

Underskrifter

Dato for aflevering: 28/05-2014

Gruppemedlemmer:

Bjørn Sørensen (BS) 201270432

Jakob Schmidt (JS) 201270402

Jeppe Stærk (JSA) 201271201

Jesper Christensen (JC) 201270478

Mick Kirkegaard (MK) 201370970

Poul Overgaard (PO) 201370794

Simon Kirchheimer (SK) 201270090

Vejleder: Torben Gregersen

Resumé (BS)

Den følgende rapport beskriver arbejdet med gruppens semester-projekt for 2. semester på Aarhus School of Engineering. Formålet med projektet er at afprøve de metoder og emner som semesterets fag har introduceret. Rapporten beskriver arbejdsmetoderne der er brugt fra idé til produkt, det egentlige produkt og udviklingen af dette samt de overvejelser og værktøjer som er benyttet.

Projektet tager udgangspunkt i et system til at beskytte børn i hjemmet mod stød og forbraændinger fra elektriske apparater. Fra et computer program kan en bruger tænde og slukke enheder som er koblet til forsyningssel-nettet uden behov for ekstra kabler. En kodelås forhindrer børnene i selv at tænde for strømmen igen og forældre kan dermed slukke for farlige elektriske apparater centralt, så børn ikke kommer til skade hvis de ikke er under opsyn.

Udviklingsforløbet er styret efter ASE-modellen som er en halv-iterativt projektleddelses-metode. Produktet er udviklet på de platforme der er gjort brug af på 1. og 2. semester. Som enheder til modtagelse og afsendelse af data over el-nettet bruges STK500-kittet fra Atmel samt egen udviklet hardware, til at interface til forsyningsnettet og en almindelig PC bruges som bruger interface. Som kodelås anvendes DE2 boardet fra Altera.

Projektet er endt ud i et funktionelt bruger interface, en X10 afsender enhed og en næsten funktionel X10 modtagerenhed.

Abstract (BS)

The following document describes the work and process of the groups 2nd term project at Aarhus School of Engineering. The purpose of the project is to use and evaluate the methods and subjects taught at this terms courses. The report describes how the product came from an idea to a physical product as well as the details of the product and the methods used.

The product developed is a child security system protecting unattended children from electrical shock and heat burns as a course of dangerous home appliances. From a computer program the user can turn on and off the power of mains outlets without other cables than that of the mains. A code lock secures that only authorised users can access to program and that way parents can disable dangerous home appliances when they leave their children unattended.

Development is managed with the ASE-model, which is a semi-iterative project management process. Further more is it done on different known platforms introduced first and second term. As transmitting and receiving units Atmel's STK500 development board is used along with specialised hardware. A PC is used as the user interface and the DE2 board from Altera is used as the code lock.

The results include a functional user interface, an X10 transmitter and an almost functional X10 receiver.

Indholdsfortegnelse

Kapitel 1 Indledning (BS, JS)	7
Kapitel 2 Opgaveformulering (PO)	9
Kapitel 3 Projektafgrænsning (JS)	11
Kapitel 4 Systembeskrivelse (MK)	13
Kapitel 5 Krav (JS)	15
Kapitel 6 Kravspecifikation (SK)	17
6.1 Usecases	17
6.1.1 Bruger	18
6.1.2 Eksterne enheder	18
6.1.3 Barn	18
6.1.4 SMS modtager	18
6.1.5 DE2 Board	18
Kapitel 7 Arbejdsproces	21
7.1 Udviklingsmodeller (JS, PO)	21
7.2 Møder, tidsplan, logbog og referater (PO)	21
7.3 Arbejdsfordeling (PO)	22
7.3.1 Hardwareteamet (PO)	22
7.3.2 Softwareteamet (BS)	22
7.3.3 Andet (PO)	22
Kapitel 8 Udviklingsværktøjer	23
8.1 LaTex og Texmaker (JS, PO)	23
8.2 Microsoft Visual Studio (BS)	23
8.3 Atmel Studio (BS)	23
8.4 National Instruments Multisim (PO)	23
8.5 Microsoft Visio (JS)	24
8.6 Altera Quartus II (PO)	24
8.7 Electronic ToolBox App (JS)	24
8.8 Filhåndtering (JS, PO)	24
8.8.1 GitHub	24
8.8.2 Dropbox	24
8.8.3 Google Drev	24
Kapitel 9 Systemarkitektur	25
9.1 Hardware arkitektur	25
9.2 Domænemodel	25

9.2.1	BDD Hardware	26
9.2.2	Plantegning over HW	26
9.3	Software arkitektur	27
9.3.1	Applikationsmodel for PC (JC)	27
9.3.2	Applikationsmodel for CSS-hovedenhed (BS)	28
9.3.3	Applikationsmodel for X10-udtag (BS)	28
Kapitel 10 Design		29
10.1	Hardwaredesign	29
10.2	Encoder (JS)	29
10.2.1	Højpasfilter (JS)	30
10.2.2	Zero Crossing Detector (SK)	31
10.3	Decoder (JS)	31
10.3.1	Båndpasfilter (JS)	32
10.3.2	Envelope Detector (SK)	32
10.3.3	Dipswitch (JS)	34
10.4	Software design	35
10.4.1	PC (JC)	35
10.4.2	CSS-hovedenhed (BS)	36
10.4.3	X10-udtag (JSA)	37
Kapitel 11 Implementering		39
11.1	Hardware implementering	39
11.2	Software implementering (BS)	39
Kapitel 12 Resultater		41
Kapitel 13 Opnåede erfaringer (JS)		43
Kapitel 14 Fremtidigt arbejde (SK)		45
Kapitel 15 Konklusion (PO, SK)		47
Kapitel 16 Individuel konklusion		49
16.1	Bjørn Sørensen	49
16.2	Jakob Schmidt	49
16.3	Jeppe Stærk	50
16.4	Jesper Christensen	50
16.5	Mick Kirkegaard	50
16.6	Poul Overgaard	51
16.7	Simon Kirchheimer	51
Kapitel 17 Litteraturliste		53
17.1	Bøger	53
17.2	Hjemmesider	53
17.3	Opslagsværker	53
Kapitel 18 Bilags-CD indhold		55

Indledning (BS, JS)

1

Med udgangspunkt i børnesikkerhed i hjemmet er der blevet udviklet et produkt, som kan hjælpe familier med børn, til at få et mere sikkert hjem.

Af problemstillinger som kan opstå i en almindelig husholdning kan nævnes:

- Fare for at et barn tænder for en kogeplade, eller andre elektriske varme aggregater, og efterfølgende kan brænde sig
- Fare for at et barn kan skære sig på køkkenknive som ligger i en skuffe

Den anden del af systemet er en babyalarm. Næsten alle mennesker i Danmark har deres mobiltelefon i nærheden hele tiden, så i stedet for at skulle have en babyalarm med rundt også, så kan man koble sin mobil til systemet og få besked når barnet giver lyd fra sig.

Dette ender ud i tre produkter:

- Afbryder til valgt 230 Vac stikkontakt
 - Beskyttelse mod kogeplader og lignende
- Låsemekanisme til at låse skabe og skuffer
 - Aflåsning af skuffe med køkkenknive
- Babyalarm til lyddetektering
 - SMS-beskeder i stedet for en ekstra ”boks” i lommen

Systemet skal være nemt at sætte op og skal kommunikere over det eksisterende 230 V vekselspændings netværk i hus installationen.

En central computer håndterer styringen mellem enhederne og systemet kan aktiveres med et kodetryk.

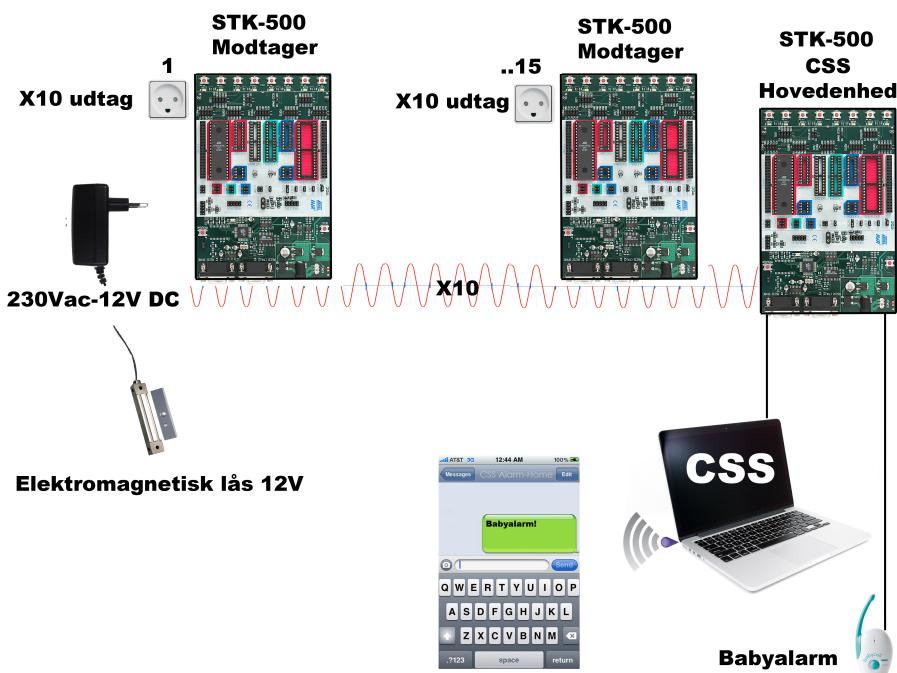
For at udarbejde det endelige produkt er vi startet i fællesskab med at få lavet de nødvendige dokumentationer der skal bruges for at kunne udvikle sådan et produkt. Herunder er bl.a. use case diagram, kravspecifikation og accepttest. Herefter har gruppen delt sig i et hardware og software team. Hardware har taget til opgave at få udviklet de nødvendige elektroniske komponenter mens software har skrevet koden.

Begge teams har stået for den respektive dokumentation for det de fik udviklet gennem forløbet.

Opgaveformulering (PO) 2

Konstruer et system der via en PC serielt kommunikere med en hovedenhed. PC'en er afhængig af et valideret login for at benytte alle funktioner. Dette login kommer fra en kodelås som har direkte kontakt til hovedenheden. Hovedenheden skal kommunikere med nogle perifere enheder ved at benytte X10 kommunikation. Denne kommunikation skal kunne aktivere eller deaktivere ønskede 230 Vac udtag på de perifere enheder. Det skal være muligt at tilkoble en magnet lås, til en køkkenskuffe, til den perifere enhed. Der skal også være en babyalarm som registrerer barnegråd. Hvis denne aktiveres skal der kunne afsendes en SMS via PC'en.

Nedenstående figur 2.1 illustrerer ovenstående.



Figur 2.1. Systemoversigt

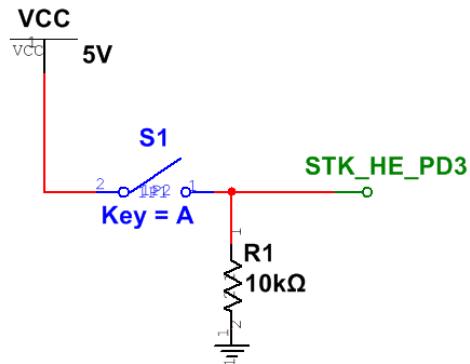
Projektafgrænsning (JS)

3

Grundet begrænset tid og ressourcer er det nødvendigt fra start at sætte nogle begrænsninger til hvilke dele af systemet der ønskes realiseret, som det ligeledes har været nødvendigt under forløbet at skære ned på hvad vi har ønsket realiseret.

X10 opererer normalt på 230 Vac el-nettet, men da vi ikke har autorisation til at arbejde med 230 V og af sikkerhedsmæssige årsager foregår realiseringen ved 18 V 50 Hz. Dette ændrer ikke på funktionaliteten eller virkemåden af systemet.

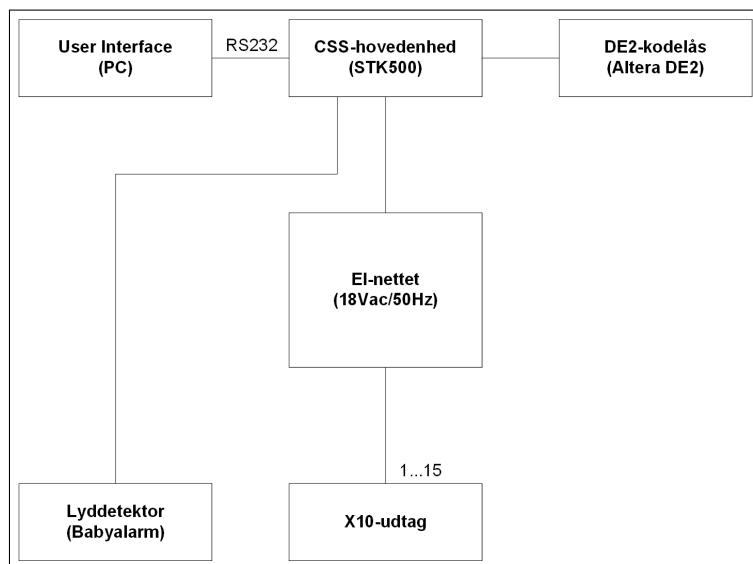
Lyddetektionen og magnetlåse er desværre ikke nået realiseret som ønsket. For lyddetektionen er der er i stedet lavet med en knap der giver et højt signal som skal imitere at lyd er detekteret. Se figur 3.1. Der er ikke lavet nogen alternativ løsning til magnetlås.



Figur 3.1. Diagram over knap for lyddetektion

Systembeskrivelse (MK) 4

Fra kundens synspunkt består systemet af en computer og nogle kontrollerbare stikdåser rundt i huset samt en babyalarm. Her beskrives hele systemet mere detaljeret.



Figur 4.1. Systemoversigt

I figur 4.1 ser man, at systemet består af en PC, der har forbindelse til CSS-hovedenheden; denne er kundens kontrolenhed(User Interface). Her har kunden adgang til et simpelt menusystem, hvori kunden kan kontrollere stikdåserne(X10-udtagene). Der er mulighed for at aktivere, deaktivere eller udlæse status på systemet. Der er endvidere også mulighed for at ændre mobilnummeret til den person, som skal modtage babyalarm-SMS'en. Hele systemet kræver en 3 cifret adgangskode, der indtastes ved hjælp af et eksternt hardwaremodul(DE2-kodelås).

Når en kommando bliver eksekveret i User Interfacet, sendes der data serielt ud via RS232 til CSS-hovedenheden. Denne data bliver encodet til en X10-bitstrøm og sendt ud på elnettet(18Vac). Denne bitstrøm bliver så aflæst af X10-udtagene, og hvis et X10-udtag har den korrekte adresse, vil kommandoen blive udført på denne.

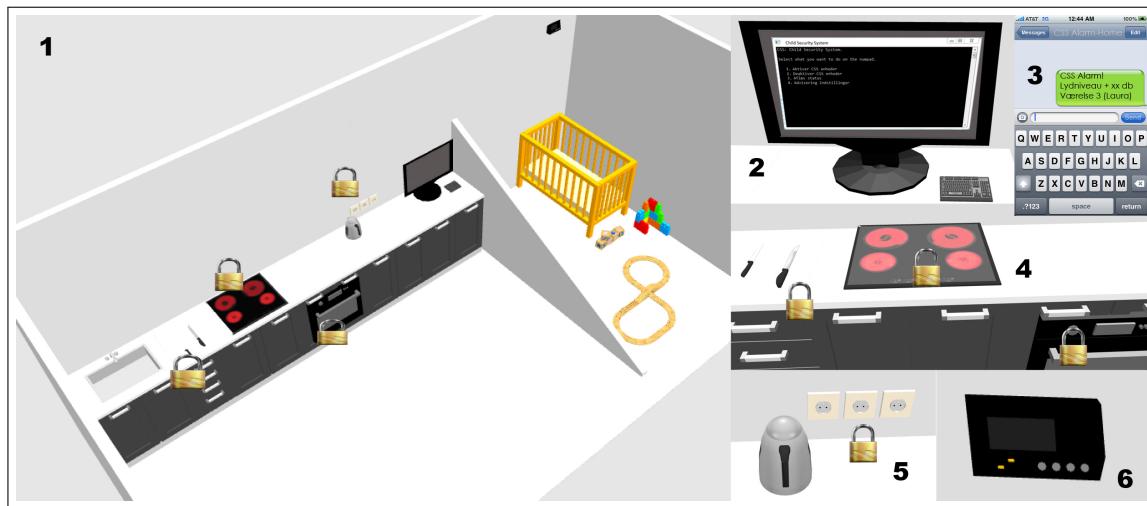
Babyalarmens funktion er at give CSS-hovedenheden besked om støj i det værelse, hvor lyddetektoren sidder placeret. Lyddetektoren er kablet direkte til CSS-hovedenheden og virker ved, at den ved et givent lydniveau¹ vil sende et signal til CSS-hovedenheden, som via

¹Se side 20-21

dennes serielle RS232 port fortæller PC'en, at der skal sendes en SMS til det forudbestemte mobilnummer med en advarsel.

