

INGENIØRHØJSKOLEN ÅRHUS

ELEKTRO-INGENIØR LINIEN

SEMESTERPROJEKT E4PRJ4

Bias Reducing Operating System

Skrevet af:

Nicolai GLUD

Studienummer: 11102

Johnny KRISTENSEN

Studienummer: 10734

Rasmus LUND-JENSEN

Studienummer: 11111

Mick HOLMARK

Studienummer: 11065

Jacob ROESEN

Studienummer: 10095

Vejleder:

Carl Jakobsen



13. december 2012

Resume 1

Abstract 2

Indholdsfortegnelse

Kapitel 1	Resume	3
Kapitel 2	Abstract	5
Kapitel 3	Forord	9
Kapitel 4	Indledning	11
4.1	Læsevejledning	11
Kapitel 5	Opgaveformulering	13
Kapitel 6	Projektformulering	15
Kapitel 7	Systembeskrivelse	17
Kapitel 8	Kravspecifikation	19
Kapitel 9	Afgrænsning	21
Kapitel 10	Projektbeskrivelse	23
10.1	Projektgennemførelse	23
10.1.1	Rollefordelinger	23
10.2	Metoder	23
10.2.1	SCRUM	24
10.2.2	V-model	25
10.3	Analyse	25
10.3.1	Hvordan måler vi hældning?	25
10.4	Systemarkitektur	25
10.4.1	Systemkomponenter	25
10.5	Design og Implementering	26
10.5.1	VBTE	26
10.5.2	SM	27
10.5.3	Kontrolinterfacet	29
10.6	Design og Implementering	31
10.6.1	Serveren	32
10.7	Resultater	32
10.8	Opnåede erfaringer	32
10.8.1	Generelt	32
10.8.2	Udvikling af hældningssensor	33
10.8.3	Metoder	33
Kapitel 11	Konklusion	35

Kapitel 12 Referencer	37
12.1 Artefakter	37
12.1.1 Kravspecifikation	37
12.1.2 Accepttestspecifikation	37
12.1.3 Systemarkitektur	37
12.1.4 Integrationstestspecifikation	37
12.1.5 Detaljeret design	37
12.1.6 Enhedstestspecifikation	37
12.2 Hjemmesider	38
12.3 Liste over bilag på CD	38
12.3.1 Kode	38
12.3.2 Dokumentation	38
12.3.3 Datablade	38
12.3.4 Billeder	38

Forord 3

Denne rapport er udarbejdet af gruppe 3, gennem 4. semester på elektroingeniør studiet ved IHA 2012. Rapporten gennemgår overordnet gruppens besvarelse og gennemførelse af projektet. For yderligere detaljer henvises der til projektdokumentationen. Rapporten er skrevet med henblik på at læseren er af samme faglige niveau som gruppen, derfor tages visse ting som givet. Tak til undervisere og vejledere.

Indledning 4

Denne rapport omhandler 4. semesters projekt. Projektets emne er "slagsideregulering af bulkskib", som er et selvvalgt emne. Rapporten beskriver processen af projektet herunder hvordan forløbet har været, hvilke metoder der er anvendt og hvilke overvejelser der ligger til grund for de valgte løsninger. I forbindelse med, at de valgte løsninger bliver beskrevet vil der også blive fremlagt alternativer og begrundelser for at de ikke blev valgt.

Der har været stillet enkelte krav til projektet og det system der skulle udvikles. Nogle af disse krav afspejler derfor også hvordan arbejdsprocessen har været og ikke mindst nogle af de komponenter som igår i det færdige system.

Formålet med projektet er at anvende de teorier og metoder, som er blevet tilegnet gennem studiet, og ikke mindst tilegne sig ny viden på egen hånd, for at fuldføre gennemførelsen af et komplet projektforløb.

Projektet har været inddelt i et antal udviklingsfaser. Udviklingsfaserne er som følger:

- Analysefase
- Struktureringsfase
- Designfase
- Implementeringsfase

4.1 Læsevejledning

Rapportens opbygning er struktureret således at den giver den bedste gennemgang af hele projektforløbet. Rapporten er i hovedtræk delt op i 2 dele. De første afsnit beskriver det overordnede system og projekt. Tilblivelsen af systemet og projektet med tilhørende overvejelser beskrives i de efterfølgende afsnit. Denne adskillelse sker mellem afsnit 9 og 10. Alle afsnit er skrevet så de som udgangspunkt godt kan stå alene, hvorfor der igennem rapporten vil komme gentagelser hvis man læser denne fortløbende. Rapportens egentlige indhold begynder fra afsnit 6 – opgaveformulering. Her gennemgås opgaveformuleringen givet fra vejledere til gruppen, som indeholder minimumskrav til projektet. I projektformuleringen bliver der defineret præcist hvad dette projekt kommer til at dreje sig om, og hvordan gruppen har formuleret dette. Herefter følger en beskrivelse af det samlede, tænkte system. I afsnit 8 – krav, fremlægges kravene der fra gruppen er stillet til projektet. Herefter beskrives projektafgrænsningen samt arbejdsmetoder og fremgangsmåde i afsnit 8 og 10.1. Afsnit 10.2 beskriver det analyse arbejde projektet har gennemgået. I afsnittene fra 10.2-10.5 nedbrydes hele projektet fra øverste abstraktion og ned til implementering. Der er her gået i dybden med de vigtige aspekter i forhold til dette projekt. Disse afsnit har samtidig også en naturlig overgang til hinanden ud fra

