemer til det endelige System-To-Be, Dronecopteren. Jonas og Thomas har efter planen arbejdet videre med analyse af et muligt kredsløb til at realisere den aktive ensretter. Jacob har indledningsvist lavet et samlet strømbudget for System-To-Be for at skabe et bedre overblik. Derefter har han arbejdet videre med analyse/simulering af to spændingsregulator kredsløb, hvor hovedformålet har været at nedbringe den omsatte effekt væsentligt fra det tidligere forslag i timebox 3 for at undgå tab i regulatoren. Søren har designet og realiseret en driver til servomotoren, der skal styre gasspjældet på brushless DLE motor. Simon har arbejdet med en samlet analyse af softwareudviklingen til system controlleren – PID regulering af motorstyringen mm.

Timebox 4 er inddelt i afsnit til hver af de ovenstående emner, og afsnittene er beskrevet ud fra formen til Realisation Phase i EUDP. Rapporten er sammensat af Thomas, og formateret til LaTeX af Simon/Jonas.

2 Ensretter (Jonas)

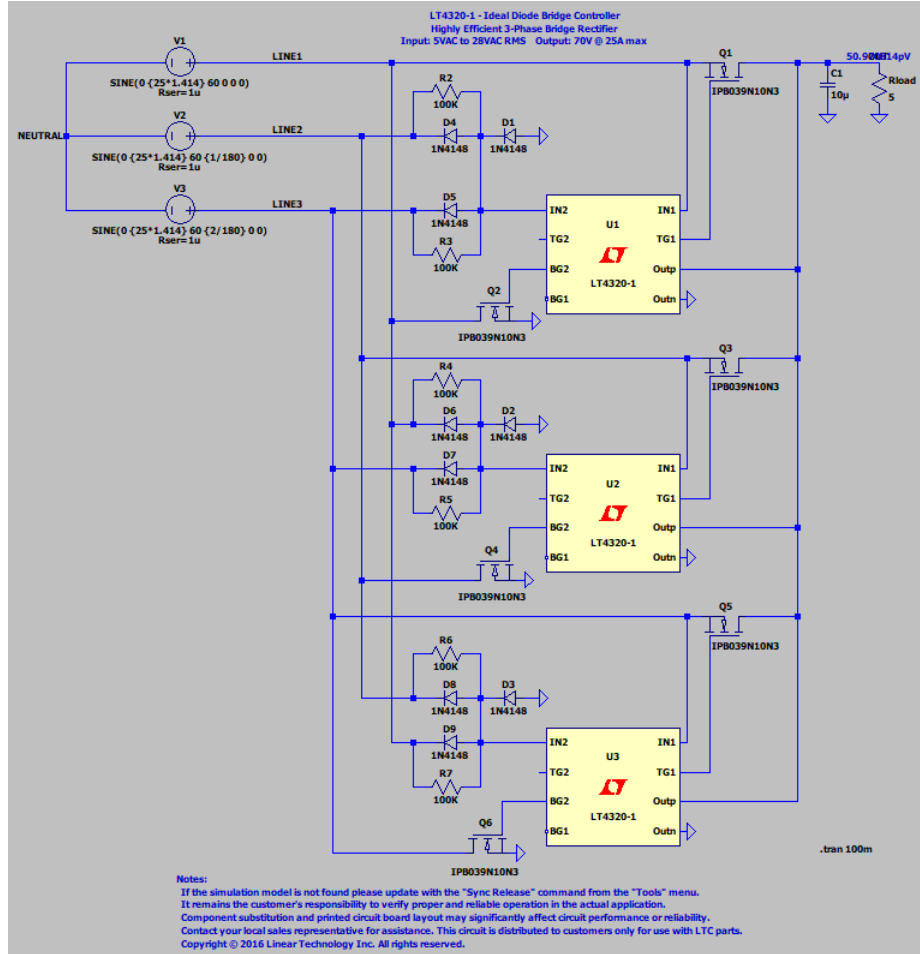
Med udgangspunkt i, at ensretteren skal tage imod tre faser, hver på ca. 36 A @ op til ca. 22 V (RMS-værdier¹), har vi valgt at gå videre med LT-4320-1 ideal diode controller IC'en jf. tidligere argumentation i Timebox 2. Inden vi vælger at bestille komponenter, mangler vi at udføre en sidste test af forbrændingsmotor/generator-subsystemet. Mere om dette i afsnittet "Design og implementering".

