AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

GRUPPE 4

E2PRJ2 - Rapport

Home Automation System

28-05-2014

Steen Tingager Kristensen _____

Anders Christensen _____

Freddie Jørgensen _____

Felix Blix Everberg _____

Niels Petur Svanholt Midjord_____

Martin Bo Augustinus _____

Ruben Pedersen _____

Hakon Primdahl _____

Resumé

Rapporten indeholder 2. semesters semesterprojekt på IHA. Her er der givet en problemstilling, som hedder tyveriforebyggelse. For at lave dette, bliver der lavet et home automation system, hvor der kan oprettes scenarier, der simulerer, at der altid er aktivitet i hjemmet. Til dette formål bliver der lavet en lysdæmper, der kan regulerer lysstyrken på en glødepære. Der vil også blive udarbejdet en General Purpose enhed med fire stikkontakter, hvor det er muligt at tilkoble et anlæg, fjernsyn eller lignende, der så kan scenariestyres til at være tændt/slukket efter ønske. For at styre enhederne bliver der lavet en brugergrænseflade, hvor brugeren har mulighed for at oprette personlige scenarier samt ændre de gemte scenarier. Disse scenarier kan aktiveres/deaktiveres efter brugerens behov.
For at realisere projektet skal der bruges en PC til brugergrænsefladen, et Arduino Nano board indeholdende en ATmega328P Micro Controller, der fungerer som controller, hvor simuleringsscenarierne bliver gemt. Ydermere benyttes et DE2-board, som fungerer som kodelås til systemet. Kodelåsen skal bruges for at sikre, at kun brugere bekendt med den fastsatte adgangskode kan tilgå systemet.
I udviklingsprocessen er ASE-modellen blevet brugt for at danne et godt overblik over de forskellige processer i udviklingen.

Abstract

This report contains the second semester's project at ASE. There has been given a thesis, which is theft prevention. To implement this, a home automation system has been made, where you can create and maintain different scenarios which simulate constant activity in the house. For this purpose we use a light dimmer which controls the intensity of a lamp. There has also been implemented a General Purpose Unit with 4 power sockets, where a sound system, a TV etc. can be powered and controlled whether they should be on or off in an active scenario.
To control the units there will be an user interface, where the user can choose to change the different scenarios, create a new scenario or simply switch the scenarios on and off.
To realize the project we have to use a PC for the user interface, an Arduino Nano Board with a ATmega328P micro controller mounted, which functions as a controller that stores the scenarios an also a DE2 board that works as a code lock. The code lock prevents unwanted people to access the system. In the developing process we have been using the ASE-model to improve the overview of the different processes and progresses in the project.

Indhold

1. Indledning

Selvom antallet af private indbrud er faldende, er dette stadigvæk et stort problem i Danmark. Sammenlignet med vores nordiske naboer, Norge og Sverige, har vi 3-4 gange så mange indbrud pr. indbygger. Svenskerne og Nordmændene er væsentligt bedre til at tyverisikre deres huse, hvilket man i Danmark kunne lære en del af.
Formålet med dette projekt er, at udvikle et nemt og brugbart system, som skal kunne forhindre indbrud. Der findes mange forskellige alarmer og alarmselskaber, der er egnet til private boliger, men der er få, som rent faktisk har tyveriforebyggelse som hovedpunkt.
Dette produkt specialiserer sig i at nedsætte raten for indbrud ved hjælp af et simuleringssystem, der benytter aktive scenarier til at simulere aktivitet i hjemmet, så huset aldrig fremstår som tomt.
Produktet består af en kodelås, der åbner for et interface, som brugeren har adgang til gennem en computer. Derfra skal brugeren kunne vælge om der skal oprettes et nyt scenarie eller om et gemt scenarie skal aktiveres eller deaktiveres. Der er også mulighed for at redigere i de allerede gemte scenarier. Scenarierne kan udspilles med de funktioner i et ønsket rum efter brugerens ønske. I dette system er der implementeret en lysdæmper og en General Purpose Enhed (betegnet GP-enhed). Lysdæmperen kan regulere lysintensiteten på en glødepære, så der kan være forskelligt lys alt efter om man er i soveværelset, stuen eller køkkenet. GP-enheden har 4 stikkontakt-udgange, hvor der kan monteres op til 4 enheder som f.eks. et TV eller et anlæg. Hver stikkontakt i GP-enheden kan tændes og slukkes enkeltvis.
Med udgangspunkt i brugeren og systemet er der opstillet en række use cases, der viser brugerens interaktion med systemet. Disse use cases er en del af kravspecifikationen. Herefter kommer struktureringen, hvor der beskrives hvordan software og hardware bliver koblet sammen. Ud fra struktureringen udarbejdes et konkret design, der skaber grundlag for implementeringen. Når alt er implementeret udfyldes accepttestspecifikationen for at sikre, at det implementerede system overholder de krav fastsat i kravspecifikationen.

