

Resume 1

Abstract 2

Indholdsfortegnelse

Kapitel 1	Resume	1
Kapitel 2	Abstract	3
Kapitel 3	Forord	7
Kapitel 4	Indledning	9
4.1	Læsevejledning	9
Kapitel 5	Opgaveformulering	11
Kapitel 6	Projektformulering	13
Kapitel 7	Projektbeskrivelse	15
7.1	Projektgennemførelse	15
7.1.1	Rollefordelinger	15
7.2	Metoder	15
7.2.1	SCRUM	16
7.2.2	V-model	17
7.3	Analyse	17
7.3.1	Hvordan måler vi hældning?	17
7.4	Systemarkitektur	17
7.4.1	Systemkomponenter	17
7.5	Design og Implementering	18
7.5.1	Kontrolinterfacet	18
7.5.2	VBTE	19
7.5.3	SM	24
7.5.4	Kontrolinterfacet	27
7.6	Design og Implementering	28
7.6.1	Serveren	28
7.6.2	Web-side	30
7.7	Resultater	31
7.8	Opnåede erfaringer	31
7.8.1	Generelt	31
7.8.2	Udvikling af hældningssensor	31
7.8.3	Metoder	32
Kapitel 8	Konklusion	33
Kapitel 9	Referencer	35
9.1	Artefakter	35
9.1.1	Kravspecifikation	35

9.1.2	Accepttestspezifikation	35
9.1.3	Systemarkitektur	35
9.1.4	Integrationstestspezifikation	35
9.1.5	Detaljeret design	35
9.1.6	Enhedstestspezifikation	35
9.2	Hjemmesider	36
9.3	Liste over bilag på CD	36
9.3.1	Kode	36
9.3.2	Dokumentation	36
9.3.3	Datablade	36
9.3.4	Billeder	36

Forord 3

Denne rapport er udarbejdet af fem ingeniørstuderende ved Aarhus Universitet, Ingeniørhøjskolen. Rapporten er hovedproduktet i et obligatorisk projektforløb på 4. semester og gennemgår overordnet gruppens besvarelse og gennemførelse af projektet. Derudover er der blevet lavet en række projektdokumentationsdokumenter og et fysisk produkt. For yderligere detaljer henvises der til projektdokumentationen.

Rapporten er skrevet med en forventning om at læseren er på samme faglige niveau som gruppen, hvilket vil afspejle sig i det faglige sprog.

Indledning 4

Denne rapport omhandler gruppens projektforløb på 4. semester. Projektets emne er selvvalgt og hedder "slagsideregulering af bulkskib". Rapporten beskriver processen af projektet herunder hvordan forløbet har været, hvilke metoder der har været 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. Arbejdsprocessen og nogle af de komponenter der indgår i det færdige system afspejler disse krav. Kravene kan ses i 5

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:

- Kravspecifikation
- Analyse og arkitektur
- Detaljeret design
- Implementering
- Test

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 10¹. Alle afsnit er skrevet så de som udgangspunkt godt kanstå 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

¹Fixme Note: lav dynamiske referencer

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.²

²**FiXme Note:** Lave refrencerne til de enkelte afsnit.

Opgaveformulering

5

Gruppen skal fremstille et system der overholder følgende krav:

- Systemet skal interagere med omverdenen vha. af sensorer og aktuatorer.
- Der skal anvendes relevante faglige elementer fra semestrets kurser.
- Systemet skal omfatte pålidelig transmission af data mellem udvalgte enheder.
- Systemet skal kunne interagere med en bruger.

Projektformulering 6

Gruppen har valgt at arbejde med lastning og losning af et bulkskib. Det er formålet at lave et system, der aflaster skibets ansvarshavende officer. Denne står på broen under hele lastningen/losningen og holder øje med at hele operationen bliver gjort ordentligt. Med et system der automatisk sørger for at skibet altid er i vatter er det med projektet tilsigtet at hjælpe officeren med denne opgave. Officeren vil blive alarmeret af systemet hvis hældningen er større end den accepterede værdi. Ligeledes vil han kunne se hvis systemet er blevet utilregneligt (som følge af manglende opdateringer eller forbindelser). Derudover skal officeren manuelt kunne vælge at flytte ballast til den ene side, hvis han ved at der kommer en tung last, der vil tynge ned i modsatte side. Som en ekstra sikkerhedsforanstaltning har systemet mulighed for at kunne sende statusbeskeder til en ekstern database - eksempelvis placeret på terminalen. Således vil man også eksternt kunne opfange faretruende udvikling.