systemarkitekturen. Hernæst samles der op på de opnåede resultater i afsnit 10.6. Efter resultaterne er præsenteret, fremlægges de erfaringer gruppen har opnået igennem hele projektforløbet, samt hvilke ting der har fungeret godt. I afsnit 10.8 snakkes der ganske kort om de anvendte udviklingsværktøjer. Afslutningsvist konkluderes der på hele projektet på godt og ondt i afsnit 11. Det anbefales dog at læse rapporten fortløbende for at få den bedste samlede forståelse for projektet og produktet.¹

¹FiXme Note: Lave refrencerne til de enkelte afsnit.

Opgaveformulering 5

Gruppen skal fremstille et system der overholder nogle krav. Kravene er at systemet skal kunne interagere med omverdenen vha. af sensorer og aktuatorer. Endvidere skal der også anvendes faglige elementer fra fjerde semesterets kurser. Transmission af data mellem enheder i projektet skal være pålidelig. Til slut skal systemet indeholde brugerinteraktion.

Projektformulering 6

Gruppen har valgt at arbejde med lastning og losning af et bulkskib. Vi vil gerne lave et system der aflaster skibets ansvarshavende officer, som står på broen under hele lastningen/losningen. Hans job er at holde øje med at hele operationen bliver gjort ordentligt. Denne opgave vil vi gerne gøre nemmere. Med et system der automatisk sørger for at skibet altid er i vatter skal kaptajnen kunne interagere med systemet hvis systemet kommer med en alarm. Kaptajnen kan manuelt vælge at flytte ballast til den ene side, hvis han ved at der kommer en række tunge containere, der skal stå i modsatte side. For at sikre at alle skibe i havnen er i vatter, sender systemet statusbeskeder til en database på havnekontoret. Kontoret kan derfor sende bemanning eller tage kontakt til skibet, der har en alarm.

Systembeskrivelse 7

BROS er et sikkerhedssystem til skibe. Systemet tages i brug ved lastning eller losning. Her er det systemets opgave at sørge for at skibet ikke får slagsside - heraf navnet: Bias Reducing Operating System (Slagsidereducerende Operativt System). I systemet er der indbygget en hældningssensor og to vandballasttanke - en i hver side af skibet. På baggrund af målinger fra hældningssensoren vil indholdet af tankene blive justeret således at der korrigeres for en slagsside af skibet.

Hele systemet styres fra Skibsførens kontor hvor et grafisk brugerinterface er installeret. Her kan der aflæses skibets hældning af skibet, vandindholdet af tankene og statusmeldinger for systemet. Som udgangspunkt vil systemet automatisk opretholde en hældning på nul grader, men hvis man ønsker det kan man her manuelt give skibet en mindre slagsside. Dette kan gøres for at imødekomme en større slagsside til modsatte side påført af en forestående ændring i skibets last.

For at indsætte et ekstra sikkerhedselement vil systemet under hele processen løbende sende værdier for systemet til en ekstern database. Dermed kan en repræsentant fra terminalen følge skibets status.

Kravspecifikation 8

Afgrænsning 9

En oversigt over de afgrænsninger dette projekt er udarbejdet med.

Afgrænsninger sat på forhånd:

- a Systemet skal interagere med omverdenen vha. sensorer og aktuatorer.
- b Systemet skal omfatte pålidelig datatransmission.
- c Systemet skal have brugerinteraktion.

Afgrænsninger sat af gruppen selv:

Som en del af dette projekt har vi opdelt komponenter i to grupper: Grundsystem og udvidelser. Grundsystemet er det basale system med de funktionalitet der skaber et færdigt produkt. Udvidelser er funktionaliter der kunne være rare at have eller ville kunne øge værdien af produktet. Disse kan dog også være funktionalitet som kunden ikke har gavn af og derfor er ubrugelige. Grundsystem og udvidelser ses i *Tabel 9.1*

Grundsystem:	Elektronisk måling af hældning Automatisk regulering af niveau i ballasttanke Niveaumåling i ballasttanke Mulighed for brugerinteraktion Advarselssignaler
Udvidelser:	Måling af afstand til terminal-kaj Måling af dybdegang Manuel styring af ballast niveau Pålidelig kommunikation med ekstern enhed

Tabel 9.1. Grundsystem og Udvidelser til BROS

10.1 Projektgennemførelse

Projektet er udført af en gruppe på fem personer. Gruppen er en fortsættelse fra et tidligere projektforsøg. Dette har givet en fordel i kommunikation og samarbejde. Gruppen valgte fra start at dele ansvar ud til personer med interesse for ansvarsområdet. Gruppen har dog stadigvæk, som følge af det velfungerende samarbejde, i store træk været fælles om opgaverne i projektet. Tidsplanen for projektet blev udarbejdet i den første fase og er blevet overvåget siden. Gruppen var opmærksom på at være realistisk frem for optimistisk. Det har derfor været nødvendigt senere hen at revurdere tidsplanen i forhold til længden af faserne. Projektets overordnede tidsplan for faserne og de eksterne milestones ligger som bilag.