2. Opgaveformulering

I dette projekt ønsker vi at lave et hjemmeautomatiseringssystem som forebygger mod indbrud, når man ikke er hjemme. Der skal laves en lysdæmper samt en general purpose enhed. Lysdæmperen skal tilsluttes en stikkontakt, hvor det er muligt at dæmpe en glødepære. General purpose enheden (GP-enheden) skal styre fire disponible stikkontakter, hvor disse enten kan være tændte eller slukkede. Derved kan man sætte en ekstra lampe, radio eller lignende til, så simulationen kan blive mere personlig. Systemet skal have et default program installeret, som ikke kan slettes eller redigeres. Der skal kunne programmeres nogle personlige scenarier, hvor der sættes et interval for hvornår de forskellige enheder er aktiverede/tændte, så det forekommer realistisk.

I det færdige produkt skal brugeren være i stand til at:

- Oprette simulerings scenarier.
    - Bestemme lysintensitet.
    - Bestemme tidsintervallet for hvornår enhederne skal være tændte/slukkede.
    - Navngive scenariet og lave en kort beskrivelse.
- Redigere i scenarierne (med undtagelse af default scenarier).
    - Redigere i navn, beskrivelse og scenarie.

- Slette scenarierne (med undtagelse af default scenarier).
- Aktivere/deaktivere scenarierne.

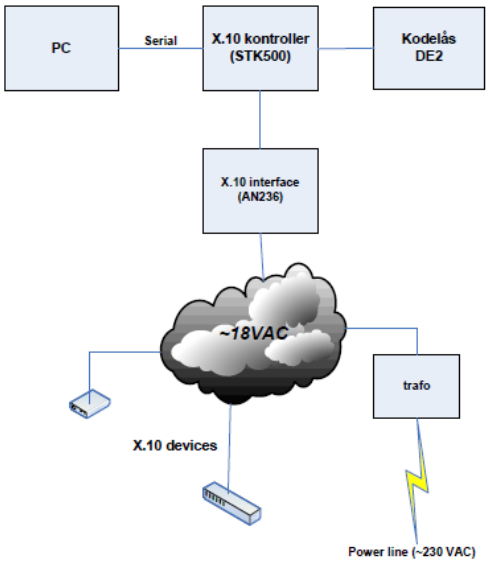
For at få rettigheder til at ændre i scenarierne skal man indtaste en kode på et tilhørende DE2 board, som fungerer som en kodelås for systemet.

3. Projektafgrænsning

Fra starten af projektet har der været defineret nogle enkelte krav fra skolen/kunden, disse har været:

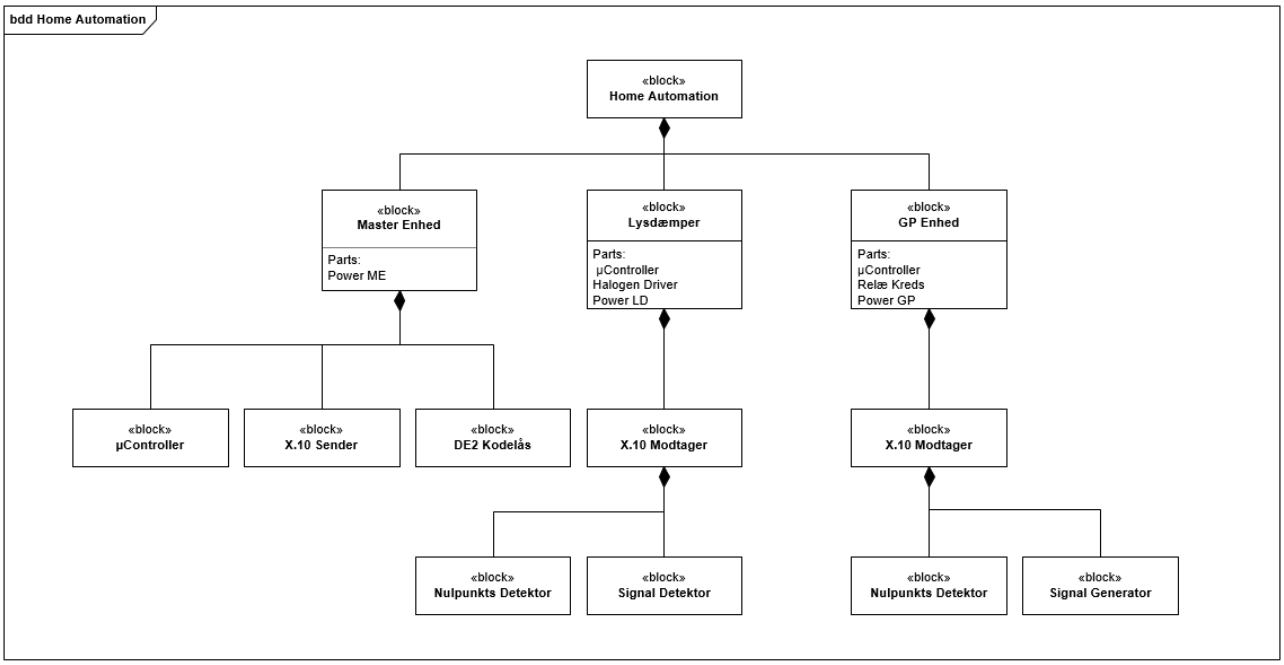
- Der skal minimum indgå en computer
- Der skal være et udviklingsboard som fungerer som X.10 controller
- Der skal være en DE2 kodelås
- Kommunikationsteknikken X.10 skal bruges