Projektbeskrivelse

7

7.1 Projektgennemførelse

Projektet er udført af en gruppe på fem personer. Gruppen er ny og sammensat af personer fra fire forskellige projektgrupper. Dette har medbragt både gode som dårlige ting. Det er godt at opdage nye måder at anskue et projektforløb og nye vinkler på udfordringer i processen. Det har dog desværre kostet en del energi at skulle tilpasse sig hinanden. En proces hvor alle har måtte gå på kompromis på et eller flere punkter.

Længden af faserne har vist sig at være skæve da processen ikke har været lige så gnidningsfri som først håbet. Som følge heraf er tidsplanen blevet revirderet flere gange og opgaver fra ét SCRUM har måttet videreføres i det næste.

Projektets overordnede tidsplan for faserne og de eksterne milestones ligger som bilag.

7.1.1 Rollefordelinger

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

Tabel 7.1. Tabel over rollefordelinger

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

7.2 Metoder

En kort præsentation af de to mest dominerende arbejds metoder 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.

7.2.1 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 7.1* vises det 2. sprint.

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

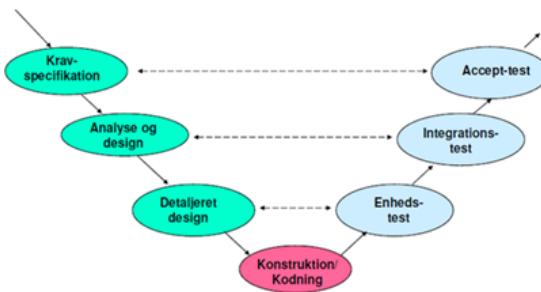
Figur 7.1. Sprint 2

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 7.2*



Figur 7.2. Forklaring af sprint points

7.2.2 V-model



Figur 7.3. V modellen

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

7.3 Analyse

7.3.1 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 7.8.

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

7.4 Systemarkitektur

Dette afsnit beskriver systemarkitekturen 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.

7.4.1 Systemkomponenter

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

Brugeren integrerer med systemet igennem KI. KI er styringsmodulet 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 vognen deres tank. For at kunne kontrollere hvor meget vand der er i tankende dette gøres ved hjælp af to ultra lydssensore som hele tiden mäter og vidergiver denne information til SM som så sender dette videre til KI der kan advare om vandstanden i tankene i tilfælde af at systemet afsat 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.

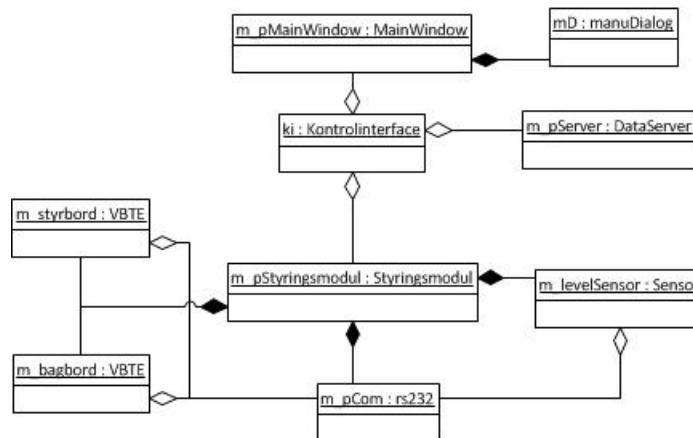
7.5 Design og Implementering

7.5.1 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 7.3

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

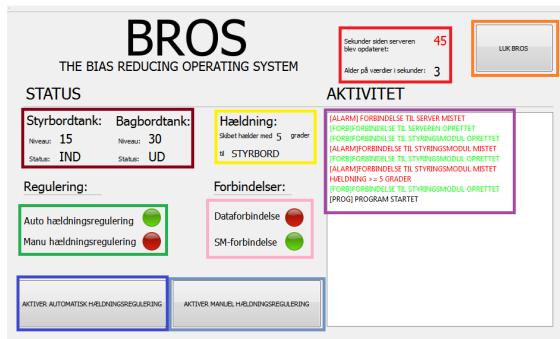
Kontrolinterfacets klasser

Kontrolinterface:	Programmets hovedklasse. Eksisterer for at rydde op i main-funktionen.
DataServer:	Står for alt TCP-kommunikationen med databasen. Oprettes af KI-klassen
Styringsmodul:	Oprettes af KI og opretter VBTE-, Sensor- og RS232klasserne.
Sensor:	Oprettes af SM og er ansvarlig for hældningsværdien.
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:	Objektet oprettes af SM-klassen og VBTE- og Sensorobjekterne har en delt association til den. Objektet har ansvaret for kommunikationen med det fysiske SM-modul. Objektet formidler sig på en protokol forstået den kommunikationsansvarlige kode på SM-modulet. Protokollen kan ses i dokumentet for det detaljerede softwaredesign.
MainWindow:	Oprettes af KI-klassen. Kontrollerer og overvåger den grafiske brugergrænseflade.
manudialog:	Oprettes af MainWindow og styrer den dialog, der fremkommer når man ønsker en manuel hældningsregulering.