10.1.1 Rollefordelinger

Projektleder:	Jacob Roesen
Projektkoordinator:	Nicolai Glud Jacob Roesen
Scrummaster:	Johnny Kristensen

Tabel 10.1. Tabel over rollefordelinger

Vi har valgt at lave roller ud fra vores udviklingsmetode, SCRUM, der er beskrevet senere. Projektlederen har haft som ansvar at strukturere arbejds og scrummøder. Projektkoordinatorens ansvar har ligget i at planlægge møder og bestille lokaler. Scrummasteren er ansvarlig for udviklingsplatformen, SCRUM, og sørge for at metoden anvendes mest optimalt.

10.2 Metoder

En kort præsentation af de to mest dominerende arbejdsmetoder der er anvendt.

I dette projekt er der anvendt metoder indlært gennem et tidligere projekt. Værktøjerne er de værktøjer som gruppen føler sig trygge ved og som gruppen føler bidrager mest til processen.

10.2.1 SCRUM

I gruppens implementering af SCRUM startes der med at lave en produktbacklog, som er den kunden ser. Derefter planlægges det første sprint. Et sprint spænder over 2 uger. Når et sprint starter bliver opgaver overført fra backloggen til sprintet. Når nye opgaver bliver sat på sprintet bliver opgaven vurderet for hvor stort et omfang den har. Derefter diskuteres der hvilke opgaver de forskellige dele af gruppen skal lave. En gang om ugen laver man et SCRUM-Meeting. Her bliver fulgt op på opgaver lavet i løbet af ugen samt tilføjelse af nye opgaver. Alle de opgaver der ikke er færdige, når sprintet er slut, overføres til næste sprint.

I projektet er der i alt 7 sprint. På *Figur 10.1* vises det 2. sprint.

Sprint nr:	Scrummaster:	JK	Koordinator:	NG		
2	Projektleder:	JR				
Overordnet opgave:	Opgaver:	Arbejdsvægtning:	Resource #1	Resource #2	Process (% done):	Kommentar:
	Videreført: Jura	3	MH	RLJ	100%	
	Videreført: Snak m. Arne vedr. sensor	1	NG	JK	100%	NG, JK
Videreført Accepttest		2	RLJ	NG	100%	NG, JR, JK, MH
Systemarkitektur						
	Systemkomponenter - Enheder	2	JR	RLJ	80%	JR, JK
	Komponentvalg	3	ALLE		100%	RLJ: er afsnittet skrevet og læst af alle?
	Strukturel systemarkitektur	4	MH	JK	60%	MH: Tilrettelse af tegning, tror vi skal sætte os ned os diskutere, flere ting
	Behavior systemarkitektur	5	JK	MH	60%	JK, NG
	HW Systemarkitektur	0	JR	JK	20%	
	Grænseflader / interface	3	NG	JR	80%	NG, JK
	Forside	1	RLJ		25%	
Teknologiundersøgelse		5	ALLE		100%	
Integrationstest		3	NG		80%	NG, JK
	Total:	32				
				Procentvis færdig:	81.25 %	

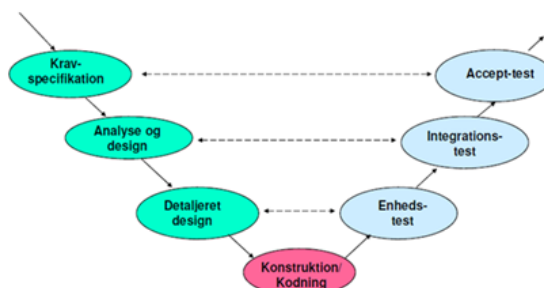
Figur 10.1. Sprint 2

Sprintet er afsluttet og man kan se hvordan nogle opgaver er færdige og hvordan resten skal overføres til næste sprint. Billedet illustrerer også hvordan planlægningen er opbygget. Forklaring af statusfarver er vist på *Figur 10.2*

0%	Ikke startet
20%	Påbegyndt
40%	Godt i gang
60%	Umiddelbart færdigt. Mangler 2. opinion
80%	Mangler gennemlæsning af alle
100%	Gennemlæst af alle

Figur 10.2. Forklaring af sprint points

10.2.2 V-model



Figur 10.3. V modellen

Vi har valgt at anvende V modellen som udviklingsmodel. Dette muliggør iterative processer hvilket er optimalt for vores udviklingstil.

10.3 Analyse

10.3.1 Hvordan måler vi hældning?

Før vi kunne begynde på at lave en niveau sensor blev vi nødt til at finde ud af hvilke muligheder der var niveaumåling. En af mulighederne var at anvende et pendul eller en libelle. Disse løsninger blev undersøgt og en redegørelse er findes i afsnit 10.8.