Installation i hjemmet (PO)

Her ses, hvordan installationen kunne laves i en kundes hjem. Hængelåsene angiver hvilke enheder i hjemmet, der kan interageres med.



Figur 4.2. Installationsoversigt

1. Samlet oversigtstegning af CSS.
2. CSS-programmet med tilhørende DE2-kodelås.
3. SMS-besked udsendt af systemet, idet lydniveauet i værelse 3 (Laura) har været over det tilladte.
4. Overblik over, hvad systemet er tiltænkt at børnesikre. Køkken skuffe med skarpe genstande, kogeplader, ovn.
5. 230V udtag. X10 styret, således at det bestemmes, om udtaget skal være aktivt.
6. Babyalarm. Illustrationen vil variere i forhold til virkeligheden.

Krav (JS) 5

Child Security System er et system vi har udviklet med udgangspunkt i at gøre den almindelige husstand mere sikker for børn. Samtidig skal systemet have en integreret baby overvågning som er tiltænkt som erstatning for den almindelige babyalarm. Via en hovedenhed styres udvalgte enheder, hvorfra disse kan slukkes og tændes. Disse enheder er almindelige 230 V udtag som ønskes at kunne afbrydes eller styre en magnetlås til aflåsning af døre eller skuffer.

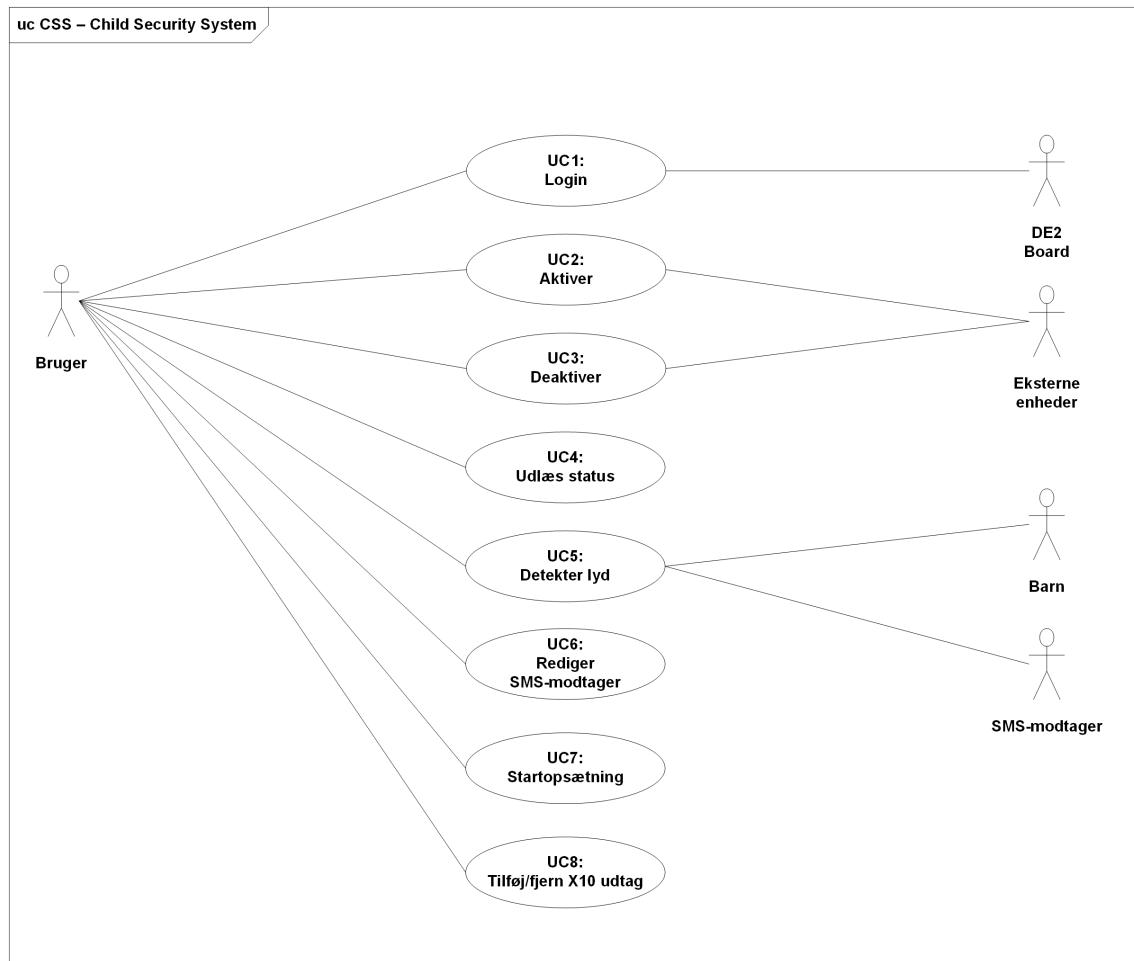
Baby overvågningen består af en sensor der mäter lydniveauet i et rum som sender et signal til hovedenheden hvis grænseværdien overskrides. Dette signal fortæller hovedenheden at den skal sende en SMS til et telefonnummer efter kundens ønske.

Kravspecifikation (SK)

6

Der er blevet udarbejdet en kravspecifikation ud fra følgende use cases se figur 6.1, her er der beskrevet hvilke use cases brugeren, DE2 boardet, eksterneenheder, barn og SMS-modtager har kontakt med.

6.1 Usecases



Figur 6.1. Usecase diagram

For yderligere beskrivelse af hver enkelt use case og større forståelse henvises til projektdokumentation¹

¹Projektdokumentation s 9-18

Herunder ses en beskrivelse af hver enkelt aktør i systemet.

6.1.1 Bruger

Type Beskrivelse	Bruger aktøren er ejeren af systemet eller den voksne med adgang til Computeren. Vil typisk være forældre, barnevige osv. (Primær)
-------------------------	--

6.1.2 Eksterne enheder

Type Beskrivelse	Eksterne enheder, omfatter hvad man ønsker at aflåse eller slukke for. Vil typisk være skabe, komfur, el-kedel osv. (Sekundær)
-------------------------	--

6.1.3 Barn

Type Beskrivelse	Barnet eller børnene i huset, som systemet skal beskytte. (Sekundær)
-------------------------	--

6.1.4 SMS modtager

Type Beskrivelse	Typisk forældrene eller barnevigen. Den person der skal have besked om gråd eller anden støj fra børneværelset. (Sekundær)
-------------------------	--

6.1.5 DE2 Board

Type Beskrivelse	DE2 Board programmeret som kodelås i DSD øvelse 7 (Sekundær)
-------------------------	--

Ikke-funktionelle krav

Herunder ses en beskrivelse af ikke-funktionelle krav.

Brugbarhed (Usability)

1. UI skal kunne bruges efter gennemlæst manual.

Pålidelighed (Reliability)

2. Levetid: 5 år uden hardware nedbrud
3. Software oppetid: Minimum 1 måned før genstart

Ydeevne (Performance)

4. System respons må maksimalt være 2,5 sekunder
5. Startuptid fra power-off til funktionel tilstand maksimalt 2 minutter
6. Systemkapaciteten er på maksimalt 15 CSS udtag
7. Ved lyddetektion må der maksimalt gå 1 minut før SMS-besked er afsendt

Vedligeholdelse (Supportability)

8. X10 udtag kan udskiftes separat ved simpel omkodning ved hjælp af addresseswitchen
9. Systemet er plug'n'play i en almindelig husholdning
10. X10 udtag kan tilføjes og installeres løbende

Generelle krav

11. Systemet skal virke på det eksisterende 230 Vac netværk i almindelige husstande
12. Kommunikationen mellem X10 udtag og hovedenheden skal ske på X10 protokollen
13. Systemet skal kunne afsende SMS-beskeder
14. Systemet skal automatisk logge ud efter 1min uden aktivitet

CSS enheder

15. Udtag skal kunne være i en 1,5 moduls Fuga stikdåse
16. Udtag skal have en LED indikator som viser at den er aktiv
17. Hovedenheden skal kunne virke på 230 Vac/13 A tilslutning

Eksterne enheder

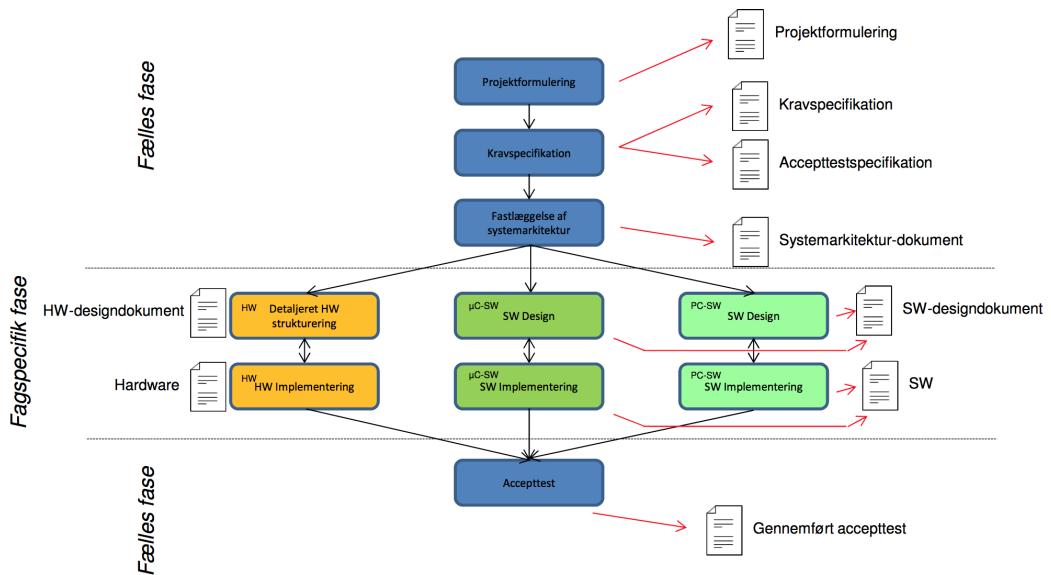
18. Lyddetektoren skal registrere lyde på over 68 dB
19. Der må maksimalt afsendes 1 SMS-besked pr. minut ved gentagende reaktion fra lyddetektoren
20. Låse enheder må maksimalt være 8x5x3 cm
21. Låse enhederne skal kunne holde 5 kilogram

Arbejdsproces

7

7.1 Udviklingsmodeller (JS, PO)

Med udgangspunkt i ISE-undervisningen¹ er projektarbejde opbygget omkring ASE-modellen som ses på figur 7.1. ASE-modellen er opbygget i 2 faser. En fællesfase og en fagspecifik fase.



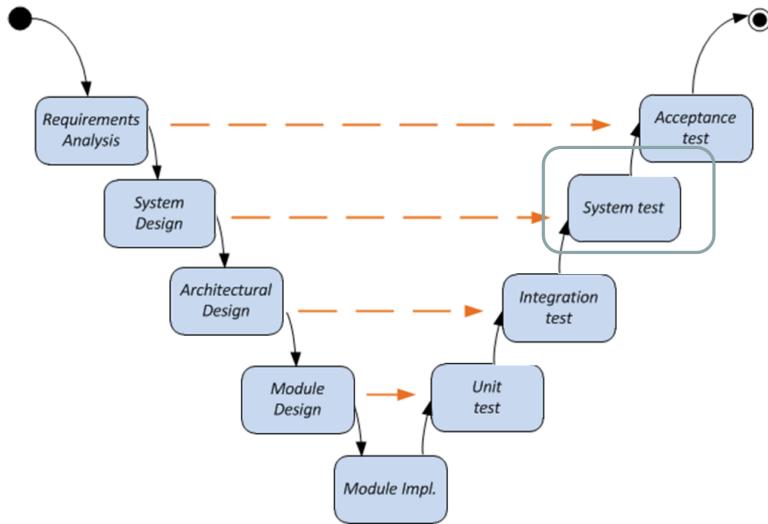
Figur 7.1. ASE-modellen

I fællesfasen arbejde hele gruppen sammen omkring udarbejdelse af de forskellige deldokumenter. I den fag specifikke fase deles gruppen op i mindre teams for at udvikle de fag specifikke deldokumenter.

ASE-modellen tager udgangspunkt i V-modellen som ses på figur 7.2.

Ved at benytte V-modellens opbygning færdiggøres en fase inden en ny påbegyndes. Og ydermere planlægges testen af alle faserne parallelt med at fasen udarbejdes. F.eks udarbejdes accepttesten samtidigt med at kravspecifikationen udarbejdes.

¹Indledende System Engineering - ASE, fag på 2. semester



Figur 7.2. V-modellen

7.2 Møder, tidsplan, logbog og referater (PO)

I forbindelse med projektforløbet er der afholdt en række møder. Vejledermøder, gruppemøder samt reviewmøder.

Vejledermøder er forbindelsen mellem gruppen og gruppens vejleder. Her har det været muligt at få løbende feedback samt et indblik i om det der forventes også er det gruppen forventer. Vejledermøder har været fastlagt til ét i ugen. Det er næsten opretholdt, dog med enkelte aflysninger.

Gruppen har hver uge holdt mindst et, nogle gange flere møder. Disse møder er brugt til at afklare uoverensstemmelser og planlægning af den kommende uge. Under gruppemøderne er der 2 gange brugt tid på en trivsels runde. Her har det været muligt at give ris/ros til gruppen og eller enkelte gruppemedlemmer. Gruppemøderne startede lidt løst, men dette blev hurtigtændret til at have en fast mødeholder, som styrede mødets gang. Tidsplanen er under gruppemøderne blevet revideret, således at den altid var opdateret til vejledermøderne. Gruppen har fra start udarbejdet en gruppekontrakt². Gruppekontrakten har dannet grundlag for forventningerne til gruppen og dens medlemmer.

Reviewmøder har fungeret således at gruppen enten udførte review på en anden gruppe og herefter fremlagde dette. Omvendt modtog gruppen lignende review fra andre grupper. Disse review førte ofte til uklarheder, som gruppen herefter måtte tage stilling til i gruppemødet.

Alle møder blev ajour-ført med logbog og mødereferat. Her har gruppen haft en fast sekretær.

²Gruppekontrakten findes på bilags cd under navnet: Gruppekontrakt.pdf

7.3 Arbejdsfordeling (PO)

Kapitler og afsnit er angivet med forfatter, hvor gruppen ikke har været fællesforfatter. Oftest har der været flere om emnet, men kun én eller to forfattere. Derfor beskrives der herunder hvem der har haft hvilke arbejdsområder.

Arbejdsfordelingen er herunder beskrevet.

7.3.1 Hardwareteamet (PO)

Hardwareteamet bestående af: Jakob, Mick, Poul og Simon. Samarbejde er nøgleordet for dette team. Hele hardware teamet har deltaget på lige fod i den foranliggende proces mht. til arkitektur og design. Herefter har enkelte været forfattere på dele af hhv. rapport og dokumentation. Heraf ses at Jakob og Simon har stået for en del af indskrivningen, mens Mick og Poul har stået for implementeringen og den tilhørende test. Vi har som team været samlet omkring fejlfinding af de enkelte moduler.