Tabel 7.2. Kontrolinterfacets klasser

Grafisk brugergrænseflade

I denne sektion vil der kort blive gennemgået det vigtigste vindue i den grafiske brugergrænseflade; hovedvinduet. En gennemgang af de resterende vinduer vil kunne findes i Softwaredesign Appendix A: Kontrolinterface. På figur 7.5.4 er hovedvinduet vist. Her er vinduets elementer indrammet. En beskrivelse af hvert element vil kunne findes i tabel 7.5.4.



Figur 7.5. På figuren ses hovedvinduet for Kontrolinterface-programmet

7.5.2 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 ultralydtransmitter og receiver. Hardware er blevet udfærdiget i to dele. Den ene er

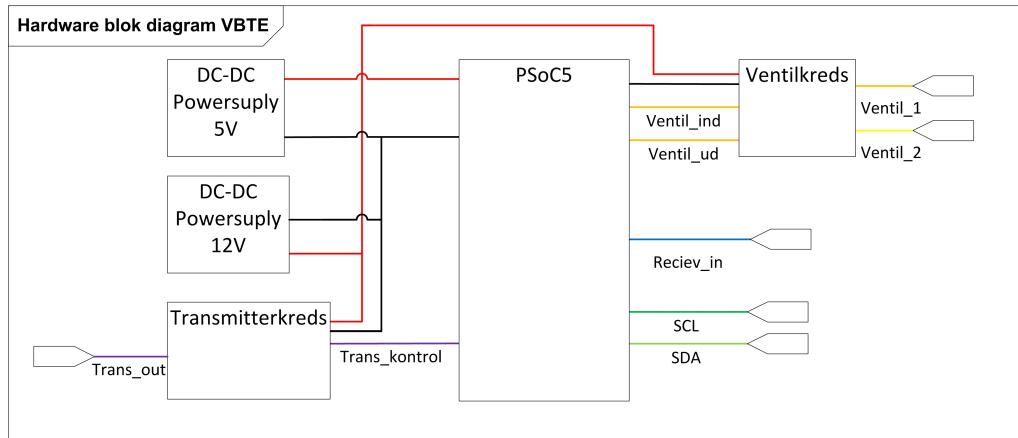
Hovedvinduets elementer

1: Forsinkelse i sekunder	Det øverste tal fortæller tiden i sekunder siden serveren sidst er blevet opdateret succesfuldt. Nedenunder udskrives tiden i sekunder siden værdierne i boks tre og fire er blevet opdateret.
2: Nedlukningsknap	Anvendes til at lukke programmet. Programmet åbner dialogen som ses i ??
3: Vandballasttankene	Her kan status for vandballasttankene aflæses. Niveauet er hvor fyldt tanken er angivet i procent. Hvis niveauet er over 70% skrives tallet med rødt. Status angiver vandets flow i tanken: IND/UD/LUKKET.
4: Hældningssensor	Værdien for hældningen af skibet angives i antal grader og i hvilken side skibet hælder.
5: Reguleringsstatus	Her angives hvorvidt automatisk eller manuel hældningsregulering er aktiveret. Der vil altid kun være en og kun en af disse aktiveret. Derfor vil der altid være en rød og en grøn indikator tændt. I dette eksempel er den automatisk hældningsregulering aktiveret.
6: Forbindelser	Indikerer hvorvidt der er forbindelse til Styringsmodulet og serveren. Dataforbindelse er rød hvis det ikke lykkedes at oprette forbindelse til serveren ved sidste forsøg. SM-forbindelse er rød hvis det ikke lykkedes at få de ventede svar fra Styringsmodulet. I denne situation er der forbindelse til styringsmodulet, men ikke serveren.
7: Automatisk reguleringsknap	Ved tryk på denne knap vil man komme til dialogen på ?? såfremt automatisk styring ikke er aktiveret. Hvis den er aktiveret og man trykker på knappen vil dialogen på figur ?? fremkomme.
8: Manuel reguleringsknap	Bringer dig til dialogen på figur ??
9: Aktivitetslog	Her udskrives vigtige hændelser i programmet med farvekoder. I dette eksempel kan det ses hvordan alarmer skrives med rødt og oprettede forbindelser skrives med grønt.