Valget faldt på at anvende et accelerometer. Det var en fordel af at anvende det accelerometer, der er monteret på PSoC'en.

10.4 Systemarkitektur

Dette afsnit beskriver systemarkitekturen for for projektet "BROS" som formuleret i projektbeskrivelsen og specificeret i kravspecifikationen. Afsnittet indeholder beskrivelse af systemkomponenter, systemarkitektur, SW-komponenter, HW-komponenter og interfaces, i den givende rækkefølge.

10.4.1 Systemkomponenter

Ud fra kravspecifikationen er der udvalgt disse beslutninger om komponenter til systemet og deres placering. Der refereres derfor til kravspecifikationen, ikke-funktionelle krav og krav generelt.

Brugeren integrere med systemet igennem KI. KI er styringsmodul for hele systemet. På KI har brugeren mulighed for at til- og frakoble systemet, justere den ønskede hældning på skibet. KI giver mulighed for at brugeren kan aflæse handlinger foretaget i systemet samtidig med at denne modtager advarsler i tilfælde at hældningen bliver for stor eller vandbalast tanke bliver overfyldte.

KI kommunikere til SM modulet igennem en uart. for at denne kommunikation kan foregå er der lavet en protocol for denne kommunikation. Kommunikationsformen er ved I₂C. SM står for at måle skibets hældning og sende denne til KI. KI kan så informere dette

til brugeren. Når SM har målt hældningen på skibet giver denne besked til VBTE1 og VBTE2 om at åbne og lukke for ventilerne til tankene. VBTE1 og VBTE2 styres fra en PSoC. VBTE1 og VBTE2 er placeret på være deres tank. For at kunne kontrollere hvor meget vand der er i tankene dette gøres ved hjælp af to ultra lydssensore som hele tiden måler og videregiver denne information til SM som så sender dette videre til KI der kan advare om vandstanden i tankene i tilfælde af at systemet af sat på manuel styring. KI sender data om skibet til databasen som lagre disse data i en mySQL database som så kan tilgås via et web interface.

10.5 Design og Implementering

10.5.1 VBTE

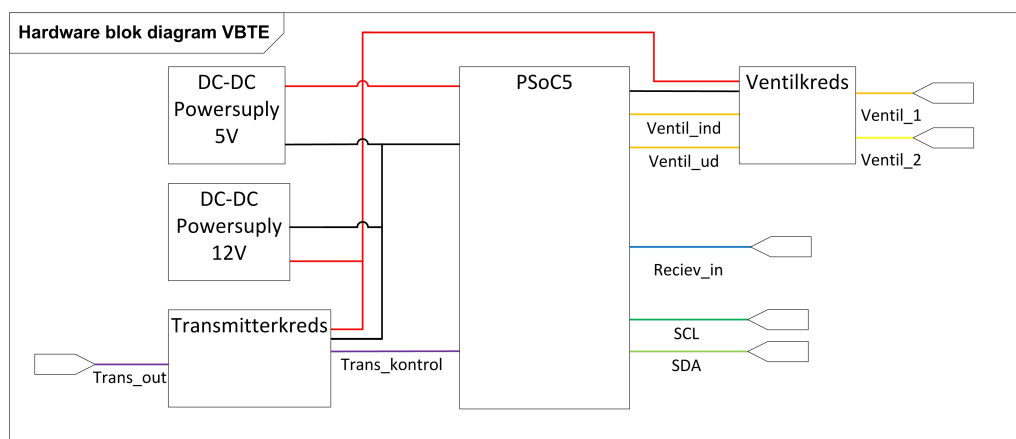
I dette afsnit beskrives design og implementering af VBTE for både software og hardware.

Hardware

Til VBTE'en er der blevet designet hardware til at styre de to ventiler samt den keramiske ultralydstransmitter og receiver. Hardware er blevet udfærdiget i to dele. Den ene er programmeret hardware på PSoC'en, den anden er hardware uden for PSoC'en. Hardware designprocessen til VBTE'en gik igennem 3 faser:

1. Overordnet design
2. Nedbrydning af blokke
3. Opbygning af design

Gennem disse faser er designet blevet udfærdiget. Fremgangsmåden er anvendt for at overskueliggøre systemet og lette arbejdet ved at dele systemet op i små dele. På *figur 10.4* ses det overordnede design af VBTE'en. Der vil i rapporten tages udgangspunkt i ventilkredsen samt receiverdriveren på PSoC'en.

