

TESTSTAND TIL BØRSTELØS DC MOTOR

BACHELORPROJEKT

PROJEKTRAPPORT

IBRAHIM ÖZTOPRAK — 201270314

SIMON L. MADSEN — 201081227

Titel:

Teststand til Børsteløs DC Motor

Projekt nummer:

15162

Projekt:

EP Bachelorprojekt, 7. semester

Projektperiode:

August 2015 - December 2015

Gruppemedlemmer:

Ibrahim Öztoprak - 201270314

Simon L. Madsen - 201081227

Vejleder:

Emir Pasic

Antal sider: 34

Resumé

Resumé

Denne rapport præsenterer bachelorprojektet, der er udarbejdet i samarbejde med Ingeniørhøjskolen Aarhus Universitet i anledning af Shell Eco-Marathon. Projektet omhandler design af en teststand til ASEs udviklede elbil. Rapportens formål er at give en forståelse for problematikken bag design af en teststand samt at præsentere en løsning for denne problematik. Løsningen indebærer udvikling af fire delsystemer. De fire delsystemer er betegnet *Belastningskreds*, *Brugergrænseflade*, *Kontrolenhed* og *Sensorer*. Disse delsystemer kan i samarbejde med hinanden belaste elbilens drivlinje, mens der foretages målinger, som gemmes til brug ved senere analyse.

Produktet er udviklet med en PSoC5 som den kontrollerende enhed til måling og beregning af data. Ydermere anvendes kontrolenheden til styring af strømmen gennem *Belastningskredsen*. Interaktionen med systemet sker gennem en *Brugergrænseflade*, der er implementeret på en computer. Baseret på målinger fra *Sensorer* vil brugeren kunne tilgå data gennem *Brugergrænsefladen*. Herigennem er det muligt at styre tests. Kommunikationen mellem PSoC5 og *Brugergrænsefladen* er realiseret gennem en TCP forbindelse.

Resultatet blev en prototype med hardware til måling og elektrisk belastning, da den generiske hardwareplatform forblev på et konceptplan. Softwaren til *Kontrolenheden* er i stand til at håndtere dataopsamling, regulering og kommunikation. Kommunikationen mellem *Brugergrænsefladen* og *Kontrolenheden* virkede ikke ved projektets afslutning, men en terminal emulator er anvendt til simulering af *Brugergrænsefladen*.

Abstract

This report presents the bachelor project that is developed in cooperation with Aarhus School of Engineering on the occasion of the Shell Eco-Marathon. The project deals with design of a test bench for ASE's developed electric vehicle. The purpose of the report is to provide an understanding of the design of a test bench, together with a solution to this issue. The solution includes a development of four subsystems. The four subsystems consist of the following: *Load Circuit*, *User Interface* and *Control Unit* and *Sensors*. Together, these subsystems can perform loading of the drive train of the electric car, while measurements are saved for use in later analysis.

The product is developed with a PSoC5 as the controlling unit for measurement and calculation of data. Furthermore, the *Control Unit* is used to control the current flow through the *Load Circuit*. The interaction with the system is done through a user interface that is implemented on a computer. Based on measurements from the *Sensors*, the user will be able to access data through the *User Interface*. Through this it is possible to control the tests. The communication between the *Control Unit* and the *User Interface* is realized through a TCP connection.

The resulting prototype includes hardware for measuring and electrical loading. The generic hardware is kept at a concept level. The software for the *Control Unit* is able to collect data, perform regulation and communicate with the *User Interface*. The communication between the *User Interface* and the *Control Unit* did not work at the end of the project, but a terminal emulator was used to simulate the *User Interface*.

Arbejdsfordeling

201270314
Ibrahim Öztoprak

Hovedansvarlig for udviklingen af *Kontrolenhed* med fokus på design & implementering, herunder implementering af softwaren. Hovedansvarlig for udvikling & implementering af *Kommunikation*. Medansvarlig i udvikling og implementering af *Brugergrænsefladen*.



16.12.2015

Dato

Underskrift

201081227
Simon L. Madsen

Hovedansvarlig for udvikling af *Belastningskreds* og *Sensorer*, med fokus på design & implementering. Medansvarlig i udvikling og implementering af *Brugergrænsefladen*. Medvirkende til udvikling af *Belastningsstyringen* på *Kontrolenhed*.

A handwritten signature.

16.12.2015

Dato

Underskrift

Indholdsfortegnelse

Arbejdsfordeling	iii
Resumé	iii
Kapitel 1 Indledning	1
1.1 Opgaveformulering	1
1.2 Systembeskrivelse	2
1.3 Projektafgrænsning	3
1.3.1 Fokuspunkter & afgrænsning	3
1.4 Læsevejledning	4
1.4.1 Skriftlige bemærkninger	4
Kapitel 2 Fremgangsmåde	5
2.1 Udviklingsproces	5
Kapitel 3 Definering af krav	8
3.1 Aktørbeskrivelse	8
3.2 Use Case beskrivelse	9
Kapitel 4 Systemarkitektur	10
4.1 Blokidentifikation for Teststand	10
4.1.1 Blokbeskrivelse for Teststand	11
4.1.2 Allokering af logisk funktionalitet	12
Kapitel 5 Udvikling af Belastningskreds	13
5.1 Overvejelser omkring DC-generatoren	13
5.2 Design af Effektkredsen	13
5.3 Design af Strømovervågning	15
5.4 Implementering af Belastningskredsen	16
Kapitel 6 Udvikling af Brugergrænseflade & Kommunikation	17
6.1 Valg af kommunikation	17
6.2 Design af Brugergrænseflade	18
6.2.1 Datalogning	19
6.3 Implementering af Brugergrænseflade og kommunikation	19
Kapitel 7 Udvikling af Kontrolenhed	20
7.1 Valg af Kontrolenhed	20
7.2 Design af programmet til PSoC5	20
7.3 Dataopsamling & behandling	20
7.4 Belastningsstyring	21
7.5 Implementering af programmet til PSoC5	22

Kapitel 8 Udvikling af Sensorer	24
8.1 Effekttransducer	24
8.1.1 Design	24
8.1.2 Implementering	25
8.2 Momenttransducer	25
8.2.1 Overvejelser	25
8.2.2 Design	26
8.2.3 Implementering	26
Kapitel 9 Resultater	28
9.1 Opsummering af integrationstest	28
9.2 Måling af elektrisk tilført effekt	28
9.3 Måling af mekanisk effekt	28
9.4 Beskyttelse mod overbelastning med Strømovervågning	29
9.5 Styring af strøm i Belastningskreds	29
9.6 Regularering	29
9.7 Kommunikation mellem Brugergrænseflade og Kontrolenhed	29
Kapitel 10 Diskussion & Evaluering	30
10.1 Diskussion af opnåede resultater	30
10.1.1 Elektrisk belastning	30
10.1.2 Måling	30
10.1.3 Brugerinteraktion	31
10.1.4 Dataopsamling og regulering	32
10.2 Opnåede erfaringer	32
10.3 Fremtidigt arbejde	33
Kapitel 11 Konklusion	34
Referencer	i
Litteratur	i

Indledning 1

Shell Eco-Marathon er en konkurrence, hvor studerende fra hele verden konkurrerer om at designe, udvikle og køre den mest energieffektive bil. De konkurrerende biler inddeltes i to hovedkategorier. Den første er **Prototype** klassen, hvor der fokuseres på den størst mulige energieffektivitet, mens passagerkomfort ikke prioriteres. Den anden kategori er **UrbanConcept** klassen, hvor fokus punktet ligger i praktiske anvendelser. Ydermere er bilerne inddelt efter energitype¹:

- Forbrændingsmotorer
- Elektriske motorer

Ingeniørhøjskolen Aarhus Universitet (ASE) stiller op til Shell Eco-Marathon 2016 med en egenudviklet bil. Bilen, som ASE stiller op med, er i **Prototype** klassen. Herunder i kategorien **Elektriske motorer**. Bilen er en videreudvikling af Team AUs elbil fra Eco-Marathon 2014.



Figur 1.1. Team AUs elbil fra Eco-Marathon 2014

Der ønskes udviklet en teststand til test af bilens ydeevne, uafhængigt af kørsel på en testbane. På denne måde kan test og optimering af elbilen foretages hjemme på værkstedet, således at elbilen er finindstillet til konkurrencestart.

1.1 Opgaveformulering

I dette bachelorprojekt ønskes der udviklet en teststand, hvis funktionalitet er at kunne belaste elbilens drivlinje². Systemet skal være i stand til at kunne måle drivlinjens effektivitet og energioptag ved forskellige belastninger.

De optagne målinger ved test, skal benyttes i udviklingsprocessen af elbilens drivlinje og elmotor med dens tilhørende motorstyring. Formålet er at kunne finde motorens optimale driftsbetingelser, således der opnås maksimal energiudbytte.

¹Shell Global 26.11.2015. <<http://www.shell.com/global/environment-society/ecomarathon/about.html>>

²Begrebet Drivlinje dækker over bilens elmotor, gearudveksling og kobling til bilens hjul

Derfor er det nødvendigt at undersøge følgende:

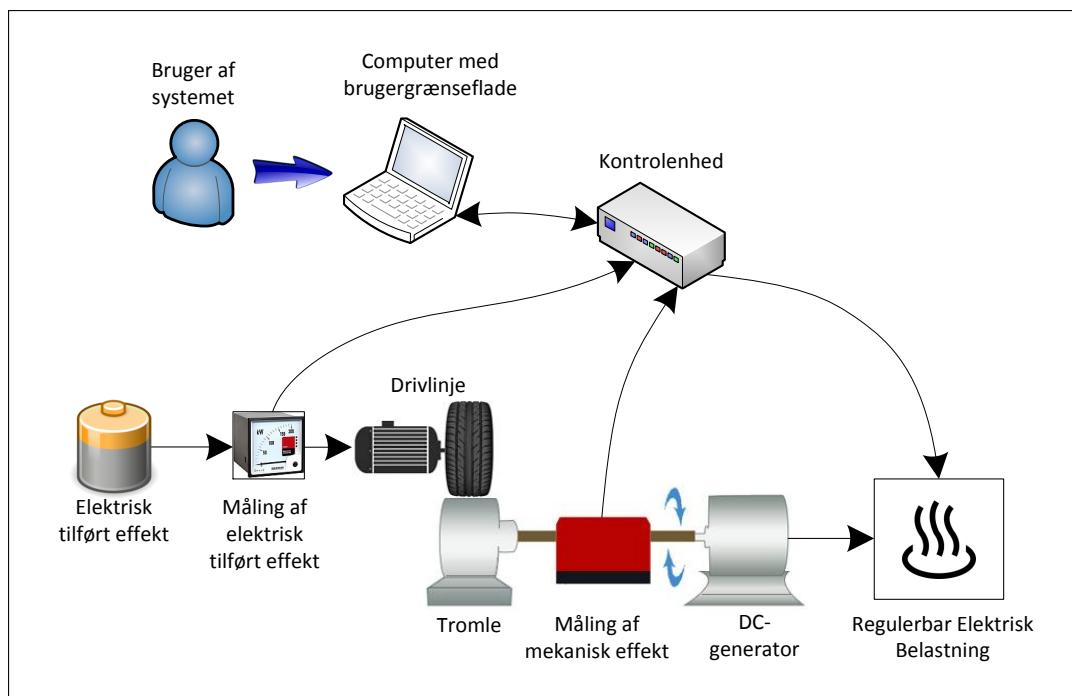
Hvordan vil et system kunne designes, hvor der ved test på elbilens drivlinje kan optages målinger, således at motorens optimale driftsbetingelser findes? Kan systemet desuden designes, så man kan belaste elmotoren for at simulere den belastning, som bilen vil udsættes for under kørsel?

Hvis det er muligt at udvikle en teststand, der kan optage målinger, forventer vi, at det er muligt at optimere elbilen op til konkurrencestart.

1.2 Systembeskrivelse

Med udgangspunkt i opgaveformuleringen implementeres det færdige system som et rullefelt. Rullefeltet består af en tromle, som drives af elbilens drivlinje. Tromlen er mekanisk forbundet til en aksel, som driver en DC-generator. For at opnå den ønskede funktionalitet kan DC-generatoren belastes elektrisk, således at elbilens drivlinje belastes mekanisk. For at kunne optage målinger ved test er der koblet en transducer på den mekaniske aksel. Transduceren kan måle hastigheden og momentet på akslen. Ligeledes måles den optagne elektriske effekt på elbilens motor. Gennem maling af den mekaniske og elektriske effekt kan drivlinjens effektivitet og energioptag findes. Systemet er illustreret på figur 1.2.