programeret 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

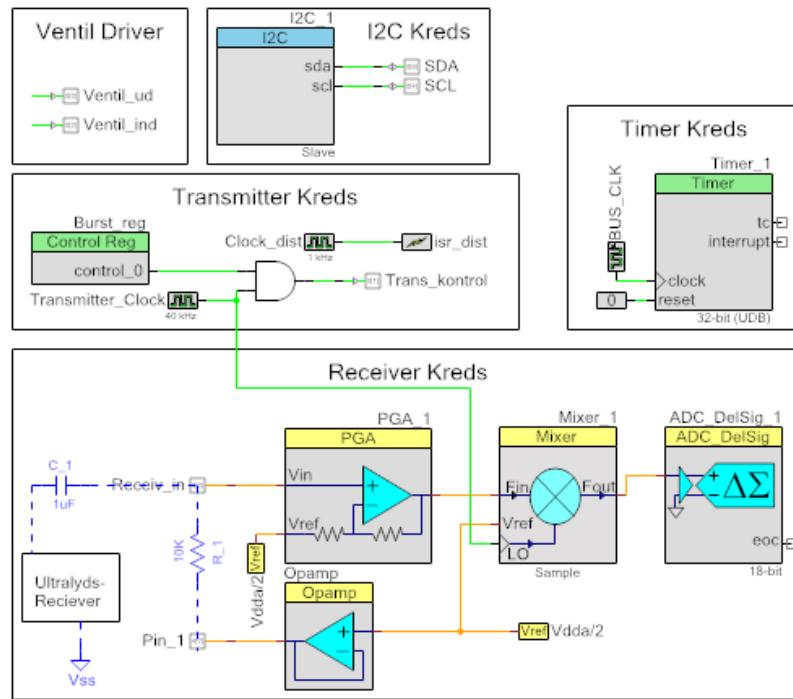
Gennem disse faser er designet blevet udfærdiget. Fremgangsmåden er anvendt for at overskueliggøre systemet og lette arbejdet ved at dele systemet op i små dele. På *figur 7.6* ses det overordnede design af VBTE'en. Der vil i rapporten tages udgangspunkt i ventilkredsen samt receiverdriveren på PSoC'en.



Figur 7.6. 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. Der vil i rapporten ligges vægt på transmitterkredsen samt receiverkredsen. På figur 7.6 ses hardware designet på PSoC'en.



Figur 7.7. Hardware på PSoC'en.

Transmitterkreds

Transmitterkredsen står for at sende burst samtidigt med hvert burst. Dette opnåes ved at bruge to clocks, en AND gate, en output pin samt et kontrol register. Transmitterclocken er indstillet til 40kHz da ultralydstransmitteren, ifølge databladet, opererer ved $40\text{ kHz} \pm 1\text{ kHz}$.

Clocken er blevet målt på oscilloskop til 40,3 kHz. Burst kontrolregisteret er anvendt til at AND'e clocken ud på trans_kontrol pinden.

Clock_dist interruptrutinen sørger for at tælle en variabel op der anvendes i main til at kalde burst funktionen. Dette gøres for at lave et nonblocking delay så andet kan køres på PSoC'en mens en burst er sendt afsted og der afventes detektion.

Receiverkreds

Receiverkredsen modtager signalet fra ultralydsreceiveren og omsætter det til en detektion. Dette sker efter følgende opskrift:

Løfte signalet → Forstærkning → Mixning.

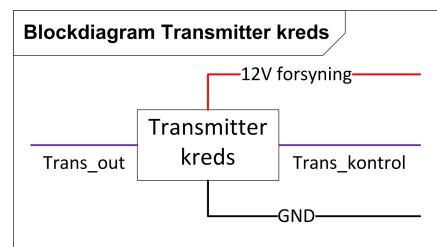
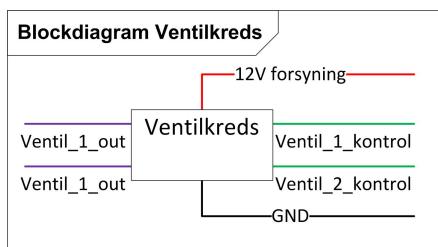
Signalet bliver løftet på til $\frac{V_{dda}}{2}$ fordi PSoC'en ikke kan arbejde med værdier under Vssa (GND). Dette gøres ved hjælp af en kondensator, en modstand og en spændingsfølger med $\frac{V_{dda}}{2}$ på det positive ben.

Efter signalet er løftet bliver det forstærket op af en PGA. Efter undersøgelser, hvor der blev sendt burst, er der valgt en forstærkning på 16, da der ca. modtages et signal på 200 mVp – p.

Herefter bliver signalet mixet med 40 kHz og filtreret. Efter mixeren giver det, stort set, en DC og summen af de to signaler. Filteret er inbygget i Delta-Sigma ADC'en og har en knækfrekvens ved $\frac{\text{Samplefrekvens}}{3}$. For at være sikker på at det meste af signalet er kommet over regnes der en opladningstid på filteret til 1/4 ms. Efter undersøgelser af signalet ind på ADC'en blev en spænding på 0,3 V.