Med følgende krav bliver der udviklet et system, som kan oprette simulerings scenarier for home automation systemet. Der vil blive udviklet 2 enheder, som kan styres fra X.10 kontrolleren; en lysdæmper samt en General Purpose Enhed. Lysdæmperen er dog det eneste, som bliver implementeret.
Da vi ikke har mulighed for at bruge 220V, skal det hele implementeres med en 18V AC adapter.
Projektets prototype kommer til at bestå af et Arduino Nano Board(udviklingsboard), PC som sender kommandoer til boardet, en kodelås som giver adgang til systemet og en lysdæmper som bliver kodet til, at kunne dæmpe og øge lysintensiteten.



Figur 1 Hardware setup fra projektoplæg.

4. Systembeskrivelse



Figur 2 - BDD

Dette system giver mulighed for at automatisere forskellige elektriske enheder i hjemmet via lysnettet, f.eks. med henblik på indbrudsforebyggelse. Forebyggelsen består i en scenariestyret simulering af de elektriske enheder, der skal simuleres som menneskeligaktivitet i hjemmet og dermed mindske risikoen for indbrud.

Systemet kommer med et standard scenarie (indbruds-forebyggelses-program), men tillader også en bruger, at specificere et eller flere scenarier som kan aktiveres enkeltvis. Når et scenarie er aktivt styres de valgte enheder via X.10 industristandarden for lysnet kommunikationen. Systemet som er beskrevet her, består af en masterenhed, en kodelås, og to modtager enheder. Systemet kan modificeres ved tilføjelse af flere modtagerenheder.

Masterenheden står for al styring i systemet og kan tilgås via en terminal på en PC. Når kodelåsen er låst op, kan brugeren tilføje, fjerne, redigere, aktivere og deaktivere scenarier. Når et scenarie er aktiveret sørger masterenheden for at sende passende kommandoer ud fra det givne scenarie.

Systemet er således udelukkende et automatisk system, og er ikke designet til at brugeren tilgår specifikke enheder gennem X.10 nettet. Ønsker brugeren at styre en enhed manuelt (Hvis f.eks. masterenheden er slukket), er der en overwrite knap på modtagerenhederne, som tillader tænd/sluk funktionalitet uafhængigt af X.10 nettet.

I dette system er der to modtagere. En lysdæmper og en fire-kanals general purpose enhed.

Lysdæmperen tillader dæmpning af en glødepære fra slukket til fuld intensitet i 10 trin.

General purpose enheden har fire stikkontakter, hvor almindelige hus apparater kan tilkobles. Hver af de 4 stikkontakter kan tændes og slukkes individuelt.

Konsollen er en PC, som kører en ASCII baseret terminal klient som fx. Tera Term. Den hovedmenu, hvorfra brugeren kan tilgå systemet, som fremstår når konsollen åbnes, vil blive udformet lignende Figur 3.