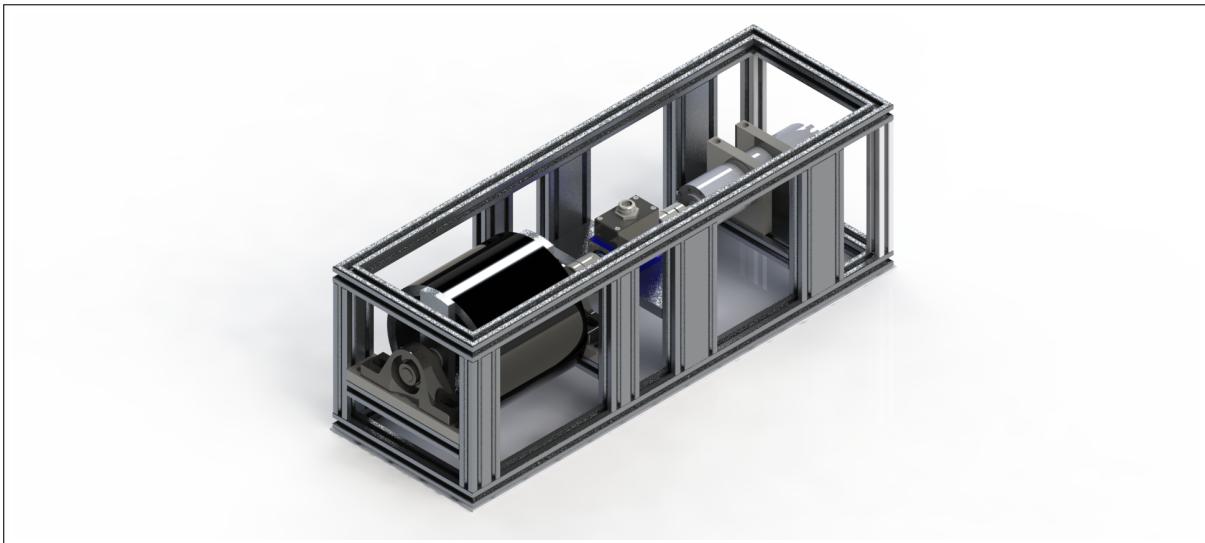
For at kunne optage målinger af den elektriske og mekaniske effekt samt styre belastningen, indføres der en kontrolenhed. Der benyttes en Brugergrænseflade til interaktion med systemet. Gennem Brugergrænsefladen har brugeren adgang til de optagne målinger og til styring af systemet.



Figur 1.2. Illustration af systemet

Grundarkitekturen for projektet er fastlagt på forhånd af den hovedansvarlige for Eco-Marathon elbilen på ASE. Grundarkitekturen for teststanden er et rullefelt, en momenttransducer og en DC-generator. Figur 1.3 viser grundarkitekturen for teststanden. Design og opbygning af det mekaniske system er foretaget af

en gruppe maskiningeniørstuderende på ASE. Det betyder, at opgaven der tilfalder bachelorgruppen, er at designe og opbygge det elektriske system.



Figur 1.3. Grundarkitekturen for teststanden

På baggrund af projektbeskrivelsen og ovenstående afgrænsning, er hovedkravene for dette bachelorprojekt:

1. At måle den elektrisk tilførte effekt på elbilen
2. At måle den mekaniske effekt som elbilen afgiver
3. At regulere belastningen elektrisk
4. At udvikle software til en kontrolenhed
5. At udvikle en brugergrænseflade, der kan interagere med systemet

1.3 Projektafgrænsning

Vi har valgt at udvikle en skræddersyet løsning til Team AUs elbil, og dermed er der ikke lavet en generisk løsning til alle projekter med lignende problemstillinger. Det vil blive efterstræbt at implementere en funktionsdygtig prototype af systemet. På baggrund af figur 1.2 og systembeskrivelsen kan det konkluderes, at systemet skal indeholde følgende:

1. Et delsystem til elektrisk belastning
2. Et delsystem til måling
3. Et delsystem til brugerinteraktion
4. Et delsystem til dataopsamling og regulering

1.3.1 Fokuspunkter & afgrænsning

Følgende punkter vil være rammesættende ved udvikling af systemet:

1. **Høj præcision:** Da formålet med teststanden er at kunne optage målinger på elbilen med henblik på optimering af bilens effektivitet, tilstræbes det at udvikle et system med høj præcision af hardware såvel som software.

2. Implementering af teststanden: For at overholde tidsrammen, der er stillet for bachelorprojektet, må projektet afgrenses. Der vil ikke blive fokuseret på at lave et færdigimplementeret system. Det efterstræbes i stedet at lave en funktionsdygtig prototype, hvor indholdet senere kan færdigimplementeres.

- **Grafisk brugergrænseflade:** Da produktet er til internt brug på ASE, og ikke et produkt der skal på markedet, er det valgt at nedprioritere det visuelle design af brugergrænsefladen. I stedet fokuseres der på brugervenligheden af brugergrænsefladen.
- **Implementeringen af de elektriske systemer:** Det er valgt at implementere de elektriske systemer på veroboard frem for PCB. Dette giver ikke systemet den robusthed, som et færdigimplementeret system skal have, men det giver den ønskede funktionalitet.

1.4 Læsevejledning

Rapporten starter med at definere problemet, lave en systembeskrivelse, definere de afgrensende elementer samt beskrive den anvendte udviklingsproces. Herefter vil de tekniske aspekter følge, i form af et kapitel for systemarkitektur og dernæst kapitler for hvert af de fire udviklede delsystemer. Kapitlerne for de fire delsystemer indeholder dybdegående forklaringer omkring design, implementering samt tests. Til sidst afsluttes rapporten med de opnåede resultater, som diskuteres, samt en evaluering af projektet og dets fremtidige arbejde.

1.4.1 Skriftlige bemærkninger

I denne rapport er henvisninger til blokke, Use Cases og aktører benævnt med **fremhævet** skrift. Ydermere er funktionsnavne og hardwarekomponenter fremhævet med *kursiv* skrift. Dette er gjort for at gøre teksten mere læsbar og for at gøre læseren opmærksom på væsentlige begreber.

Fremgangsmåde 2

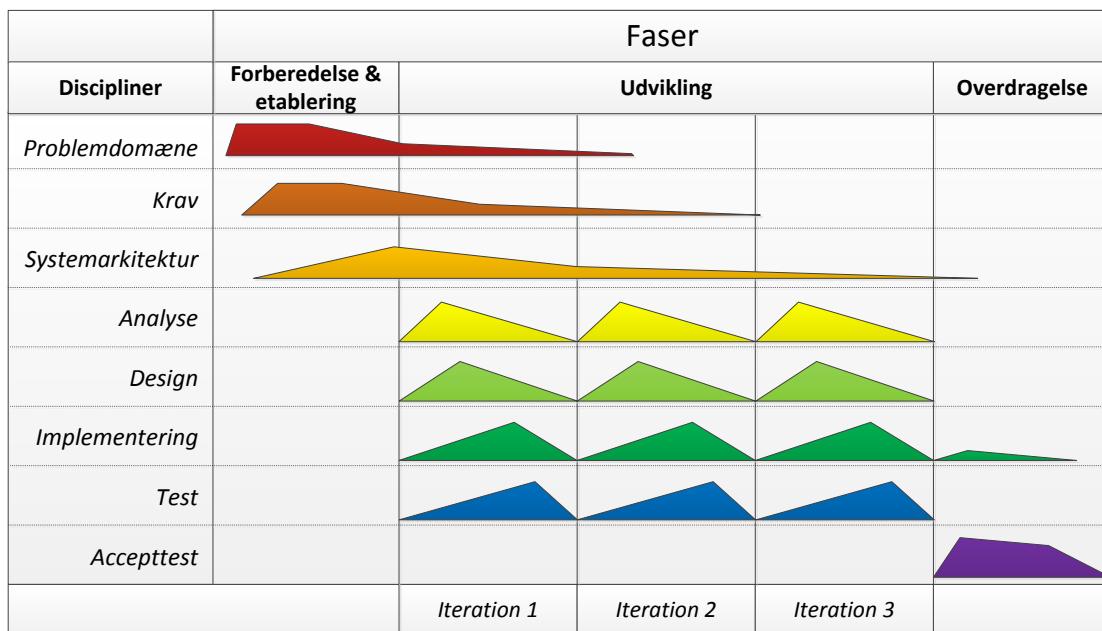
I dette kapitel beskrives udviklingsprocessen, som anvendt under udvikling af systemet. Ydermere beskrives de forskellige udviklingsværktøjer, der er anvendt under projektforløbet.

2.1 Udviklingsproces

Udviklingsprocessen for projektet er inspireret af forskellige udviklingsprocesser, som gruppen har stiftet bekendtskab med i tidligere projekter. Dette gør sig gældende både for den teoretiske del og den praktiske anvendelse.

Fokuspunktet for projektet har været at udvikle et system ved brug af en iterativ udviklingsproces. Ved at arbejde iterativt skabes der mulighed for at kunne designe, implementere og teste samtidig.

Ved produktudvikling er det oplagt at arbejde iterativt. Ydermere er det også vigtigt, at udvikle i en inkrementel proces, da det gør det muligt at fremvise noget funktionelt til møderne med kunden. Ved inkorporering af en inkrementel udvikling til den iterative proces, fås et mere færdigudviklet system efter hver iteration. Det betyder, at systemet vil kunne fremvises med nye funktionaliteter efter hver iteration. I dette projekt er udviklingsprocessen forløbet som illustreret i figur 2.1.



Figur 2.1. Illustration af projektets udviklingsproces

Ud fra figur 2.1 kan udviklingsprocessen opdeles i tre overordnede faser:

- Forberedelse og etablering
- Udvikling
- Overdragelse

Disse faser er nøjere beskrevet i de efterfølgende afsnit.

Forberedelse og etablering

I denne fase er fokuspunktet at forstå opgaven og fastlægge problemdomænet, for derved at udarbejde en analyse af problemet.

Dernæst bliver der fokuseret på, at udarbejde et udkast til en kravspecifikation for derved at formulere en systemarkitektur. Det er særligt væsentligt at udarbejde en solid kravspecifikation, da der tages udgangspunkt i denne under udvikling af systemet.

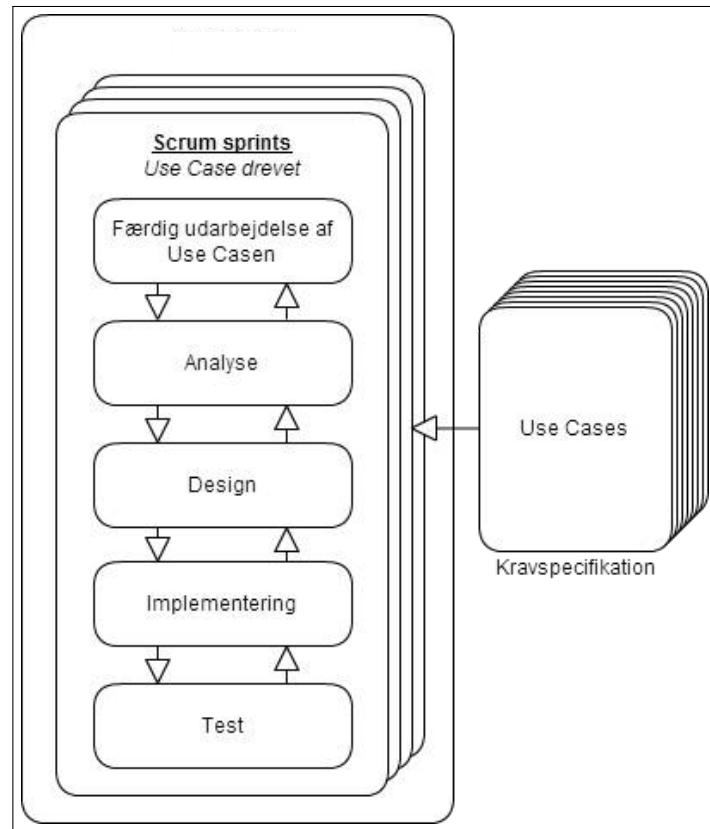
Udvikling

Udviklingsfasen er den længste af de tre faser. Det er i denne fase der primært bliver fokuseret på udviklingen af systemets funktioner. Ved fasens afslutning er alle use cases implementeret, og systemet betragtes som værende fuldt fungerende.

Til koordinering af udviklingsfasen bliver SCRUM¹ anvendt, sideløbende med en use case drevet udvikling som illustreret på figur 2.2. Der fokuseres udelukkende på SCRUM aktiviteter, som er relevant for projektet, dvs. der er anvendt elementer af SCRUM. Følgende er en beskrivelse af de SCRUM aktiviteter, som er anvendt i projektet.

¹SCRUM er en projektstyringsmetode, der understøtter iterativ og inkrementel udvikling.

- **Produkt Backlog:** En *Produkt Backlog* består af use cases fra kravspecifikationen. Hver use case betragtes som en enkelt opgave.
- **Sprintplan:** Planlægningen af de enkelte sprints, specificeres et fokusområde for det pågældende sprint. Derved bliver det planlagt, hvilke opgaver, der skal varetages ud fra *Produkt Backlog*'en.
- **Dagligt SCRUM møde:** Et dagligt møde mellem projektgruppen er blevet afholdt, hvor der er givet status over arbejdet.



Figur 2.2. Oversigt over udviklingsfasen

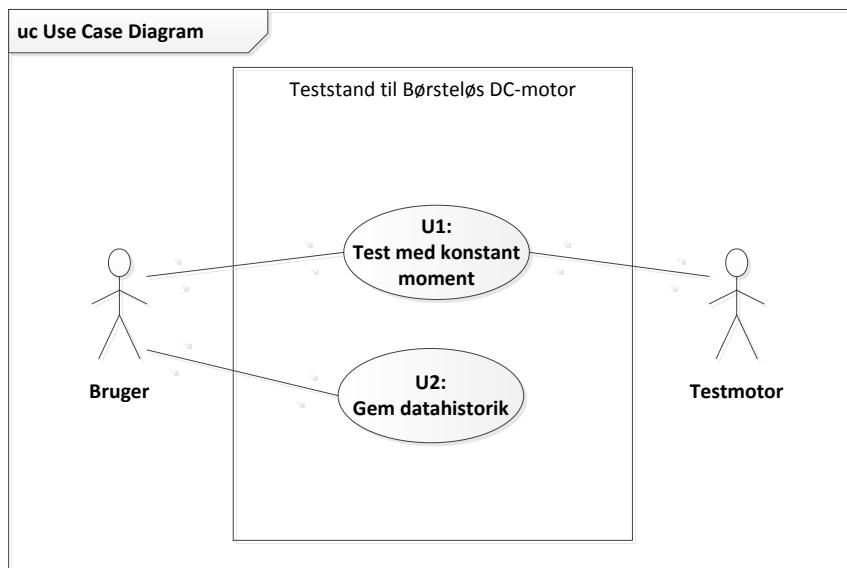
Overdragelse

I denne fase bliver der fokuseret på at overdrage produktet til de ansvarlige for projektet. Dette foregår ved, at acceptteste produktet i samarbejde med kunden. I denne anledning kan der opstå behov for finjusteringer på produktet. Derfor vil der i overdragelsesfasen forekomme mindre implementering og finjusteringer på systemet.