7.3.2 Softwareteamet (BS)

Softwareteamet bestående af Bjørn, Jeppe og Jesper. I den indledende fase af arkitekturen har software teamet arbejdet sammen, men forholdsvis hurtigt herefter delte teamet op i tre områder ud fra de hardware platforme der skulle programmeres på. Jesper har haft ansvar for PC, Bjørn for CSS-hovedenhed og Jeppe for X10-udtag. Jeppe har også lavet ClickATell klassen til PCen. Vi har arbejdet individuelt, men dog med regelmæssige møder og afklaring for at sikre at interface aftaler og lignende stadig blev overholdt.

7.3.3 Andet (PO)

Opsætning af L^AT_EX, både dokumentation og rapport

Hovedansvarlige: Mick, Poul

Rettelse af projektdokumentation Hovedansvarlig: Mick

Sekretærarbejde, bestående af logbog samt møderefiler

Hovedansvarlig: Simon

Mødeleder, afholdelse af møder, status, samt tavlestyring

Hovedansvarlig: Poul

Lokalebooking samt vejlederkontakt

Hovedansvarlig: Jeppe

Udviklingsværktøjer 8

Gennem hele projektforløbet er der anvendt forskellige programmer og værktøjer til de respektive opgaver. Nogle programmer havde vi kendskab til på forhånd hvor andre var helt nye for enkelte eller alle gruppemedlemmer.

8.1 LaTex og Texmaker (JS, PO)

Hele rapporten er skrevet i L^AT_EX. Dette valg kom i starten af projektet da IDA havde et tilbud om et gratis endags kursus, hvor hele gruppen blev enige om at deltage.

L^AT_EXer et kodebaseret tekstdredigerings sprog som er designet netop til større rapporter. Formålet er at gøre forfatteren fri for at skulle bekymre sig om formateringer således at han/hun kan rette al fokus på indholdet i rapporten.

Texmaker er benyttet som teksteditor. Heri er alt L^AT_EXkoden skrevet. Programmet har indbygget pdf-viewer, der gør det muligt at se det endelige produkt vha. Texmakers kompiler. Texmaker er desuden udstyret med dansk stavekontrol.

Det krævede dog lidt tid i starten at komme i gang med L^AT_EX, men da det var på plads fungerede det rigtig godt.

8.2 Microsoft Visual Studio (BS)

Udviklingsprogrammet Microsoft Visual Studio 2012 er brugt til softwareprogrammeringen til PC.

8.3 Atmel Studio (BS)

Atmel Studio 6.1 er det brugte værktøj til programmering af software til CSS-hovedenheden og X10-udtaget.

8.4 National Instruments Multisim (PO)

National Instruments Multisim er benyttet i forbindelse med design af kredsløbsdiagrammer.

8.5 Microsoft Visio (JS)

Som del af ISE-undervisning er der blevet undervist og anvendt Microsoft Visio til udarbejdelse af diverse diagrammer. Herunder UML og SysML. SysML er anvendt til at designe blok- og internalblok-diagrammer for hardwaren. UML er anvendt til klasse- og sekvensdiagrammer.

8.6 Altera Quartus II (PO)

Altera Quartus II er anvendt til VHDL programmeringen af DE2 kodelåsen.

8.7 Electronic ToolBox App (JS)

Electronic ToolBox er en applikation der er udviklet til iOS. Den indeholder informationer omkring det meste elektronik og værktøjer til diverse udregninger af kredsløbsdesign. Vi har primært anvendt det til udregninger på knækfrekvenser i forbindelse med høj-, lav- og båndpasfiltre.

8.8 Filhåndtering (JS, PO)

Til håndtering af filer er nedenstående 3 løsninger brugt.

8.8.1 GitHub

GitHub er et sky-basseret versionsstyringsprogram. Det er brugt til de produktmæssige dokumentationer, dvs. software kode, hardware diagrammer og projektdokumentation samt projektrapporten.

8.8.2 Dropbox

Dropbox benyttes som cloud løsning. Dropbox har fungeret som fælles harddisk. Primært benyttet i forbindelse med de 2. afholdte reviews. Ydermere er dropbox benyttet til deling af litteratur.

8.8.3 Google Drev

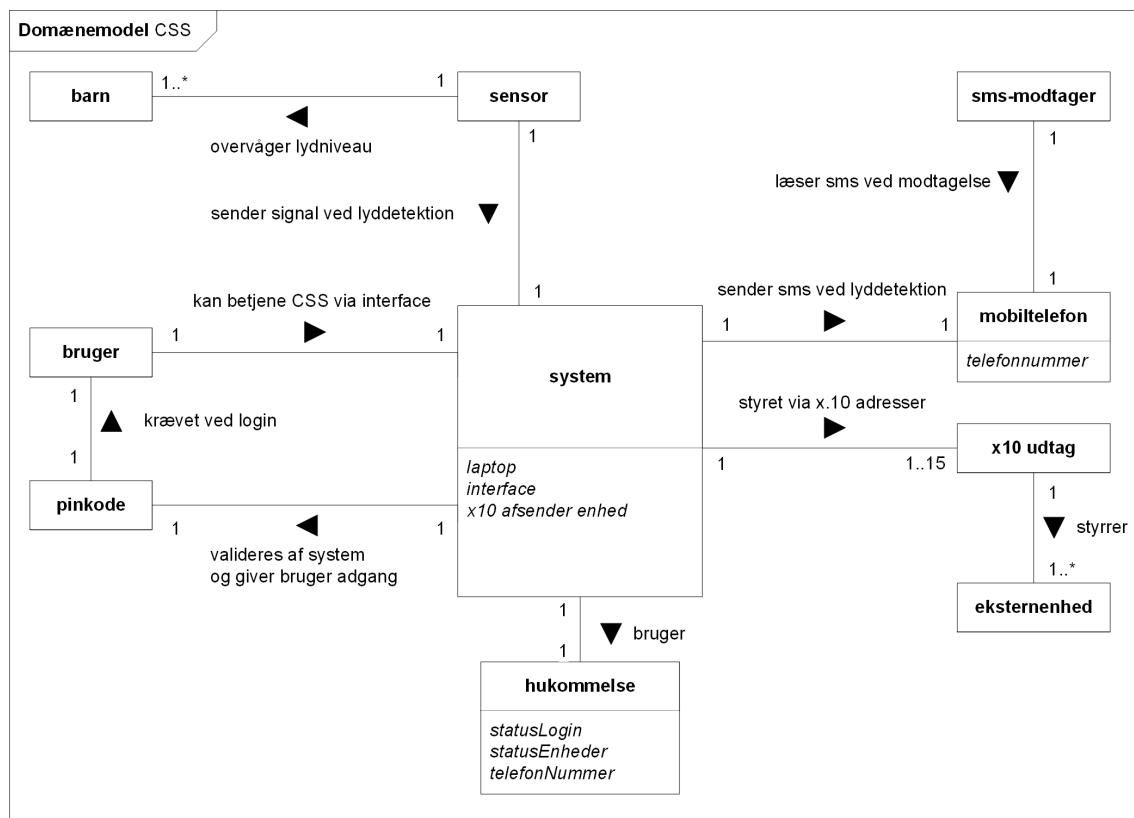
Logbogen og møderefiler er udarbejdet i Google Drevs dokument funktion. Og tidsplanen er udarbejdet i regneark funktionen. På den måde kan alle se og rette i det samme dokument samtidigt.

Systemarkitektur 9

9.1 Hardware arkitektur

Følgende diagrammer viser hvordan hardwarearkitekturen er opbygget.¹

9.2 Domænemodel

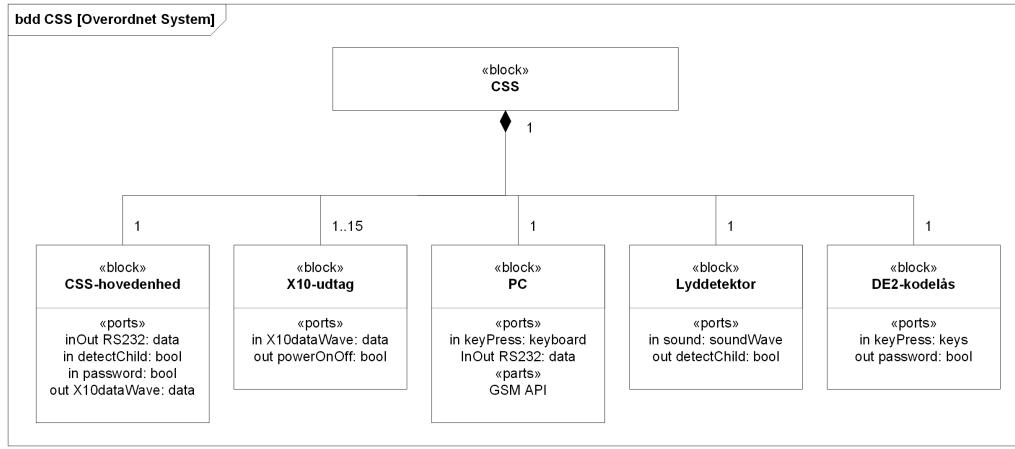


Figur 9.1. Domænemodel

Domænemodellen er udarbejdet i samarbejde med kunden. Denne har til opgave at give et struktureret billede af systemets funktionalitet og sammenhæng. Domænemodellen gør ikke brug af fagudtryk, men pile og kortfattede samt præcise sætninger anvendes for at beskrive sammenhængen mellem blokkene. Dette er med til at opnå en højere forståelse af systemet som helhed for kunden.

¹For yderligere BBD/IBD se projektdokumentation side 28-30.

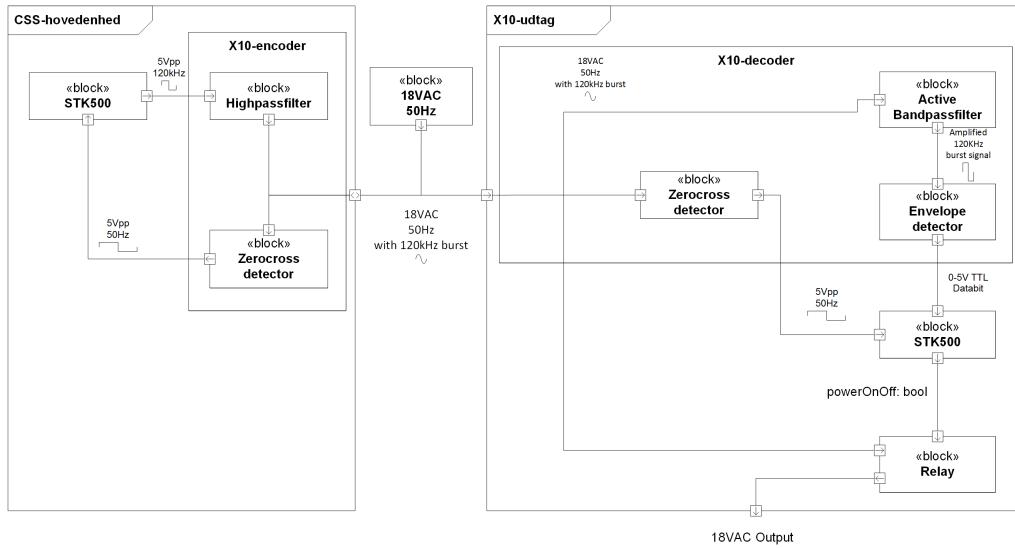
9.2.1 BDD Hardware



Figur 9.2. BDD Hardware

BDD diagrammet giver et overblik over hvad det samlede system består af. Her ses en portbeskrivelse som viser hvilke signaler hver blok består af.

9.2.2 Plantegning over HW



Figur 9.3. Plantegning over HW

Plantegningen over HW giver et overblik over hvordan CSS-hovedenheden og X10-udtaget er forbundet, samt hvilke type signaler der bliver sendt imellem dem.

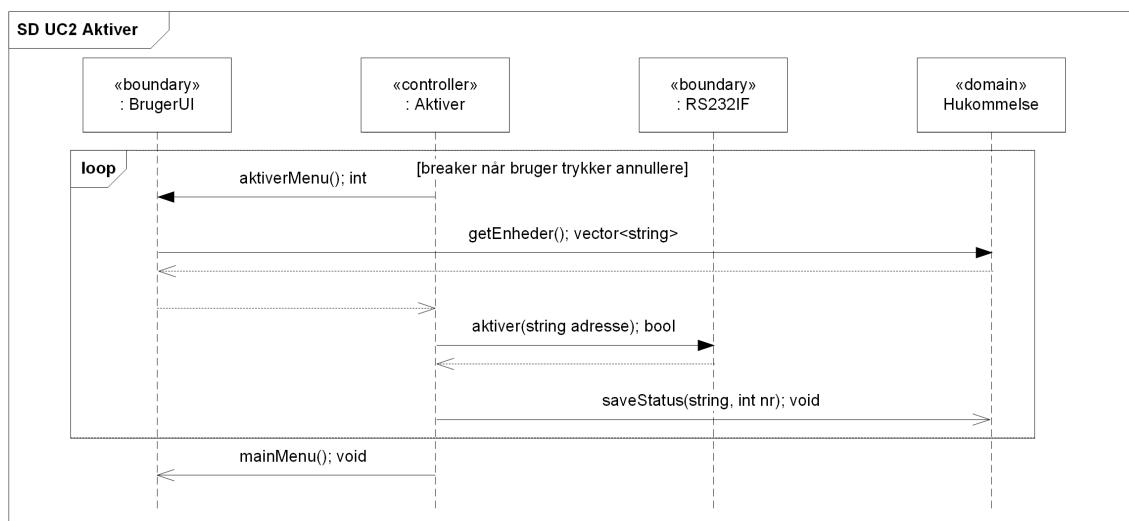
9.3 Software arkitektur

Med udgangspunkt i domænemodellen udviklet i arkitekturfasen er der udviklet applikationsmodeller for hver computer i systemet. Dette giver overblik over de funktionaliteter som skal implementeres på de forskellige platforme.

Applikationsmodellen består af at beskrive hvordan information fordeles i hvert UC. Dette opnåes med tre diagram typer. Sekvensdiagrammer som viser hvordan information bevæger sig sekventielt igennem systemets klasser, et klassediagram som sammenfatter de metoder og relationer som er fundet i sekvensdiagrammet og et tilstandsmaskinediagram som viser et systems forskellige tilstande. Det sidste er udeladt da det ikke er aktuelt for det opbyggede system.