Vi ønsker at designe et test-kredsløb som kan implementeres, og testes med meget lavere strømforhold end i det endelige kredsløb, for at demonstrere funktionaliteten først. Første opgave har været at finde frem til et bud på hvordan ensretteren kan designes. Denne timebox (ift. ensretter) omhandler netop dette.

¹RMS-værdier for spændingsniveauer er afhængige af forbrændingsmotorens RPM.

2.1 Analyse

Leverandøren af LT-4320-1, foreslår kredsløbet set på figur 1², som en yderst energieffektivt 3-faset aktiv ensretter.



Figur 1: Diagram: Aktiv 3-faset ensretter, konstrueret med 3 x LM4320-1 IC'er.

Kredsløbet anvender 1 x LT4320-1, samt 2 N-channel MOSFET pr. fase. Primære kilde til regulering af ripple er C1 (øverste, højre hjørne). I databladet³ for IC'en angiver producenten ligning 1, som en vejledning i at bestemme størrelsen på load kondensatoren.

Fra denne isoleres ripplespænding, for at få et indtryk af kondensatorens betydning for ripple på output (Se ligning 2). Vi forventer derfor at kunne reducere ripple-niveauer, ved at øge størrelse på load kondensatoren.

$$C_{Load} \geq \frac{I_{AVG}}{V_{ripple} * 2 * f} \quad (1)$$

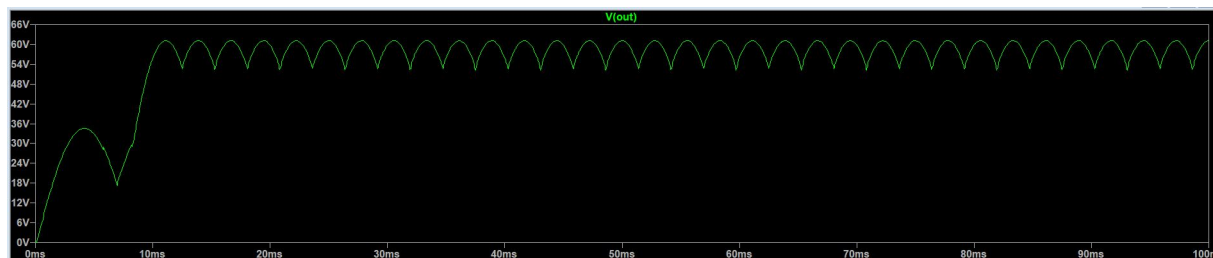
$$V_{ripple} = \frac{I_{AVG}}{C_{Load} * 2 * f} \quad (2)$$

²Kilde: <https://www.analog.com/en/products/lt4320.html#product-documentation>

³Kilde: <https://www.analog.com/media/en/technical-documentation/data-sheets/4320fb.pdf>

2.2 Simulering

Kredsløbet er simuleret i simulationsprogrammet LTspice XVII, producentens (af kredsløbsdiagrammet, figur 1) værdier er anvendt i simulering for at demonstrere funktionaliteten. Som det kan ses, på figur 2 nedenfor, leverer kredsløbet ved input på $3 \times 25V_{AC_{RMS}}@60Hz$, et DC-output på ca $60V_P$. $R_{Load} = 5\Omega$, $C_{Load} = 10\mu F$. Der bemærkes et betragteligt niveau af ripplespænding.



Figur 2: Simulering: Udgangsspænding på ensretterkredsløbet. Simulering er foretaget i LTspice XVII. Udgangsspænding er målt før R_{load} med reference til jord.

2.3 Design og implementering

Formålet er som nævnt, at demonstrere funktionaliteten af LT4320-1. Vi vil bygge en kraftigt nedskalaret udgave af det kredsløb, som vi forestiller os kommer til at være det endelige design. Vi har endnu ikke fastlagt en grænseværdi for ripplespænding på udgangen af ensretteren, og kan ikke opstille et krav som relaterer til ripplespænding. Dog håber vi med denne test, at kunne demonstrere en kontrol over denne ripplespænding. Som udgangspunkt vil vi forsøge med forskellige størrelser for load kondensatoren.