Det beskrevne projekt er således båret af en iterativ og inkrementel proces, hvor udviklingsdelen koordineres af elementer fra SCRUM.

Definering af krav 3

For at specificere systemets funktionalitet, er der defineret en række krav til systemet, som skal udvikles. Kravene er specificeret ved en række Use Cases, som definerer systemets funktionelle krav, og derudover er der specificeret en række ikke-funktionelle krav. Use Casene udformes som scenarier af interaktion mellem systemet og systemets aktører.¹



Figur 3.1. Use Case diagram for systemet

3.1 Aktørbeskrivelse

Systemet har to aktører: Brugeren og Testmotoren, som systemet skal kunne belaste og foretage målinger på. På figur 3.1 ses Use Case diagrammet for systemet. Her er vist aktørerne, der findes i systemet samt hvilke Use Cases disse er en del af.

¹Den fulde Kravspecifikation forefindes i Dokumentationen kapitel 1

Brugeren

Brugeren er systemets primære aktør. Det er brugeren, som vil benytte systemet til at foretage test på Testmotoren.

Testmotoren

Alternativ reference: Drivlinje, Bilens elmotor

Testmotoren er den aktør, systemet skal foretage test på. Aktøren påvirker systemet med en mekanisk effekt. Derudover optages der målinger af Testmotorens optagne elektriske effekt.

3.2 Use Case beskrivelse

Use Casene på figur 3.1 er de forskellige scenarier, der beskriver de funktionelle krav til systemet. De enkelte use cases beskriver et scenario, hvor aktører interagerer med systemet.

Use Case 1: Test med konstant moment

I Use Case 1 foretager Brugeren test af Testmotoren. Brugeren kan via Brugergrænsefladen se status på testen, indtaste det moment som Testmotoren skal belastes med, og herefter starte systemet. Systemet vil regulere således, at Testmotoren yder et konstant moment. Mens testen er i gang vises følgende på brugergrænsefladen; Testmotorens elektriske optagne effekt, afgivne mekaniske effekt, effektivitet, det elektriske energiforbrug, samt hastighed og moment på systemets aksel. Efter endt test kan Brugeren stoppe systemet via Brugergrænsefladen.

I tilfælde af overbelastning af DC-generatoren under test, afbrydes testen automatisk og status på Brugergrænsefladen vil vise "Stoppet".

Use Case 2: Gem Datahistorik

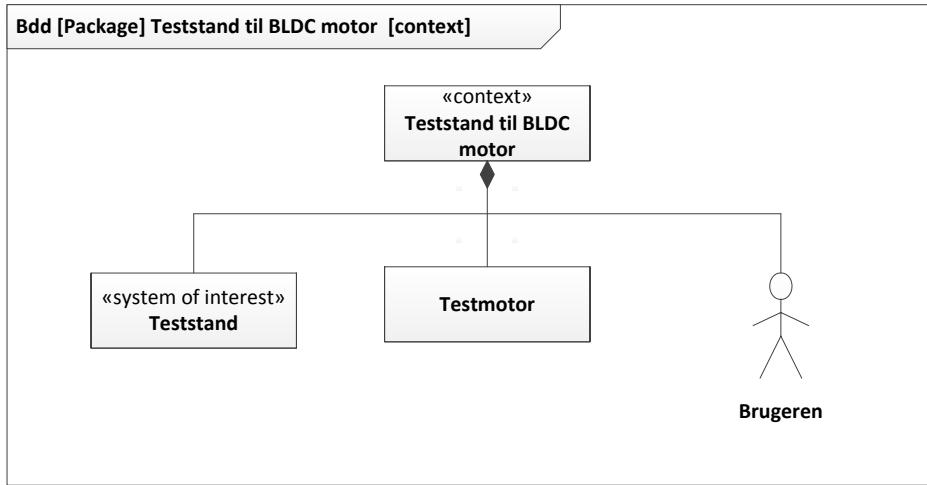
I Use Case 2 udtrækker Brugeren måledata, som er optaget under test. Bruger kan efter endt test, via Brugergrænsefladen, gemme måledata på sin computer. Måledata gemmes for hhv. momentet, hastigheden, mekanisk effekt, elektrisk tilført effekt, virkningsgrad og elektrisk energiforbrug. De udtrukne data kan anvendes til videre analyse af testmotoren.

Disse krav forsøges imødekommet i systemarkitekturen, hvorfor udarbejdelse og beskrivelse af denne er næste led i arbejdsprocessen.

Systemarkitektur 4

I dette kapitel beskrives systemarkitekturen for systemet. Systemarkitekturen er rammesættende for design og implementering af systemet, som beskrives i de efterfølgende kapitler.

Det er her, systemets funktionalitet, der er beskrevet i Systembeskrivelsen kapitel 1.2 på side 2, nedbrydes til overordnede moduler.



Figur 4.1. BDD context for systemet

Systemet kan illustreres i et block definition diagram (BDD) bestående af tre moduler, og er vist på figur 4.1.

Brugeren er vedkommende, der ønsker at benytte systemet. Via brugergrænsefladen kan brugeren interagere med systemet.

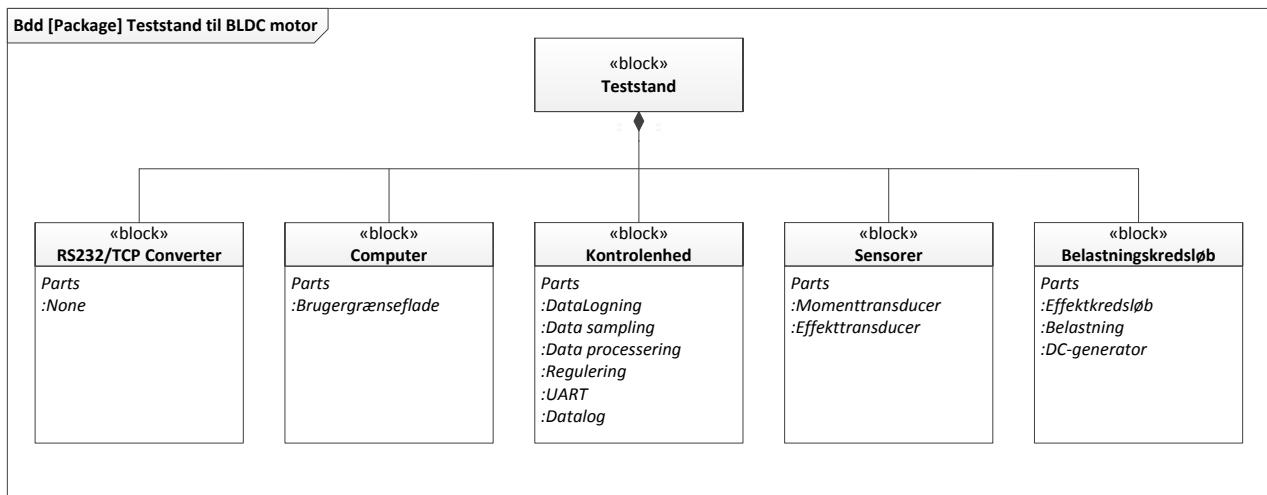
Testmotoren er elbilens motor, der ønskes testet.

Teststanden er systemet, der ønskes udviklet for at kunne foretage test af Testmotoren.

4.1 Blokidentifikation for Teststand

For at systemet får den ønskede funktionalitet, som er beskrevet i kapitel 3, er der forskellige moduler som systemet må indeholde. Dette indebærer moduler, som skal muliggøre interaktion mellem Brugeren

og systemet, samt moduler, der kan foretage målinger og belaste Testmotoren. De identificerede moduler, der skal opfylde Teststandens funktionalitet, er illustreret i følgende BDD på figur 4.2.



Figur 4.2. BDD diagram for Teststand

4.1.1 Blokbeskrivelse for Teststand

Sensorerne i systemet er en moment- og en effektransducer. Momenttransduceren optager måling af den mekanisk overførte effekt i form af moment og hastighed. Målingerne sendes til systemets kontrolenhed. Effektransduceren optager måling af den elektrisk tilførte effekt til testmotoren. Målingerne er i form af spændings- og strømmåling. De målte data sendes til systemets kontrolenhed til beregning og regulering.

Belastningskredsløbet bruges til at trække effekt ud af systemet for derved at styre den mekaniske belastning på akslen. Belastningskredsen består af en DC-generator, der omsætter den mekaniske effekt til elektrisk effekt. Vha. effektelektronik kan mængden af elektrisk effekt, der skal afsættes, kontrolleres. Ydermere tilføjes en strømmåling, der overvåger strømmen gennem belastningskredsløbet for at beskytte DC-generatoren mod overbelastning.

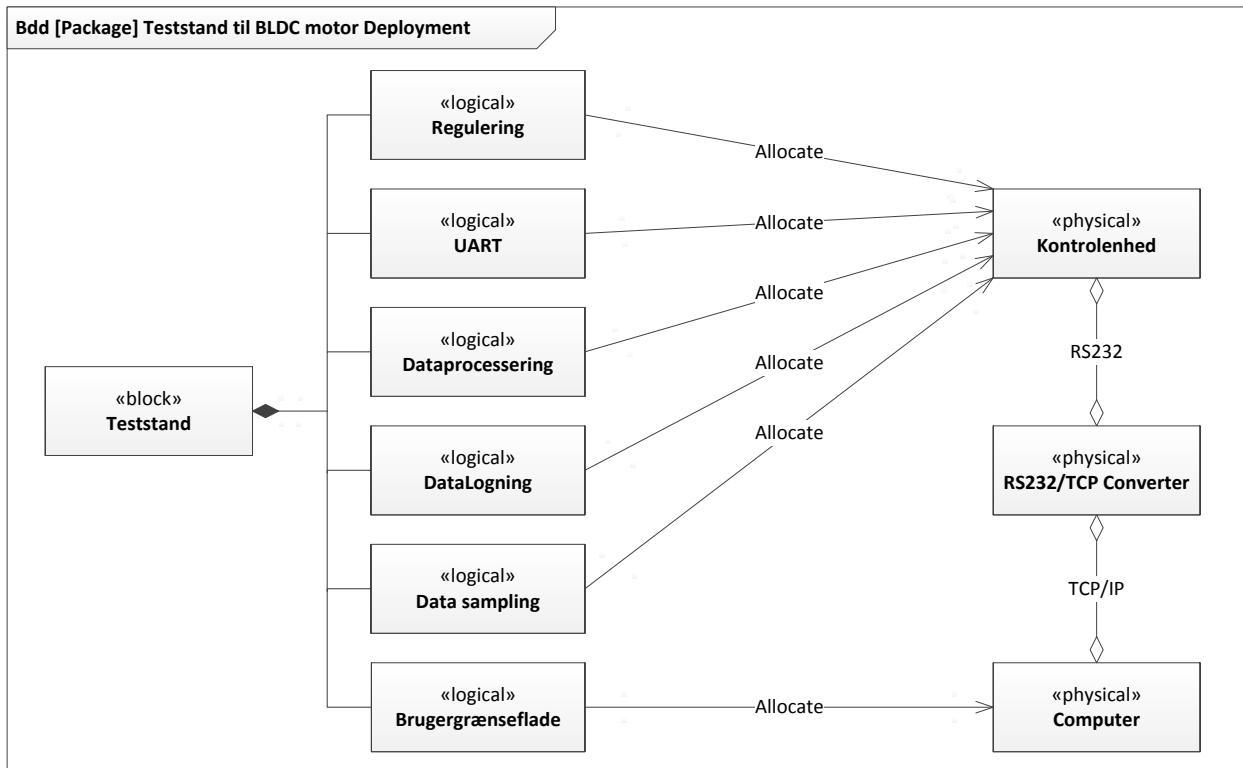
Computeren vil være Brugerens PC, som vil være forbundet til kontrolenheden. Herfra kan Brugeren interagere med systemet via Brugergrænsefladen. Det er her Brugeren kan foretage tests og se måledata fra test.

Kontrolheden er ansvarlig for reguleringen og logningen af data. Reguleringen foretages på baggrund af målinger fra momenttransduceren, og effektafsættelsen i belastningskredsløbet styres med et reguleringssignal.

RS232/TCP Converteren er tilføjet for at kunne oprette en kommunikation mellem computeren og kontrolheden.

4.1.2 Allokering af logisk funktionalitet

Teststanden består af en række blokke, der har en logisk funktionalitet, som er allokeret på en række fysiske enheder. Nedenstående figur 4.3 viser, hvor de logiske blokke er allokeret på de fysiske.



Figur 4.3. Deployment diagram for Teststand

Det ses af diagrammet, at de logiske blokke **Regulering**, **UART**, **Dataprocessering**, **DataLogging** og **Data sampling** alle er allokeret på Kontrolenheden. I dette projekt er det besluttet, at Kontrolenheden skal implementeres som en *PSoC5LP μController*.

Brugergrænsefladen er allokeret på brugerens computer. Det er besluttet, at computeren skal køre med et Linux baseret operativsystem.

RS232/TCP Converter benytter sin egen indbyggede logiske funktionalitet.

Med redegørelsen i Kapitel 3 og 4 er processens første fase, Forberedelse og etablering, afsluttet. Projektets anden fase, Udvikling, kan nu påbegyndes, og vil blive beskrevet i de efterfølgende fire kapitler.