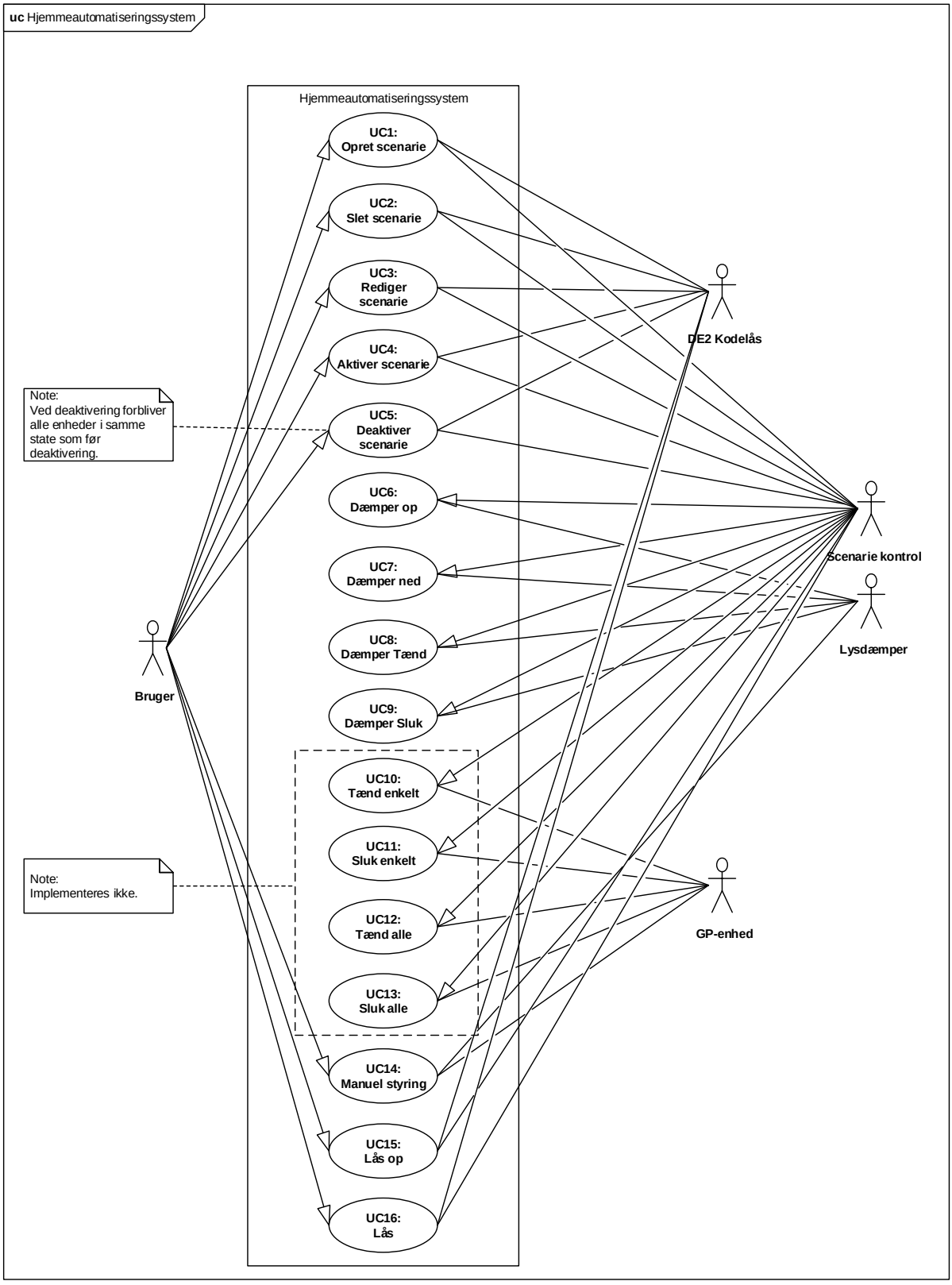
```
Hovedmenu:

   1. Opret scenarie
   2. Slet scenarie
        (Scenarier)
   3. Rediger scenarie
        (Senarier)
   4. Aktiver scenarie
        (Scenarier)
   5. Deaktiver scenarie
        (Scenarier)
```

Figur 3 - Eksempel på hovedmenu

5. Kravspecifikation

5.1. Generelt

Ud fra opgavebeskrivelsen er der udarbejdet en kravspecifikation baseret på et use case diagram (Figur 4). Use case diagrammet giver et overordnet overblik over systemet samt hvordan der kommunikeres heri. I efterfølgende afsnit er use casene beskrevet på et abstrakt niveau. For mere information omkring disse henvises til projektdokumentationen.



Figur 4 - Use case diagram

5.2. Aktører

På Figur 4 ses aktørerne for systemet. Disse er beskrevet nedenfor.

Bruger
Brugeren er ejeren af systemet og derfor primær aktør. Brugeren ønsker at kunne styre og tilpasse sit system, så det passer ind i brugerens behov.

DE2-Kodelås
Kodelåsen er en sekundær aktør. Denne har ansvaret for at verificere adgangen til systemet således, at der ikke kan opstå tilgang fra uønskede brugere. Brugeren har en kode kendt af systemet, det bruges til at oplåse/låse systemet.

