

# 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 forbrændinger fra elektriske apparater. Fra et computer program kan en bruger tænde og slukke enheder som er koblet til 230 Vac el-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 projektledelses-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 230 Vac el-nettet 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 modtager.



# 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 the 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 introduces 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

---

<b>Kapitel 1</b>	<b>Indledning</b>	<b>7</b>
<b>Kapitel 2</b>	<b>Projektafgrænsning (JS)</b>	<b>9</b>
<b>Kapitel 3</b>	<b>Systembeskrivelse (MK)</b>	<b>11</b>
<b>Kapitel 4</b>	<b>Krav (JS)</b>	<b>13</b>
<b>Kapitel 5</b>	<b>Kravspecifikation (SK)</b>	<b>15</b>
5.1	Usecases . . . . .	15
5.1.1	Bruger . . . . .	16
5.1.2	Eksterne enheder . . . . .	16
5.1.3	Barn . . . . .	16
5.1.4	SMS modtager . . . . .	16
5.1.5	DE2 Board . . . . .	16
<b>Kapitel 6</b>	<b>Arbejdsproces</b>	<b>19</b>
6.1	Udviklingsmodeller (JS, PO) . . . . .	19
6.2	Møder, tidsplan, logbog og referater (PO) . . . . .	20
6.3	HW/SW teams . . . . .	21
6.3.1	Hardware teamet (PO) . . . . .	21
6.3.2	Software teamet (BS) . . . . .	21
<b>Kapitel 7</b>	<b>Udviklingsværktøjer</b>	<b>23</b>
7.1	LaTeX og Texmaker (JS, PO) . . . . .	23
7.2	Visual Studio (JC) . . . . .	23
7.3	Atmel Studio (BS) . . . . .	23
7.4	National Instruments Multisim (BS, PO) . . . . .	23
7.5	Microsoft Visio (JS) . . . . .	23
7.6	Altera Quartus II (PO) . . . . .	24
7.7	Electronic ToolBox App . . . . .	24
7.8	Filhåndtering (JS, PO) . . . . .	24
7.8.1	GitHub . . . . .	24
7.8.2	Dropbox . . . . .	24
7.8.3	Google Drev . . . . .	24
<b>Kapitel 8</b>	<b>Systemarkitektur</b>	<b>25</b>
8.1	Hardware arkitektur . . . . .	25
8.2	Domænemodel . . . . .	25
8.2.1	BDD Hardware . . . . .	26
8.2.2	Plantegning over HW . . . . .	26
8.3	Software arkitektur . . . . .	27

8.3.1	Applicationsmodel for PC (JC)	27
8.3.2	CSS hovedenhed (BS)	28
8.3.3	Applikationsmodel for X10 udtag (BS)	28
<b>Kapitel 9</b>	<b>Design</b>	<b>31</b>
9.1	Hardware design	31
9.2	Encoder	31
9.2.1	Højpasfilter	32
9.2.2	Zero Crossing Detector	33
9.3	Decoder	33
9.3.1	Båndpasfilter	34
9.3.2	Envelope Detector	35
9.3.3	Dipswitch	35
9.4	Software design	37
9.4.1	PC delen (JC)	37
<b>Kapitel 10</b>	<b>Implementering</b>	<b>39</b>
10.1	Hardware implementering	39
10.2	Software implementering	39
<b>Kapitel 11</b>	<b>Resultater</b>	<b>41</b>
<b>Kapitel 12</b>	<b>Konklusion</b>	<b>43</b>
<b>Kapitel 13</b>	<b>Individuel konklusion</b>	<b>45</b>
13.1	Bjørn Sørensen	45
13.2	Jakob Schmidt	45
13.3	Jeppe Stærk	46
13.4	Jesper Christensen	46
13.5	Mick Kirkegaard	46
13.6	Poul Overgaard	47
13.7	Simon Kirchheiner	47
<b>Kapitel 14</b>	<b>Litteraturliste</b>	<b>49</b>
14.1	Bøger	49
14.2	Hjemmesider	49
14.2.1	Opslagsværker	49



# Indledning

---

# 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 i mellem enhederne og systemet kan aktiveres med et kodetryk.





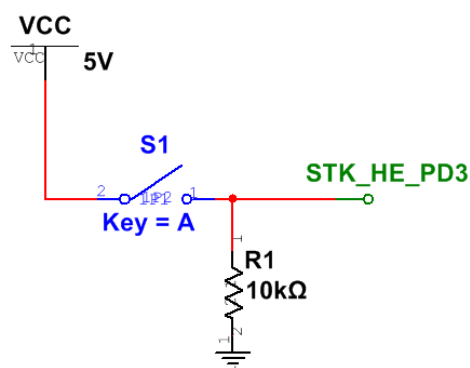
# Projektafgrænsning (JS) 2

---

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

X10 opererer normalt på 230 V nettet, men da vi ikke har autoritet 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 det er i stedet lavet med en knap der giver et højt signal som skal imitere at lyd er detekteret. Se figur 2.1. Der er ikke lavet nogen alternativ løsning til magnetlås.



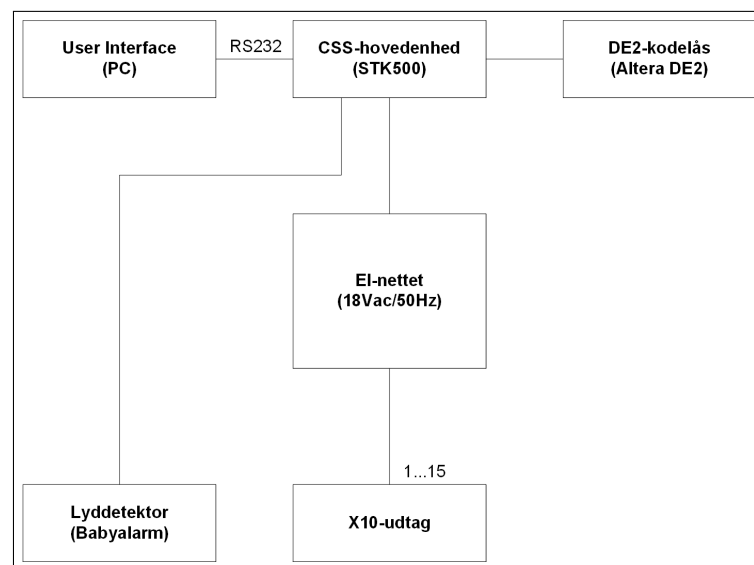
*Figur 2.1.* Schematic over knap for lyddetektion



# Systembeskrivelse (MK)

# 3

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 3.1.* Systemoversigt

I figur 3.1 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 encoderet til en X10-bitstrøm og sendt ud på el-nettet (18Vac). Denne bitstrøm bliver så aflæst af X10-udtagene, og hvis et X10-udtag har den korrekte adresse, vil kommandoen blive udført på denne.