Krav til test-kredsløb:

1. Indgangsspænding skal være på 9-72V jf. datablad for LT4320-1.
2. Frekvensen af test-signalet skal være 500 Hz, for at ligge i nærheden af den frekvens vi forventer at generatoren leverer.

Vi har desuden besluttet os for at teste spændingsniveauerne på indgangssignalerne til ensretteren ift. motorens RPM. Før vi bestiller komponenter, vil vi have bekræftet, at ensretteren ikke modtager signaler under 9V, i det som kommer til at omfatte forbrændingsmotor/generator-subsystemets driftsområde, eller at vi på anden vis kan sikre, at ensretteren ikke kommer til at modtage signaler under 9V. Såfremt dette bekræftes, bestilles komponenter, analysen færdiggøres, og konstruktion af testkredsløb forventes påbegyndt i Timebox 5.

3 Spændingsregulator (Jacob)

Vi har behov for en spændingsregulator i vores system, da batteripakken leverer 22 V, mens vi skal bruge 5 V til styringerne og vores FRDMKL25Z, som vores board. I det nedenstående følger en analyse af spændingsregulatoren. Spændingsregulatoren indgik også i Timebox 3. Sidenhen er vi dog blevet opmærksomme på væsentlige effekttab på den tidligere omtalte løsning. I denne timebox vil der udelukkende være design af 2 forskellige kredsløb, hvorefter kredsløbene testes og bygges i næste timebox.

3.1 Analyse

3.1.1 Structural Analysis

Som nævnt herover, får vi behov for en spændingsregulator, som en del af systemet, for at kunne levere den nødvendige forsyningsspænding til de logiske kredsløb. Herunder listes kravene til spændingsregulatoren.

3.1.2 General Requirements

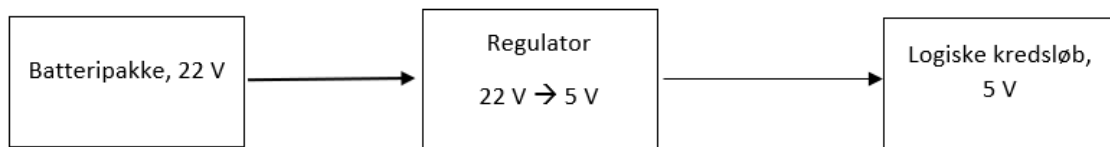
1. Spændingsregulatoren skal kunne regulere en spænding fra 22 V ned til 5 V, med en pålydende strøm af 1 A.
2. Spændingsregulatoren skal være vejrbestandig.

3.1.3 Behavioural Analysis

Den ønskede opførsel af komponenten er, at den muliggør en regulering i spænding fra 22 V til 5 V.

3.1.4 Interface Analysis and Design

Spændingsregulatoren vil være koblet til batteripakken i den ene ende og de logiske kredsløb i den anden ende. Som både indgang og udgang til spændingsregulatoren, vil der være forbundet almindelige ledere - 1 kvadrat som indgang og 0,5 kvadrat som udgang. Se figur 3.



Figur 3: Viser strømmens retning i de subsystemer, som relaterer til spændingsregulatoren.

3.1.5 Dimensioning

Da spændingsregulatoren er en blivende del af systemet, er vi afhængige af, at holdbarheden er god, og som minimum lever op til kvaliteten af de resterende dele af produktet. Ét af de problemer vi kan støde på under spændingsregulering af temperaturforøgelse, der kan brænde komponenten af.

Bruges der blot en LM7805 til at regulere fra 22 V til 5 V, med en strøm på 1 A, vil der være et effekttab på 17 W.⁴

3.2 Designfasen

Nedenstående gennemgår de valg der er truffet omkring design af spændingsregulatoren på baggrund af analysen.