9.3.1 Applikationsmodel for PC (JC)

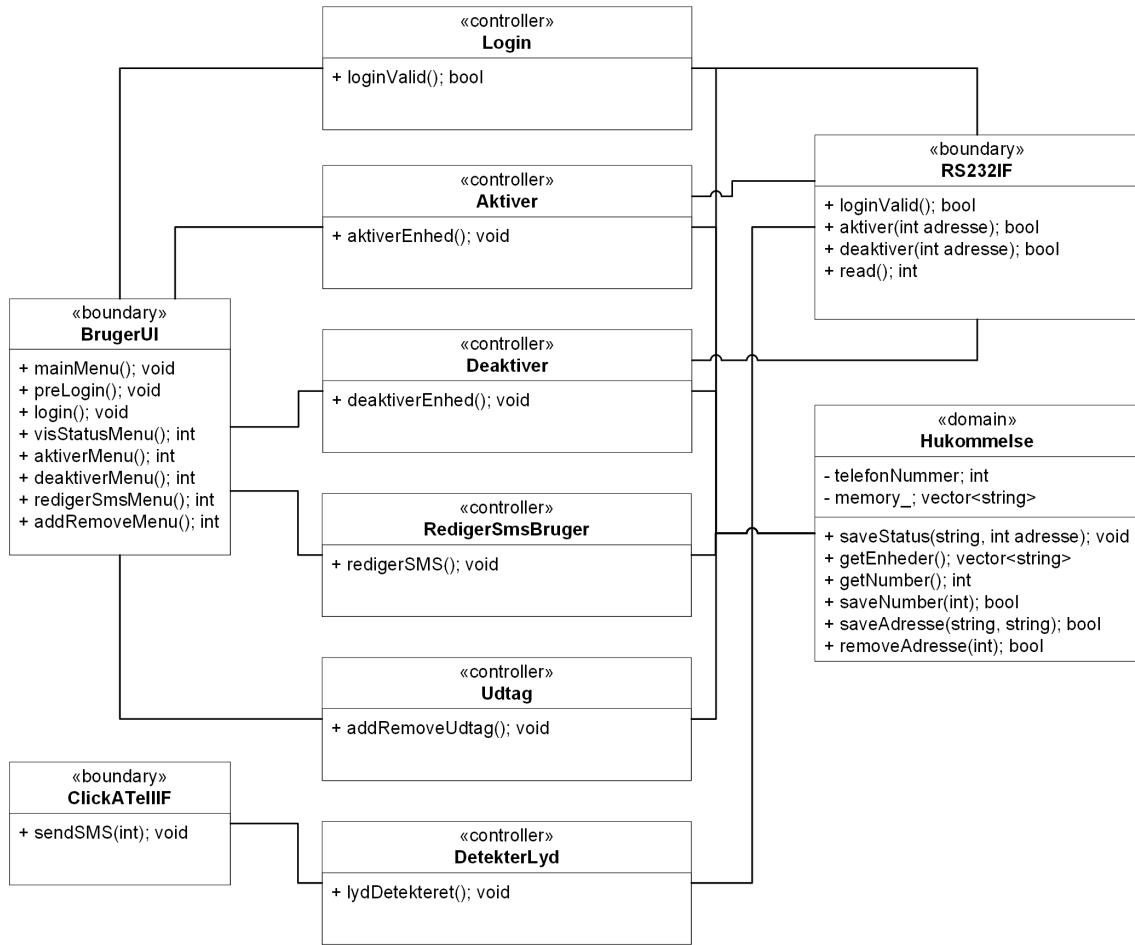
Ud fra domænemodellen og de forskellige UC er der oprettet sekvens diagrammer for hver use case (UC4 og UC7 undtaget, UC4 fordi den er utrolig simpel og UC7 fordi den refererer til UC8) Sekvensdiagrammet for UC2 Aktiver kan ses på nedenstående gifur 9.4. Resten af sekvensdiagrammerne kan ses i projektdokumentationen².



Figur 9.4. Sekvensdiagram UC2

Efter at alle sekvensdiagrammerne er lavet har vi alle vores konceptuelle klasser med metoder som vi nu kan overføre til et klassediagram som ses på figur 9.5.

²Se side 33-34 i projektdokumentationen

**Figur 9.5.** PC klasse diagram

Når det konceptuelle klassediagram er lavet, kan man arbejde videre med et statisk klassediagram samt tilhørende klassebeskrivelser. Herefter kan designet begynde.

9.3.2 Applikationsmodel for CSS-hovedenhed (BS)

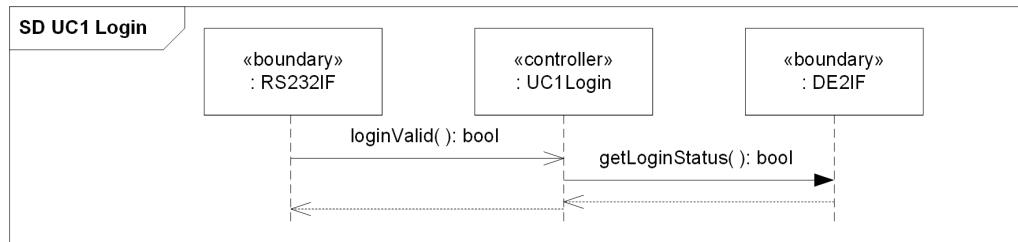
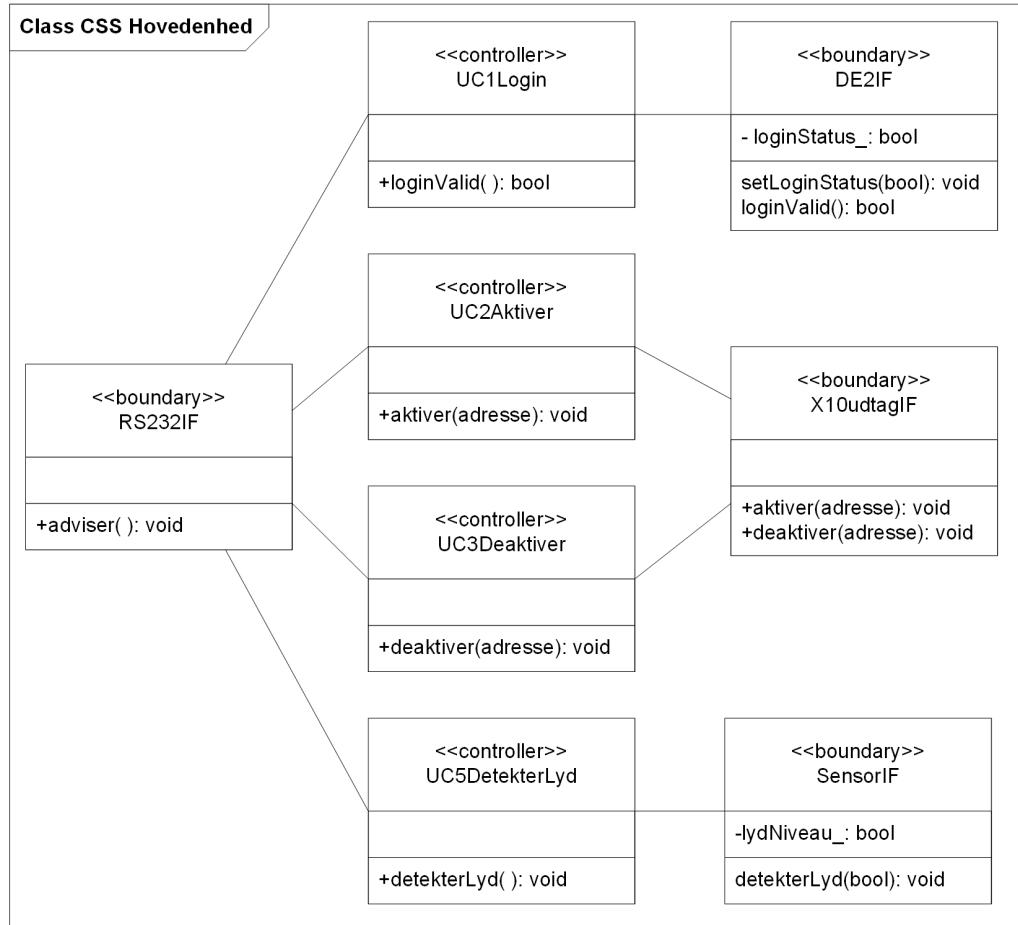
Med udgangspunkt i systemdomænemodellen, figur 9.1, er de konceptuelle klasser for CSS-hovedenheden udledt.

Med dette udgangspunkt laves der sekvensdiagrammer for hvert UC. Disse er vist i figur 9.6. De viser hvordan metodekald imellem de konceptuelle klasser fra domænemodellen og giver et overblik over den basale funktionalitet. Her er kun medtaget UC1. Se projektdokumentationen³ for resten.

Dette resulterer i et klassediagram med grundfunktionaliteterne, se figur 9.7. Denne bruges under implementeringen og ender ud i et statisk klassediagram som beskriver det endelige program med alle hjælpemetoder.

Efter denne analyse er der et klart overblik til implementeringsfasen.

³Se side 43 i projektdokumentationen

**Figur 9.6.** Sekvensdiagram for CSS-hovedenhed**Figur 9.7.** Klassediagram for CSS-hovedenhed

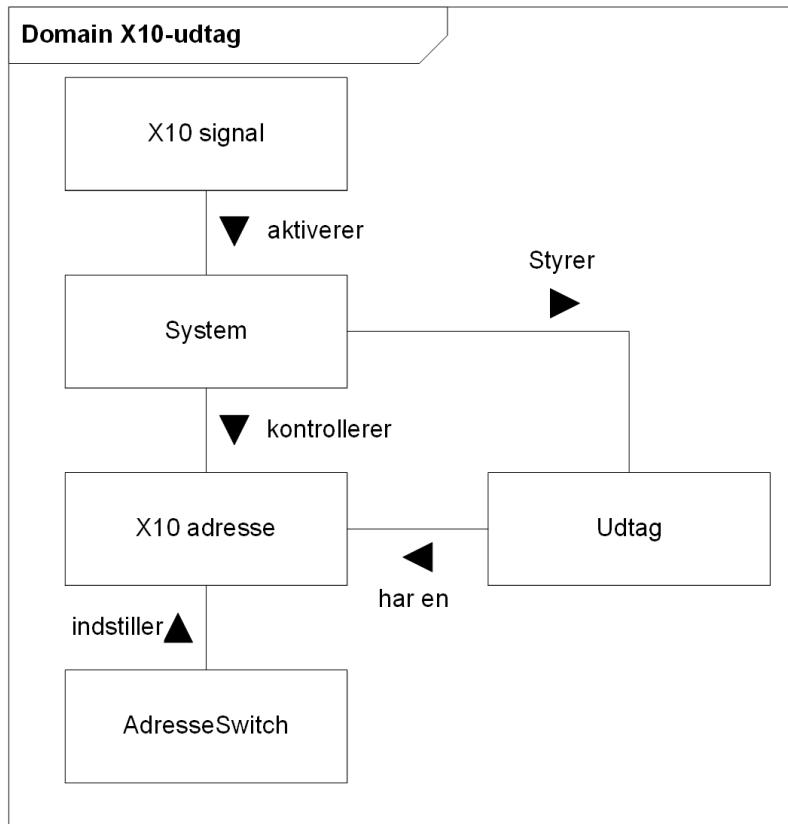
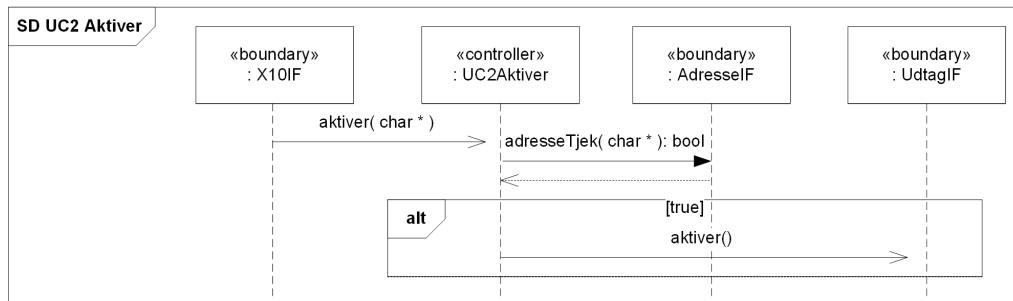
9.3.3 Applikationsmodel for X10-udtag (BS)

Først er der lavet en detaljeret domænemodel for X10 udtaget. Denne er vist i figur 9.8. Denne laves ved at gennemgå UC beskrivelserne og finde de ting som har indflydelse på netop denne del af systemet.

Med dette udgangspunkt laves der sekvensdiagrammer for hvert UC. Disse er vist i figur 9.9. De viser metodekald i mellem de konceptuelle klasser og giver et overblik over den basale funktionalitet. Her er kun medtaget UC2 Aktiver. Se projektdokumentationen⁴ for resten.

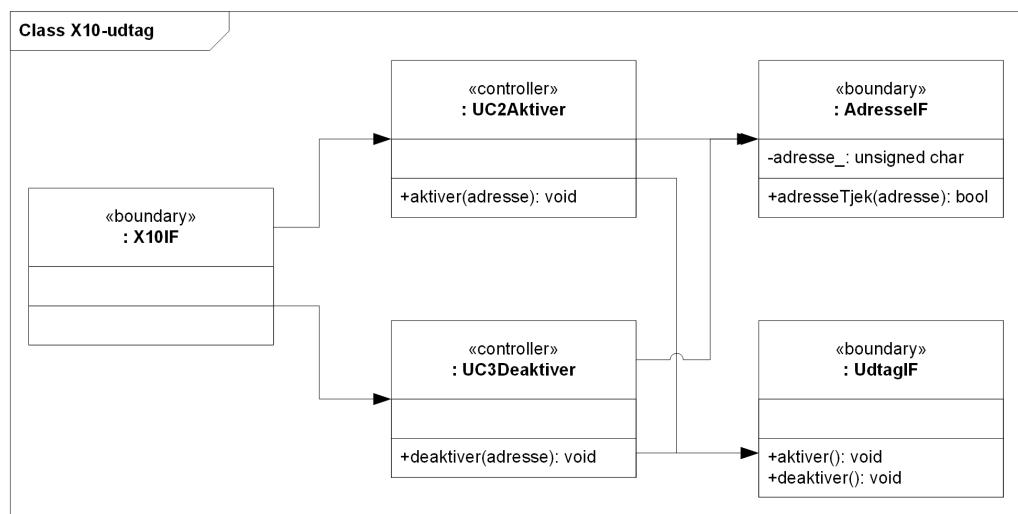
Dette resulterer i et klassediagram med grundfunktionaliteten beskrevet, se figure 9.10.

⁴Se side 57 i projektdokumentationen

*Figur 9.8.* Domænemodel for X10-udtag*Figur 9.9.* Sekvensdiagram for X10-udtag

Denne bruges under implementeringen og ender ud i et statisk klassediagram som beskriver det endelige program med alle hjælpemethoder.

Denne analyse af funktionalitet giver et klart overblik til implementeringsfasen.



Figur 9.10. Klassediagram for X10-udtag

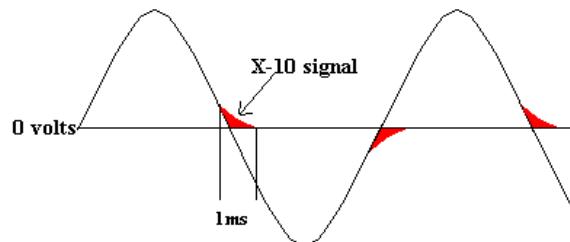
Design 10

10.1 Hardwaredesign

Generelt kan det overordnede hardwaredesign beskrives som noget elektronik der sender information ud på el-nettet og modtages samt analyseres af noget elektronik i den anden ende. Disse vil blive beskrevet herunder som encoder og decoder¹.

10.2 Encoder (JS)

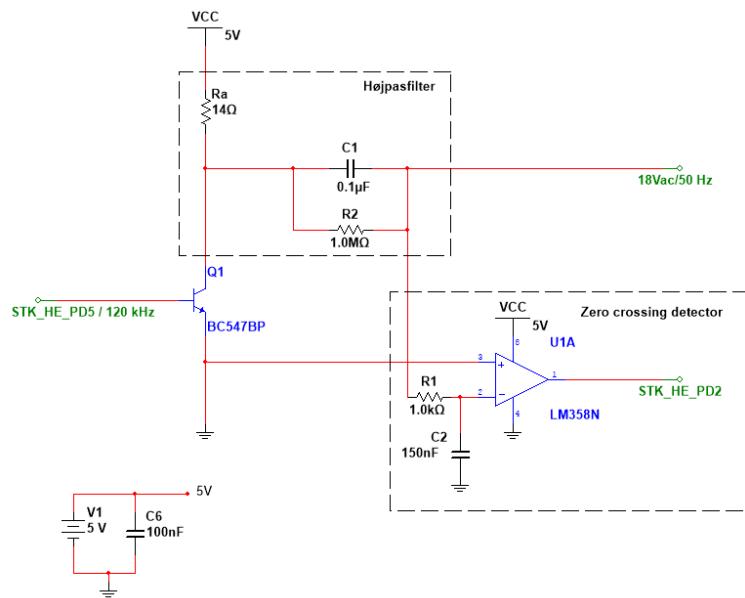
X10-encoderen, er en del af CSS hovedenheden. CSS-hovedenheden genererer X10-bitstrømmen og sender den ud på det eksisterende el-net. Encoderen består af et højpasfilter der lader 120 kHz burst passere mens det blokerer for nettets 50 Hz signal, og en zero crossing detector der detekterer nulgennemgang.