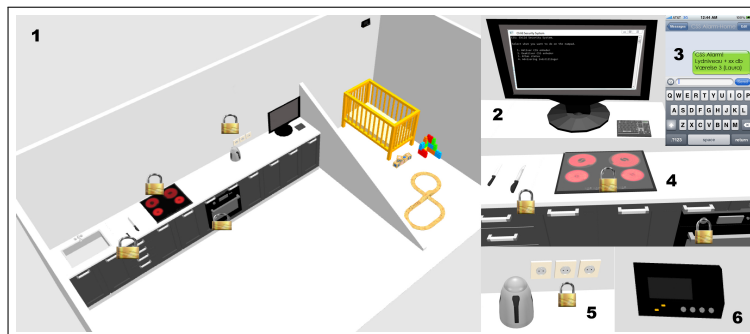
Babyalarmens funktion er at give CSS-hovedenheden besked om støj i det værelse, hvor lyddetektoren sidder placeret. Lyddetektoren er kablet direkte til CSS-hovedenheden og virker ved, at den ved et givent lydniveau<sup>1</sup> vil sende et signal til CSS-hovedenheden, som via

<sup>1</sup>se ikke-funktionelle krav i produktdokumentationen

dennes serielle RS232 port fortæller PCen, 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 3.2.* 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) 4

---

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 baby alarm. 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 syre en magnetlås til aflåsning af døre eller skuffer.

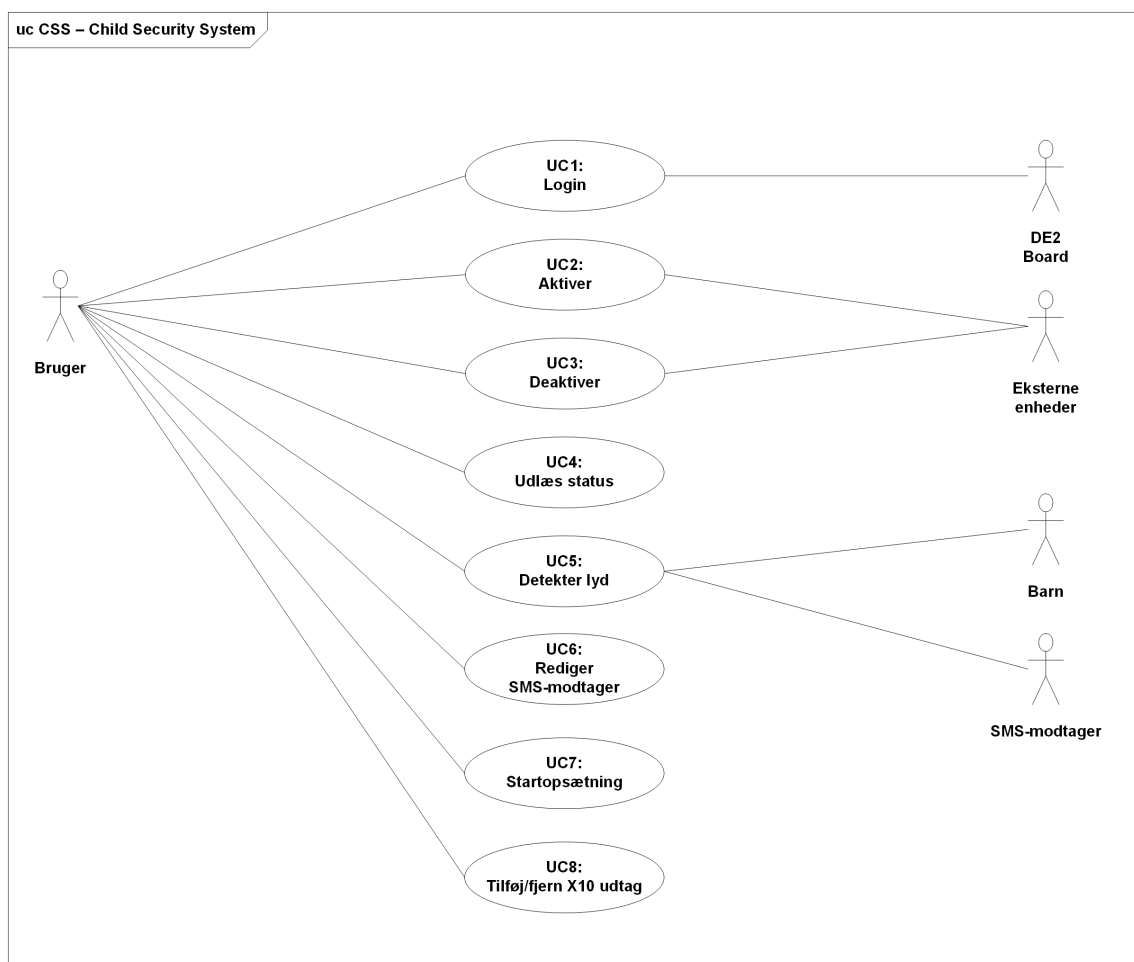
Baby overvågningen består af en sensor der måle lydniveauet i et rum som sender et signal til hovedenheden hvis grænseværdien overskrides. Dette signal fortæller hovedenheden at den skal sende en SMS til et telefonnummer efter kundes ønske.



# Kravspecifikation (SK) 5

Der er blevet udarbejdet en kravspecifikation ud fra følgende use cases se figur 5.1, her er der beskrevet hvilke use cases brugeren, DE2 boardet, eksterneenheder, barn og sms-modtager har kontakt med.

## 5.1 Usecases



*Figur 5.1.* Usecase diagram

For yderligere beskrivelse af hver enkelt use case og større forståelse henvises til projektdokumentation<sup>1</sup>

<sup>1</sup>Projektdokumentation s 9-14

Herunder ses en beskrivelse af hver enkelt aktør i systemet.

#### 5.1.1 Bruger

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

#### 5.1.2 Eksterne enheder

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

#### 5.1.3 Barn

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

#### 5.1.4 SMS modtager

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

#### 5.1.5 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 opetid: 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 lydtektion 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 adresseswitchen
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**

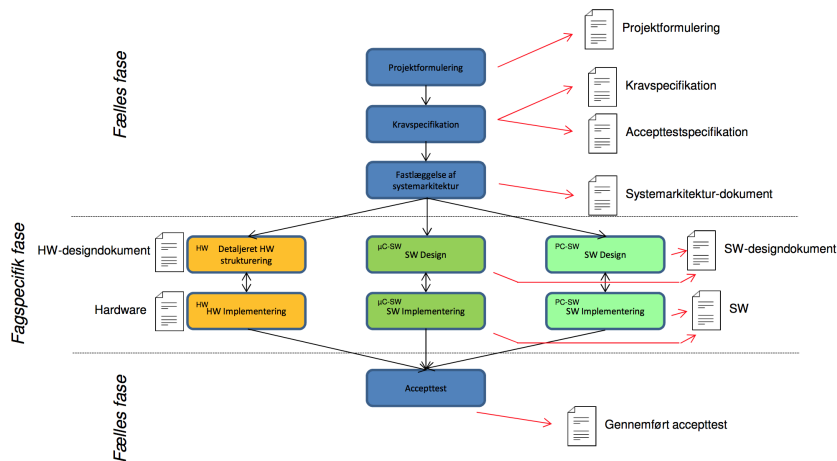
18. Lyddetektoren skal registrere lyde på over 68 dB
19. Der må maksimalt afsendes 1 SMS-besked pr. minut ved gentagende reaktion fra lyddetektoren
20. Låse enheder må maksimalt være 8x5x3 cm
21. Låse enhederne skal kunne holde 5 kilogram



# Arbejdsproces 6

## 6.1 Udviklingsmodeller (JS, PO)

Med udgangspunkt i ISE-undervisningen<sup>1</sup> er projektarbejde opbygget omkring ASE-modellen som ses på figur 6.1. ASE-modellen er opbygget i 2 faser. En fællesfase og en fag specifik fase.