Scenariekontrol
Scenariekontrol er en sekundær aktør. Denne er masterenheden for systemet, der har til ansvar, at administrere og validere de forskellige scenarier samt sending af kommandoerne til modtagerenhederne.

Lysdæmper
Lysdæmper er en sekundær aktør. Denne er styret af scenariekontrollen og udfører de kommandoer, der vedrører lysdæmperen (UC6-9), tilhørende det aktive scenarie.

GP-enhed
GP-enhed er en sekundær aktør. Denne er styret af scenariekontrollen og udfører de kommandoer, der vedrører GP-enheden (UC10-13), tilhørende det aktive scenarie.

5.3. Use cases

Nedenfor beskrives de use cases der optræder i systemet (for use case diagram se Figur 4). For yderligere information om Use casene henvises til dokumentationen.

UC1 - Opret scenarie
I denne UC er det muligt for brugeren at oprette et personligt scenarie med ønskede kommandoer. Brugeren skal åbne konsollen og vælge "Opret scenarie" i hovedmenuen, hvorpå de ønskede kommandoer indskrives. Når brugeren er færdig, gemmes disse i scenariekontrollen og brugeren sendes tilbage til hovedmenuen.

UC2 - Slet scenarie
I denne UC er det muligt for brugeren at slette et personligt oprettet scenarie. Brugeren skal åbne konsollen og vælge "Slet scenarie" i hovedmenuen, hvorpå en liste over de gemte scenarier fremstår. Der vælges nu det scenarie, der ønskes slettet. Herefter slettes scenariet fra scenariekontrol og brugeren sendes tilbage til hovedmenuen. Bemærk det er ikke muligt at slette det preinstallerede default scenarie.

UC3 - Rediger scenarie

I denne UC er det muligt for brugeren at redigere et personligt oprettet scenarie. Brugeren skal åbne konsollen og vælge "Rediger scenarie" i hovedmenuen, hvorpå en liste over de gemte scenarier fremstår. Der vælges nu det scenarie, der ønskes redigeret. Herefter vælger brugeren det element, der skal redigeres og opdaterer denne med nye værdier. Efter redigeringen gemmes ændringerne i scenariekontrol og brugeren sendes tilbage til hovedmenuen. Bemærk det er ikke muligt at redigere det preinstallerede default scenarie.

UC4 - Aktiver scenarie
I denne UC er det muligt for brugeren at aktivere et scenarie. Brugeren skal åbne konsollen og vælge "Aktiver scenarie" i hovedmenuen, hvorpå en liste over gemte scenarier fremstår. Der vælges nu det scenarie, der ønskes aktiveret. Brugeren bliver bedt om at bekræfte valget, og herefter aktives scenariet. Brugeren sendes nu tilbage til hovedmenuen.

UC5 - Deaktiver scenarie
I denne UC er det muligt for brugeren at deaktivere et scenarie. Brugeren skal åbne konsollen og vælge "Deaktiver scenarie" i hovedmenuen, hvorpå brugeren skal bekræfte deaktiveringen. Det aktive scenarie deaktiveres og brugeren sendes tilbage til hovedmenuen.

UC6 - Dæmper op
I denne UC justerer lysdæmperen pærens lysintensitet et trin op.

UC7 - Dæmper ned
I denne UC justerer lysdæmperen pærens lysintensitet et trin ned.

UC8 - Dæmper tænd
I denne UC tænder lysdæmperen pæren på fuld lysintensitet.

UC9 - Dæmper sluk
I denne UC slukker lysdæmperen for pæren.

UC10 - Tænd enkelt
I denne UC tænder GP-enheden den fra brugeren ønskede stikkontakt.

UC11 - Sluk enkelt
I denne UC slukker GP-enheden den fra brugeren ønskede stikkontakt.

UC12 - Tænd alle
I denne UC tænder GP-enheden for alle stikkontakterne.

UC13 - Sluk alle
I denne UC slukker GP-enheden for alle stikkontakterne.

UC14 - Manuel styring

I denne UC er det muligt for brugeren at overstyre det aktiverede scenarie. Brugeren trykker på den manuel overwrite kontakt tilhørende den ønskede enhed, hvorpå dennes tilstand inverteres (tændes/slukkes). Bemærk det er ikke muligt at ændre på lysintensiteten i lysdæmperen.

UC15 - Lås op
I denne UC er det muligt for brugeren at låse op for systemet, så det kan tilgås. Brugeren indtaster kode 1 og trykker Enter. Herefter indtaster brugeren kode 2 og trykker på Enter. Afsluttende indtaster brugeren kode 3 og trykker på Enter, hvorefter kodelåsen sender signal til scenariekontrol om, at systemet er låst op (tilgængeligt).