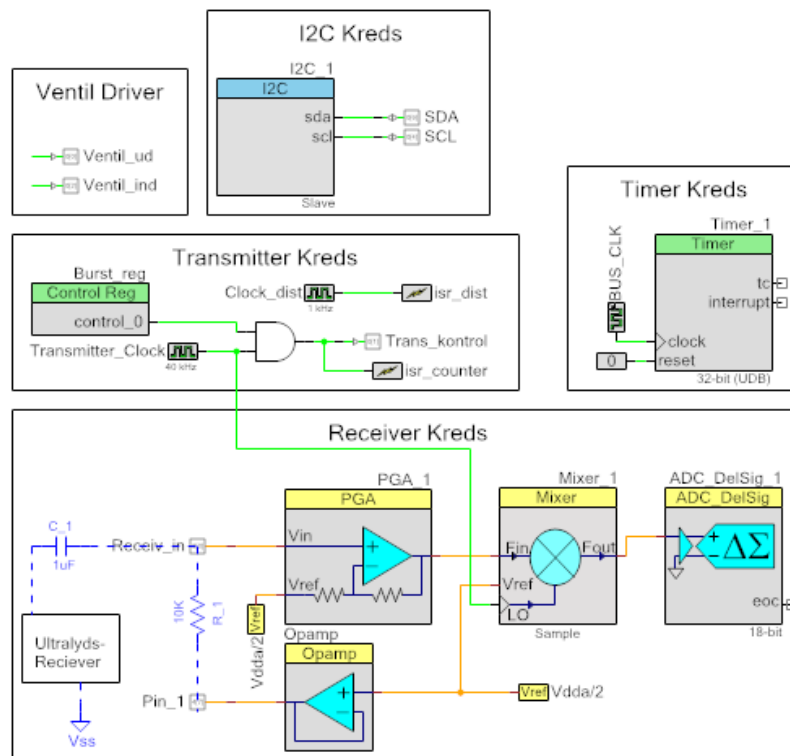


Figur 10.4. Illustrering af overordnet design af hardware på VBTE.

Hardware på PSoC'en

Hardware opbygget på PSoC'en kunne uden problemer være blevet designet uden for PSoC'en, men PSoC miljøet gør det meget nemt at arbejde med mange forskellige

elementer. På figur 10.4 ses hardware designet på PSoC'en.



Figur 10.5. Hardware på PSoC'en.

Software

10.5.2 SM

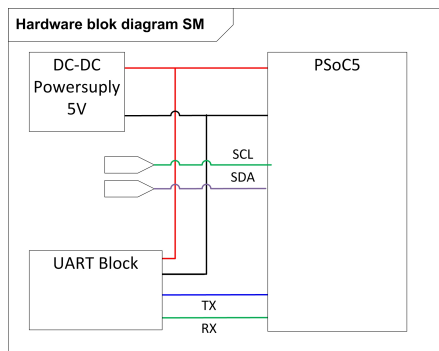
I dette afsnit beskrives design og implementering af SM modulet. SM modulet består af både software og hardware.

Hardware

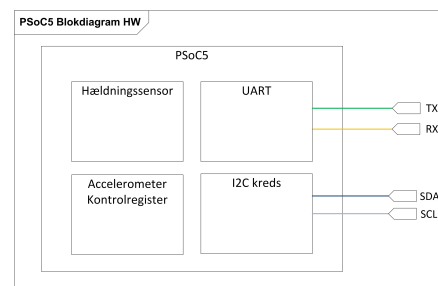
SM modulets hardware består af en konverteringskreds og en PSoC. På PSoC'en er monteret et Kionix KXSC7-2050 accelerometer. Konverteringskredsen anvendes til at sende UART signaler fra PSoC til en KI modulet. Accelerometerets x-akse anvendes til hældningsmålinger for hældningssensorblokken. Designfasen til SM er delt op i 3 faser:

1. Overordnet design
2. Nedbrydning af blokke
3. Opbygning af design

Denne fremgangsmåde gør det muligt for en udefrakommende at følge med i processen og at kunne implementere modulet så det overholder de krav der er stillet. Ligeledes gør fremgangsmåden det lettere at overskue flere løsninger til hvert problem. på *Figur 10.6* ses det Overordnede design og på *Figur 10.7* ses PSoC blokken i SM. Der bliver efterfølgende taget udgangspunkt i Hældningssensoren.