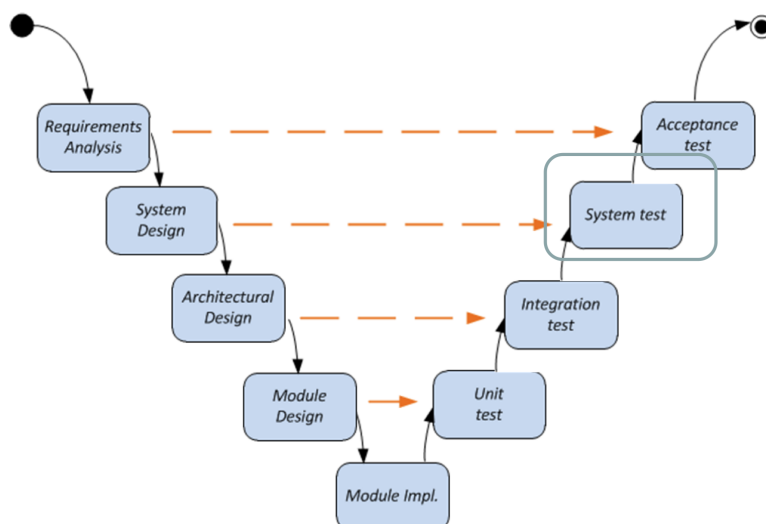
**Figur 6.1.** ASE-modellen

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 6.2.

V-modellen ses på figur 6.2. 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 kravespecifikationen udarbejdes.

<sup>1</sup>Indledende System Engineering - ASE, fag på 2. semester



Figur 6.2. V-modellen

## 6.2 Møder, tidsplan, logbog og referater (PO)

I forbindelse med projektforsløbet 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 én 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 trivselsrunde. Her har det været muligt at give ris/ros til gruppen og eller enkelte. Gruppemøderne startede lidt løst, men dette blev hurtigt ændret 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 gruppekontrakt<sup>2</sup>. Gruppekontrakten har dannet grundlag for forventningerne til gruppen og dens medlemmer.

Reviewmøder har fungere således at gruppen enten udførte review på en anden gruppen 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.

<sup>2</sup>Gruppekontrakten findes på bilags cd under navnet: Gruppekontrakt.pdf

## **6.3 HW/SW teams**

### **6.3.1 Hardware teamet (PO)**

Hardware teamet bestående af: Jakob, Mick, Poul og Simon har arbejdet meget sammen om opgaven. Samarbejde er nøgleordet for dette team. Alle fire har hovedsagligt deltaget i alle opgaver herunder.

### **6.3.2 Software teamet (BS)**

I software gruppen bestående af Bjørn, Jeppe og Jesper har vi arbejdet sammen under de indledende faser og først i den detaljerede designfase har vi delt opgaverne op. Derfra har vi arbejdet individuelt, men dog med regelmæssige møder og afklaring for at sikre at interface aftaler og lignende stadig blev overholdt.



# Udviklingsværktøjer 7

---

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

## 7.1 LaTeX og Texmaker (JS, PO)

Hele rapporten er skrevet i  $\text{\LaTeX}$ . 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.

$\text{\LaTeX}$  er et kodebaseret tekstredigerings 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. Det er heri al  $\text{\LaTeX}$ kode er 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  $\text{\LaTeX}$ , men da det var på plads fungerede det rigtig godt.

## 7.2 Visual Studio (JC)

## 7.3 Atmel Studio (BS)

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

## 7.4 National Instruments Multisim (BS, PO)

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

## 7.5 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 diagrammer og internal blok diagrammer for hardwaren. UML er anvendt til software relaterede diagrammer.

## 7.6 Altera Quartus II (PO)

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

## 7.7 Electronic ToolBox App (JS)

Electronic ToolBox er en applikation der er udviklet til IOS. Den indeholder information 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 båndpasfilter.

## 7.8 Filhåndtering (JS, PO)

Til håndtering af filer er nedenstående 3 løsninger brugt.

### 7.8.1 GitHub

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

### 7.8.2 Dropbox

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

### 7.8.3 Google Drev

Logbogen, mødereferater 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.

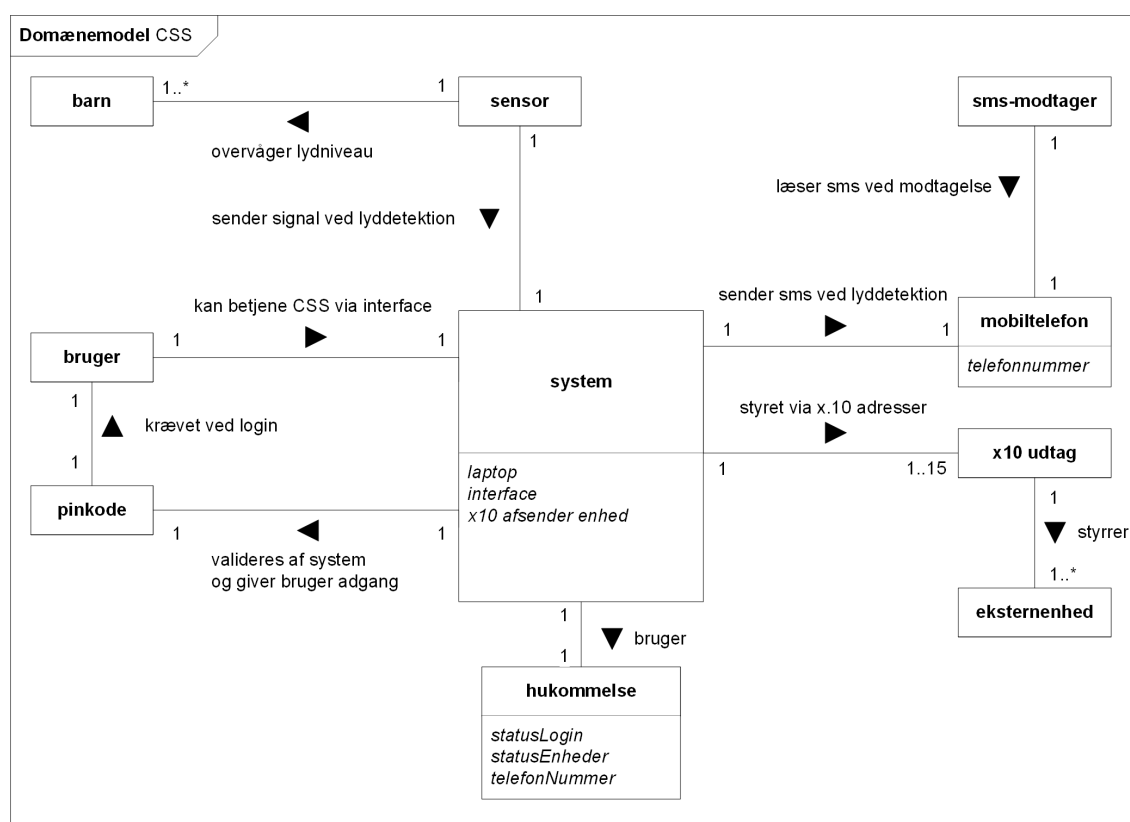


# Systemarkitektur 8

## 8.1 Hardware arkitektur

Efterfølgende diagrammer viser hvordan hardware arkitekturen er opbygget.<sup>1</sup>