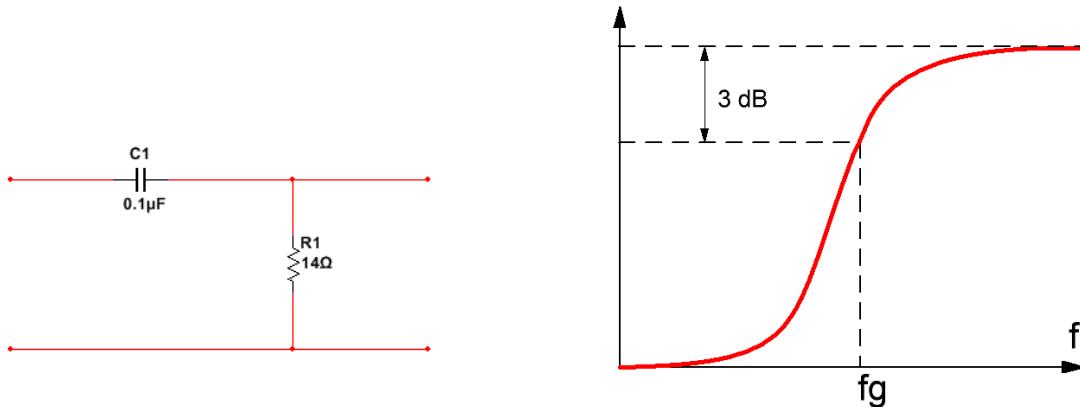
Figur 10.1. 120 kHz burst i zero crossing

Ideen med at sende burst ud på nettet i zero crossing kan ses illustreret på figur 10.1

¹For yderlige beregninger se projektdokumentation side 67-75

**Figur 10.2.** Samlet Encoder

10.2.1 Højpasfilter (JS)

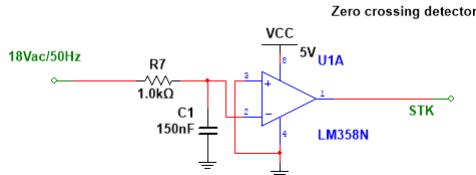
**Figur 10.3.** Højpasfilter med værdier**Figur 10.4.** Kurvekarakteristik for højpas-filter

For at sende X10 kommandoer ud på det eksisterende 50 Hz el-net er det nødvendigt at koble elektroniken direkte herpå og eftersom det elektronik ikke tåler de høje spændinger fra nettet, er det nødvendigt at blokere det signal, men stadig at kunne sende de 120 kHz ud. Dette løses med et højpasfilter.

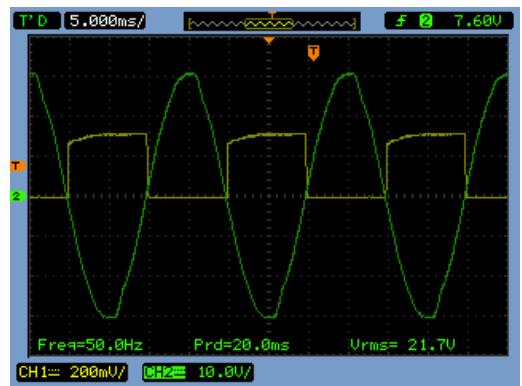
Den ønskede knækfrekvens skal ligge omkring de 120 kHz for at opnå mindst dæmpning herpå. Ved denne knækfrekvens skulle 50 Hz signalet være ubetydelig lille efter filteret.

Kondensatoren forudbestemmes via beregningerne til en værdi på 0,1 nF.

10.2.2 Zero Crossing Detector (SK)



Figur 10.5. Zero crossing detector med værdier



Figur 10.6. Scope billede af Zero crossing detector, CH1 (udgangssignal), CH2 (indgangssignal)

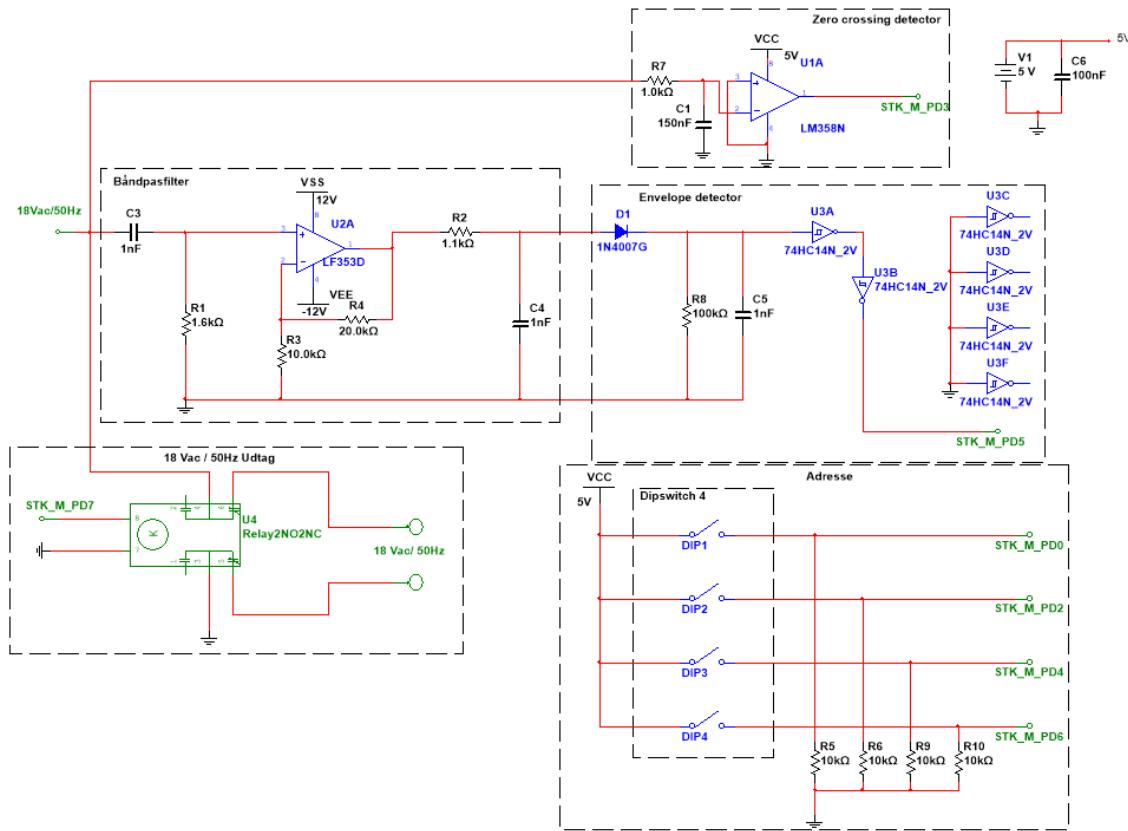
Zero crossing detectoren har til opgave at detektere nulgennemgang, det er et krav for at X10 protokollen kan virke. Der er placeret en zero crossing detector både på encoderen og decoderen, da begge disse består af et STK500-kit der kræver information om nulgennemgang. Opbygning kan ses på overstående figur 10.5, der er anvendt en operationsforstærker af typen LM358N, som toggler udgangssignalet ved hver nulgennemgang se figur 10.6. Operationsforstærkerens positive ben er koblet til stel for at lave et trigger niveau til 0 V.

Modstanden R_7 sidder der bl.a. for at beskytte zero crossing detectoren mod 18 Vac nettet, men sammen med kondensatoren udgør den også et lavpasfilter. Under implementeringen kunne det konstateres at der kom støj ind på zero crossing detectoren, og dette problem løste lavpasfilteret.

Der ønskes at dæmpe 120 kHz signalet, derfor designes lavpasfilteret ud fra en knækfrekvens på 1,0 kHz.

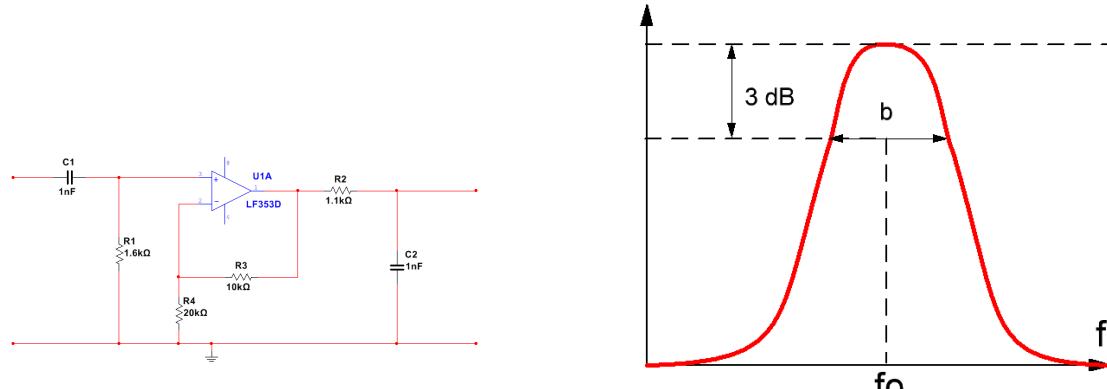
10.3 Decoder (JS)

Decoderen, som er den del i systemet der omdanner burst fra encoderen til X10 kommando, er opbygget af et båndpasfilter, zero crossing detector og en envelope detector. I starten af kredsløbet sidder båndpasfilteret, og dette blokerer for el-nettet og forstærker 120 kHz signalet der kommer fra encoderen. Herefter ledes signalet gennem envelope detector som omdanner burstet til et TTL signal.



Figur 10.7. Samlet Decoder

10.3.1 Båndpasfilter (JS)



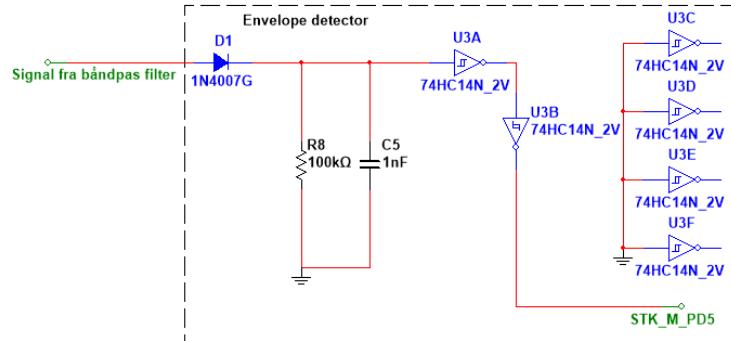
Figur 10.8. Båndpasfilter med værdier

Figur 10.9. Kurvekarakteristik for båndpasfilter

Båndpasfilteret har til opgave at filtrere alle signaler med frekvenser over og under 120 kHz fra samtidig med at det forstærker signalet i båndpasset. Forstærkningen opnås ved at koble en ikke inverterende operationsforstærker med modstande forbundet på tilbagekoblingen, hvis størrelse afhænger af den ønskede forstærkning, mellem et højpasfilter og lavpasfilter som illustreret på figur 10.8. Kurvekarakteristikken er illustreret på figur 10.9.

Da vi ønsker at forstærke 120 kHz signalet, designes højpasfilteret således at knækfrekvensen udregnes til 110 kHz, og for lavpasfilteret beregnes en knækfrekvens til 130 kHz.

10.3.2 Envelope Detector (SK)



Figur 10.10. Envelope detector

Envelope detectorens opgave er at udglatte burstsignalet fra båndpasfilteret og lave det om til et TTL signal som STK500 kan aflæse.

Den er opbygget af en diode, et RC-led og en schmitt-trigger. Dioden har til opgave at sortere alle de negative halvperioder fra og kun sende de positive halvperioder til RC-leddet. Kondensatoren vil derfor kun opfange de positive halvperioder. Kondensatoren er med til at udglatte signalet da den ikke kan nå at aflades på en periode.

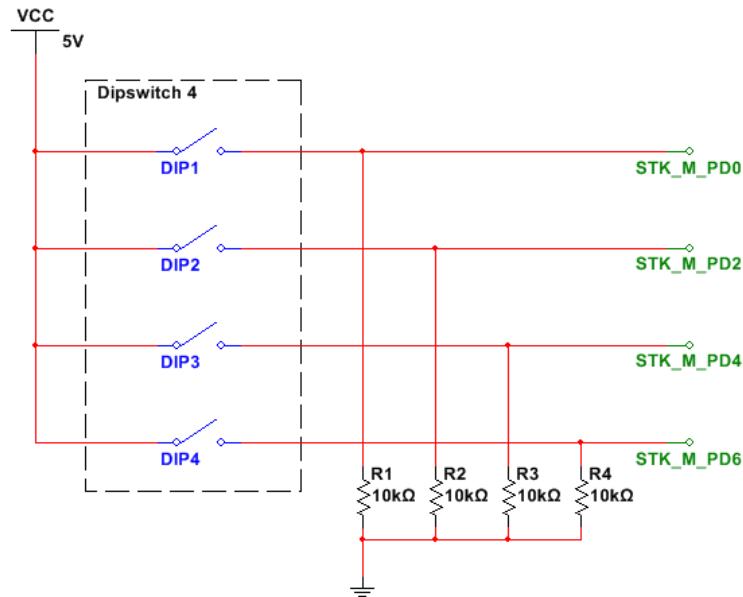
Modstanden R_8 er en afladnings modstand, og den bestemmer hvor hurtigt kredsløbet skal aflades, jo højere modstand jo langsommere går det med at aflade.

Der er anvendt en schmitt-trigger til at lave signalet om til et firkantet signal som STK500 kan aflæses. Grunden til at der er brugt 2 er fordi de er inverterende.

10.3.3 Dipswitch (JS)

For at en X10-kommando bliver udført af det korrekte udtag medsendes en adresse på fire bit. Til at angive den korrekte adresseringen for et udtag, er der lavet fire dipswitches der er forsynet med 5 V og forbundet til STK500 på X10-udtaget, som det er illustreret på figur 10.11.

En åben kontakt vil give LAV(0V) og en lukket kontakt vil resultere i HØJ(5V)



Figur 10.11. Dipswitch for adressering af X10-udtag

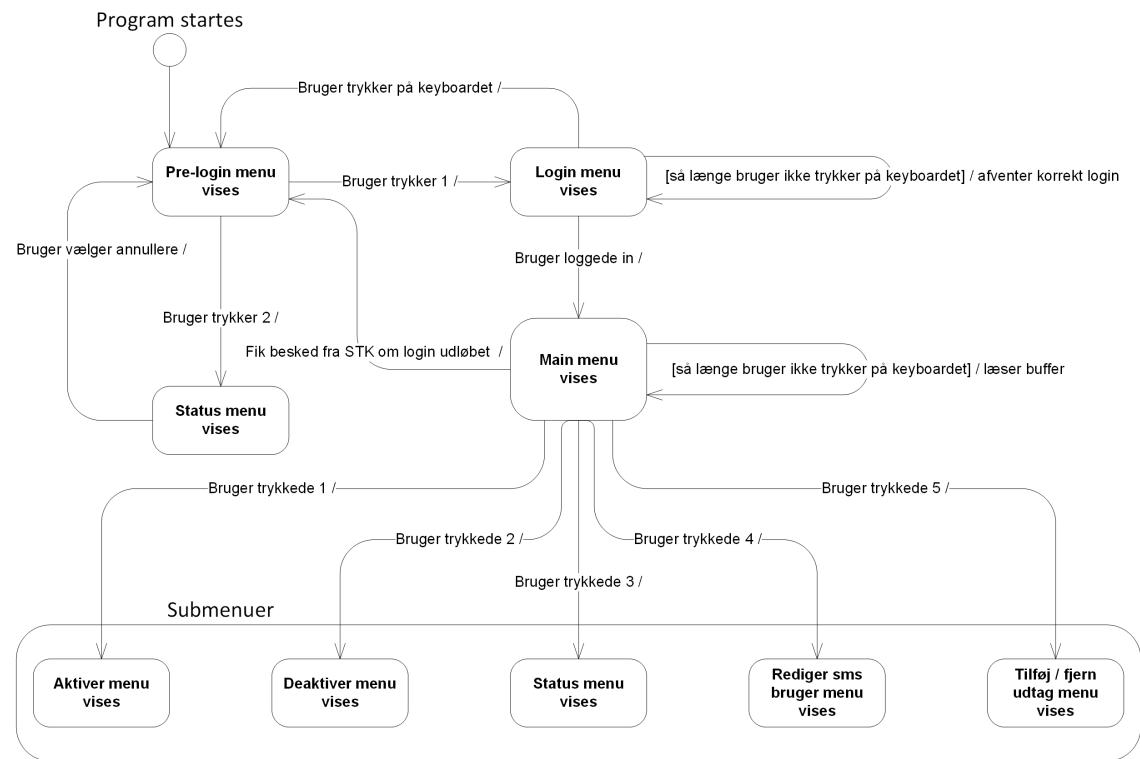
10.4 Software design

Herunder er de forskellige dele af softwaren kort beskrevet.