3.2.1 Structural Design

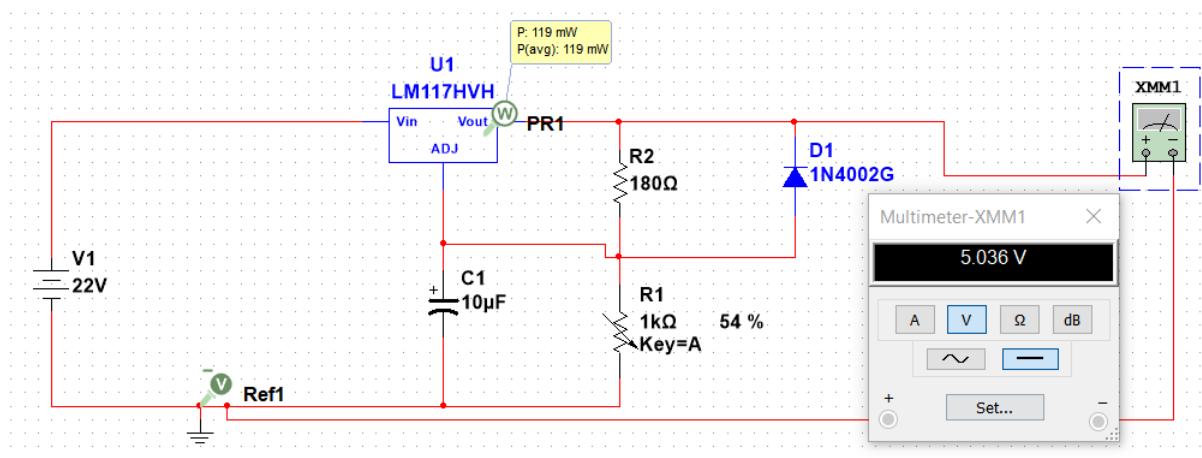
Spændingsregulatoren skal være vejrbestandig, og sikres efter IP66 standard. Problemet med dette er, at regulatoren udvikler en hel del varme. Derfor skal spændingsregulatoren være i en kasse for sig, hvor heatsink monteres udvendigt på kassen.

3.2.2 Behavioural Design

I Timebox 3 introduceres en spændingsregulator der bygger på 2 IC'ere fra LM78xx familien. Vi havde i første gang valgt denne løsning, da disse IC'ere er billige og driftsikre, men efter nyere overvejelser i forhold til effekttab, er løsningen ikke effektiv. Derfor introduceres nu 2 nye muligheder for spændingsregulatordesign.

LM117

Herunder ses en simulering af opsætningen af en spændingsregulator på baggrund af en LM117. Opstillingen er lavet på baggrund af 1 af standardopstillingerne i datasheetet for LM117/LM317T.⁵



Figur 4: Simulering: Spændingsregulering vha. LM117.

Som det ses på figur 4, opnås den ønskede spænding, og den effekt der afsættes i LM117 er meget acceptabel. Dette effekttab vil give en temperaturstigning i LM117 på 22,1 grader, hvilket følger af de temperaturkarakteristika der gælder for LM117 iflg. dataarket.

Ulemperne ved LM117 er, at vi ikke har mulighed for at styre strømmen på outputtet. LM117 håndterer op til 1,5 A på outputtet, men vi ønsker kun 1 A. Derfor kan det være nødvendigt med et current sink kredsløb efterfølgende. Endvidere er LM117 IC'en cirka 10 gange dyrere end en

⁴Effekt: $P = V \cdot A$

⁵TI, 2018

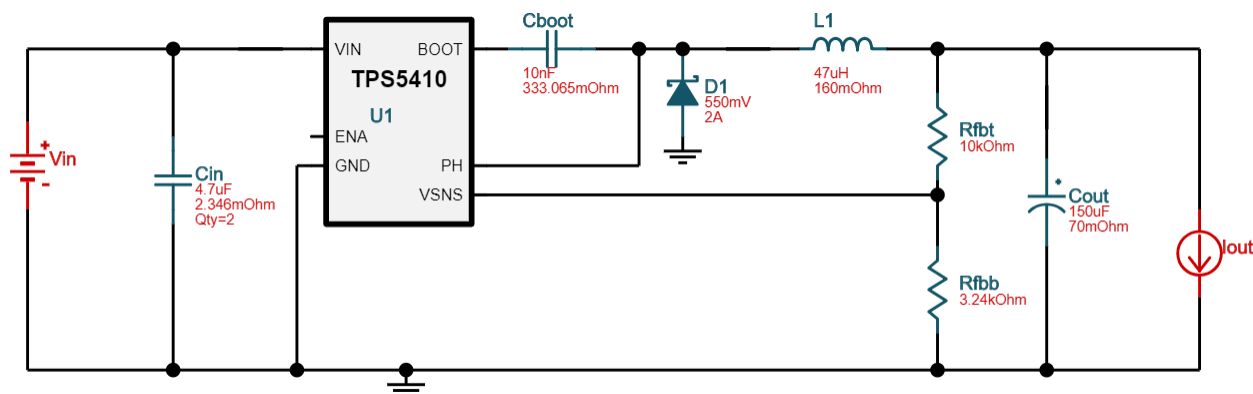
opsætning med to stk. LM78xx IC'ere, idet ét styk LM117 koster omkring 100 kr.