## 8.2 Domænemodel

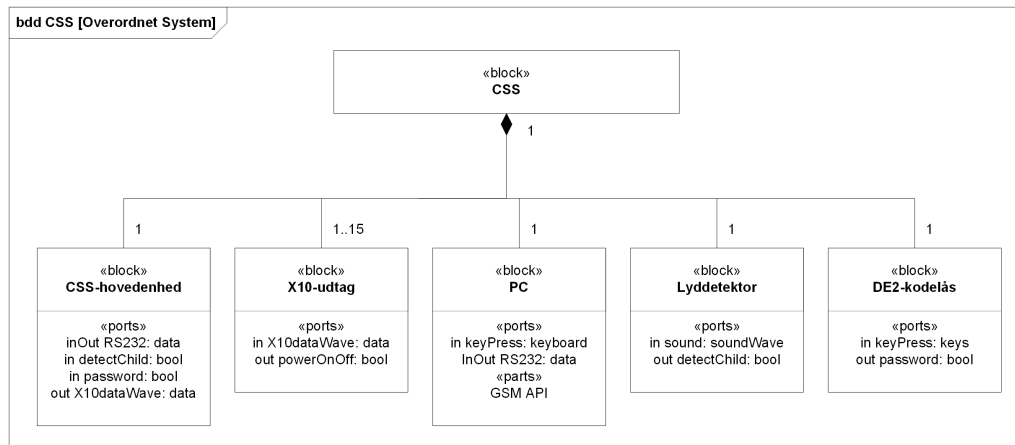


Figur 8.1. Domænemodel

Domænemodel 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.

<sup>1</sup>For yderligere BBD/IBD se projektdokumentation afsnit System Arkitektur.

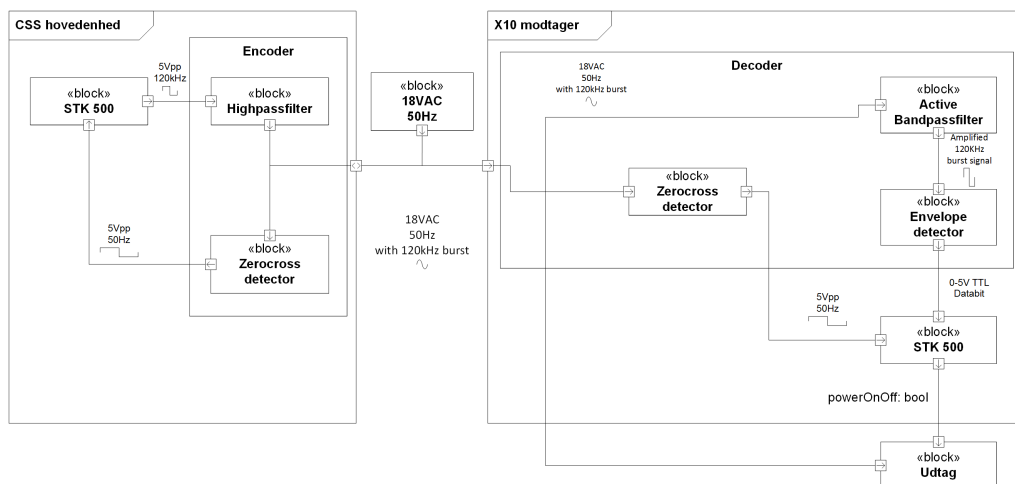
### 8.2.1 BDD Hardware



Figur 8.2. BDD Hardware

BDD diagrammet giver et overblik over hvad det samlede system består af. Vi ser en port beskrivelse som viser hvilke signaler hver blok består af.

### 8.2.2 Plantegning over HW



Figur 8.3. Plantegning over HW

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

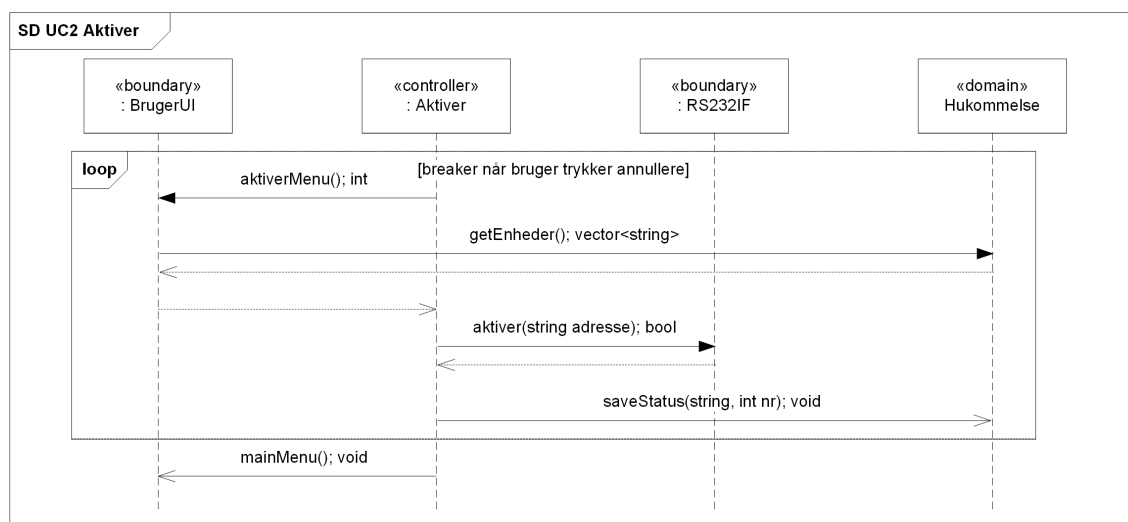
## 8.3 Software arkitektur

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.