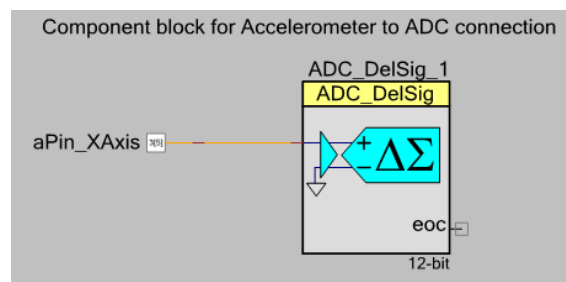


Figur 10.6. Hardware blok for SM



Figur 10.7. PSoC blok for sm

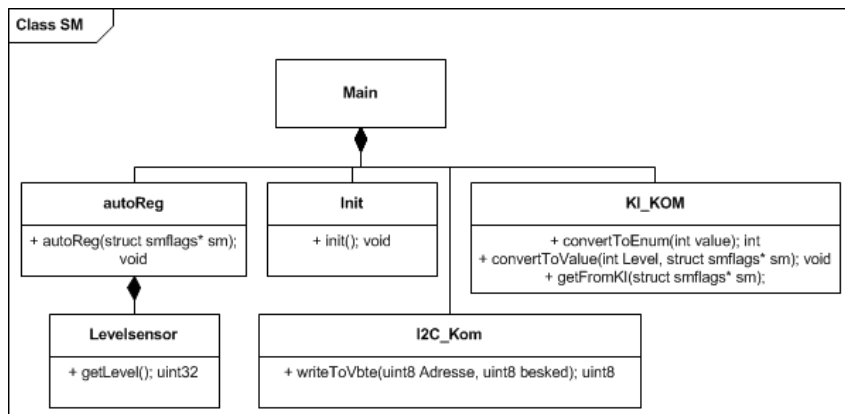
Hældningssensoren består af 2 komponenter: det førnævnte accelerometer samt en DelSig ADC internt i PSoC'en. Valget af accelerometer kommer af at have lavet en række prototyper der ikke mødte vores krav, hvilket accelerometeret i PSoC'en gjorde. Valget af ADC faldt på en DelSig da, den er meget støj immun grundet det indbyggede lavpas filter og har en høj opløsning. Valgte komponenter er illustreret på *Figur 10.8*. ADC konverterer det analoge signal fra, en pin forbundet til, accelerometer til en digital værdi der så senere bliver anvendt i softwaren. For ADC og accelerometerets opsætning se da afsnit 12.1.5 *Detaljeret hardware design* i Bilag.



Figur 10.8. Hældningssensorens implementering

Software

SM software behandler den konverterede data fra ADC'en samt kommunikation med VBTE og KI modulerne. Softwarens udvikling har fulgt de samme faser som hardwaren, hvilket har ført til letforståelig og læselig kode. Softwaren er illustreret på *Figur 10.9*. Der vil efterfølgende blive taget udgangspunkt i funktionerne `autoReg` og `getFromKI`.



Figur 10.9. Klassediagram for SM

getFromKI

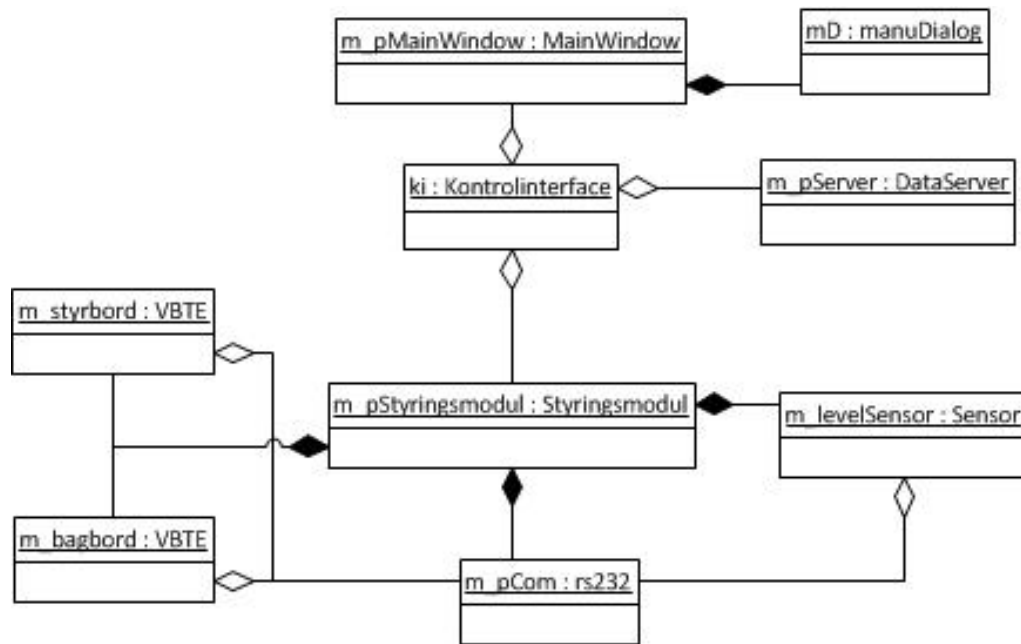
Funktionen er bedst beskrevet med et flowdiagram:

10.5.3 Kontrolinterfacet

I dette afsnit beskrives design og implementering af Kontrolinterfacet som lavet på baggrund af kravspecifikationen og systemarkitekturen.

Designet af Kontrolinterfacet afspejler meget den generelle opbygning af systemet. Således er hvert element af systemet implementeret som en klasse. Derudover er der nogle hjælpeklasser. En oversigt over klasserne og deres ansvar kan ses i tabel ??.

Det gælder for VBTE-, SM- og Sensor-klasserne at når der efterspørges en af de værdier, klassen har ansvaret for, så benyttes RS232-objektet til at fremskaffe disse værdier ved hjælp af den serielle kommunikationsprotokol.