Udvikling af Belastningskreds

5

I dette kapitel beskrives *Belastningskredsen*. Designet tager udgangspunkt i kravspecifikationen og systemarkitekturen. Designprocessen samt hvilke overvejelser og valg der er taget, vil desuden blive beskrevet¹.

5.1 Overvejelser omkring DC-generatoren

Som beskrevet i Systembeskrivelsen afsnit 1.2, er grundarkitekturen for projektet fastlagt på forhånd. Dette indebærer, at det er fastlagt at der skal benyttes en *DC-generator* til at omsætte den mekaniske effekt til elektrisk effekt, hvorefter systemet kan belastes elektrisk. *DC-generatoren*, som er indkøbt til teststanden, er den samme DC-motor som Team AUs elbil fra Eco-Marathon 2014, benyttede som elmotor. Med denne type *DC-generator* kan elbilens drivlinje fuldt belastes, da effekterne på både generator og motor er identiske.

Gruppen i Team AU, der er ansvarlige for udvikling af elbilens elektriske systemer, ønsker at benytte en anden type elmotor til Team AUs elbil 2016, end den der blev benyttet i 2014. Elmotorerne, der har været i forslag, har en større nominel effekt end den elmotoren, der blev benyttet i 2014. Dette betyder at den indkøbte *DC-generator* til teststanden, ikke er stor nok til fuldt ud at belaste elbilens drivlinje.

For at imødekomme denne udfordring er andre størrelser og typer af *DC-generatorer* undersøgt med fokus på at benytte samme type motor, drevet som generator, som de forslædede elmotorer til elbilen. Det var dog ikke muligt at finde en *DC-generator*, der var i passende overensstemmelse med ønskerne på daværende tidspunkt.

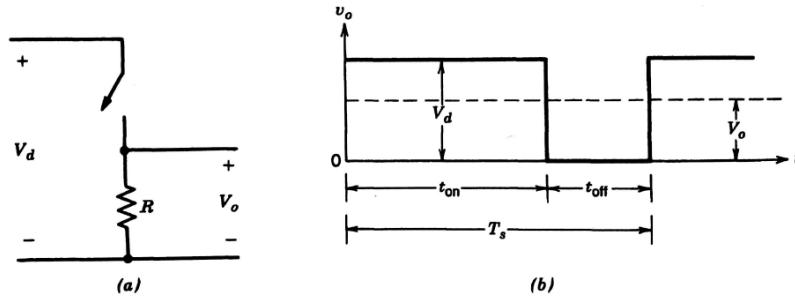
I samarbejde med den ansvarlige for udviklingen af elbilens elektriske systemer, blev det besluttet at benytte den indkøbte *DC-generator*, vedviedende at denne ikke kan belaste de foreslædede elmotorer til elbilen, fuldt ud. Valget er bl.a. truffet for at undgå konstruktionsændringer i grundarkitekturen af teststanden.

For at give systemet fleksibilitet til at kunne fuldt belaste de forslædede elmotorer, er det besluttet at den resterende del af *Belastningskredsen* skal designes til at kunne belaste den største foreslædede elmotor. På denne måde kan der eftermonteres en *DC-generator* med en nominel effekt der tilsvarer elmotorens effekt.

5.2 Design af Effektkredsen

Som tidligere beskrevet skal *Belastningskredsen* kunne belaste *DC-generatoren* elektrisk, og belastningen skal være regulerbar. Det er valgt at benytte en effektmodstand til at afsætte den elektriske energi i. For at kunne regulere effektafsættelsen er der taget udgangspunkt i principippet for en Step-Down converter, som ses på figur 5.1 [4].

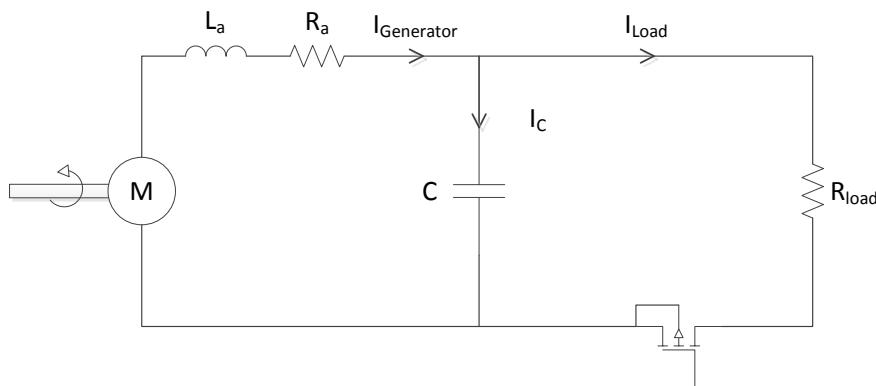
¹For en detaljeret gennemgang af *Belastningskredsen* henvises der til Hardware Design & Implementering, kapitel 3 i Dokumentationen



Figur 5.1. Princip af switchmode DC-DC Step-Down converter

Inputspændingen til converten V_d , er spændingen leveret af *DC-generatoren*. Outputspændingen V_o er spændingen over belastningsmodstanden. V_o kan varieres ved at styre dutycyclen på switchen, hvor middelværdien af outputspændingen er proportionel med dutycyclen. Ved brug af denne metode kan effektafsættelsen i belastningsmodstanden reguleres vha. af et PWM-signal.

Der er tilføjet en kondensator for at dæmpe rippetrømmen i DC-generatoren. Ved at indføre kondensatoren fås et RLC-lavpasfilter, bestående af *DC-generatoren*s ankermodstand, selvinduktion og den tilføjede kondensator. Dette lavpasfilter vil dæmpe rippetrømmen, der opstår grundet switching, og dermed er strømmen fra *DC-generatoren* en DC-strøm, som varierer med dutycyclen. På figur 5.2 ses den principielle opbygning af *Effektkredsen*.



Figur 5.2. Princip af effektkredsen

Strømmen i kondensatoren vil over en switchperiode variere, det vil sige der løber en rippetrøm. Den valgte kondensator, implementeret i systemet, tåler dog kun den rippetrøm der forekommer ved brug af den indkøbte *DC-generator*. Dette valg er truffet af økonomiske hensyn, da en kondensator, der tåler en større rippetrøm, er vurderet til at være for bekostelig. Denne kan dog nemt udskiftes, såfremt der anvendes en *DC-generator*, hvor rippetrømmen i kondensatoren bliver større.

Udfordringen med denne løsning findes blandt andet under opstart, hvor kondensatoren endnu ikke er opladet. Der vil løbe en større strøm fra *DC-generatoren* under opstart, end der vil under stationær tilstand, da kondensatoren skal oplades. Dette medfører et opstartsforløb hvor elbilens drivlinje ikke belastes jævnligt.

Ligeledes er der en udfordring når systemet stoppes. Når systemet har kørt, er kondensatoren opladet,

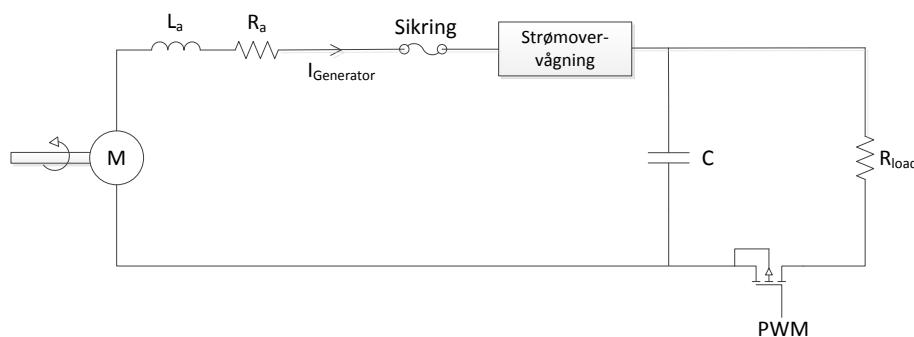
og da systemet er stoppet genererer *DC-generatoren* ingen spænding. Dette vil medføre at strømmen vil løbe fra kondensatoren og tilbage i *DC-generatoren*, som vil køre som motor, indtil kondensatoren er afladet. Denne udfordring kunne løses ved at indsætte en diode, for at sikre strømretningen. På grund af systemets design vil spændingsfaldet over en diode medføre, at systemet ikke vil kunne afsætte den effekt, som er specificeret med den indkøbte *DC-generator*. Det er derfor valgt ikke at indføre en diode i designet af systemet.

5.3 Design af Strømovervågning

For at beskytte *DC-generatoren* mod overbelastning, ønskes det at systemet stopper, hvis strømmen fra *DC-generatoren* overstiger dens maksimale mærkestrøm. For at kunne overvåge strømmen fra *DC-generatoren*, skal der indføres en *strømovervågningskreds*.

Der er flere forskellige muligheder for at kunne måle strømmen. Én mulighed er at indføre en shuntmodstand, og måle spændingsfaldet over shuntmodstanden. Udfordringen ved at benytte denne metode, er at effektafsættelsen i shuntmodstanden fører til varmeafsættelse, der vil få modstanden i shuntresistoren til at stige. Dette giver en ulineær sammenhæng mellem strømmen fra *DC-generatoren* og spændingsfaldet over shuntmodstanden. Denne metode vil også føre til et spændingsfald, der stiger med strømstyrken.

Den valgte metode til at måle den tilførte strøm, er ved brug af Hall-sensor baseret strømtransducer. Denne type transducer har en høj præcision, og har kun en lille påvirkning på systemet der måles på. Transducertypen måler magnetfeltet omkring lederen og omsætter det til spænding. Magnetfeltet omkring lederen, afhænger af strømstyrken gennem lederen og derved fås en outputspænding, der afhænger af strømmen i lederen. Transduceren er designet således at outputspændingen er proportionel med strømmen i transducerens måleområde. På figur 5.3 ses den principielle opbygning af *strømovervågningen*.

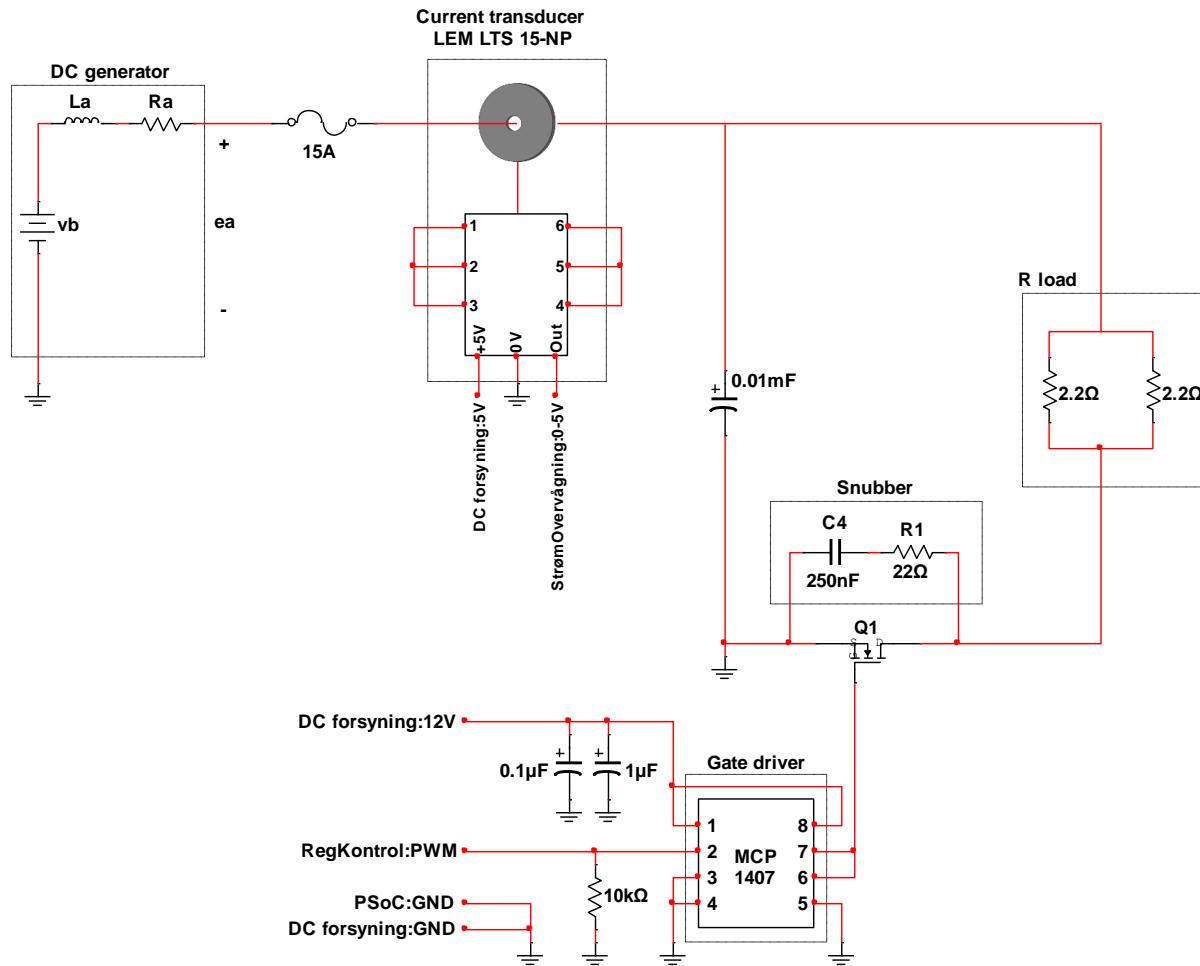


Figur 5.3. Princip af Strømovervågningen

Som en ekstra foranstaltning, til at beskytte *DC-generatoren* mod overbelastning, har gruppen valgt at implementere en smeltesikring. Sikringen har en højere mærkestrøm end grænsen hvor *strømovervågningen* vil få systemet til at stoppe. Dermed er strømovervågningsfunktionen den primære beskyttelse og sikringen er backupbeskyttelse af *DC-generatoren*.