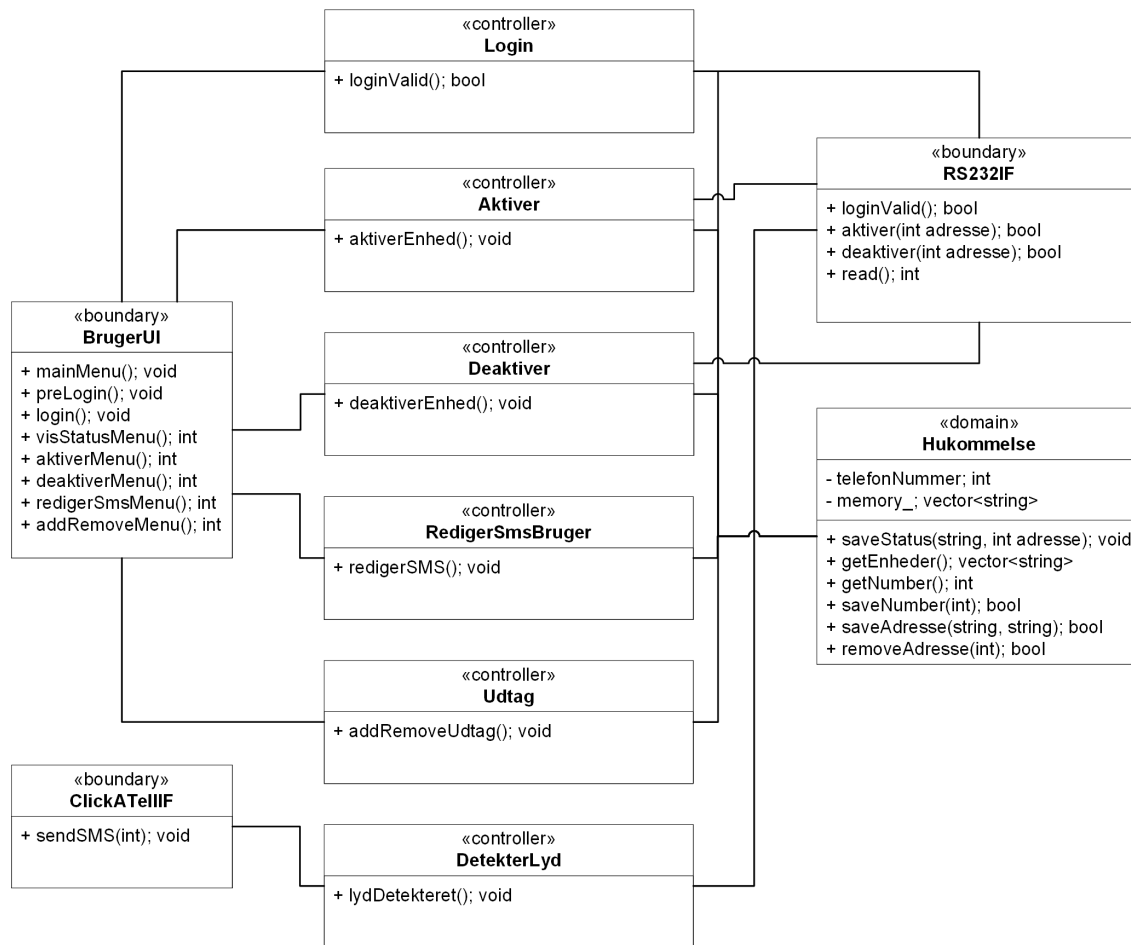
### 8.3.1 Applicationsmodel for PC (JC)

Ud fra domænemodellen og de forskellige Use cases er der oprettet sekvens diagrammer for hver use case (UC4 og UC7 undtaget, UC4 fordi den er utrolig simpel og UC7 fordi den referere til UC8) Sekvensdiagrammet for UC2 Aktiver kan ses på nedenstående billede. Resten af sekvensdiagrammerne kan ses i rapport dokumentationen



**Figur 8.4.** Sekvensdiagram UC2

Efter at alle sekvens diagrammerne er lavet har vi alle vores conceptuelle klasser med metoder som vi nu kan overføre til et klasse diagram som ses nedenfor



*Figur 8.5. PC klasse diagram*

Når man så har lavet sit conceptuelle klassediagram kan man arbejde videre med det for at få et statisk klassediagram og klassebeskrivelser hvorefter man kan begynde sit design.

### 8.3.2 CSS hovedenhed (BS)

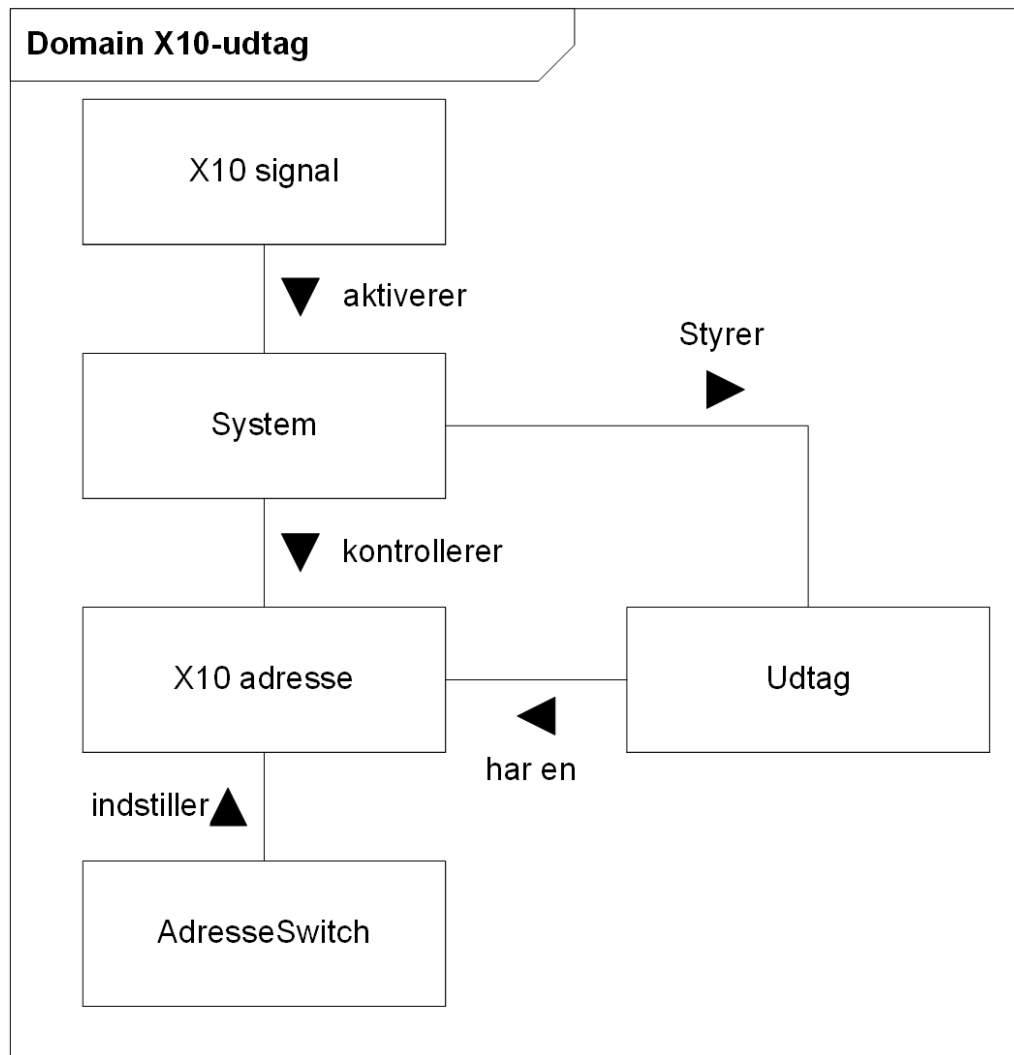
#### 8.3.3 Applikationsmodel for X10 udtag (BS)

Først er der lavet en detaljeret domænemodel for X10 udtaget. Denne er vist i figur 8.6. Denne laves ved at gennemgå UC beskrivelserne og finde de ting som har indflydelse på netop denne del af systemet.