Hardware uden for PSoC'en

Uden for PSoC'en er der lavet 2 hardwareblokke. Disse har til ansvar at omsætte et kontrolsignal til en større spænding over de respektive enheder.



Figur 7.8. Hardwareblok for ventil

Figur 7.9. Hardwareblok for transmitteren

Ventilkreds

Ventilkredsen får to kontrolsignaler fra PSoC'en og disse skal styre de to ventiler. Ventilerne monteret på kredsen er fra Danfoss og er af modellerne EV210A-1.2 og EV210A-4.5. Disse ventiler drives ved 12V 0.4A. Der er valgt en BD139 transistor til at forstærke signalet op. Denne har en forstærkning mellem 40 og 160 (aflæst fra databladet¹). IO's på PSoC'en kan maksimalt trække en strøm på 4 mA og det vil i værste tilfælde ikke være nok til at trække ventilerne.

Der er derfor lavet en darlington kobling for at mindske belastningen af PSoC'en og for

¹Se bilag/BD139.pdf

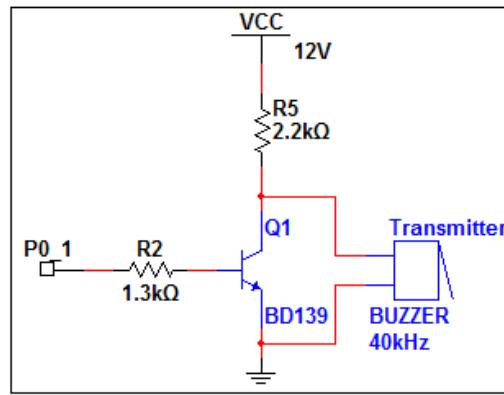
at sikre at der bliver lukket nok op for transistoren. Dette er gjort med en transistor af typen BC547. Transistorne er valgt ud fra pris og tilgængelighed. Der var først valgt en MOSFET IRLZ44n men denne var både for dyr og kunne klare en unødvendig stor effekt. Det smarte ved MOSFET'en er dog at den er spændingsstyret i forhold til de to andre transistorer.

Transmitterkreds

Transmitterkredsen anvender ultralydstransduceren som en højtaler. Denne er koblet over transistoren (Her anvendes også en BD139).

Via kontolsignalet fra PSoC'en, der bliver sendt ud ved 40 kHz, bliver der sat en spænding over transduceren.

Forsyningen hertil trækkes fra samme forsyning som ventilkredsen der bliver leveret fra powersuply'en. Ud fra databladet er der aflæst en ohmsk modstand på omtrent 10 k Ω . Derfor kan PSoC'en uden problemer selv trække denne kreds.

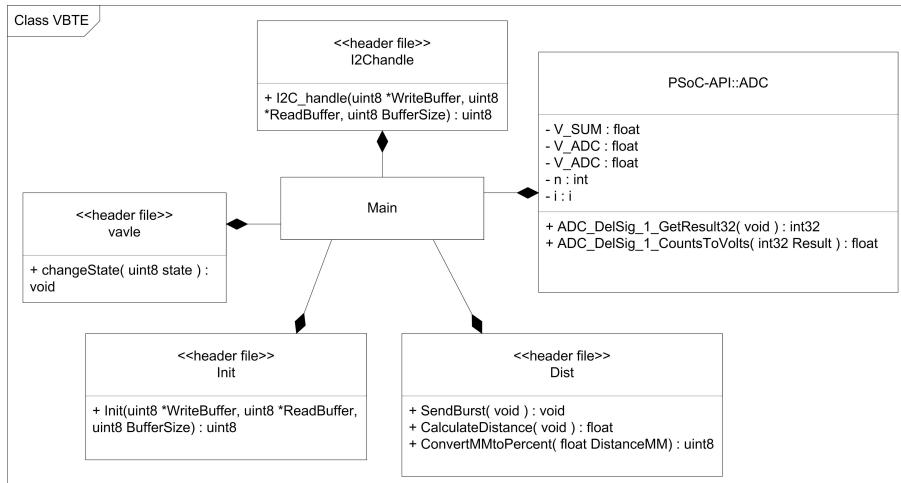


Figur 7.10. Transmitterkreds

For yderligere detaljer vedrørende hardware på VBTE henvises til *Deltajeret hardware design*.

Software

Softwareen til VBTE modulet er blevet designet til at kunne opfylde kravene i kravspecifikationen samt arkitekturen i systemarkitekturen. Den er blevet designet til at passe til hardwaren så den kan kontrollere de enkelte elementer. I designfasen er der blevet udfærdiget et klassediagram samt en statemachine over funktionaliteten. På *figur 7.5.2* ses klassediagrammet for VBTE modulets software.