UC16 - Lås
I denne UC er det muligt for brugeren at låse systemet, så det ikke kan tilgås. Brugeren trykker på Enter, hvorefter kodelåsen sender signal til scenariekontrol om, at systemet er låst (ikke tilgængeligt).

6. Projekt beskrivelse

6.1. Projektgennemførelse

Tidligt i projektforløbet havde gruppen forskellige overvejelser om arbejdsfordelingen i gruppen. En idé om opdeling i 2 grupper opstod; en gruppe til at håndtere hardware delen og en gruppe til at håndtere software delen. Den idé blev imidlertid hurtigt tilsidesat, da det under forrige semesterprojekt viste sig at være en kilde til forvirring blandt gruppens medlemmer, da man hurtigt kunne miste overblikket over grænsefladen mellem software og hardware.
I dette semesterprojekt blev det derfor besluttet at benytte en anden fremgangsmåde, modul-opdeling. Projektet består i teorien af tre forskellige dele: **Masterenhed, GP-enhed** og **Lysdæmper**. I gruppen blev disse tre moduler uddelt, to personer på hver. Dog blev det bestemt at opdele det en smule mere, sådan at X10 enhederne, som kan forstås som et konceptuelt modul, der spænder sig over alle de andre moduler, blev et arbejds-modul for sig selv. Sådan endte det ud med, at der var fire grupper hver bestående af to mand, som havde hænderne i både software og hardware delen. Det har i modul-grupperne skabt en bedre helhedsforståelse, og misforståelser mellem grænseflader er på denne måde blevet nedsat betydeligt, i forhold til sidste semesterprojekt.

Som noget af det første i projektet blev der nedsat en tidsplan, for at give et overblik over progressionen og målene. Tidsplanen er blevet revideret nogle få gange hovedsagligt på grund af fejl eller misforståelser. Overordnet set har det været en god retningslinje for at blive færdig i tide, et eksempel på tidsplanen ses herunder:

E2PRJ Gr. 4 Tidsplan Version 3:
Uge 14

UGE	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Problemformulering			21.												
Kravspecifikation			21.												
Accepttest			21.												
System Arkitektur								25.							
Design								25.							
Implementering												23.			
Test												23.			
Rapport færdiggørelse												23.			
Færdig	Igangværende			Deadline	evt	Undervisningsfri periode							Planlagt		

6.2. Metoder

6.2.1. V-modellen

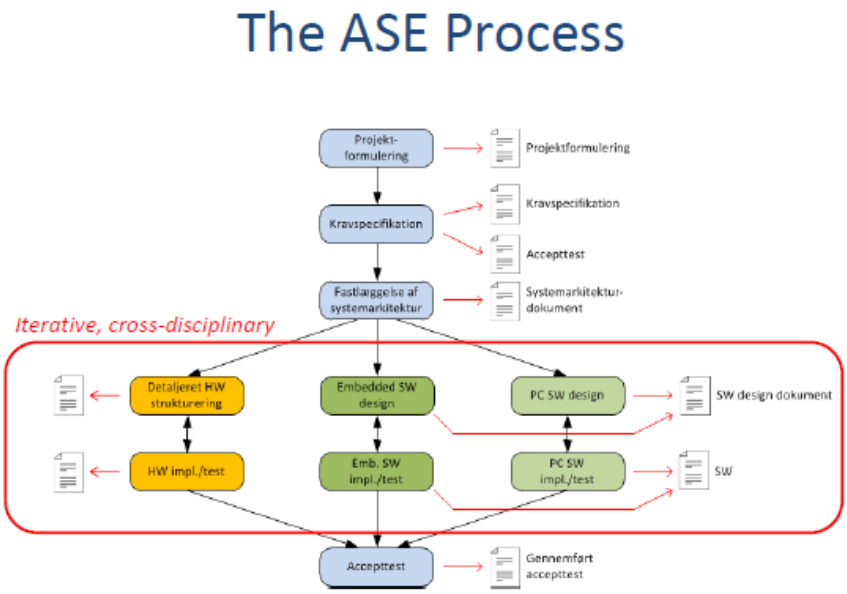
Vi har delvis benyttet os af V-modellen, der står for verifikation og validerings model. Denne model tester om blokkene fungerer som forventet samt om det samlede system fungerer efter hensigten. Dog afviges der fra modellens principper, da der kun er blevet udarbejdet en accepttest efter kravspecifikationen blev lavet. Der er blevet udført en Unit test på modulerne, men der er dog kun blevet foretaget en begrænset dokumentering af disse test.