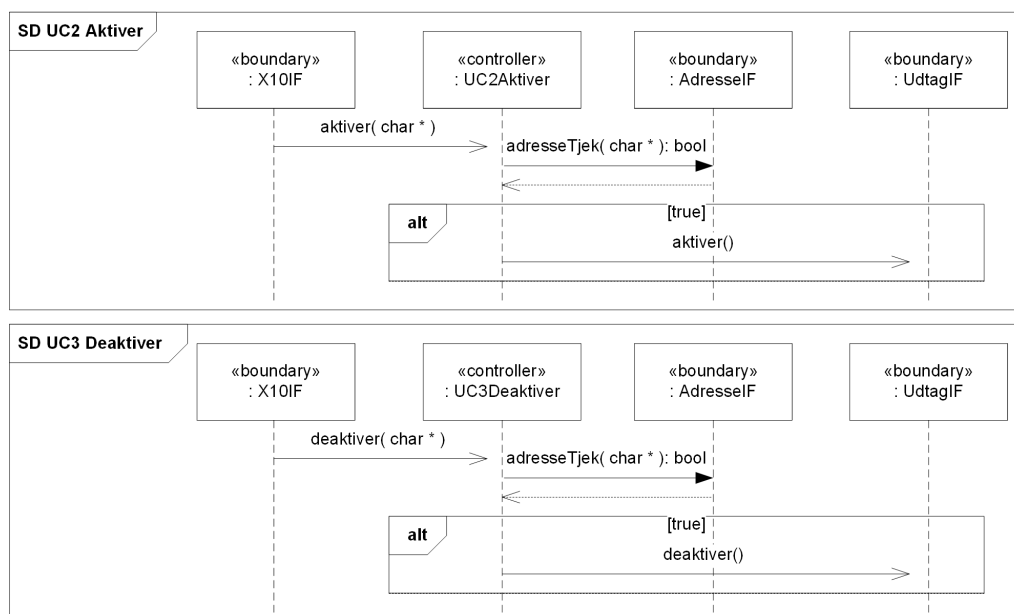
Med dette udgangspunkt laves der sekvensdiagrammer for hvert UC. Disse er vist i figur 8.7. De viser hvordan metodekald i mellem de conceptuelle klasser og giver et overblik over den basale funktionalitet.

Dette resulterer i et klassediagram med grundfunktionaliteten beskrevet, se figure 8.8. Denne bruges under implementeringen og ender ud i et statisk klassediagram som beskriver det endelige program med alle hjælpemetoder.

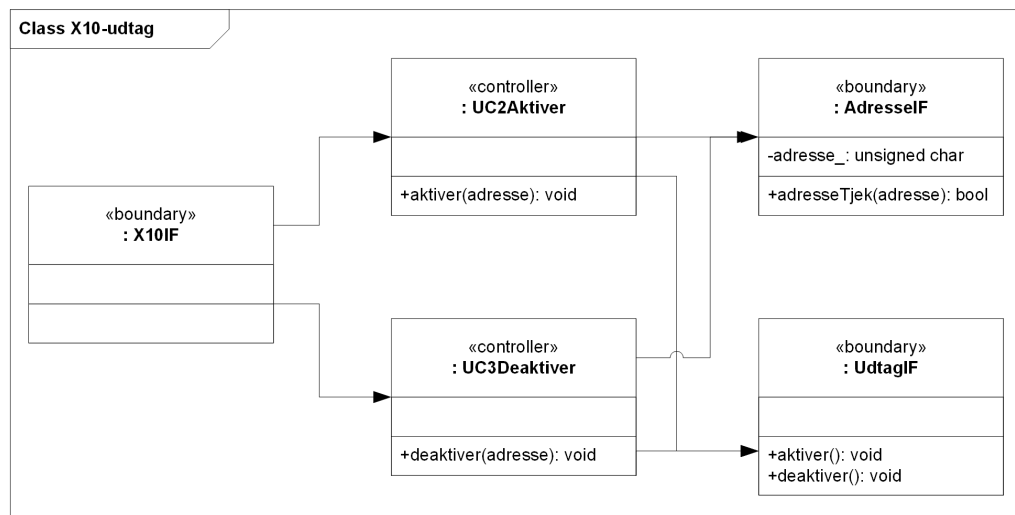
Denne analyse af funktionalitet giver et klart overblik til implementeringsfasen.



Figur 8.6. Domænemodel for X10 udtag



Figur 8.7. Sekvensdiagram for X10 udtag



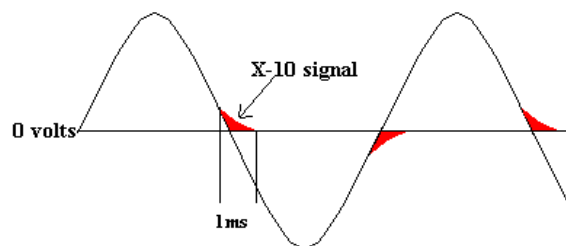
*Figur 8.8.* Klassediagram for X10 udtag

## 9.1 Hardware design

Generelt kan det overordnede hardware design beskrives som noget elektronik der sender information ud på nettet, som sendes og analyseres af noget elektronik i den anden ende. Disse vil blive beskrevet herunder som encoder og decoder<sup>1</sup>.

## 9.2 Encoder

Encoderen, også omtalt som senderenhed, er den del af systemet der genererer X10 kommandoen og sender den ud på det eksisterende el-net. Et højpasfilter der lader 120 kHz burst passere mens det blokerer for nettets 50 Hz signal, udgør sammen med en zero crossing detector hele hardwaren for encoderen.

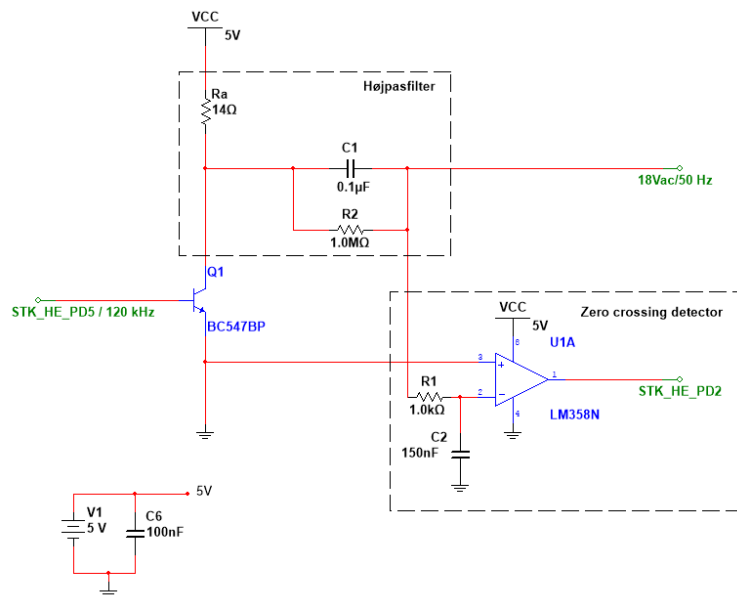


*Figur 9.1.* 120 kHz burst i zero crossing

Ideen med at sende burst ud på nettet i zero crossing kan ses illustreret på figur 9.1