5.4 Implementering af Belastningskredsen

På figur 5.4 ses det færdige design af *Belastningskredsen*. Kredsløbstegningen indeholder de tre hovedkomponenter, som er beskrevet i de tre forgående sektioner, *DC-generatoren*, *Effektkredsen* og *Strømovervågningen*. Udover de tre hovedkomponenter indeholder kredsløbet også en *Gate driver*, hvis formål er at drive switchen, som er implementeret som en MOSFET. Det primære formål med *Gate driveren* er at få MOSFET'en til at lede strømmen så hurtigt så muligt. Kredsløbet indeholder også en *Snubberkreds*, hvis formål er at dæmpe de overspændinger, der kan opstå over MOSFET'en.



Figur 5.4. Implementering af Belastningskredsen

Udvikling af Brugergrænseflade & Kommunikation 6

I dette kapitel beskrives *Brugergrænsefladen* og *Kommunikationen* samt deres funktionalitet. Kapitlet tager udgangspunkt i de vigtigste aspekter i *Brugergrænsefladen*¹.

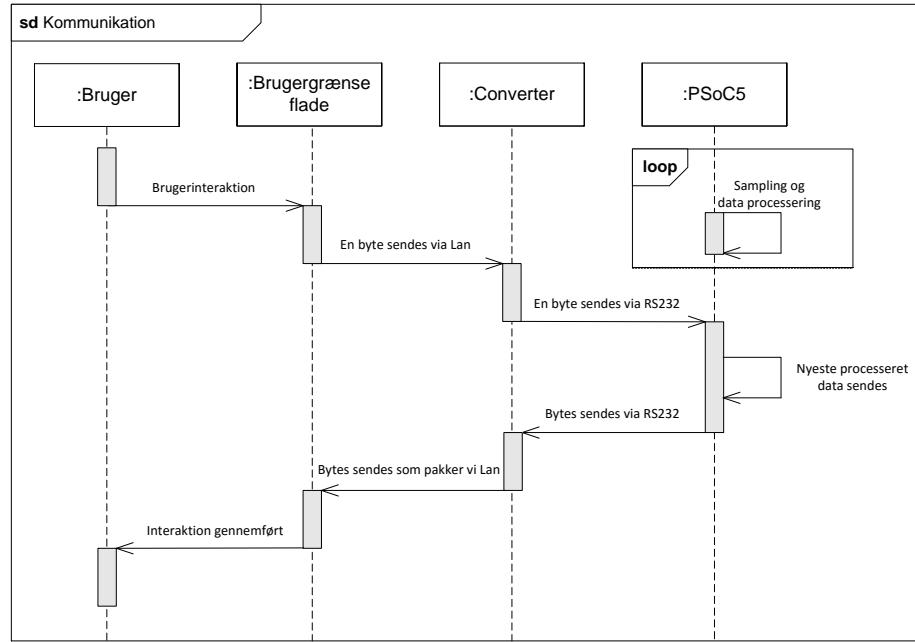
6.1 Valg af kommunikation

Jævnfør Systembeskrivelsen afsnit 1.2 skal *Brugergrænsefladen* kunne kommunikere med *Kontrolenheden*. Det vil sige der skal implementeres en kommunikation mellem disse. Kommunikationsprotokollen er jf. Kravspecifikationen specificeret til at være en TCP/IP-forbindelse.

Kontrolenheden har ikke en integreret ethernet port, og kan derfor ikke udelukkende kommunikere via en TCP/IP-forbindelse. *PSoC5* har i stedet en onboard COM-Port, og denne er udnyttet under udviklingen af kommunikationen. Det er valgt at implementere en UART-forbindelse i form af UART. Fordelen ved UART er, at det er nemt at implementere. Dog har det den ulempe, at det er en upålidelig forbindelse, da der ikke føres noget tjek på evt. mistet data under transmission. Ved god afskærmning, og tilpas lav hastighed på forbindelsen, kan chancen for at miste data mindskes.

For at de to forskellige kommunikationsprotokoller kan kommunikere med hinanden, indføres der en *RS232 til TCP Converter*. Denne har ansvaret for at konvertere UART-forbindelsen fra *Kontrolenheden* til TCP/IP. Den fulde kommunikation for systemet er beskrevet på sekvensdiagrammet, som er vist på figur 6.1.

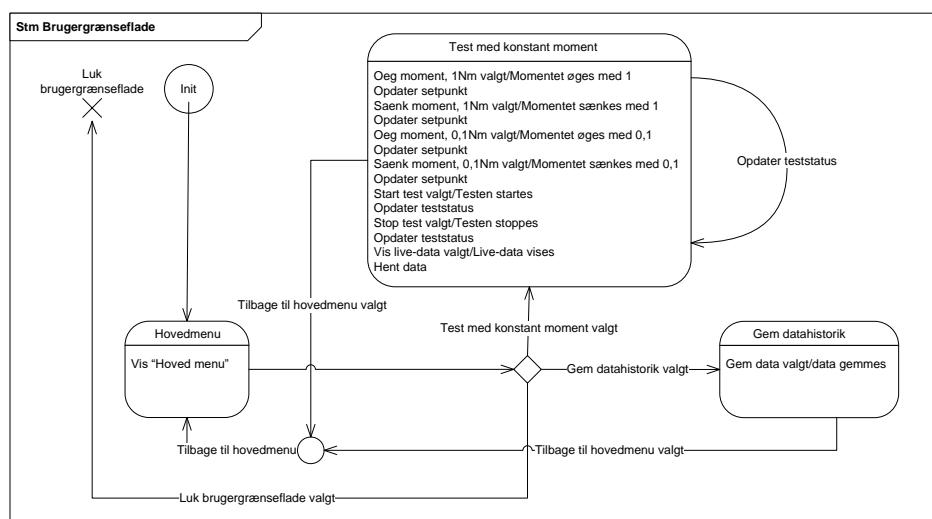
¹For en detaljeret gennemgang af *Brugergrænsefladen* og *Kommunikationen* henvises der til Software Design & Implementering, kapitel 4 i Dokumentationen



Figur 6.1. Sekvensdiagram over kommunikationen i systemet

6.2 Design af Brugergrænseflade

Som nævnt i Projektafgrænsningen afsnit 1.3 på side 3 vil det visuelle design af *Brugergrænsefladen* være nedprioriteret. Dels fordi produktet er til internt brug, og dels fordi det ikke skal på markedet. Der fokuseres i stedet på brugervenligheden af *Brugergrænsefladen*, og dette er realiseret ved et terminalprogram. Der er udviklet et terminalprogram, som Brugeren kan interagere med, til regulering og logning af data. Statemachine diagrammet på figur 6.2 viser opbygningen af *Brugergrænsefladen*.



Figur 6.2. State machine diagram over brugergrænsefladen

6.2.1 Datalogning

Jf. Systembeskrivelsen afsnit 1.2 på side 2, skal brugeren kunne tilgå de optagne målinger fra *Brugergrænsefladen*. For at kunne tilgå de optagne målinger fra *Brugergrænsefladen*, er der opstillet følgende to løsningsforslag:

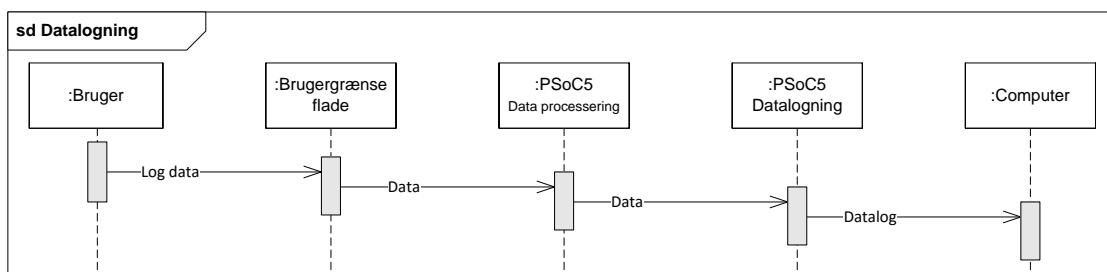
1. Logning af data på et eksternt SD-kort

Denne løsning går ud på at implementere en ekstern SD-kortlæser på *Kontrolenheden*, der ved brug af en SPI kommunikation kommunikerer med denne. Brugeren vil herefter, fra *Brugergrænsefladen*, kunne sende en besked til *Kontrolenheden* om at gemme de optagne målinger på SD-kortet.

2. Logning af data lokalt på Computer

Denne udvikling går ud på at implementere et sideløbende program til *Brugergrænsefladen*, der sørger for at sende og modtage beskeder fra *Kontrolenheden*. På baggrund af indtastninger på *Brugergrænsefladen*, vil de optagne målinger blive gemt i tekstmapper lokalt på *Computeren*.

Det blev i første omgang forsøgt at implementere løsningsforslag 1, men grundet komplikationer med implementeringen af SPI-kommunikationen, blev denne metode skrinlagt. Alternativt blev forslag 2 anvendt som løsning til datalogningen. Fordelen ved denne løsning er, at de loggede data ligger lokalt på *Computeren*, og er klar til viderebehandling. Sekvensdiagrammet på figur 6.3 viser programforløbet for datalogningen.



Figur 6.3. Sekvensdiagram over datalogningen

6.3 Implementering af Brugergrænseflade og kommunikation

Brugergrænsefladen er som nævnt et terminalprogram. Terminalprogrammet er udviklet som et shell script, som er et sprog designet til at køre på en Unix kerne. Sideløbende med *Brugergrænsefladen* er der udviklet et program til håndtering af TCP/IP-forbindelsen. Dette er en videreudvikling af en øvelse, fra et tidligere semesterfag kaldet *Introduktion til kommunikationsnetværk*. Udvikling af UART-forbindelsen er løst ved, at implementere sekvensdiagrammet på figur 6.1 i en switch case struktur. Switch casen modtager en besked og handler på baggrund af denne. Det vil sige alt efter hvilken besked der modtages, vil den pågældende case blive bevirket. Tilsammen udgør TCP/IP- og UART-forbindelsen *Kommunikationen* mellem *Brugergrænsefladen* og *Kontrolenheden*.

Udvikling af Kontrolenhed 7

I dette kapitel beskrives *Kontrolenheden* og dens funktionalitet. Kapitlet tager udgangspunkt i de vigtigste aspekter omkring *Kontrolenheden*¹.

7.1 Valg af Kontrolenhed

Jf. Systembeskrivelsen afsnit 1.2, bliver der stillet en række krav til funktionaliteten af *Kontrolenheden*. Disse funktionaliteter indebærer blandt andet at *Kontrolenheden* skal kunne optage målinger fra diverse transducere. De optagne målinger skal derefter kunne viderebehandles. Ydermere skal *Kontrolenheden* være i stand til at regulere belastningen på den mekaniske aksel. Til sidst skal *Kontrolenheden* kunne kommunikere med en *Brugergrænseflade*.

Det blev besluttet at *Kontrolenheden* skal være en *PSoC5 μController*. Grunden til dette er, at *PSoC5* opfylder de ovennævnte krav til en *Kontrolenhed*. En anden grund til at *PSoC5* er den valgte *μController*, er grundet tidligere erfaring med PSoC familien fra tidligere undervisning og projekter. På daværende tidspunkt blev der anvendt en PSoC3 i undervisningen, den *PSoC5* der anvendes i dette projekt har en anden hardware arkitektur, hvilket har været nødvendigt at undersøge. Det er erfaringer med CyPresses blokbaserede hardwareprogrammering, lært fra tidligere undervisning, der anvendes.

Med valget af *PSoC5* som *Kontrolenhed*, kan der nu designes et program, der opfylder de ovennævnte krav.

7.2 Design af programmet til PSoC5

Programmet til *PSoC5* er designet ud fra systemarkitekturen. Programmet skal opfylde den logiske funktionalitet for måling og styring og, skal muliggøre at *PSoC5* kan kommunikere med *Brugergrænsefladen*.

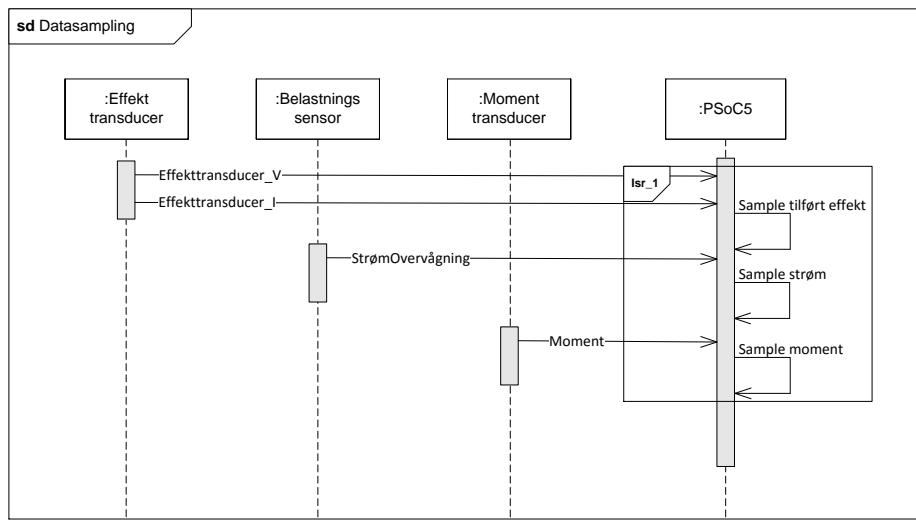
Programmet til *PSoC5* er designet og implementeret ud fra sekvensdiagrammer. De enkelte sekvensdiagrammer illustrerer programflowet, for de forskellige funktionaliteter som *PSoC5* skal udføre. De væsentligste funktionaliteter er beskrevet i de efterfølgende afsnit.