BUCK converter spændingsregulator med TPS5410

En anden mulighed for konvertering af spændingen er med en BUCK converter. Opstillingen herunder kommer fra et designværktøj udviklet af Texas Instruments, kaldet WEBENCH⁶. I nævnte værktøj indtastes blot den ønskede input og outputstrøm og spænding, hvorefter der tegnes forskellige mulige kredsløb, som man kan vælge imellem. På baggrund af effektivitet og kostpris, valgtes følgende (Se figur 5).

Værktøjet indeholder også et analyseværktøj. For at bekræfte, at kredsløbet giver den ønskede outputspænding, har jeg kigget på karakteristikken i steady state. Se figur 6.

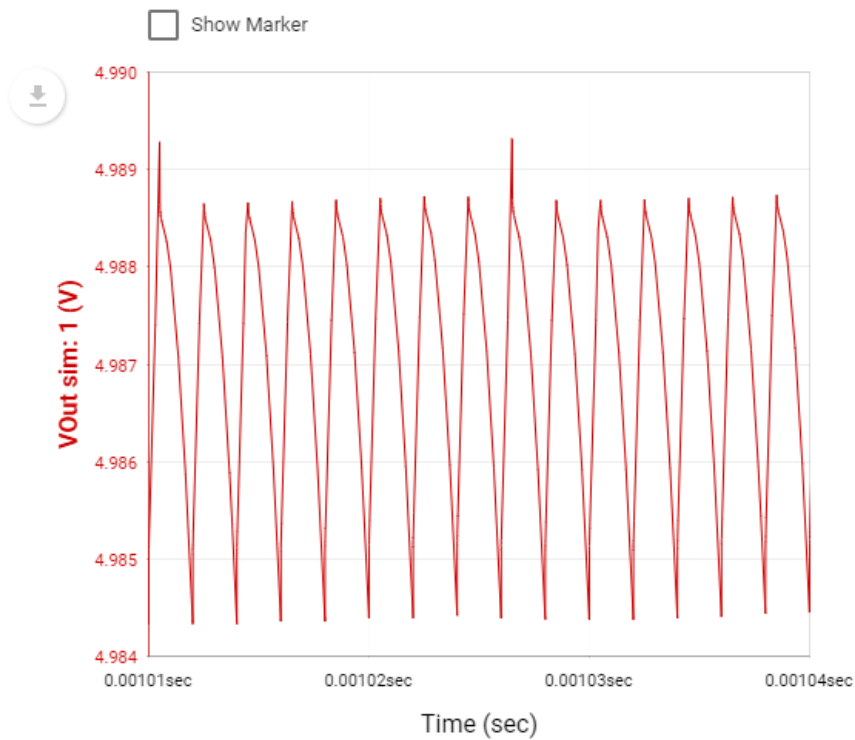
Som det ses, opnås den ønskede spænding. Billedet viser en del ripple, men denne er underordnet, så længe den ikke har større udsving end det vises her. Effektiviteten for ovenstående konverter er på 86,1 %, og effekttabet i IC'en udregnes af programmet til 208 mW. Dette giver, i forhold til de i dataarket⁷ oplyste temperaturkarakteristika, en temperaturstigning på 22,02 grader, hvilket er acceptabelt, også uden heatsink. Endvidere er denne løsning væsentlig billigere da ét styk TPS5410DR kan erhverves for cirka 20 kr.



Figur 5: Diagram: BUCK converter vha. TPS5410.

⁶Instruments, 2019

⁷Texas Instruments, 2019.