10.4.1 PC (JC)

Softwaren til PCen, er brugerens grænseflade til at kontrollere systemet. Her kan brugeren manøvrere rundt i de forskellige menuer og udføre de forskellige ting som er beskrevet i UC beskrivelserne.

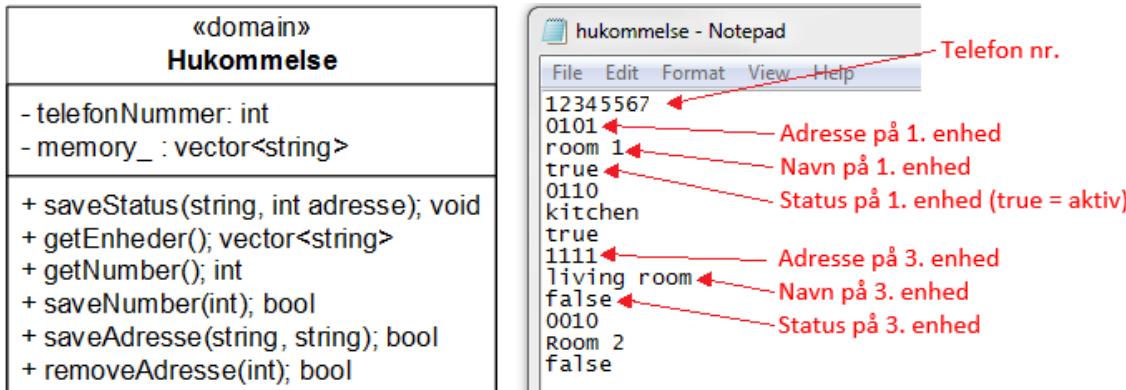
Nedenfor er der illustreret hvordan man kommer frem og tilbage i brugerinterfacet. Udover bruger-input så kan CSS-hovedenheden give PCen besked om at der ikke længere er logget ind hvilket vil sende brugeren fra main menu og tilbage til pre-login menuen. Kun så længe brugeren står i main menu og ikke har trykket på keyboardet, er det muligt for programmet at afsende en sms eller sende brugeren til pre-login menu pga. udløbet login.



Fra alle sub-menuerne kan brugeren gå tilbage til main menu ved at annullere

Figur 10.12. State-machine diagram over brugerflade

En vigtig del af PC softwaren er, at kunne gemme alle de enheder brugeren har og hvilket telefonnummer brugeren har. Dette håndterer Hukommelse klassen som gemmer alt i en text fil og henter dataen ind under opstart. Således kan brugeren lukke programmet, åbne det igen og stadig have de samme enheder samt deres status og adresse. Figur ?? Illustrerer hvordan enhederne og telefonnummeret er gemt i tekstu filen.



Figur 10.13. Hukommelses header og udklip af text filen

10.4.2 CSS-hovedenhed (BS)

Her følger beskrivelse af to hovedfunktionaliteter for CSS-hovedenheden. For at se resten af implementeringen henvises til kildekoden på den vedlagt CD.

Klasse CircBuffer (BS)

Efter som X10 kommunikationen er meget langsom (50 bits/s) bruges en buffer til at holde på kommandoerne indtil de er klar til at blive afsendt. Dette er CircBuffer klassens opgave. Denne er udformet som en cirkulær buffer som kan holde 2 fulde kommandoer. Bemærk at i følge X10-protokollen skal alle kommandoer afsendes to gange. Så der er plads til 4 kommandoer, hvor to af dem altid er de samme.

Klassen er bygget til selv at holde styr på hvilket bit der skal sendes næste gang. Dette forenkler brugen væsentligt, da udtagning af data fra bufferen sker fra en interrupt service rutine (ISR). Denne rutine er beskrevet i detaljer senere.

Alle kommandoer termineres med '\0' karakteren.

Klasse X10IF (BS)

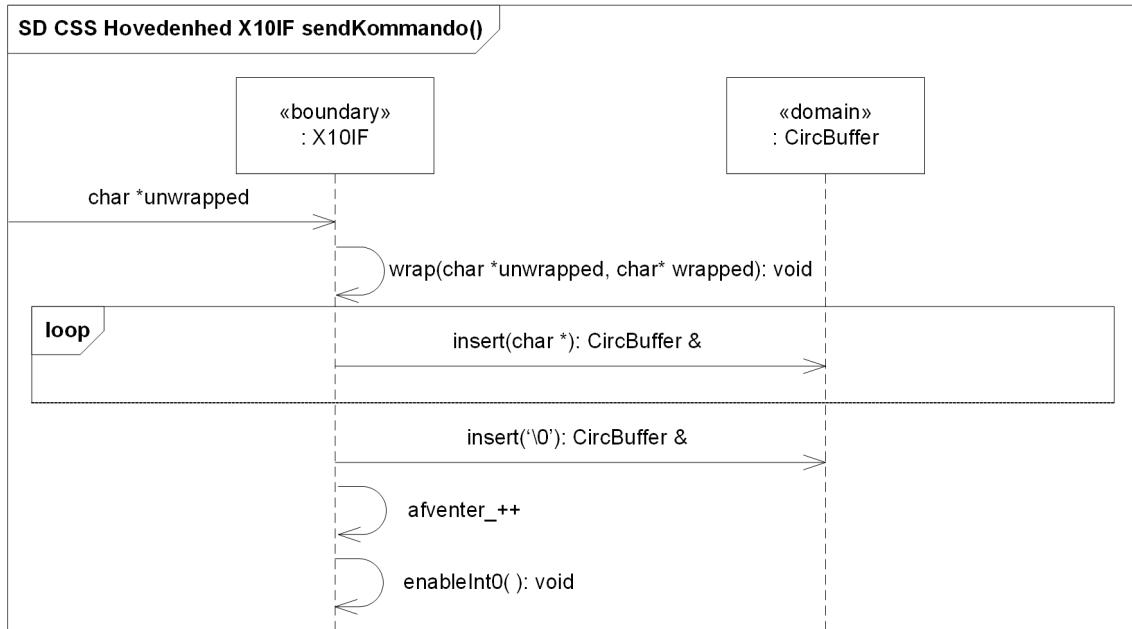
En af de kritiske og avancerede klasser er X10IF klasserne på hhv. CSS-hovedenheden og X10-udtaget. I designfasen er der udviklet sekvensdiagrammer som beskriver nogle af metoderne og sammenspillet til andre klasser.

Funktionaliteten for metoden sendKommando() i X10IF klassen, på CSS-hovedenheden, har som ansvar at afsende en fuld X10-kommando, iht. protokollen, ud fra en parameter, formateret som vist i tabel 10.1.

Tabel 10.1. Parameter opbygning til sendKommando() metoden i X10IF i CSS-hovedenhed

Byte	0	1..4
Data	Kommando	Adresse

Metoden skal først omskrive den modtaget kommando til en X10 bit-streng. Her efter indsættes alle bits i en buffer, hvorfra de afsendes når der detekteres et zero cross. Sekvensen er vist i figur 10.14.

**Figur 10.14.** Sekvensdiagram for metoden sendKommando() i X10IF klassen på CSS hovedenheden

I figur 10.15 er vist kildekoden for sendKommando()-metoden.

ZeroCrossInt funktion (BS)

I tilfælde af et interrupt signal på INT0 benet på CSS-hovedenheden køres en bestemt funktion i microcontrolleren. Dette er kaldet en ISR. Forløbet for denne er beskrevet i sekvensdiagrammet på figur 10.16. Først kontrollerer den om der er nogle kommandoer i kø. Der efter henter den det næste bit der skal afsendes fra bufferen. Ud fra værdien bestemmer den om der skal tændes for 120 kHz frekvensen i timeren. Når en kommando er helt sendt nedskriver den køen og hvis køen er tom slår den interruptet fra.

10.4.3 X10-udtag (JSA)

Objektorienteret programmeret software der styrer X10-udtaget hvorfra der læses zero cross og signal input fra hardwaren og herefter udføre funktioner som beskrevet i use case beskrivelserne.

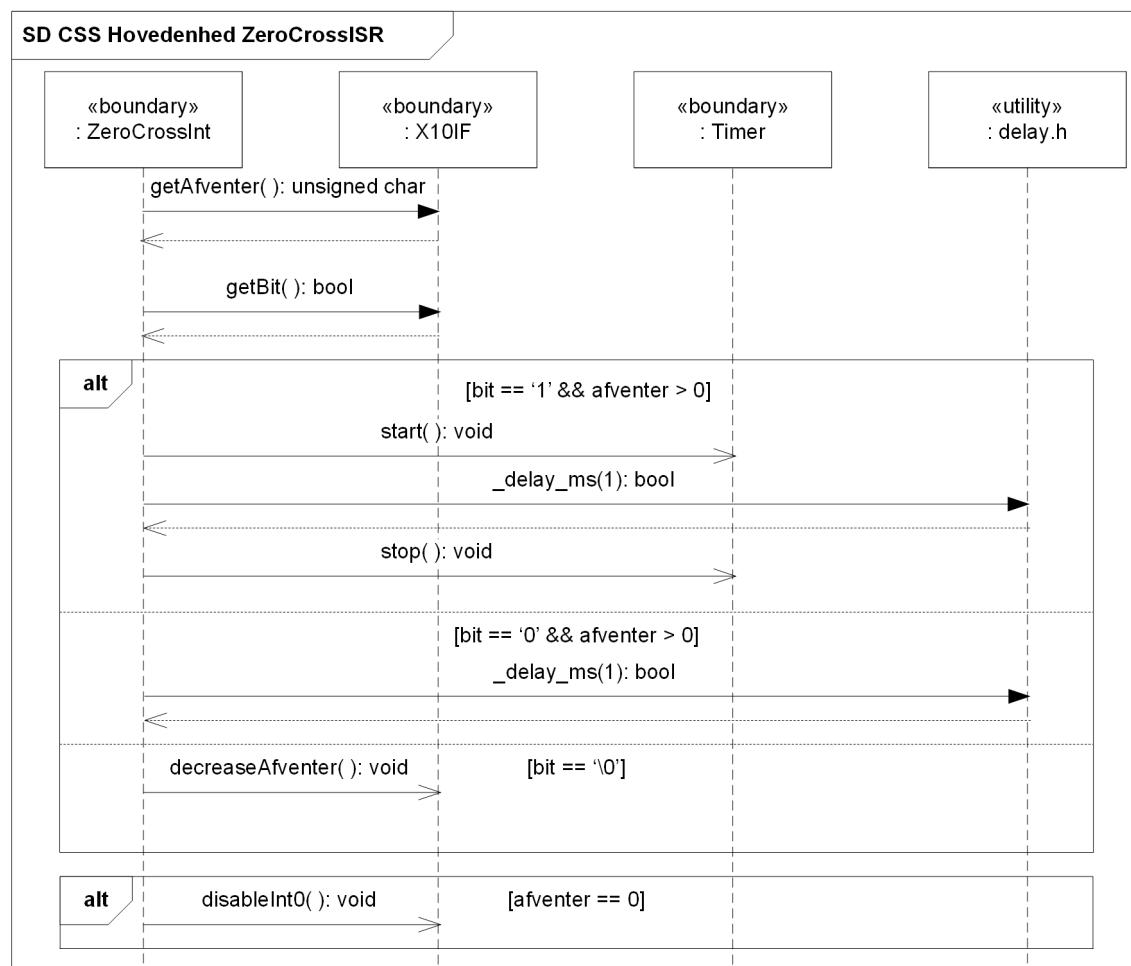
Når der læses et zero cross signal på interrupt INT1 benet køres interrupt service rutinen som læser et højt- (logisk '1') eller lavt- (logisk '0') signal input på PD5 benet. Det modtaget bit bliver sendt til insertX10bit funktionen, som er vist i figur 10.17, som lagrer de modtaget bit løbende i et char array kaldet X10Array_. Funktionen kontroller løbende

```

1 // Indskriver X10 kommando i bufferen
2 void X10IF::sendKommando(char *unwrapped)
3 {
4     // Indholder kommando i X10 format
5     char wrapped[X10_COMMAND_LENGTH];
6     wrap(unwrapped, wrapped);
7
8     // Indsaet bits i bufferen 2 gange
9     for(unsigned char j = 0; j < 2; j++)
10    {
11        // Indsaet bits
12        for(char i = 0; i < X10_COMMAND_LENGTH; i++)
13            buffer_.insert(wrapped + i);
14    }
15
16    // Terminer bufferen
17    char termChar = '\0';
18    buffer_.insert(&termChar);
19
20    // Foroeg antallet af kommandoer der venter paa at blive sendt
21    afventer_++;
22
23    // Aktiver ZeroCrossInt interrupt
24    enableInt0();
25 }

```

Figur 10.15. Kodeudsnit for sendKommando()-metoden fra X10IF klassen



Figur 10.16. Sekvensdiagram for INT0 ISR på CSS-hovedenheden

de fire første pladser i arrayet for STX kommandoen, som er X10 signalets start kommando. Når funktionen detekterer at der er modtaget en STX kommando, lagrer den de næste 24 bit der bliver modtaget i arrayét, som derefter bliver sendt til funktionen unwrapX10Array.

```

1 void X10IF::insertX10bit(char X10bit)
2 {
3     // Indsaetter X10bit i X10Array_ paa plads X10ArrayPlads_
4     X10Array_[X10ArrayPlads_] = X10bit;
5
6     // Ser efter STX match paa X10Array_[0], [1], [2] og [3], og sender fyldt X10Array_ ved match
7     if (X10ArrayPlads_ == 0 && X10Array_[0] != 1) // STX[0] kontrol af bit[0]
8     {
9         X10ArrayPlads_ = 0; // Nulstiller X10ArrayPlads_ hvis det ikke matcher med STX[0]
10    }
11    else if (X10ArrayPlads_ == 1 && X10Array_[0] != 1 && X10Array_[1]) // STX[1] kontrol af
12        bit[1]
13    {
14        X10ArrayPlads_ = 0; // Nulstiller X10ArrayPlads_ hvis det ikke matcher med STX[1]
15    }
16    else if (X10ArrayPlads_ == 2 && X10Array_[0] != 1 && X10Array_[1] != 1 && X10Array_[2]) // //
17        STX[2] kontrol af bit[2]
18    {
19        X10ArrayPlads_ = 0; // Nulstiller X10ArrayPlads_ hvis det ikke matcher med STX[2]
20    }
21    else if (X10ArrayPlads_ == 3 && X10Array_[0] != 1 && X10Array_[1] != 1 && X10Array_[2] != 1
22        && X10Array_[3] != 0) // STX[3] kontrol af bit[3]
23    {
24        X10ArrayPlads_ = 0; // Nulstiller X10ArrayPlads_ hvis det ikke matcher med STX[3]
25    }
26    else if (X10ArrayPlads_ < 27) // Taeller X10ArrayPlads_ en op ved STX match indtil der er
27        modtaget 28 bits
28    {
29        X10ArrayPlads_++; // Taeller X10ArrayPlads_ en op
30    }
31    else if (X10ArrayPlads_ >= 27) // Sender X10Array_ naar der er modtager 28 bits
32    {
33        unwrapX10Array(X10Array_); // Sender X10Array_ til funktionen unwrapX10Array();
34        X10ArrayPlads_ = 0; // Nulstiller X10ArrayPlads_ naar X10Array_ er afsendt
35    }
36 }
```

Figur 10.17. Kodeudsnit for funktionen insertX10bit() fra X10IF.cpp

I sekvensdiagrammet, som vist i figur 10.18, er der vist hvordan unwrapX10Array funktionen dekoder det modtaget array. Metoden dekoder den modtaget X10 formateret kommando og adresse. Og sender herefter den dekodet adresse via den matchet dekodet kommando til den tilhørende controller.