7.3 Dataopsamling & behandling

Jf. kravene for *Kontrolenheden*, er det et krav at der skal kunne optages målinger. Dette er realiseret ved en A/D-konvertering. De opsamlede data midles for at eliminere støj. Alternativt kunne der implementeres et lavpasfilter på *PSoC5*, hvilket vil give samme resultat. Midlingen er implementeret som software, men kunne ligeledes implementeres som en hardwareblok på *PSoC5*. De midlede data anvendes til videre

¹For en detaljeret gennemgang af *Kontrolenheden* henvises der til Software design & implementering, der kan forefindes i kapitel 4 i Dokumentationen

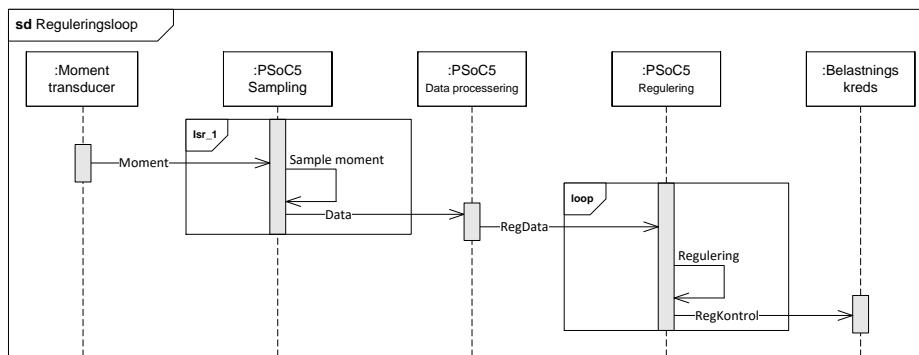
databehandling. Nedenstående sekvensdiagram på figur 7.1 illustrerer hovedscenariet for dataopsamlingen i programmet.



Figur 7.1. Sekvensdiagram over datasamlingen

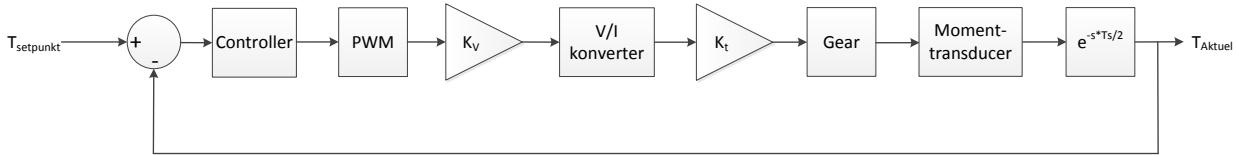
7.4 Belastningsstyring

Til styring af belastningen indføres en reguleringssløjfe, der på baggrund af et setpunkt indstillet i *Brugergrænsefladen*, vil regulere systemet så der opnås det ønskede output. Reguleringssløjfen, der skal implementeres på *PSoC5*, skal designes således, at programmet forløber som vist på sekvensdiagrammet på figur 7.2.



Figur 7.2. Sekvensdiagram over reguleringen

På baggrund af analyse af systemet, er blokdiagrammet på figur 7.3 fremstillet. Blokdiagrammet er udgangspunktet for design og implementering af reguleringssløjfen og uddybes i det følgende.



Figur 7.3. Blokdiagram af systemet

Følgende antagelser er gjort i forbindelse med designet af reguleringen:

- *DC-generatorens* hastighed holdes konstant ved rated hastighed
- Strømmen i *DC-generatoren* betragtes som DC

Setpunktet $T_{setpunkt}$ er brugerens ønskede setpunkt for momentet, som reguleringsloopet skal regulere efter.

Controlleren er den regulatoren, der skal implementeres i systemet, for at opnå en hurtig og præcis regulering.

PWM-blokken omsætter outputtet fra regulatoren til et pulsbredde-moduleret signal, der benyttes til at styre strømmen i *Belastningskredsløbet*.

Forstærkningen K_v er forstærkningsforholdet mellem middelværdien af spændingen fra PWM-blokken og spændingen fra *DC-generatoren*.

V/I konverter omsætter spændingen fra K_v blokken til en strøm.

Forstærkningen K_t er *DC-generatorens* momentkonstant, der omsætter strømmen til et moment i *DC-generatoren*.

Gear er omsætningsforholdet af momentet mellem *DC-generatoren* og systemets aksel.

Momenttransducer omsætter momentet på akslen, til et målesignal.

Tidsforsinkelsen er tilføjet, da der opstår en tidsforsinkelse på tilbagekoblingen, grundet sampling.

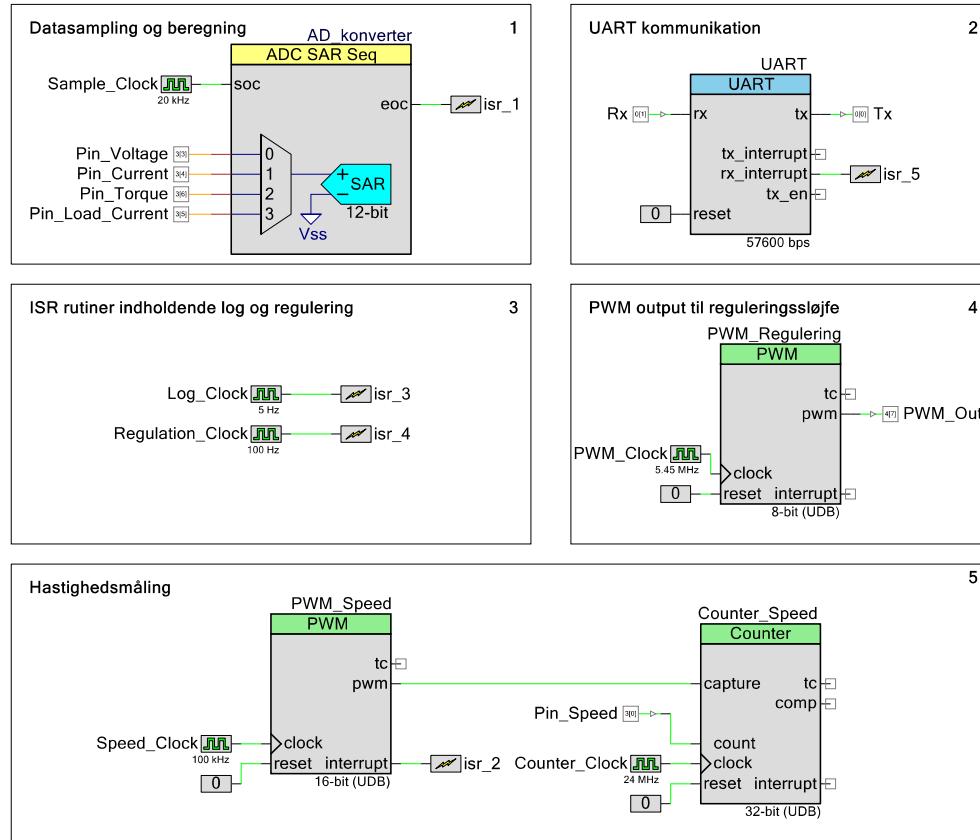
Under analyse og design af overføringsfunktionen for systemet kunne det ses, at systemets respons aldrig nåede setpunktet. Da der ønskes at outputtet af reguleringen skal følge inputtet, indføres en regulator. Udgangspunktet for designet af regulatoren er en PID regulator. Til dette formål benyttes en PI-regulator, som giver systemet en forstærkning samt fjerner den stationære fejl på outputtet. Grunden til at PI regulatoren er valgt, er fordi der ikke er behov for at fjerne overshootet på outputtet. Da formålet med det differentierende led, 'D', netop er at fjerne dette, kan det udelades for dette system. Jf. kapitel 4, afsnit 4,7 i Dokumentationen ses det at systemet har den ønskede båndbredde. Dette er en yderligere årsag til ikke at inkludere det differentierende led D.

7.5 Implementering af programmet til PSoC5

Under implementeringen af programmet, er udfordringen at skrive et program med lav kobling [9] mellem metoderne i C. Dette er løst ved at implementere sekvensdiagrammerne i main funktionen med en interrupt-rutine struktur. Det er forsøgt at opdele funktionalitet op i header- og c filer. Ydermere er

det forsøgt at implementere programmet således, at main funktionen fungerer som en kontrolfunktion, der styrer den overordnede sekventielle gennemgang i diagrammet.

PSoC5 programmet er udviklet i udviklingsværktøjet *PSoC Creator*. Til udvikling af *PSoC5* programmet er det anvendt *PSoC5* komponenter, vist på nedenstående figur 7.4.



Figur 7.4. PSoC5 Topdesign for programmet

Ramme 1 indeholder komponenterne *ADC SAR Seq*, *Clock* og *Interrupt*. *AD-konverteren* har til formål at måle på de indkomne signaler. *Clocken* angiver samplehastigheden på *AD-konverter* blokken.

Ramme 2 indeholder komponenterne *Interrupt* og *UART*, som er anvendt til kommunikation mellem *PSoC5* og *Computeren*. Hver gang data er modtaget vil *interrupt-rutinen* blive kaldt, og de modtagne data vil blive læst.

Ramme 3 indeholder to *interrupt-rutiner* samt to *Clocks*. Den første *interrupt-rutine* indeholder log-variablerne, der anvendes i ramme 2. Den anden *interrupt-rutine* indeholder reguleringen.

Ramme 4 indeholder en *PWM* blok samt en tilhørende *Clock*. *PWM* blokken giver et pulsbredde-moduleret output, på baggrund af outputtet fra reguleringen.

Ramme 5 indholder også en *PWM* blok samt en *Counter*, hver med deres egen *Clock*. Formålet med den femte ramme er at kunne foretage en frekvensmåling. På baggrund af et signal, genereret fra *PWM* blokken, måles frekvensen på et inputsignal på *Counteren*.

Designet, illustreret på figur 7.4, i samarbejde med den udviklede software, udgør *Kontrolenheden*.

Udvikling af Sensorer 8

I dette kapitel beskrives *Sensorer*. Designet tager udgangspunkt i kravspecifikationen og systemarkitekturen. Designprocessen samt hvilke overvejelser og valg der er taget, vil desuden blive beskrevet¹.

Ifølge Systembeskrivelsen afsnit 1.2 på side 2, skal systemet kunne måle den elektrisk tilførte effekt på elbilen samt den mekaniske effekt som elbilen afgiver. For at kunne foretage disse målinger er *Sensorer* opdelt i to delsystemer, en *Effekttransducer* til måling af den elektrisk tilførte effekt og en *Momenttransducer* til måling af den mekaniske effekt.

8.1 Effekttransducer

8.1.1 Design

For at kunne måle den tilførte effekt, kræver det måling af den tilførte spænding og strøm til elbilens motor, da $P_{tilført} = V_{tilført} \cdot I_{tilført}$. Da den tilførte effekt måles som hhv. en spænding og en strøm, er *Effekttransducer* opdelt i to delsystemer: En spændingssensor og en strømsensor.

Design af spændingssensor

Metoden der er valgt til at måle den tilførte spænding, er at måle spændingen over en modstand i et spændingsdeler, bestående af to modstande. Modstandsnetværket er koblet parallelt over belastningen, som i dette tilfælde er elmotoren. Forholdet mellem modstandene beregnes ved, at den tilførte spænding nedskaleres til spændingsniveauet på $\mu Controlleren$. Denne metode er valgt, da spændingen, der ønskes målt, er relativ lav. Havde spændingen været højere, skulle der overvejes en anden løsning. Fordelen ved denne løsning er, at det er en simpel konstruktion og den er billig at implementere.

Udfordringen ved løsningen er blandt andet variationer på modstandene grundet temperatur. Desuden vil, såfremt modstandene udskiftes, nye modstande sandsynligvis have en anden modstandsværdi som følge af nøjagtigheden på dem.

For at beskytte $\mu Controllerens$ indgang kunne der indføres en zenerdiode, koblet parallelt over den modstand, der måles på. Denne vil kortslutte, hvis spændingsniveauet overstiger $\mu Controllerens$ maksimale indgangsspænding. Den valgte $\mu Controller$ i dette projekt har indbygget beskyttelse af dens indgange, og da der udvikles en prototype for at vise principperne for systemet, er det undladt at implementere denne eksterne beskyttelse.

Design af strømsensor

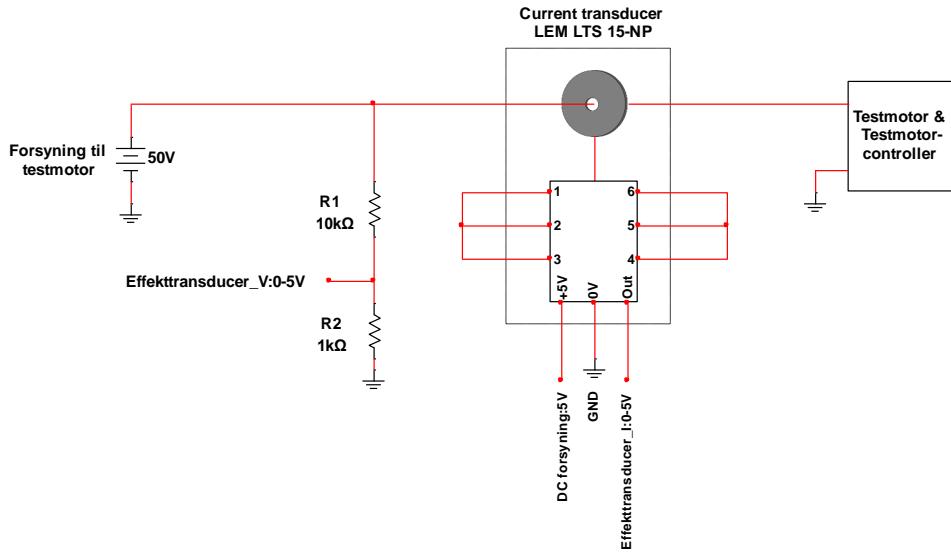
Overvejelserne omkring valg af løsning til måling af elmotorens optagne strøm er de samme som overvejelserne gjort for *Strømovervågningen* i *Belastningskredsen*. Overvejelserne findes i kapitel 5,

¹For en mere detaljeret gennemgang af *Sensorer* henvises der til Hardware Design & Implementering, der forefindes i Dokumentationen.

afsnit 5.3 på side 15.