Performance Summary

Sim ID	VOut Peak-Peak	VOut Average	Inductor Current Peak-Peak	Average Frequency
1	5.15 mV	4.99 V	0.18 A	499.92 kHz

Figur 6: Performance analyse af spændingsregulator, TPS5410.

3.3 Implementering

Systemerne implementeres og testet begge indenfor de næste 4 uger. Det konkluderes, at vi var for hurtige da vi besluttede os for en spændingsregulator baseret på LM78xx familien. Videre konklusion afventer til de 2 nye systemer er testet på board.

4 Driver (servomotor)(Søren)

4.1 Analyse/design



```
86     01 LPTPM counter increments on even LPTPM counter clock = (1UL << 3)
87     */
88     TPM0->SC |= (1UL << 3);
89
```

Figur 7: Udsnit af kode til motordriver.

Fremgår i timebox 3, hvor vi har dokumenteret, at det kræver et signal på 50 Hz og ved at justere PWM mellem 1 ms til 2 ms. Kan vi justere positionen på servo motor.

Dette signal bruges også til at justere hastigheden på vores BLDC-motor, via vores ESC. Vi viste i timebox 3, hvordan vi brugte signalet til. De værdier vi brugte, var 1 ms til stop af BLDC-motor og værdien 1.25 ms, for at generer 500 rpm +- 100 rpm.

Ud fra dette er følgende krav opstillet:

- 50  signal
- 20 ms periode
-  l kunne justere puls mellem 1 – 2 ms.

FRDM-KL25Z Pins				KL25Z128 Pins									
On-board Usage	IO Header & Pin Num	Arduino™ R3 Pin Name	FRDM-KL25Z Pin Name	KL25Z Pin #	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	Reset State/Function
Blue LED	J2 12	D13	PTD1	74	ADCR_SE5b	PTD1	SPB_SCK	FTM0_CH1					ADCR_SE5b
–	J2 08	D11	PTD2	75		PTD2	SPB_MOSI	UART2_RX	FTM0_CH2	SPB_MISO			DISABLED
–	J2 10	D12	PTD3	76		PTD3	SPB_MISO	UART2_TX	FTM0_CH3	SPB_MOSI			DISABLED

Figur 8: Tabel over kanaler, pins mv. for TPM0 modulet.

På figur 8 vises hvilke kanaler på TPM0 modulet vi bruger samt pins og mode i PCR_MUX.

Ved at læse kommentarer til koden på figur 9, vises konfigurationen, og beregningen for værdier til MOD og til CNV, for at lave de 50 Hz med 1 ms.