Figur 10.10. Oversigt over, og sammenhæng mellem, objekter i kontrolinterface-programmet

Kontrolinterfacets klasser

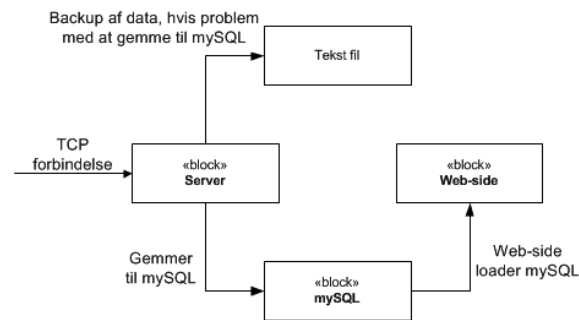
Kontrolinterface (KI)	Programmets hovedklasse. Eksisterer for at rydde op i main-funktionen.
DataSet (DS)	Står for alt TCP-kommunikationen med databasen. Oprettet af KI-klassen
Styringsmodul (SM)	Oprettet af KI og opretter VBTE-, Sensor- og RS232klasserne.
Sensor	Oprettet af SM og er ansvarlig for hældningsværdien. Når objektet bliver spurgt til den værdi af SM-klassen vil Sensor-klassen anvende sin delte association til RS232-objektet til at få denne af det fysiske SM-modul.
VBTE	Der eksisterer et VBTE-objekt for hvert fysisk VBTE-modul. Det er objektets ansvar at holde styr på værdierne for sit VBTE-modul.
RS232	Driveren til kommunikationen med det fysiske SM-modul. Objektet formidler sig på en protokol forstået af det tilsvarende objekt på SM-modulet. Objektet oprettes af SM-klassen og VBTE- og Sensorobjekterne har en delt association til den.
MainWindow	Oprettet af KI-klassen og kontrollerer og overvåger den grafiske brugergrænseflade.
manudialog	Oprettet af MainWindow og styrer den dialog, der fremkommer når man ønsker en manuel hældningsregulering.

Tabel 10.2. Kontrolinterfacets klasser

10.6 Design og Implementering

Denne section beskriver design og implementering af database delen som er lavet på baggrund af kravspecifikationen og systemarkitekturen.

Databasen er blevet opdelt i 2 dele som håndterer denne side. Databasen er opdelt i serveren og web-siden.



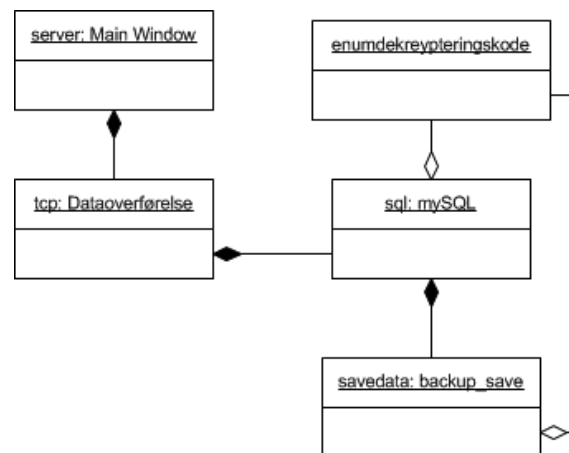
Figur 10.11. Illustrere overordnet hvordan databasen fungerer

10.6.1 Serveren

Denne section beskriver design og implementering af serveren. Opbygningen af serveren er gjort i klasser sådan at hver element i serveren er implementeret som en klasse se figur ??.

Ud over dette benyttes der nogle hjælpeklasser.

Det gælder at når KI forsøger at kontakte databasen, tilsluttes denne via tcp som benytter sig af socket. Når tilslutningen er gjort kan der overføres data fra KI til databasen. Dataerne bliver gemt i mySQL databasen. I tilfælde af at der bliver problemer med at gemme, laves der en tekst fil hvori dataerne bliver lagt og kan så forsøgt gemt af web siden.



Figur 10.12. Illustrere overordnet hvordan serveren fungerer

10.7 Resultater

10.8 Opnåede erfaringer

10.8.1 Generelt

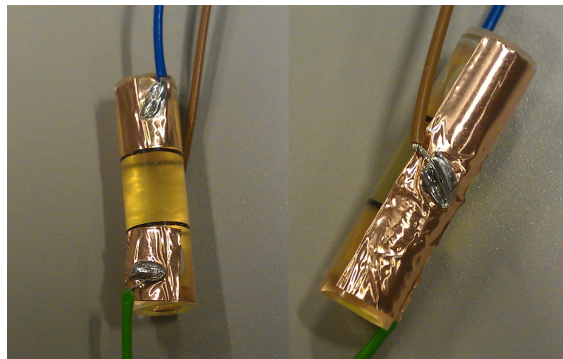
I projektet "BROS" blev systemet delt op i komponenter der så blev tildelt forskellige ansvarspersoner. Gruppens medlemmer valgte selv hvilken enhed de ville arbejde med. Dette har dog skabt en situation hvor gruppemedlemmer alene sidder med den fulde viden omkring moduler. Denne situation resulterede i at alle skulle være tilstede hver gang

modulerne skulle samles til det fulde system. Gruppen blev sendt opmærksom på denne problematik og kunne derfor ikke nå at reagere tidsnært.