8.1.2 Implementering

På figur 8.1 ses det færdige design af *Effekttransduceren*. Figuren viser placeringen af hhv. spændings- og strømsensoren. Spændingssensoren er principielt placeret før strømsensoren, så den kun mäter den strøm elmotoren optager. Strømmen i spændingssensoren er desuden så lille, at såfremt den var med i målingen, ville man stadig være inden for den fastsatte tolerance på strømmålingen.



Figur 8.1. Implementering af Effekttransducer

8.2 Momenttransducer

8.2.1 Overvejelser

Som beskrevet i Systembeskrivelsen afsnit 1.2 på side 2, var grundarkitekturen for projektet fastlagt på forhånd. Dette indebar beslutning om at benytte en momenttransducer til måling af den mekaniske effekt, leveret af elbilens motor. Efter anbefaling fra den hovedansvarlige for udviklingen af Teststanden til Eco-Marathon elbilen, er det valgt at anvende en momenttransducer med et USB-interface, da denne type momenttransducer er benyttet på andre Motorstande på ASE, med god erfaring. Til den foreslæde momenttransducer medfølger software, der giver mulighed for en grafisk præsentation, samt mulighed for at logge måledata.

Da der i dette projekt, jf. Udvikling af Kontrolenhed kapitel 7 på side 20, ønskes det at benytte en *μController* til at optage målingerne fra *Momenttransduceren*, er det valgt at finde en *Momenttransducer* med et analogt interface. Dette er valgt da det ikke ønskes at fokusere på, at skulle udvikle en USB-driver til *μControlleren*, men i stedet kan benytte dens indbyggede A/D-konverter.

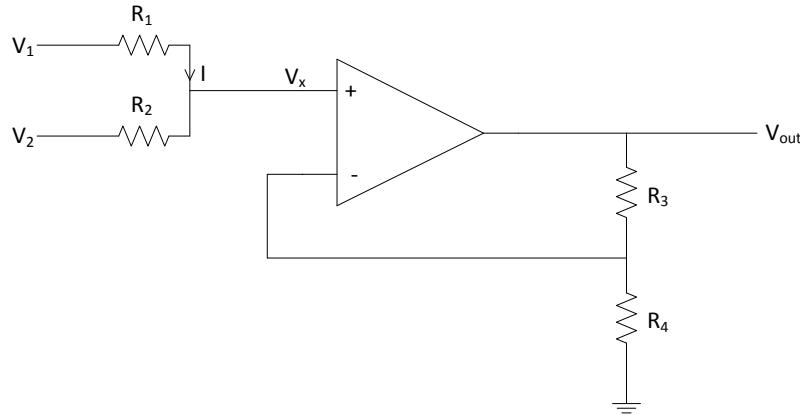
Efter analyse af forskellige typer *Momenttransducere*, valgte gruppen en *Momenttransducer* fra samme fabrikant, med samme måleområde og samme fysiske størrelse, som den foreslæde momenttransducer. Ved at vælge denne *Momenttransducer* undgås desuden konstruktionsændringer på Teststanden.

8.2.2 Design

Med den valgte Momenttransducer måles den mekaniske effekt, ved at måle momentet på akslen og dennes rotationshastighed, da den mekaniske effekt er givet ved $P_m = T_{aksel} \cdot \omega_{aksel}$. Outputsignalet for momentet er en analog spænding, hvor spændingen er proportional med momentet. Signalet for hastigheden er et firkantsignal, hvis frekvens er proportional med akselhastigheden.

Momenttransduceren kan måle momentet på akslen i begge omløbsretninger, hvor outputspændingen i den ene omløbsretning er negativ og i den anden omløbsretning positiv. Da $\mu Controlleren$ kun kan arbejde med positive spændinger, er der lavet et *Signalomsætningskredsløb*, for at sikre at spændingerne til $\mu Controlleren$ er positive. På denne måde er det uden betydning hvilken retning Teststandens aksel drives i.

Princippet for *Signalomsætningskredsløbet* er en ikke-inverterende adder som vist på figur 8.2 [10]. Kredsløbets funktion er, at lægge et offset til momentsignalet for derved at løfte signalet op i det positive område, så det kan benyttes med $\mu Controlleren$.

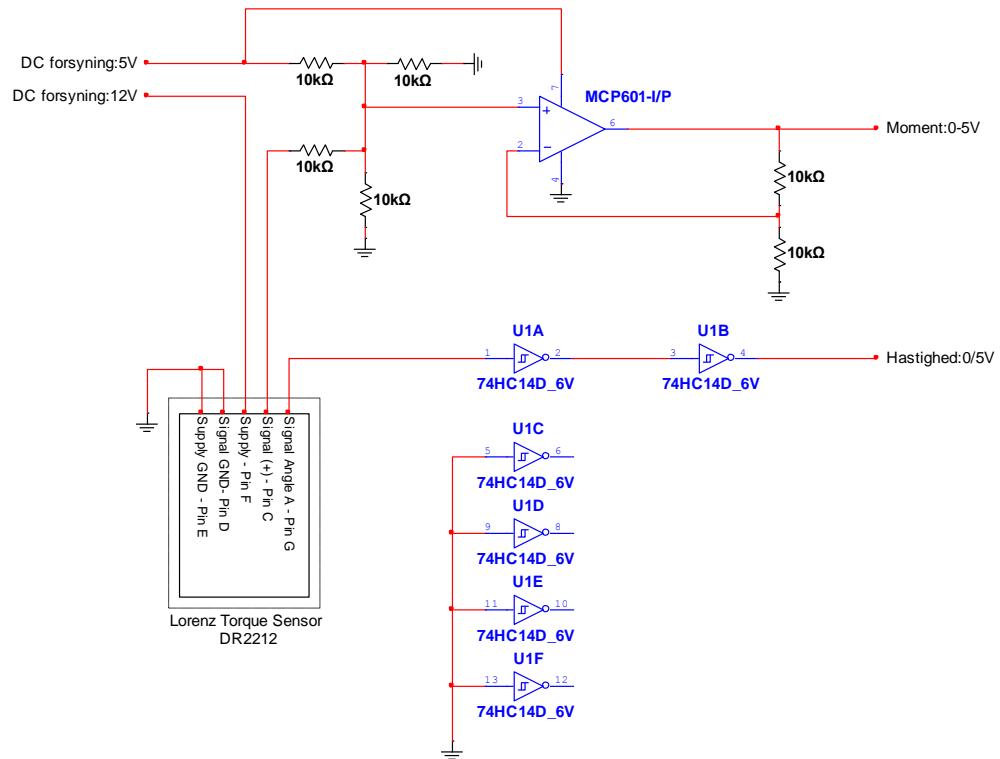


Figur 8.2. Princip af ikke-inverterende adder

Som beskrevet er signalet for hastigheden et firkantsignal, hvis frekvens er proportional med akselhastigheden. Spændingsniveauet på firkantsignalet stemmer overens med spændingsniveauet på $\mu Controlleren$, og derfor startede projektgruppen med at koble dette signal direkte på $\mu Controllerens$ indgang. Under de indledende tests kunne det observeres, at signalet ikke blev målt korrekt af $\mu Controlleren$, som målte en højere frekvens end det reelle hastighedssignal. Det blev observeret, at der var støj overlejret på hastighedssignalet, og der konkluderedes at dette var grunden til fejlmålingen. For at imødekommne dette problem, er der tilføjet en Schmitt-trigger, for at indføre en hysterese. Dette løste problemet, og ved måling af hastighedssignalet, kunne det ses at støjen var fjernet.

8.2.3 Implementering

På figur 8.3 ses det færdige design af *Momenttransduceren*. Figuren indeholder en oversigt over forbindelserne til *Momenttransduceren* samt de to kredsløb, som er beskrevet i de forgående sektioner, til hhv. signalomsætningen af momentsignalet, og fjernelse af støj på hastighedssignalet.



Figur 8.3. Implementering af Momenttransducer

Resultater 9

Dette kapitel beskriver de opnåede resultater i forbindelse med den udviklede prototype. Resultaterne vil tage udgangspunkt i afsnittet *Integrationstest*, der forefindes i Dokumentationen.

9.1 Opsummering af integrationstest

Integrationstesten for projektet er udført og gennemgået af gruppens medlemmer. Projektet opfylder delvist Kravspecifikationen, hvilket har medført, at Accepttesten ikke er godkendt. Kravene for de enkelte delsystemer forefindes i Dokumentationen i kapitlet Kravspecifikationen.

Gruppen af maskiningeniørstuderende, der var ansvarlige for udviklingen af den mekaniske teststand, var ikke færdige med at implementere teststanden, da projektgruppen nåede til testfasen. Dette har resulteret i, at den mekaniske teststand indeholder momenttransducer og DC-generator, samt bilens elmotor blev derfor simuleret under test.

Resultaterne i de efterfølgende afsnit er test af funktionaliteten af de forskellige delsystemer, inkluderende hardware og software på Kontrolenheden.

9.2 Måling af elektrisk tilført effekt

Effekten måles med effekttransduceren og verificeres ved måling med et volt- og amperemeter. Tabel 9.1 giver et overblik over resultatet af testen.

Funktionalitet	Resultat
Hent dataloggen for den elektriske effekt.	OK
Datalog stemmer overens med målt effekt, inden for angivet præcision	OK

Tabel 9.1. Resultater af målinger af elektrisk tilført effekt

9.3 Måling af mekanisk effekt

Da Momenttransduceren ikke har været til rådighed ved test, er signalerne simuleret med en spændingsforsyning og frekvensgenerator. Resultaterne verificeres ved måling med et voltmeter og et oscilloscope.

Funktionalitet	Resultat
Hent dataloggen for den mekaniske effekt.	OK
Datalog stemmer overens med målt mekanisk effekt, inden for angivet præcision	Ikke OK

Tabel 9.2. Resultater af målinger af mekanisk effekt

9.4 Beskyttelse mod overbelastning med Strømovervågning

En strømforsyning leverer effekten, der afsættes i belastningsmodstanden. Strømmen måles af strømovervågningen og verificeres ved måling med et amperemeter.

Funktionalitet	Resultat
Systemet slår fra ved en strøm over angivet grænse	OK
Systemet er stoppet, og teststatus på brugergrænsfladen er ændret	OK

Tabel 9.3. Resultater af beskyttelse mod overbelastning

9.5 Styring af strøm i Belastningskreds

Dutycyclen på PWM-signalen måles med et oscilloscope og strømmen måles med et amperemeter. Spændingen som strømforsyningen er sat til, er tilsvarende DC-generatorens spænding ved nominel hastighed.

Funktionalitet	Resultat
Kredsløbet påtrykkes et PWM signal med en given dutycycle, og den tilsvarende strøm måles	OK
Dutycyclen på PWM-signalen tredobles, og det forventes at strømmen ligeledes tredobles	OK

Tabel 9.4. Resultater af styring af strømmen i belastningskredsen

9.6 Regulering

Da momenttransduceren ikke har været til rådighed ved test, er momentsignalen simuleret med en spændingsforsyning. Dutycyclen på PWM-signalen måles med et oscilloscope.

Funktionalitet	Resultat
Der indstilles et momentsetpunkt i brugergrænsefladen, og feedbacket fra spændingsforsyningen sættes til nul ⇒ Dutycyclen er 100%	OK
Feedbacket fra spændingsforsyningen øges ⇒ Dutycyclen begynder at falde	OK
Feedbacket justeres, indtil PWM-signalen ikke ændrer sig	OK
Feedbacket fra spændingsforsyningen svarer til momentsetpunktet	Ikke OK

Tabel 9.5. Resultater af reguleringen

9.7 Kommunikation mellem Brugergrænseflade og Kontrolenhed

Funktionalitet	Resultat
Navigation mellem menuerne på Brugergrænsefladen	OK
Sende kommandoer til og modtage beskeder fra Kontrolenheden	Ikke OK

Tabel 9.6. Resultater af kommunikationen

Diskussion & Evaluering

10

I dette kapitel diskutes de opnåede resultater og projektets helhed evalueres. Først diskutes de opnåede resultater og herefter de opnåede erfaringer igennem projektforløbet. Til sidst beskrives det fremtidige arbejde for *Teststanden*.

10.1 Diskussion af opnåede resultater

Formålet med projektet er at udvikle en prototype af en teststand til Team AUs elbil til Eco-Marathon 2016. Jf. projektafgrænsningen er der defineret delsystemer til implementering af prototypen. Disse delsystemer er følgende:

1. Et delsystem til elektrisk belastning
2. Et delsystem til måling
3. Et delsystem til brugerinteraktion
4. Et delsystem til dataopsamling og regulering

10.1.1 Elektrisk belastning

Overordnet set er det lykkedes, at udvikle en *Belastningskreds*, der kan omsætte mekanisk effekt til elektrisk effekt. Det er dog ikke lykkedes at designe kredsen, så den kan afsætte den mængde effekt, der er specificeret for den største *DC-generator*, som kan kobles på systemet. Dette skyldes valget af kondensator i systemet, men det er muligt at udskifte kondensatoren til en model, der kan håndtere en større rippelstrøm.