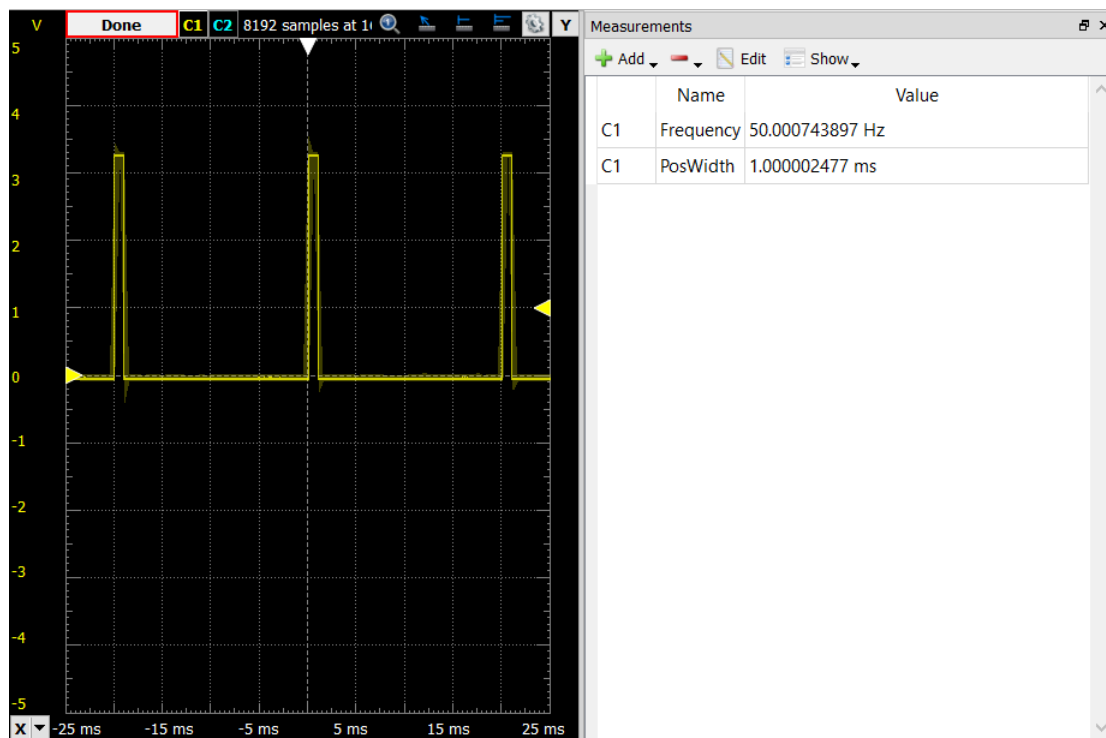
Ved at slutte Analog Discovery's oscilloskop til pin PTD3, vises at vores init funktion virker efter hensigten, med 50 Hz og 1 ms puls. Se figur 10

```

13 #define PERIOD 60000
14 #define MIN_VAL_SERVO 3000 // 1 ms to start servo
15 #define MAX_VALUE_SERVO 6000 // 2 ms to turn max on servo
16 #define START 3750 // Set duty cycle to 1.25 ms for 500 cranking rpm
17
18 #define START_BTN (1) // PIN PTD1 Start button
19 #define SERVO_THROTTLE (2) // PIN PTD2 channel TPM0
20 #define ESC_SIGNAL (3) // PIN PTD3 channel TPM0
21
22 #define PWM_CH_SERVO (2) // PTD2 PWM Servo channel on TPM0
23 #define PWM_CH_ESC (3) // PTD3 PWM ESC channel on TPM0
31 void init_pwm() {
32     /*
33      * Vi ønsker et 50 Hz signal. og vores clock er 48 Mhz.
34      * 48Mhz/50Hz = 960000, da vores counter er en 16 bit, dvs max decimal 65536.
35      * vores prescaler værdi er = 960000/65536 = 14,648, så vi sætter den til 16.
36      * vores count værdi bliver (960000/16)-1 = 59999 for at få en 50Hz frekvens
37      */
38     SIM->SCGC5 |= SIM_SCGC5_PORTD_MASK;
39     SIM->SCGC6 |= SIM_SCGC6_TPM0_MASK;
40
41     /*
42      * enable TPM in mux (4UL << 8);
43      */
44
45     PORTD->PCR[SERVO_THROTTLE] |= (4UL << 8);
46     PORTD->PCR[ESC_SIGNAL] |= (4UL << 8);
47     /*Selects the clock source for the TPM counter clock:
48      * 01 MCGFLLCLK clock or MCGPLLCLK/2 = (1UL << 24)
49      * Selects the MCGPLLCLK or MCGFLLCLK clock for various peripheral clocking options:
50      * 1 MCGPLLCLK clock with fixed divide by two = (1UL << 16)
51      */
52
53     SIM->SOPT2 |= ((1UL << 24) | (1UL << 16));
54
55     // Set count value in reach for 2^16 = 65536
56     TPM0->MOD = PERIOD - 1;
57
58     /* Prescale Factor Selection:
59     * 100 Divide by 16 = (1UL << 0)
60     */
61     TPM0->SC = (4UL << 0);
62     /*
63     * Debug Mode:
64     * 11 LPTPM counter continues in debug mode = (3UL << 6)
65     */
66     TPM0->CONF |= (3UL << 6);
67     /*
68     * Channel Mode Select: active HIGH
69     * ELSB = (1UL << 5)
70     * MSB = (1UL << )
71     * check side number 555 in manual for channel options
72     */
73     TPM0->CONTROLS[PWM_CH_SERVO].CnSC = ((1UL << 3) | (1UL << 5));
74     TPM0->CONTROLS[PWM_CH_ESC].CnSC = ((1UL << 3) | (1UL << 5));
75     /*
76     * set duty cycle to 0
77     */
78     TPM0->CONTROLS[PWM_CH_SERVO].CnV = MIN_VAL_SERVO;
79     TPM0->CONTROLS[PWM_CH_ESC].CnV = MIN_VAL_SERVO;
80     /*
81     * Clock Mode Selection
82     * Selects the LPTPM counter clock modes. When disabling the counter, this field remain set until
83     * acknowledged in the LPTPM clock domain.
84     */
85

```

Figur 9: Udsnit af kode vedr. init funktion.