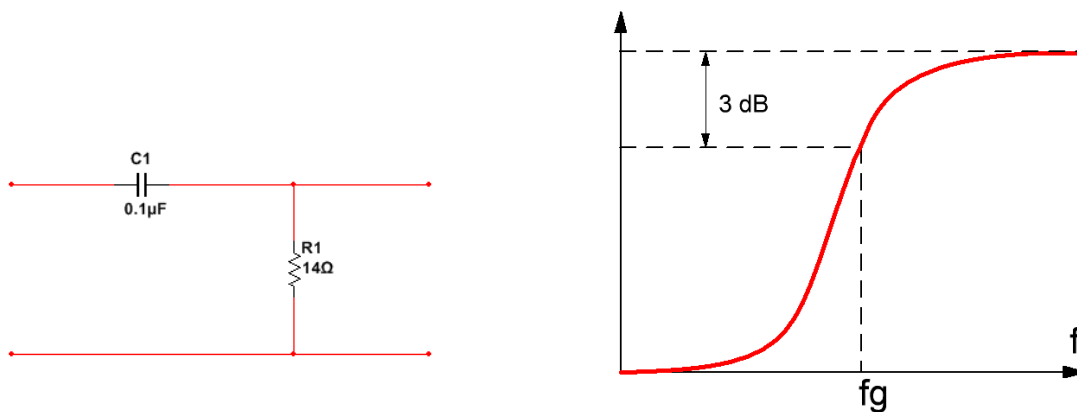
---

<sup>1</sup>For yderlige beregninger se projektdokumentation afsnit HW-design



Figur 9.2. Samlet Encoder

### 9.2.1 Højpasfilter



Figur 9.3. Højpasfilter med værdier

Figur 9.4. Kurvekarakteristik for højpasfilter

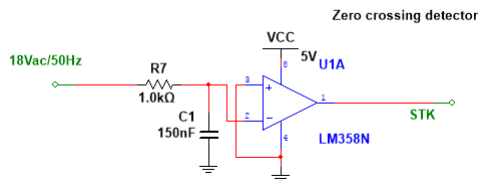
For at sende X10 kommandoer ud på det eksisterende 50 Hz el-net er det nødvendigt at koble elektronikken direkte herpå og eftersom det elektronik ikke tåler de høje spændinger fra nettet, er det nødvendigt at blokere det signal, men stadig at kunne sende de 120 kHz ud. Dette løses med et højpasfilter.

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

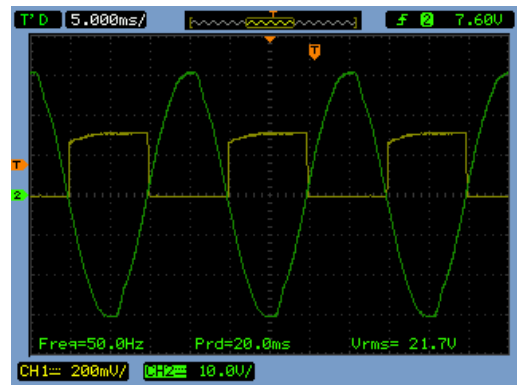
Kondensatoren forudbestemmes for beregningerne til en værdi på 0,1 nF.



### 9.2.2 Zero Crossing Detector



**Figur 9.5.** Zero crossing detector med værdier



**Figur 9.6.** Scope billede af Zero crossing detector, CH1(indgangssignal), CH2(Udgangssignal)

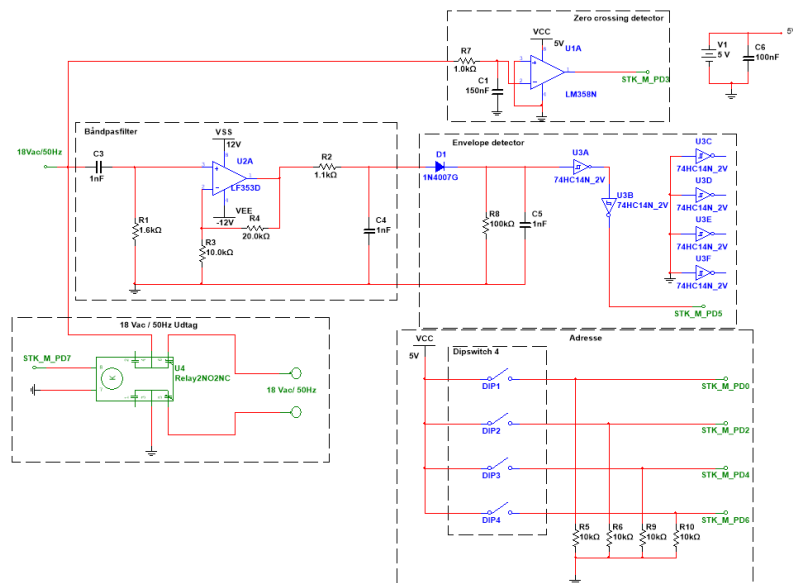
Zero crossing detectoren har til opgave at detektere nulgennemgang, det er et krav for at X10 protokollen kan virke. Der er placeret en Zero crossing detector både på Encoderen og Decoderen, da begge disse består af et STK-kit der kræver information om nulgennemgang. Opbygning kan ses på overstående figur ??, der er anvendt en operationsforstærker af typen LM358N, som toggler udgangssignalet ved hver nulgennemgang se figur 9.6. Operationsforstærkerens positive ben er koblet til stel for at lave et triggerniveau 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.

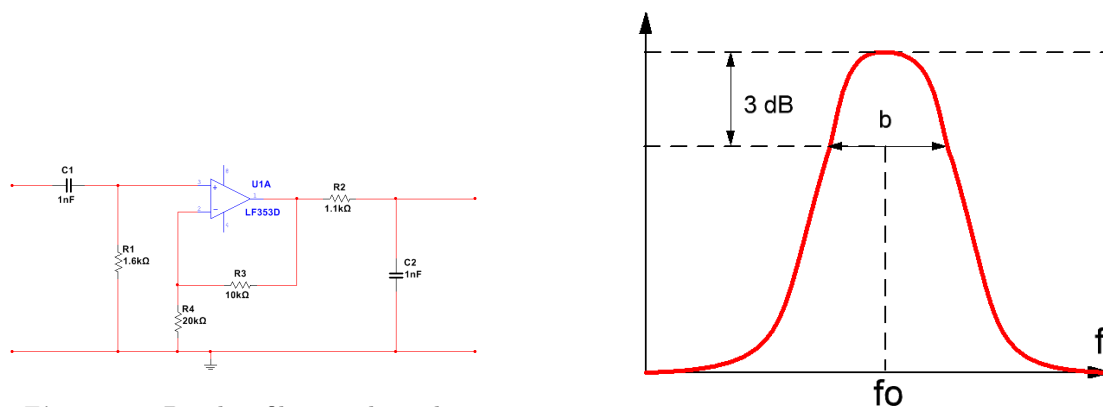
## 9.3 Decoder

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 50 Hz nettet og forstærker 120 kHz signalet der kommer fra encoderen. Herefter ledes signalet gennem envelope detector som omdanner burstet til et TTL signal.