Det er muligt at regulere mængden af elektrisk effekt i *Belastningskredsen* vha. et PWM-signal. Med resultaterne af de målinger, der er foretaget, kan kredsløbet afsætte den mængde elektrisk effekt, som er specificeret for den indkøbte *DC-generator*. Ved at benytte den indkøbte *DC-generator* skal man være opmærksom på, at *DC-generatoren* kun kortvarigt kan levere den strøm, der kan afsætte den specificerede effekt, da den ellers overbelastes.

Da den valgte kondensator har en stor kapacitetsværdi, har denne en stor indflydelse på systemets dynamik. Dette betyder, at systemet reelt set kun kan simulere kørsel ved en fast hastighed.

Der er implementeret en overvågning af strømmen fra *DC-generatoren*. Med denne overvågning stopper systemet, hvis strømmen fra *DC-generatoren* overskridt en fastsat grænse. Den valgte strømtransducer har en nominel rating, der passer til den indkøbte *DC-generator*. Hvis der benyttes en *DC-generator*, som leverer en større strøm, skal transduceren udskiftes til en model med højere strømrating.

10.1.2 Måling

Systemerne til målingen inkluderer *Effekttransduceren* til måling af elbilens optagne elektriske effekt og *Momenttransduceren* til måling af elbilens afgivne mekaniske effekt.

Effekttransduceren er designet så den opfylder de stillede krav. Med resultaterne af de målinger, der er foretaget, kan *Effekttransduceren* måle den optagne effekt inden for den angivne præcision.

Som beskrevet i kapitlet Udvikling af Sensorer 8 på side 24, kunne designet af spændingssensoren have været forbedret ved at indføre beskyttelse. Denne beskyttelse skulle sørge for, at outputspændingen af sensoren ikke overstiger den maksimal tilladelige indgangsspænding på *Kontrolenheden*.

Momenttransduceren er som beskrevet et indkøbt komponent, hvor måling af moment og hastighed optages af *Kontrolenheden*.

For at sikre at momentsignalet ligger inden for *Kontrolenhedens* arbejdsområde er *Signalomsætningskredsen* udviklet. Med resultaterne af de målinger, der er foretaget under enhedstest, opfylder kredsløbet de stillede krav. Efter de enkelte delsystemer blev sammenkoblet, og der skulle foretages integrationstest, kunne kredsløbet ikke opfylde de stillede krav til præcisionen. Det antages, at dette skyldes induceret støj i kredsløbet. Det blev forsøgt at implementere en kondensator for at eliminere støjen, dog uden succes. Det forventes, at denne problemstilling kan løses ved at implementere kredsløbet på PCB, frem for veroboard.

Kredsløbet til hastigheds måling er designet, så det opfylder de stillede krav. Med resultaterne fra de foretagede målinger, kan hastigheden måles inden for den angivne præcision.

10.1.3 Brugerinteraktion

Brugerinteraktionen indebærer udvikling af en *Brugergrænseflade*. Der er udviklet en *Brugergrænseflade*, som opfylder kravet om brugerinteraktion. *Brugergrænsefladen* er enkel og nem at benytte.

For at kunne kommunikere med *Kontrolenheden* skulle der etableres en kommunikation mellem denne og *Brugergrænsefladen*. Der er implementeret en UART-kommunikation fra *Kontrolenheden* til *Converteren*. *Converteren* er en *RS232/TCP Converter*. For at sende beskeder til, og modtage data fra *Kontrolenheden* er der sideløbende med terminalprogrammet udviklet en *TCP_Client* fil.

Ved enhedstest af *Brugergrænsefladen* og TCP/IP-kommunikationen blev *Kontrolenheden* simuleret med en terminal emulator på en anden computer. Der blev etableret kommunikation mellem enhederne samt sende og modtage beskeder.

Ligeledes blev UART-forbindelsen fra *Kontrolenheden* testet på samme måde, hvor *Brugergrænsefladen* er simuleret ved en terminal emulator. Det var ligeledes muligt at etablere kommunikation mellem enhederne samt sende og modtage beskeder.

Under integrationstesten, hvor *Kontrolenhed* og *Brugergrænseflade* er koblet sammen, kunne det ses, at *Brugergrænsefladen* ikke viste korrekte værdier, når data blev efterspurgt. Da der ikke var en klar idé om, hvor fejlen kunne ligge, var løsningen en delvis afgrænsning af de to delsystemer. Den første antagelse var, at fejlen lå i UART-forbindelsen, idet det er det "svage" led i kommunikationen. Men da dette var testet i enhedstesten, kunne fejlen ikke ligge her. Det samme gælder for TCP/IP-forbindelsen og *Brugergrænsefladen*, som også blev testet under enhedstest. Det må forventes, at fejlen ligger andetsteds, muligvis i opbygningen af *Brugergrænsefladen*. Dette er dog ikke bekræftet, da årsagen til fejlen endnu ikke er fundet.

Da ovennævnte fejl ikke kunne løses, blev integrationstesten gennemført med *Kontrolenheden* tilkoblet en terminal emulator, der simulerer *Brugergrænsefladen*.

10.1.4 Dataopsamling og regulering

Dataopsamling og regulering indeholder måling af indkomne signaler fra effekt- og momenttransducer, processering af de opsamlede data samt design af en regulerinssløjfe. Da responsen på regulerinssløjfen ikke var tilstrækkeligt, blev der designet en tilhørende regulator. Der blev udviklet software, der muliggør dette.

Under enhedstest, blev dataopsamlingen på de indkomne signaler simuleret med en frekvensgenerator. Det kunne ses at signalerne blev målt og processeret korrekt.

Enhedstest af reguleringssløjfen blev simuleret med et signal fra en frekvensgenerator samt setpunkt sat i programmet. Det var muligt at foretage regulering, der kunne verificeres på PWM-outputtet af reguleringssløjfen.

Under integrationstest hvor de enkelte delsystemer er sammenkoblet, kunne det ses, at der var uoverensstemmelse i målingen af momentet. Af samme grund som er nævnt i afsnit 10.1.2, kunne måling af momentet ikke opfylde de stillede krav til præcision.

10.2 Opnåede erfaringer

I dette afsnit beskrives de erfaringer, der er gjort under projektforløbet. De største erfaringer, der er gjort er følgende:

1. Iterativ arbejds metode
2. Udvikling med effektelektronik
3. Design af Reguleringssløjfe
4. Opbygning af et kontrolsystem på en microcontroller

Iterativ arbejds metode

Gennem projektforløbet er der blevet arbejdet iterativt med de forskellige delsystemer. Projektgruppen har tidligere benyttet en iterativt arbejdsform, men det er første gang på et projekt i denne skala. Det har fungeret godt, at det har været muligt at tage et skridt tilbage i designprocessen og kunne gentanke en løsning. Et eksempel på dette er udviklingen af metoden til at måle og beregne akselhastigheden. Det første design der blev implementeret benyttede to interrupt-rutiner. På grund af timingen mellem interrupt-rutiner blev hastighed ikke beregnet korrekt. Metoden blev derfor redesignet, hvorefter hastigheden kunne beregnes korrekt.

Udvikling med effektelektronik

Der er opnået nogen erfaring med at benytte effektelektronik i et system. Under udviklingen er der blevet trukket på erfaringer med almindelige elektronikkomponenter. Dog har det været nødvendig at sætte sig ind i, hvordan effektelektronikken adskiller sig fra traditionel elektronik. Et eksempel er brugen af MOSFET'en i *Belastningskredsen*. Det blev erfaret, at der skulle benyttes en Driverkreds, der kunne få MOSFET'en til at lede strømmen hurtig nok, for at undgå for stor effektafsættelse i denne.

Design af Reguleringssløjfe

Til at regulere momentet på systemets aksel er der blevet udviklet en regulerinssløjfe. Teorien omkring reguleringen er kendt fra kurset *Introduktion til Regulerings teknik*. Men der er opnået erfaring med at analysere et elektro-mekanisk system og opstille en overføringsfunktion for det, for derefter at designe en regulator og implementere den på en μ Controller.

Opbygning af et kontrolsystem på en microcontroller

Der er opnået en del erfaringer med at designe og implementere et kontrolsystem på den valgte μ Controller. Da der er stor forskel på opbygningen og brugen af forskellige typer μ Controllere, er den opnåede erfaring specifik for den valgte μ Controllere. Dette inkludere sampling, dataprocessering, kommunikation, sekventiel styring ved brug af interruptrutiner og meget andet.

10.3 Fremtidigt arbejde

I forlængelse af projektet er der flere muligheder for, hvordan produktet kan blive videreudvikles.

Computeren kunne erstattes med en Raspberry Pi (RPi) eller anden form for computere med μ Processor-funktioner. En μ Processor er både stærkere og hurtigere end en μ Controller. Ydermere er det muligt at føre dataprocessering på μ Processoreren, hvilket muliggør hel eller delvis erstatning af *Kontrolenheden*. Hvis ikke det er muligt at erstatte *Kontrolenheden*, kan der i stedet investeres i en billigere μ Controller, idet μ Processoren foretager den tunge del af arbejdet.

Datalogningen kunne udvides, så man anvender μ Processorens USB-port til at gemme data på en ekstern enhed. En anden mulighed er, at gemme data på nettet for eksempel i Dropbox, hvilket muliggør fleksibilitet i forhold til datadistribution. Dette vil dog kræve, at μ Processoren har en internetforbindelse.

Det fremtidige arbejde med *Brugergrænsefladen* inkluderer implementering af en grafisk *Brugergrænseflade* som erstatning for terminalprogrammet. Derudover skal der arbejdes med optimering af *Brugergrænsefladen*, så den er intuitiv og har et professionelt udseende.

Hardwaren i systemet kunne implementeres på PCB'er, således kredsløbene er mere robuste over for støj. Ydermere kunne systemet monteres i monteringskasser, så de er afskærmet korrekt og kan tåle vind, vejr og stød.

For at nedbringe strømmen i belastningskredsen kunne *DC-generatoren* erstattes med en anden *DC-generator*, der kan tåle 500 W samt en højere spænding. Ydermere kunne der laves et redesign af *Belastningskredsen*, så den ikke påvirker systemet under acceleration og deceleration.

Konklusion 11

Det udviklede system, *Teststand til Børsteløs DC-motor*, resulterede i udvikling af fire delsystemer: *Belastningskreds*, *Brugergrænseflade & Kommunikation*, *Kontrolenhed* og *Sensorer*.

Belastningskredsen resulterede i en prototype, der kan afsætte den effekt som er specifiseret for den indkøbte *DC-generator*. Prototypen opfylder kun delvist de stillede krav, grundet design og valget af komponenter.

Brugergrænsefladen og *Kommunikationen* resulterede i et fungerende terminalprogram med den ønskede funktionalitet. Det lykkedes dog ikke at få Brugergrænsefladen til at kommunikere med *Kontrolenheden*.

Kontrolenheden resulterede i udvikling af software, der fungerer på en *PSoC5-enhed*, som anvendes til at sende og modtage beskeder, processere samplet data samt regulere *Belastningskredsen*.

Sensorer resulterede i fungerende målekredsløb, der opfylder de stillede krav om måling.

I problemformuleringen blev der stillet følgende hovedspørgsmål:

Hvordan vil et system kunne designes, hvor der ved test på elbilens drivlinje, kan optages målinger således at motorens optimale driftsbetingelser findes? Kan systemet desuden designes, så man kan belaste elmotoren for at simulere den belastning, som bilen vil udsættes for under kørsel?

Baseret på resultatet af det udviklede produkt, *Teststand til Børsteløs DC-motor*, kan det konkluderes, at det er muligt at udvikle et system der muliggør testning af motorens optimale driftsbetingelser. Det er dog kun delvist lykkedes at udvikle et system, der kan simulere den belastning, som bilen vil udsættes for under kørsel, da systemet forudsætter, at bilens hastighed er i stationær tilstand.

Referencer

Litteratur

1. J. E. Carryer, R. M. Ohline & T. W. Kenny · *Introduction to Mechatronic Design* · 1st edition · Pearson 2011 · ISBN: 9780136095217
2. H. M. Deitel · *C How to Program* · 6th edition · Pearson 2010 · ISBN: 9780137059669
3. James F. Kurose & Keith W. Ross · *Computer Networking: A Top-Down Approach* · 6th edition · Pearson 2013 · ISBN: 9780273768968
4. Ned Mohan, Tore M. Undeland & William P. Robins · *Power Electronics: Converters, Applications and Design* · 3rd edition · Wiley 2003 · ISBN: 9780471226932
5. Sarma Mulukutla · *Electric Machines* · 2nd edition · Cengage 1994 · ISBN: 9780534938437
6. Norman S. Nise · *Control Systems Engineering* · 6th edition · Wiley 2011 · ISBN: 9780470646120
7. Nils Leth · *Servomekanismer* · Servolaboratoriet 1978 · ISBN: 8787950111
8. *Linux Shell Scripting Tutorial v1.05r3 - A Beginner's handbook* · 2002 · [Online] Tilgængelig fra: <http://www.freeos.com/guides/lsst/> · Besøgt: 7. november 2015.
9. Jacob Nordfalk · *Objektorienteret programmering i Java* · 5. udg. · Forlaget Globe 2014 · ISBN: 9788779006973
10. B. Basavaraj · *Basic Electronics* · 2nd edition · Vikas Publishing House 2009 · ISBN: 9788125916628