Figur 10: Måling med oscilloskop på pin PTD3; spændingsamplitude, som funktion af tid.

```

91 void angle_throttle(unsigned int procent) {
92
93     if (procent > 0 && procent < 100) {
94         int res = (MIN_VAL_SERVO * procent) / 100;
95         TPM0->CONTROLS[PWM_CH_SERVO].CnV = MIN_VAL_SERVO + res;
96     }
97
98 }

```

Figur 11: Kode til styring af servomotor

Herefter har vi implementeret en funktion til at styre vores servo motor, den tager et argument, som er i procent, hvor meget den skal dreje, dette bruges til at eftervise at servo motor kan justeres ved at ændre pulsen mellem 1 – 2 ms.

Hermed er vist, at de opsatte krav er opfyldt.



Figur 12: Servomotor tilsluttet KL25Z

5 Motorstyring (Simon)

5.1 Analyse

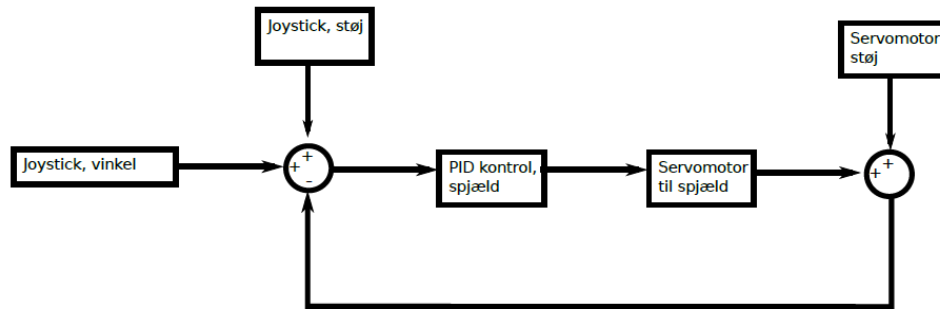
Motorstyringen vil tage udgangspunkt i vinkel af joystick som indikerer hastigheden. Dette vil tage udgangspunkt i et.

- proportionalt forhold: At den aktuelle forskel mellem den ønskede vinkel af spjældet og den aktuelle vinkel tilrettes.
- integralt forhold: At de tidligere fejl mellem ønsket position af spjæld og reelle kan korrigeres på baggrund af kendskabet til fejl.
- deriveret forhold: At pludselige ændringer i spjældets position kan afhjælpes.

Teoretisk er dette udtrykt ved:

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \frac{de(t)}{dt} \quad (3)$$

hvor $e(t)$ er fejl som funktion af tiden, $e(\tau)$ er fejl som funktion af den akkumulerede tid og k_P , k_I og k_D er konstanter svarende til henholdsvis det proportionale, integrede og deriverede forhold.



Figur 13: Motorstyring, blokdiagram

Et karakteristisk program flow kunne være:

```
while(1){
    error = theoretical_value - actual_value;
    integral = integral + (error*iteration_time);
    derivative = (error - error_prior)/iteration_time;
    output = KP*error + KI*integral + KD*derivative;
    error_prior = error;
    sleep(iteration_time);
}
```

6 Mangler

- Der mangler at finde præcision af konstanterne k_P , k_I og k_D .
- Det er muligt at jeg vil lave et simuleringsmiljø idet konstanterne kan “tunes” på plads. Under tuning verificerer man om systemet er stabilt.
- Jeg vil gerne undersøge muligheden for at benytte bl.a. interrupts.
- Der mangler udformning af diagram for program flow.
- Til næste timebox vil jeg gerne om analysetrinnet kan sættes på hos mig igen idet jeg gerne vil kunne læse mig endnu mere ind på teorien og udarbejde et ordentligt teoretisk grundlag for simuleringstrin.