Figur 7.11. Klassediagram over VBTE

Selvom softwareen er skrevet i C er der lavet klasser for alle h-filer for at gøre systemet overskueligt. For detaljerede metodebeskrivelser henvises til detaljert software design.

7.5.3 SM

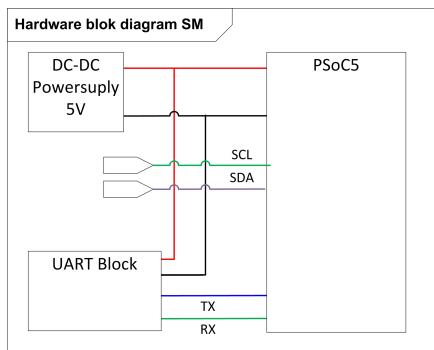
I dette afsnit beskrives design og implementering af SM modulet. SM modulet består af både software og hardware.

Hardware

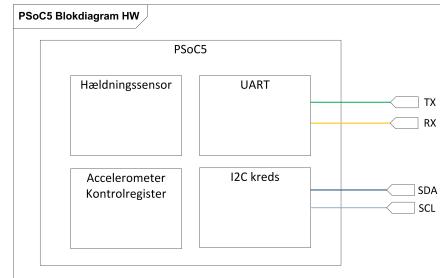
SM modulets hardware består af en konverteringskreds og en PSoC. På PSoC'en er monteret et Kionix KXSC7-2050 accelerometer. Konverteringskredsen anvendes til at sende UART signaler fra PSoC til en KI modulet. Accelerometerets x-akse anvendes til hældningsmålinger for hældningssensorblokken. Designfasen 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 7.12* ses det Overordnede design og på *Figur 7.13* ses PSoC blokken i SM. Der bliver efterfølgende taget udgangspunkt i Hældningssensoren.



Figur 7.12. Hardware blok for SM

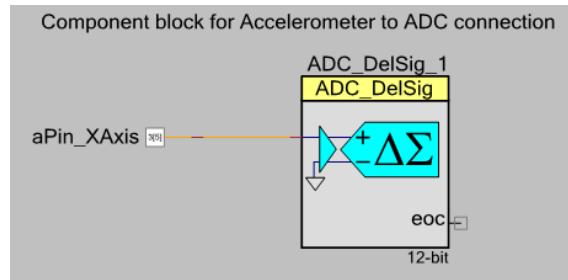


Figur 7.13. PSoC blok for sm

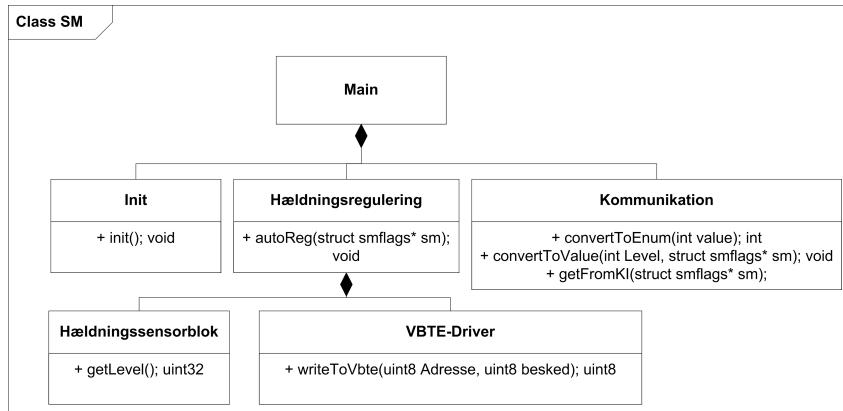
Hældningssensoren består af 2 komponenter: det førnævnte accelerometer samt en DelSig ADC internt i PSoC'en. Valget af accelerometer kommer af at have lavet en række prototyper der ikke mødte vores krav, hvilket accelerometeret i PSoC'en gjorde. Valget af ADC faldt på en DelSig da, den er meget støj immun grundet det indbyggede lavpas filter og har en høj opløsning. Valgte komponenter er illustret på *Figur 7.14*. 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 9.1.5 *Detaljeret hardware design* i Bilag.

Software

SM software behandler den konverterede data fra ADC'en samt kommunikation med VBTE og KI modulerne. Sofwarens udvikling har fulgt de samme faser som hardwaren, hvilket har ført til letforståelig og læselig kode. Softwaren er illustreret på *Figur 7.15*. Der vil efterfølgende blive taget udgangspunkt i funktionerne autoReg og getFromKI.