6.2.2. SysML

For at danne et visuelt overblik over vores design, har vi benyttet os af SysML. SysML har den gode egenskab, at det giver indsigt i både software og hardware delens opbygning. Til at beskrive det overordnede system, er der lavet et BDD (blok definitions diagram) som viser de blokke der indgår i systemet, og hvordan de er forbundet. Ud fra BDD blokkene bliver der herefter lavet et IBD (interne blok diagram), som viser, hvad de tre hovedkomponenter Masterenhed, Lysdæmper og GP-enhed indeholder og deres forbindelser indbyrdes samt til omverdenen. Disse tre hovedkomponenter bliver efterfølgende grundigt beskrevet i hvert deres IBD, hvor alle de interne signaler er påtegnet og navngivet. Den softwaremæssige del er beskrevet ved hjælp af en applikationsmodel. Applikationsmodellen er lavet ud fra en domænemodel, et sekvensdiagram og et state machine diagram. Domænemodellen er lavet ud fra use casene, hvor de relevante navneord er blevet indsat i blokke og forbundet sammen. Disse blokke bliver herefter indsat i et sekvensdiagram, hvor beskederne imellem blokkene beskriver de forskellige steps i use casene. Beskederne bliver herefter til funktioner i et klassediagram, der indeholder de tilhørende blokke.

6.2.3. ASE proces

ASE processen er den overordnede proces, der er blevet benyttet gennem hele projektet (Figur 5). ASE processen er en proces, der opbygges omkring en konkret problemformulering udarbejdet fra en opgavebeskrivelse. Ud fra problemformuleringen er der udarbejdet en kravspecifikation med dertilhørende use cases for alle mulige scenarier i systemet. Med den færdige kravspecifikation er der herefter udarbejdet en systemarkitektur, der senere splittes op i hardware- og softwaredesign efterfulgt af implementering til disse. Afsluttende er accepttesten blevet udfyldt for at sikre, at vores implementering overholder vores tidligere fastlagte kravspecifikation.



Figur 5 - Overblik over ASE-processen

6.3. Hardware strukturering

Som beskrevet i systembeskrivelsen, består systemet af tre separate moduler. For en oversigt over modulernes indbyrdes kommunikation henvises der til Figur 6, som viser forbindelserne mellem disse.

Masterenhed

Systemets masterenhed består hardwaremæssigt, af en ATmega32, en X.10 sender, en kodelås og en strømstyrings enhed. Brugeren kommunikerer med masterenheden gennem en seriel terminal (efter brugerens eget valg), og masterenheden sender beskeder til de andre to enheder over X.10 lysnettet.

Lysdæmper

Denne enhed er opbygget af en 8 bit AVR micro controller, en X.10 modtager, en halogen driver og en strømstyrings enhed. Dette modul kan modtage kommandoer via X.10 lysnettet, eller vha en lokal overwrite kontakt.

GP-enhed

General purpose enheden består af en 8 bit AVR micro controller, en X.10 modtager, en relæ kreds med 4 stikkontakter, og en strømstyrings enhed. På samme måde som lysdæmperen modtager GP-enheden styring via X.10 lysnettet eller fra et antal af overwrite kontakter.

Manuel overwrite

Disse kontakter er placeret sammen med de almindelige afbrydere til lyset i rummet. Ved tryk på kontakten for manuel overwrite, vil den pågældende enhed skifte tilstand (fx fra tændt til slukket). Manuel overwrite kontakterne optræder på lysdæmperen samt en kontakt for hver stikkontakt i GP-enheden. Det er ikke muligt at dæmpe gennem manuel overwrite kontakterne. Trykkes der på kontakten mens et scenarie er aktivt, vil enheden skifte tilstand, men kan overskrives ved en ny kommando fra de aktive scenarie.