Figur 9.7. Samlet Decoder

### 9.3.1 Båndpasfilter



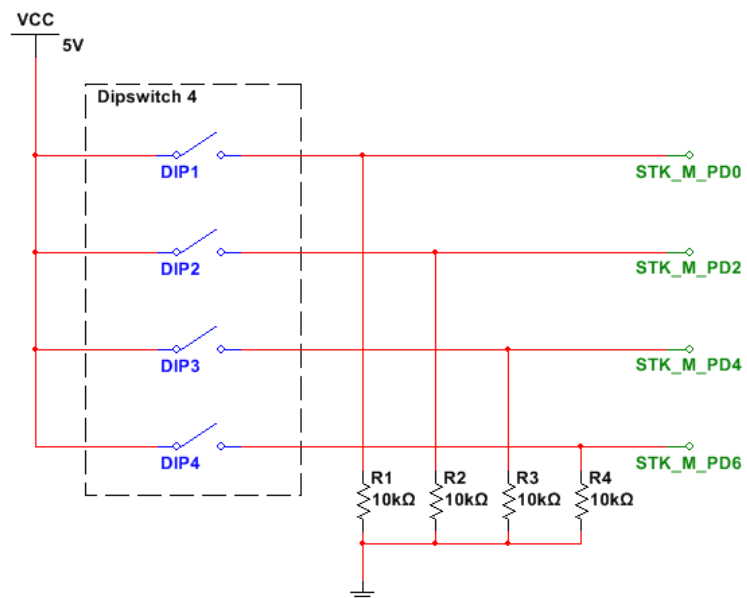
Figur 9.8. Båndpasfilter med værdier

Figur 9.9. Kurvekararakteristik 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 OpAmp med modstande, hvis størrelse afhænger af den ønskede forstærkning, mellem et højpasfilter og lavpasfilter som illustreret på figur ???. Kurvekararakteristikken er illustreret på figur 9.9.

Da vi ønsker at forstærke 120 kHz signalet, designs højpasfilteret således at knæfrekvensen udregnes til 110 kHz, og for lavpasfilteret beregnes en knæfrekvens på 130 kHz.





*Figur 9.11.* Dipswitch for adressering af udtæg

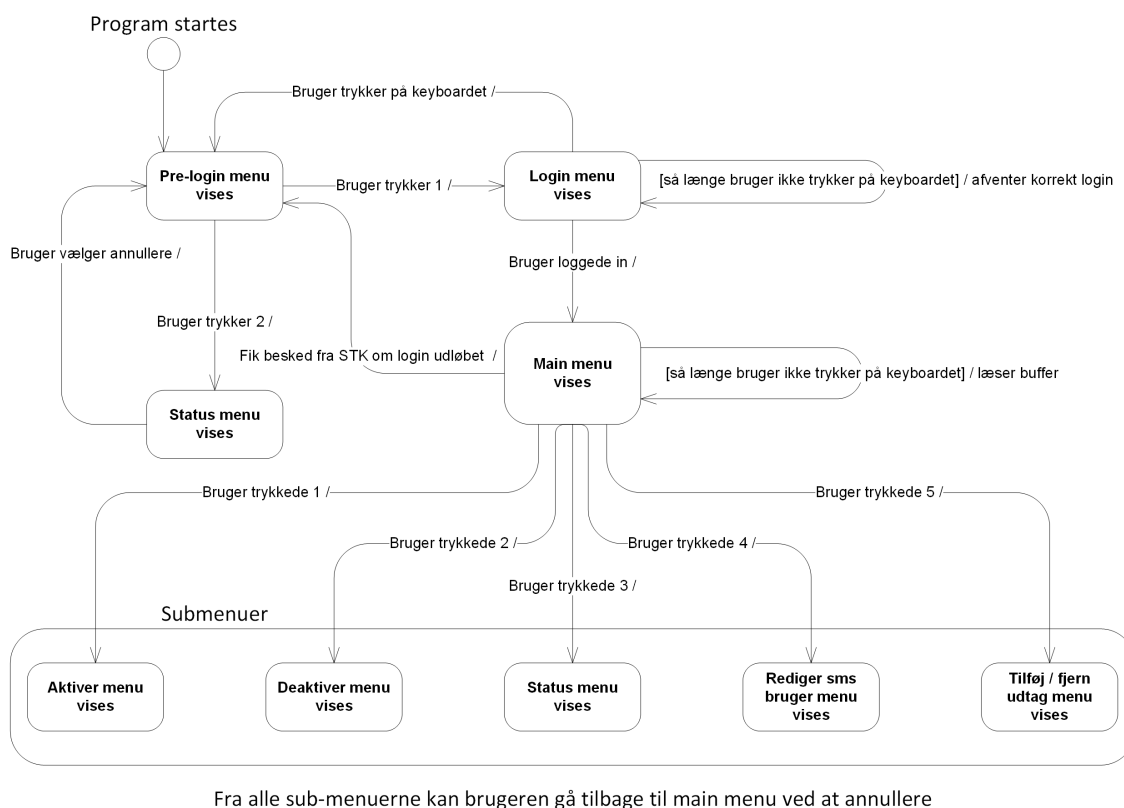
## 9.4 Software design

Design fasen var mere eller mindre lige til efter alt den foreliggende dokumentation var på plads. Opgaverne blev delt ud således at Jesper lavede PC delen, Bjørn lavede software til senderen og Jeppe til modtageren og ClickATell klassen. Vi har løbende haft møder og forsøgt at hjælpe hinanden når der var nogle spørgsmål.

### 9.4.1 PC delen (JC)

Softwaren til PC'en, er brugeren grænseflade til at kontrollere systemet. Her kan han manurære rundt i de forskellige menuer og udføre de forskellige ting som er beskrevet i use casene.

Nedenfor er der illustreret hvordan man kommer frem og tilbage i brugerinterfacet. Udover bruger input så kan CSS hovedenheden give PC'en besked om at der ikke længere er logget ind hvilket vil sende brugeren fra main menu og tilbage til pre-login menuen.



**Figur 9.12.** State machine diagram over brugerflade



# Implementering 10

---

10.1 Hardware implementering

10.2 Software implementering





# Resultater 11

---



# Konklusion 12

---



## 13.1 Bjørn Sørensen

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

Med en baggrund som elektronikmand har jeg skulle koncentrere 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 rigtig god, men må også erkende at vi havde en fast deadline som ikke gjorde det muligt at nå helt i mål med idéen.

## 13.2 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 rigtig godt. Det gode samarbejde og overblik skyldes at vi jævnligt har holdt møder, både som selvstændig gruppen, men også sammen med vejleder. Desuden er der afholdt enkelte trivsels runder, 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 rigtig tilfreds med resultatet af vores projekt, på trods af at vi langt fra fik realiseret alle de ting vi havde udtænkt 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.

### 13.3 Jeppe Stærk

### 13.4 Jesper Christensen

Semester projektet 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 hardware grupper 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 software architecturen uden enlig at have kodet noget var en udfordring, men det har været super lærerigt. Synes helt sikkert at jeg er bedre rustet til at lave system arkitektur en anden gang efter at have prøvet krafter med det her. Alt i alt har det været et rigtig godt projekt forløb.

### 13.5 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 rigtigt 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 karakter-givende projekt næste gang, så man kan ligge endnu mere sjæl i det.

## 13.6 Poul Overgaard

### 13.7 Simon Kirchheiner

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 fik vi ikke lavet vores lyddetektor, men vi fik 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 14

---

## 14.1 Bøger

## 14.2 Hjemmesider

### 14.2.1 Opslagsværker

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