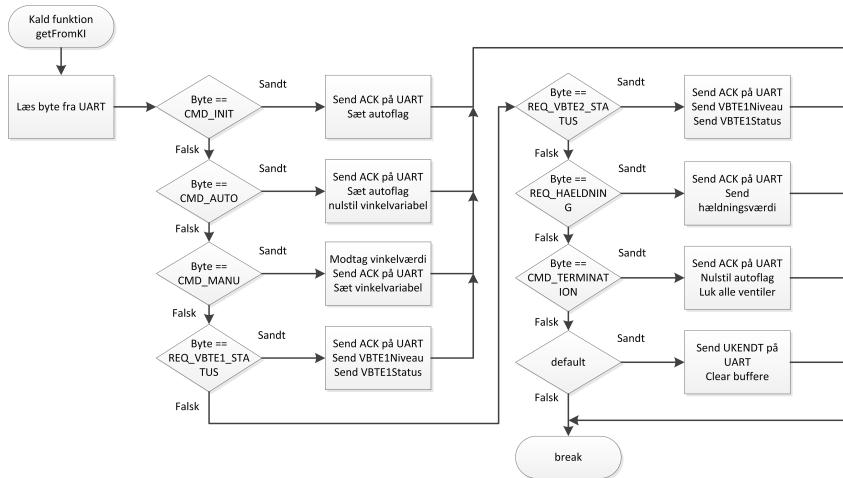
Figur 7.14. Hældningssensorens implementering



Figur 7.15. Klassediagram for SM

getFromKI

Funktionen er bedst beskrevet med et flowdiagram:



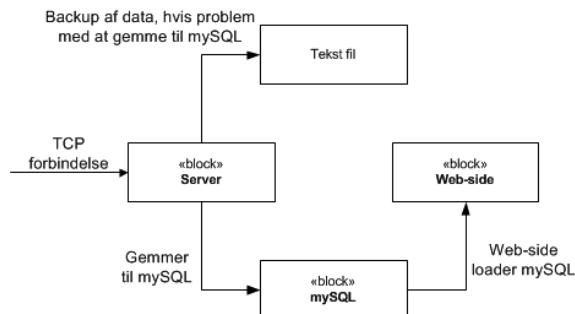
Figur 7.16. Flowdiagram for funktionen getFromKI

Funktionen har til ansvar at kommunikere med KI efter de krav opsat i Kravspecifikation. Hver ACK er relateret til den besked, der er modtaget, så KI modulet ved hvilken besked, SM har modtaget. Dette sikre pålidelig overførsel.

7.5.4 Database

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 håndtere denne side. Databasen er opdelt i serveren og web-siden.

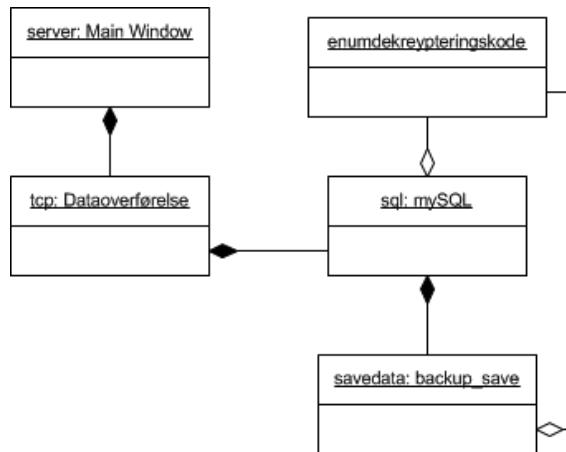


Figur 7.17. Illustrere overordnet hvordan databasen fungere

7.5.5 Serveren

Denne section beskriver design og implementering af severen. Opbygningen af severen er gjort i klasser sådan at hver element i severen er implementeret som en klasse se figur 7.20. 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 en tekst fil hvor dataerne bliver lagt og kan så forsøgt gemt af web siden.



Figur 7.18. Illustrere overordnet hvordan serveren fungere

Serverens klasser

Server	Er hovedvinduet på serveren. Information til brugeren udskrives herfra og knapper er implementeret.
TCP forbindelse (tcp)	Opretter en mulig tcp forbindelse hvor skibet kan tilslutte sig og overføre data.
Gemme data (sa-vedata)	Gemmer data til en tekst fil hvis ikke det har været muligt for serveren at få kontakt til mySQL.
De kryptering(enumeratorDeKrypteringskode)	Dekryptere level fra sm så data bliver i grader

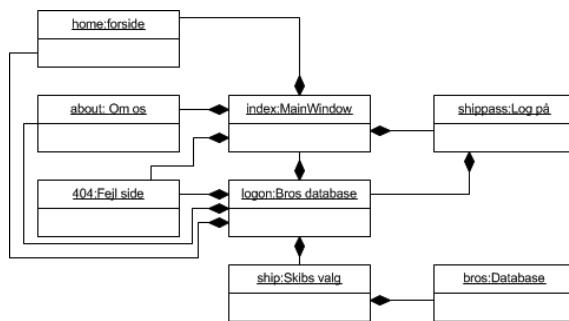
Tabel 7.3. Serverens klasser

7.5.6 Web-side

Denne section beskriver design og implementering af web siden.