Figur 10.18. Sekvensdiagram for metoden unwrapX10Array() i X10IF klassen på X10-udtaget

Implementering 11

11.1 Hardware implementering

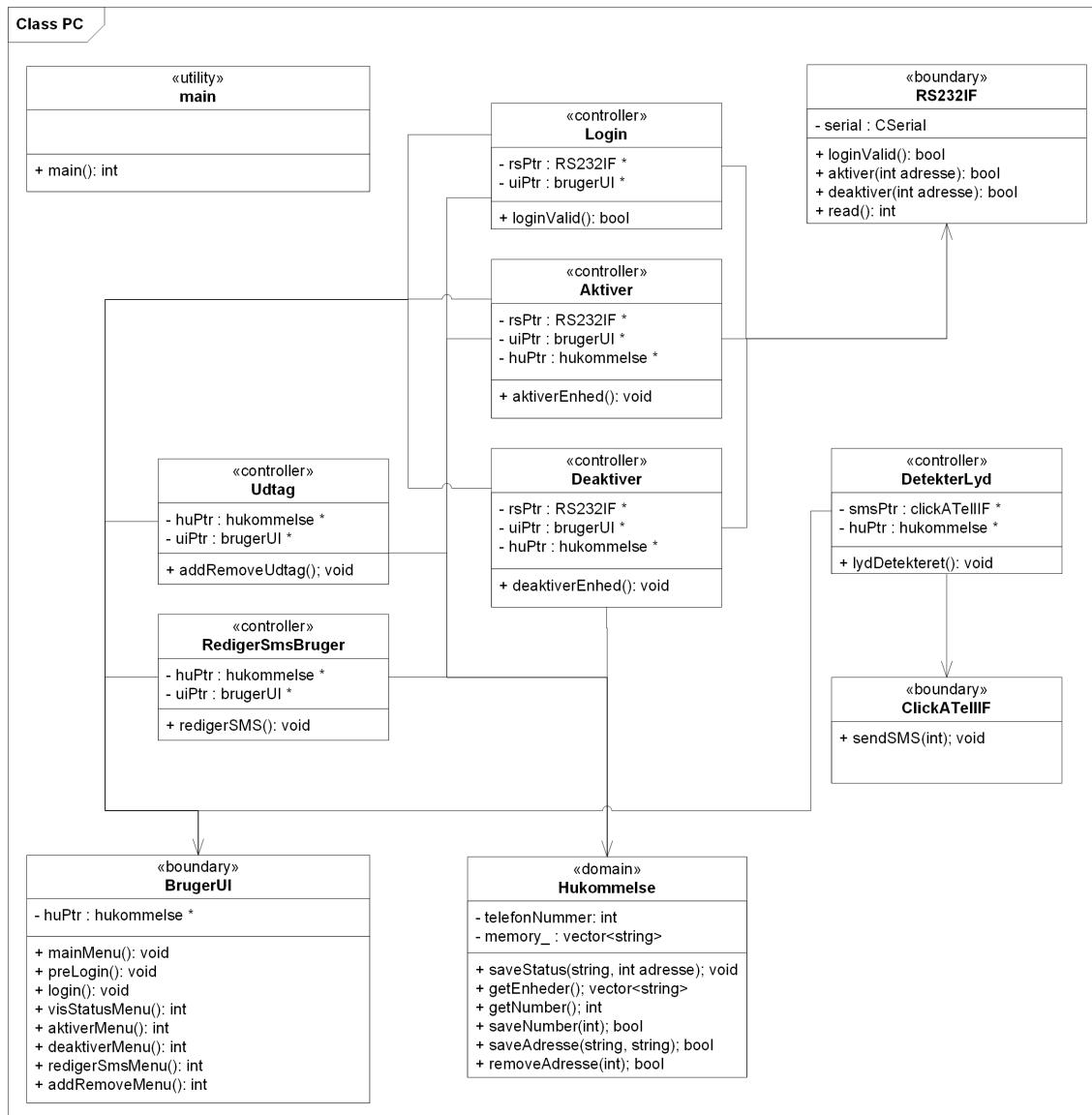
Encoder og decoder er bygget på veroboard¹ ud fra kredsløbsdiagrammerne. Knappen til babyalarmen er tilføjet på encoder printet. 18 Vac / 50 Hz udtaget med tilhørende relæ er tilføjet på decoderprintet.

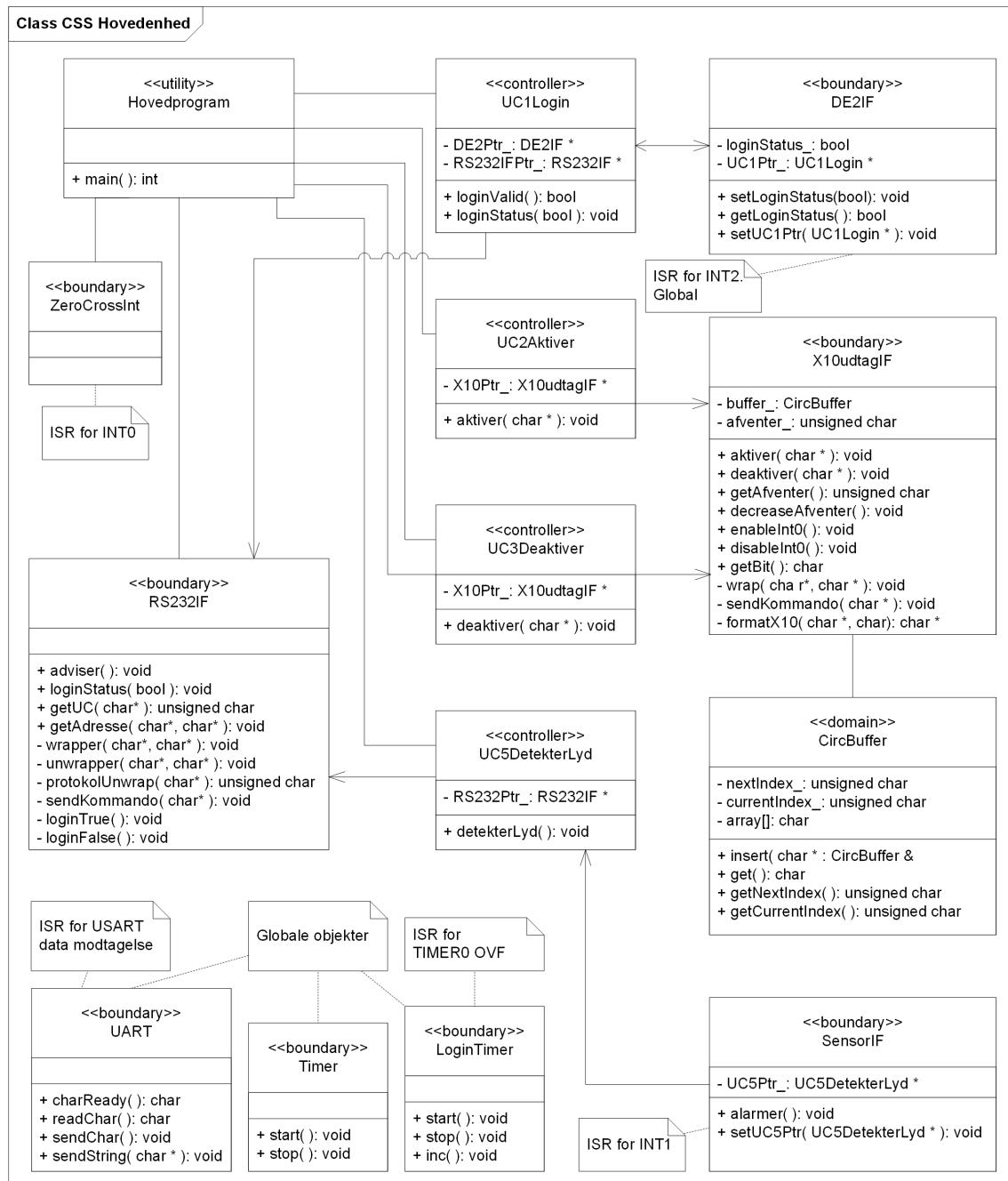
11.2 Software implementering (BS)

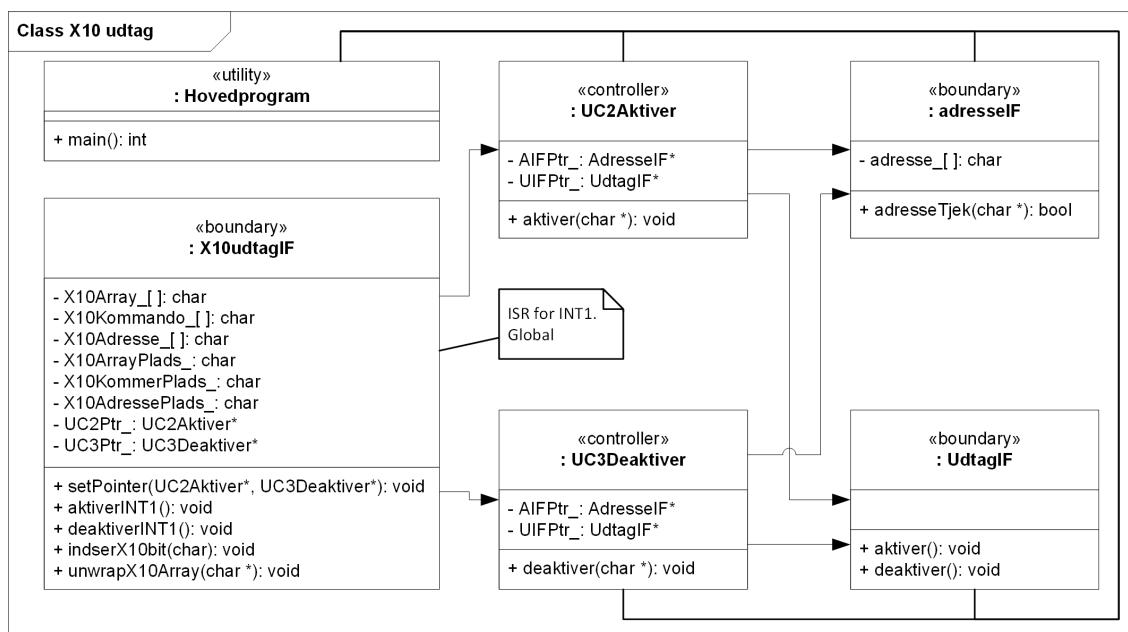
Efter designfasen er software brudt så meget ned at det er lige til at programmerer. Programmeringen er sket ud fra de klassediagrammer som blev udviklet i applikationsmødellen. Da implementeringen giver en endnu større detaljgrad bliver klassediagrammerne også opdateret til statiske klassediagrammer. De er vist i figur 11.1, figur 11.2 og figur 11.3.

Kildekode for alt software kan findes på vedlagte CD under mappen "SW Kildekode".

¹Billeder af begge veroboard ses under integrationstesten i dokumentationen, side 106

*Figur 11.1.* Statisk klassediagram for PC

*Figur 11.2.* Statisk klassediagram for CSS-hovedeneded

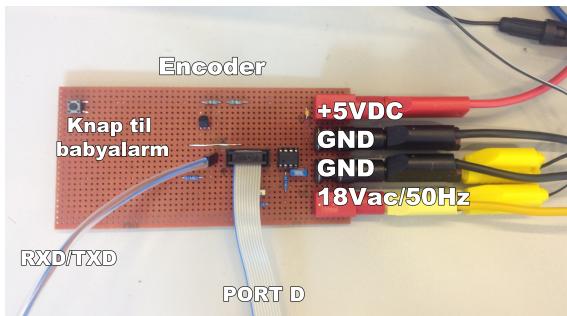


Figur 11.3. Statisk klassediagram for X10-udtag

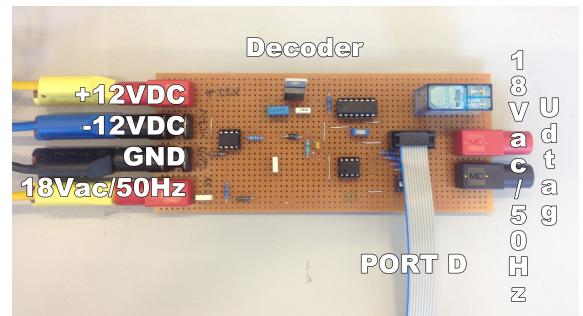
Resultater 12

Resultatet af denne opgave er endt ud i en prototype af Child Security Systemet, dokumentationen af dette system, samt denne rapport omhandlende systemet.

Igennem en velstruktureret arkitektur er det lykkedes at udarbejde et system som kan udføre de ønskede opgaver. Disse opgaver udføres af systemet dels gennem RS232 kommunikation mellem PC og CSS-hovedenhed og dels med X10 kommunikation mellem CSS-hovedenhed og X10-udtag. User-interfacet har opnået et flow som specifiseret i arkitekturen. Prototypen er opbygget på veroboarder, disse ses på nedenstående figure 12.1 og 12.2. Først er der udarbejdet printudlæg for hhv. encoder og decoder, disse er herefter produceret i loddelab.

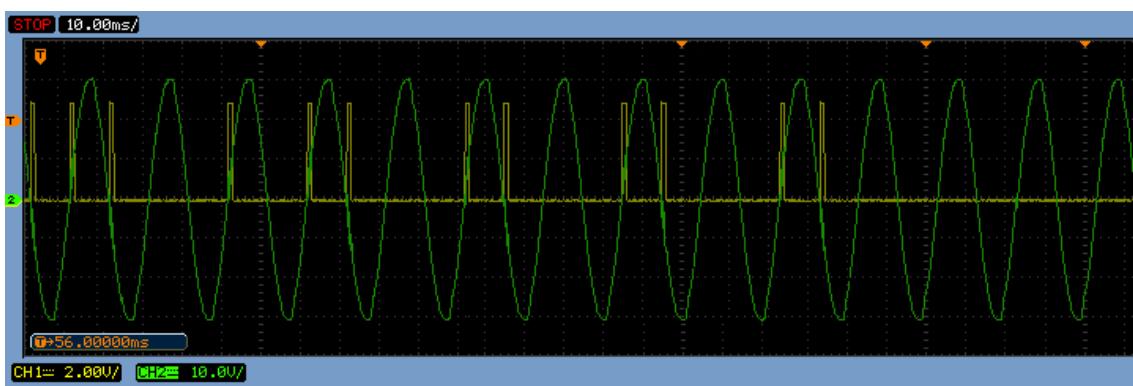


Figur 12.1. Encoder print



Figur 12.2. Decoder print

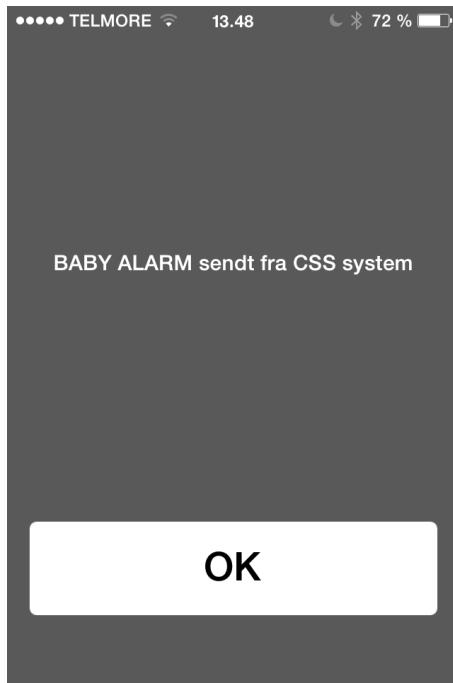
Det er ikke lykkedes at få X10-udtag softwaren til at modtage X10-bitstrømmen, derfor er der taget et scopebillede af bitstrømmen med oscilloscope indstillingen single sweep. Ses på figur 12.3.



Figur 12.3. Scopebillede af et testsignal

Babyalarmens advarsels-SMS, kommer som en flash SMS. Dvs. SMS'en vises direkte på skærmen for brugeren, herved kan det ikke undgås at brugeren ser beskeden, når mobilen

tages i brug. Skærbilledet som møder SMS-modtageren, på en iPhone, vil se ud som på figur 12.4



Figur 12.4. Flash SMS

RS232 kommunikationen virkede rigtig godt efter der blev gennemført modul test både på RS232IF klassen på PC og CSS-hovedenhed. Modultest kan ses i rapport dokumentationen¹.