Gruppen har i dette projekt bestået af 5 elektronik ingeniørstuderende. Dette har skabt nogle problemer med hensyn til programmeringsdele af projektet. Gruppen har dog været god til at søge hjælp til problematiske opgaver, hvilket har gjort at programmeringstunge moduler er blevet implementeret korrekt.

10.8.2 Udvikling af hældningssensor

Vi har startede med at udvikle på en prototype af en libellesensor vist på *Figur 10.13*. Vi



Figur 10.13. henning

kom frem til at den har en kapacitet på omkring $1 \cdot 10^{-15} [F]$. Det gør det praktisk umuligt at anvende da vores filter så har en alt for stor cutoff frekvens liggende over 3.0 MHz. Den høje frekvens giver en stor selvinduktion i vores ledning. Samtidig kan vi ikke lave sinus med denne frekvens med PSoC'en. Dette gjorde at vi måtte finde en anden løsning.

Næste prototype bestod af et potmeter og et pendul. Dog havde potmeteret en for stor friktionsmodstand, der gjorde det upræcist i forhold til vores krav.

Vi har gennem et tredje semestersfag fundet ud af at PSoC'en indeholder et accelerometer. Vi valgte så at lave en prototype med det. Dette viste sig at være en god løsning.

10.8.3 Metoder

Gruppen var opsatte på at have læst review på projektets dokumentation, men de adspurgte grupper havde ingen interesse i at lave review udvekslinger. Dette har ført til at dokumentationen blev gennemlæst på et senere tidspunkt af gruppens vejleder. Grundet udskydelsen havde gruppen set sig nødsaget til at forlænge tidsplanen for at opdatere og forbedre foreløbigt færdig dokumentation. Dette førte til at design og, i sidste ende, implementering blev udskudt, hvilket medfører et stort pres imod slutningen af projektføreløbet.

Gruppens opdeling i udviklingsmetoden har været meget flydende. Selv om der har været en projektleder og scrummaster har det ikke været nødvendigt, da gruppen har arbejdet godt sammen og diskuteret udviklingsmetoden til fuldest. Der har i forløbet været en udskiftning i koordinatorposten, hvorefter rumkoordinering har forløbet godt.

Gruppen har fået en bedre forståelse for eventuelt kommunikation med en kunde gennem flere iterationer af kravspecifikation.

Konklusion 11

Referencer 12

12.1 Artefakter

12.1.1 Kravspecifikation

Kravspecifikationsdokumentet er udarbejdet i begyndelsen af projektet og omfatter beskrivelse af Use Cases, ikke funktionelle krav samt kvalitetsfaktorer. Den fuldstændige kravspecifikation kan ses i bilag. (Kravspecifikation.pdf)

12.1.2 Accepttestspecifikation

Accepttestspecifikationsdokumentet beskriver de tests der skal laves for at undersøge om de ønskede krav er opfyldt. Den fuldstændige accepttestspecifikation kan ses i bilag (Accepttest.pdf).

12.1.3 Systemarkitektur

Systemarkitektur dokumentet beskriver systemets HW/SW opbygning og grænseflader. Den fuldstændige systemarkitektur kan ses i bilag (Systemarkitektur.pdf).

12.1.4 Integrationstestspecifikation

Integrationstestspecifikation beskriver de test der skal laves for at undersøge hvorledes de forskellige komponenter kan kommunikere. Den fuldstændige Integrationstest kan ses i bilag (Integrationstest.pdf).

12.1.5 Detaljeret design

Det detaljerede design dokument beskriver hvordan HW/SW er designet og hvordan systemets komponenter fungerer. Det fuldstændige Detaljeret design dokument kan ses i bilag (Detaljeret Hardware design.pdf og Detaljeret Software design.pdf).

12.1.6 Enhedstestspecifikation

Enhedstestspecifikation beskriver de tests der skal laves for at undersøge om de forskellige stubbe af systemet fungerer hensigtsmæssigt. Den fuldstændige enhedstestspecifikation kan ses i bilag (Enhedstest.pdf).

12.2 Hjemmesider

<http://www.docs.google.com> <http://office.microsoft.com/en-us/visio/> <http://www.maplesoft.com/>
<http://www.ni.com/multisim/>

12.3 Liste over bilag på CD

Komponentliste.pdf SCRUM.xls Logbog.pdf

12.3.1 Kode

KI

hest

SM

hestning

VBTE

honning

Server

12.3.2 Dokumentation

Accepttest.pdf

Arkitektur.pdf

Detaljeret_hardware_design.pdf

Detaljeret_software_design.pdf

Enhedstest.pdf

Integrationstest.pdf

Kravspecifikation.pdf

12.3.3 Datablade

PSoC Kionix KXSC7 datasheet (Accelerometer) ST3232 OSV! HESTE

12.3.4 Billeder

hvis vi har billeder af vores produkt!

Rettelser

Note: Lave referencerne til de enkelte afsnit. 12