Opbygningen af web siden er gjort hovedsageligt i php hvor dette gav mening. Desuden er html og styles (css) blevet benyttet til at sikre ens stil på hele web siden og for simpelt at kunne velgehoides. For at sikre mindst data transmision imellem computer og server er principippet i ajax blevet benyttet.

Når havneterminalen ønsker at se data om skibe der er opkoblet på BROS kan denne tilgå disse via den web baserede database. Når serveren køre kan brugeren tilgå databasen via knappen "Database"ellers kan denne tilgås via den locale adresse. For at komme ind i databasen skal brugeren taste sit password dette gøres for at sikre imod uaotirseret brug af databasen. Efter log ind på siden kan brugeren vælge hvilket skib der skal benyttes. Når skib er valgt kommer brugeren ind på den valgte database. Her kan brugeren se ID på skibet vandtilstand i ballast tanke, hældning og tid siden sidste opdatering (for nærmere se appendix²). Hvis serveren har haft problemer med at gemme den nyeste data vil web siden forsøge at gemme som det første inden databasen loades.



Figur 7.19. Illustrere overordnet hvordan web-siden fungere

²FiXme Note: link til appendix

Web sidens beskrivelse

Hovedvindue (index)	Den første side brugeren kommer til om ikke denne går via serveren	7.4
Databasen (login)	Denne side loader siden Forside, Om os og Log in ind i sig.	
	Denne side bliver kaldt efter at brugeren er logget på. Denne side loader Skibs valg, børs datasæn for skibet, Forside og Om os.	

Tabel 7.4. Web sidens opbygning

7.6 Resultater

7.7 Opnåede erfaringer

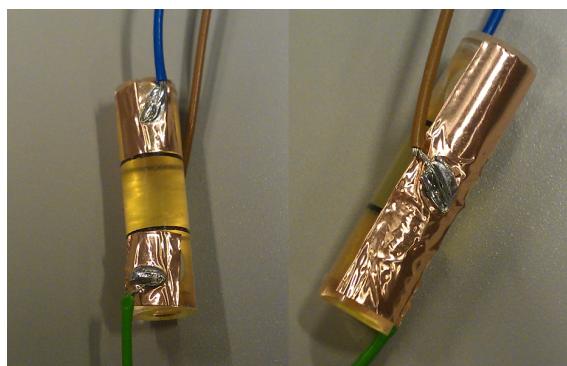
7.7.1 Generelt

Gruppen har fra start ville uddelegerede opgaver og ansvar for at lette arbejdsprocessen. Systemet blev derfor opdelt i moduler med en ansvarlig tilknyttet. Modulerne blev uddelt efter interesse. Dette har været en rigtig god løsning for flere af gruppens medlemmer tidligere, men i dette forløb har det vist sig også at være en hæmsko. Det fælles projektarbejde blev hurtigt fem personer, der sad med hvert deres lille område af projektet. Den interne kommunikation udeblev, og sparringen blev mangelfuld på grund af manglende indsigt i hvad de resterende gruppemedlemmer lavede.

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

7.7.2 Udvikling af hældningssensor

Vi har startede med at udvikle på en prototype af en libellesensor vist på *Figur 7.22*. Vi



Figur 7.20. henning

kom frem til at den har en kapacitet på omkring $1 * 10^{-15} [F]$. 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.

7.7.3 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 projektforløbet.

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 8

Referencer 9

9.1 Artefakter

9.1.1 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)

9.1.2 Accepttestspezifikation

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

9.1.3 Systemarkitektur

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

9.1.4 Integrationstestspezifikation

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

9.1.5 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).

9.1.6 Enhedstestspezifikation

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

9.2 Hjemmesider

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

9.3 Liste over bilag på CD

Komponentliste.pdf SCRUM.xls Logbog.pdf

9.3.1 Kode

KI

hest

SM

hestning

VBTE

honning

Server

9.3.2 Dokumentation

Accepttest.pdf

Arkitektur.pdf

Detaljeret.hardware_design.pdf

Detaljeret.software_design.pdf

Enhedstest.pdf

Integrationstest.pdf

Kravspecifikation.pdf

9.3.3 Datablade

PSoC Kionix KXSC7 datasheet (Accelerometer) ST3232 OSV! HESTE

9.3.4 Billeder

hvis vi har billeder af vores produkt!