¹Hhv. side 98 og 100 i projektdokumentationen

Opnåede erfaringer (JS)

13

Som en af de første ting kan det nævnes at alle i gruppen har lært og fået erfaringer med at udarbejde større rapporter i L^AT_EX hvilket kan blive rigtig brugbart senere hen blandt andet i forbindelse med bachelor projekt.

Vi har som medlemmer, og gruppen som helhed, skulle arbejde noget mere selvstændigt end sidste semesterprojekt, sammen med at alt dokumentation skulle udarbejdes fra bunden. Dette har været med at til give et overblik over sådanne projekters forløb og givet god erfaring med at arbejde struktureret som gruppe. Ydermere har vi haft rigtig god erfaring med at holde korte og konkrete møder, hvor vi hurtigt dannede et overblik over hvor langt vi hver især var nået siden sidst, og hvad der skulle fokuseres på ind til næste møde.

Fremtidigt arbejde (SK)

14

Et fremtidige arbejde kunne være at udarbejde de dele i projektet som ikke blev nået. Det er at få X10-udtaget til at fungere helt korrekt, samt at få lavet et mere kompakt design af vores print så det kunne implementeres lettere og mere enkelt.

Lyddetektoren bliver aktiveret ved hjælp af en trykkontakt, et fremtidige arbejde kunne være at færdigudvikle denne. Der skulle laves kredsløbstegninger samt implementering og test. En udvikling af de beskrevet låsemekanismer skulle også udarbejdes for at der kunne leveres et komplet system.

Brugermenuen kunne udvikles med et smartere og mere brugervenlig design. Hvis systemet skulle leveres til andre lande end Danmark ville det være en fordel at der kunne vælges sprog, det gør at systemet kan sælges i større omfang. En app til smartphones kunne laves så der var mulighed for at betjene systemet fra andre steder end i hjemmet.

Konklusion (PO, SK)

15

Formålet med projektet har været at konstruere et system for at øge børns sikkerhed i hjemmet. Det skulle være muligt at styre bestemt 230Vac udtag i hjemmet, og herigennem låse f.eks skuffer eller kogeplader. Ydermere ønskede vi at modernisere den udbredte babyalarm til en mobil løsning.

Vi har opnået en prototype, som kan afsende en X10 kommando. PC softwaren, samt CSS hovedenhedens software virker som tiltænkt. Vi har her til slut ikke fået X10-udtagets software til at virke sammen med decoder hardwaren. Vi oplever at den første bit ikke kommer på det tidspunkt den skal, mens resten af bitrækken kommer som ønsket. Dette kan skyldes en langsom opstart i hardwaren eller en software fejl. Prototypen er opbygget på veroboard, dette valg blev taget, da vi oplevede for mange løse forbindelser i fumlebrædderne. Det er lykkedes at få systemet til at afsende SMS.

Samarbejdet har fungeret godt og ud fra gruppektrontrakten har gruppen haft et fælles standpunkt og grundlag for projektgenemførelsen. Gennem ASE-modellen er gruppen delt i 2 teams, der har haft hhv. hardware og software som fokuspunkt. Dette samarbejde har fungeret godt, både i de to teams samt gruppen samlet. Gruppen har arbejdet struktureret og professionelt med projektet. Gruppemøderne har spillet en væsentlig rolle for at opretholde strukturen.

Vi har igennem L^AT_EXoplevet opstartsvanskeligheder, men tilslut er vi rigtigt glade for at have valgt den løsning.

Individuel konklusion

16

16.1 Bjørn Sørensen

Min overordnede konklusion på dette projektforløb er en riktig lærerig og succesfuld implementering af alle de metoder og arbejdsredskaber som er blevet introduceret på dette semester. Det har været en riktig spændende proces fra den indledende brainstorm til det endelige projekt og jeg syntes generelt gruppen har været riktig god til at følge den samme struktur og arbejdsgang med udgangspunkt i ASE-modellen.

Med en baggrund som elektronikmand har jeg skulle koncentrerer mig en del om ikke at blande mig i elektronikken da jeg har været i software gruppen og i øvrigt læser på IKT linjen. Dette har dog været en god lejlighed til at få afprøvet de grænseflade aftaler vi har indgået og stole på dem.

Vi havde i gruppen fra starten store ambitioner, men har i løbet af perioden fået skåret idéen helt ind til benet. Dette syntes jeg egentlig har været lidt ærgerligt da den oprindelige idé var riktig god, men må også erkende at vi havde en fast deadline som ikke gjorde det muligt at nå helt i mål med idéen.

16.2 Jakob Schmidt

Sammenlignet med 1. semester projektet, så har processen i det her projekt været meget mere struktureret. Sammenholdet og overblikket mellem hvert projektmedlem har været riktig godt. Det gode samarbejde og overblik skyldes at vi jævnligt har holdt møder, både som selvstændig gruppe, men også sammen med vejleder. Desuden er der afholdt enkelte trivselsrunder, hvor vi hver i sær har skulle fortælle hvordan vi selv følte det gik i projektet.

Inden vi fik det endelige overblik over hvor omfattende vi kunne lave vores projekt, var det tydeligt at vi havde lidt for stor ambitioner til projektes omfang. Dette blev skåret ned efter første reviewmøde da vi fra anden vejleder blev anbefalet på det kraftigste at revurdere vores use cases.

Personligt har jeg beskæftiget mig med elektroniken i projektet, det kom som naturligt valg da jeg læser på elektro linjen. Fagligt har der været nogle komplikationer med at få de enkelte moduler til at virke som tiltænkt. Projektet har været spændende men samtidig udfordrende, specielt eftersom den nødvendige teori først var helt på plads i den sidste fase af forløbet.

Overordnet set er jeg riktig tilfreds med resultatet af vores projekt, på trods af at vi langt fra fik realiseret alle de ting vi havde uttænkt fra første brainstormmøde. Efter revurdering

af projektets omfang er alle dele desværre stadig blevet færdige, men ideen er der og de vigtigste dele blev implementeret.

16.3 Jeppe Stærk

Dette semesterprojekt har været et utroligt lærerigt projekt hvor der har været stor fokus på dokumentationen af arbejdet, vi har som gruppe haft ugentlige vejleder- og gruppe møder, hvor vi har vendt det løbende arbejde og snakket om hvilken retning vi skulle i. Da vi nåde implementeringsfasen, var vi som følge af forarbejdet med dokumentationen, klar til at gå direkte i gang med hardware- og softwareudviklingen.

Da jeg er på IKT linjen valgte jeg at være med i software udviklingen, hvor jeg tog mig af X10-udtaget og funktionen til at sende SMS via. X10-udtaget blev lavet med udgangspunkt i objektorienteret programmering. I processen med at lave SMS funktion, lå der først en del forudundersøgelse i at finde en løsning der kunne implementeres i systemet, valget faldt på ClickATell som via en http API kan sende SMSer.

Jeg syntes at vores gruppe arbejde har fungeret rigtig godt og harmonisk. I løbet af projektet har vi udviklet os til at arbejde yderst struktureret. Ligeledes har det ugelige vejledermøde været rigtig godt til at give et godt overblik over hvilken retning vi skulle i og hvor vi løbende skulle lægge vores fokus. Så alt i alt er jeg yderst tilfreds med dette semesterprojekt.

16.4 Jesper Christensen

Semesterprojektet har budt på mange spændende udfordringer som har gjort det rigtig lærerigt. Vi valgte i gruppen at introducere nogle nye værktøjer til projekt skrivning i LaTeX og en anden cloud i GitHub. Det var en lidt hård process i starten da man ikke rigtig kendte programmerne, men i sidste ende har det virkelig givet pote og tror helt sikkert det er noget jeg vil fortsætte med at bruge i kommende projekt forløb.

Vi har i gruppen haft rimelig klare linjer fra starten, omkring hvem der lavede hvad. Vi delte os op i en software- og hardwaregruppe og det synes jeg har været rigtig fint og vi har haft et godt samarbejde i software gruppen og i vores ugentlige møder har vi kunnet følge med i hvad den anden gruppe fik lavet.

At lave softwarearkitekturen uden egentlig at have kodet noget var en udfordring, men det har været super lærerigt. Synes helt sikkert at jeg er bedre rustet til at lave systemarkitektur en anden gang efter at have prøvet kraftet med det her. Alt i alt har det været et rigtig godt og lærerigt projekt forløb.

16.5 Mick Kirkegaard

Vi besluttede i starten af projektfasen at bruge skriveredskabet LaTeX, og jeg blev lidt uventet primus-motor i dette værktøj. Dette har sammen med SVN-værktøjet GitHub givet en del hovedpine. Men jeg er rigtig glad for at være kommet igennem processen med disse værktøjer.

Selve projektet fik vi sparket igang med en fantastisk brainstorm med en masse gode ideér som vi hurtigt fandt ud af, var alt for optimistisk. Der måtte derfor hurtigt skrues ned for ideérne og fokuseres på hovedindholdet i projektet.

Jeg syntes processen har været rigtig sjov, det har været spændende at bruge de værktøjer man lærer på studiet til noget reelt elektronik. Bare det at forstå X10 kommunikationens virkemåde har været rigtig spændende for mig som Elektro studerende. Dette var en af nødderne der skulle knækkes før projektet virkelig blev spændende.

I hardwaregruppen har vi haft et tæt samarbejde og virkelig vendt og drejet alle aspekterne i projektet med hinanden. Og med vores ugentlige møder, har vi kunnet følge lidt med i software gruppens arbejde med projektet. Alt-i-alt et godt forløb. Savner dog et karaktergivende projekt næste gang, så man kan ligge endnu mere sjæl i det.

16.6 Poul Overgaard

Jeg er rigtigt stolt af hvad vi som gruppe har opnået i forbindelse med dette semesterprojekt. Det har været en lærerig process, at benytte kurserne tværfagligt og herved opnå ny viden. At arbejde med udgangspunkt i ASE modellen, passer mig fint. Fællesfasen med mange inputs fra gruppen samlet, og herefter en fagspecifik fase, hvor jeg har fungeret som en del af hardware gruppen.

Personligt har jeg fået stort udbytte af projektet. Vi har som gruppe fra start haft store ambitioner, som vi har måtte forventningsafstemme undervejs grundet tidsmæssigt pres. Grund idéen med vores CSS synes jeg er interessant, og er bestemt et produkt jeg ser vil blive en fast del af hjemmet på sigt.

Jeg har under møderne, især i projektes sidste fase fungeret som mødeleder/ansvarlig. Det har været rart, at jeg som gruppen yngste og med STX baggrund, har fået lov til at få ansvaret for denne del. Og det er et ansvar jeg gerne tager. Det er bestemt en gruppe jeg kan se mig selv arbejde sammen med igen.

16.7 Simon Kirchheimer

Vi har været syv mand i vores projektgruppe, og det har givet os mulighed for at øve os på at samarbejde med flere om et projekt. Vi har været en god projektgruppe med mange forskellige kompetencer, det har medført at man kan hjælpe hinanden godt. Det var en stor fordel at hele projektet var opdelt i faser og havde løbende møder og reviews, så folk ikke bare kørte på helt selv. Faserne har hjulpet med at få et overblik over projektet, og gjort at vi har tænkt over de ting vi skulle lave. I starten af projektet havde vi store ambitioner om at nå mange forskellige ting, men som tiden skred frem måtte vi se i øjnene at det ikke var realistisk.

I forhold til 1. semester har vi brugt mere tid på at dokumentere det vi skulle lave, og fået en forståelse for de forskellige arbejdsmetoder vi har lært i ISE. Jeg har været på HW delen og der har ASB/MSA fagene givet den nødvendige viden til at kunne løse de problemstillinger vi stod overfor. Jeg tog ansvaret for at skrive logbog/referater, det har

givet et godt grundlag for at få struktureret arbejde, da alle kan gå ind i logbogen/referatet og se hvad vi snakkede om.

Vi fik et produkt næsten som planlagt, grundet tidsmangel blev lyddetektoren ikke lavet og vi fik næsten kommunikation over AC nettet til at fungere korrekt. Jeg har fået et stort udbytte ud af dette projekt, og jeg synes vi har haft et godt samarbejde i projektgruppen. Generelt har gruppen fungeret godt og jeg mener vi har et rigtig godt produkt.

Litteraturliste 17

17.1 Bøger

AVR Microcontroller and Embedded Systems, Muhammad Ali Mazidi et al, Prentice Hall, 2014

C How to Program (7th Ed.), Paul Deitel & Harvey Deitel, Pearson Education, 2013

17.2 Hjemmesider

Stack Overflow: <http://www.stackoverflow.com/> [2014-05-27]

Brugt som løsning og informationskilde ifm. implementeringen

17.3 Opslagsværker

Officiel C++ reference: <http://www.cplusplus.com/> [2014-05-24]

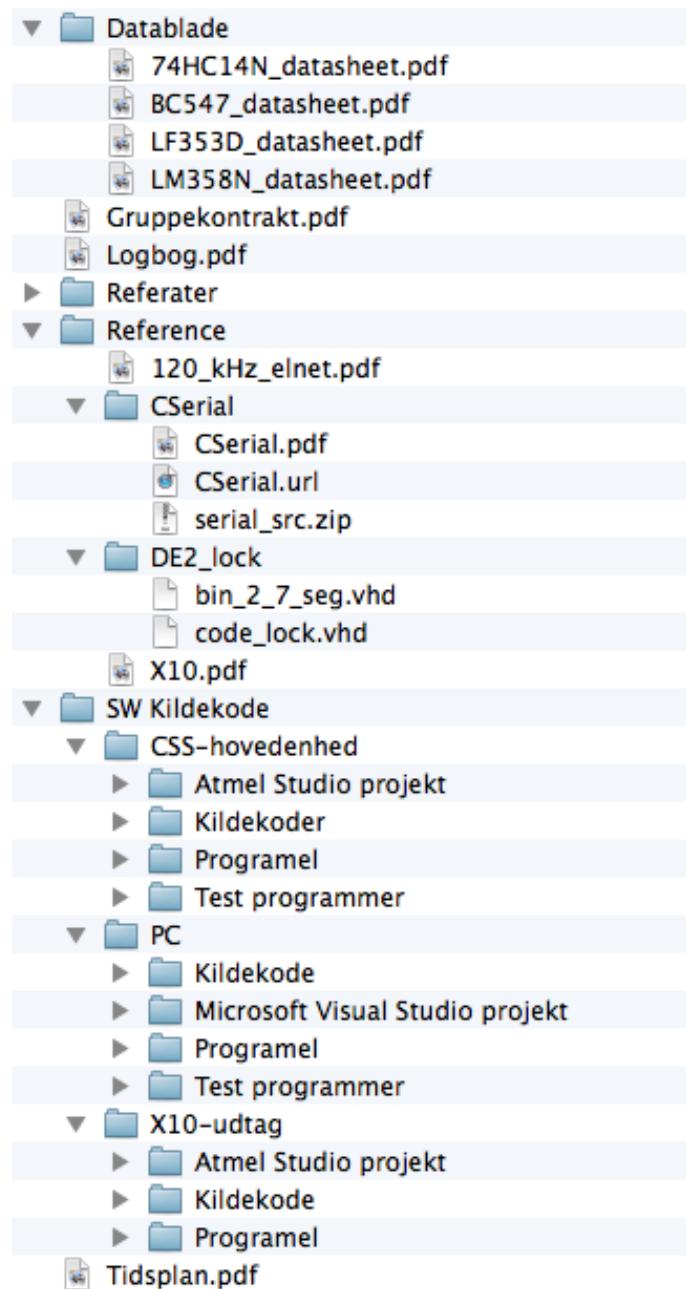
Opslagsværk til C++ standarden

UML-light, T-133, Finn Overgaard Hansen, IHA, Januar 2004

Summary from the Atmel Mega32 manual, T-495, Henning Hargaard, IHA, Juni 2011

Bilags-CD indhold 18

Dette er en oversigt over filstrukturen på den vedlagt CD med bilag.



Figur 18.1. Bilags-CD filstruktur