6.3.1. Signalbeskrivelse

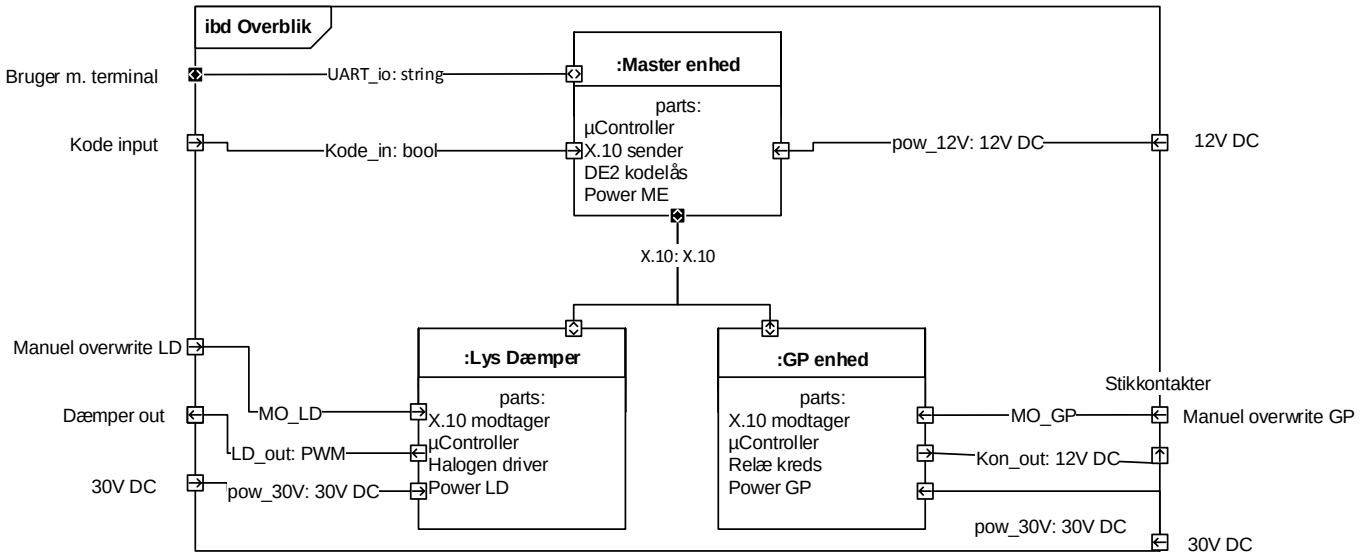
Herunder fremstår en beskrivelse af de benyttede signaler fra ovenstående diagrammer. Ved digitale signaler forstås CMOS standard, medmindre andet er anført.

Signalnavn	Signaltype	Definition	Beskrivelse
UART_io	Seriel kommunikation	USB	Forbindelsen til terminal
X.10	Data signal på X.10 lysnettet	X.10 ind. Stand.	Dette er datasignalet på X.10 lysnettet som benyttes af masterenheden til at styre de andre enheder
kode_in	Fysiske switches	Indstilling af kode på switches.	Kode indstillet på switches af bruger.
kode_val	Digital	0V eller 5V digitalt spænding	5V= låst op
X10_data	Seriel kommunikation	5V digitalt serielt signal	
X10_int	Interrupt	5V digitalt signal	Bruges til at time X.10 signaler med X.10 lysnettet
pow_5V	Analog	Analog 5V spænding	Forsyningsspænding, Kan levere 0,9A
pow_12V	Analog	Analog 12V spænding	Forsyningsspænding, Kan levere 0,9A
pow_30V	Analog	Analog 30V spænding	Forsyningsspænding Kan levere 1,8A (90% af max for lab forsyning)
X10_GPdata	Seriel kommunikation	5V digitalt serielt signal	
X10_GPint	Interrupt	5V digitalt signal	Bruges til at time X.10 signaler med X.10 lysnettet
MO1_ow	Fysisk switch	5V spænding videregives ved tryk	Manuel overwrite kontakt til stikkontakt 1
MO2_ow	Fysisk switch	5V spænding videregives ved tryk	Manuel overwrite kontakt til stikkontakt 2
MO3_ow	Fysisk switch	5V spænding videregives ved tryk	Manuel overwrite kontakt til stikkontakt 3
MO4_ow	Fysisk switch	5V spænding videregives ved tryk	Manuel overwrite kontakt til stikkontakt 4
MO_LD	Fysisk switch	5V spænding videregives ved	Manuel overwrite kontakt til lysdæmperen

		tryk	
kon_in1	Digitalt	0V eller 5V digitalt spænding	5V= Aktiveret
kon_in2	Digitalt	0V eller 5V digitalt spænding	5V= Aktiveret
kon_in3	Digitalt	0V eller 5V digitalt spænding	5V= Aktiveret
kon_in4	Digitalt	0V eller 5V digitalt spænding	5V= Aktiveret
kon_out1	Analog	0-12V spænding	
kon_out2	Analog	0-12V spænding	
kon_out3	Analog	0-12V spænding	
kon_out4	Analog	0-12V spænding	
X10_LDdata	Seriel kommunikation	5V digitalt serielt signal	
X10_LDint	Interrupt	5V digitalt signal	Bruges til at time X.10 signaler med X.10 lysnettet
LD_in	Digital	PWM signal	
LD_out	Analog	0-12V spænding	

6.3.2. IBD System Overblik

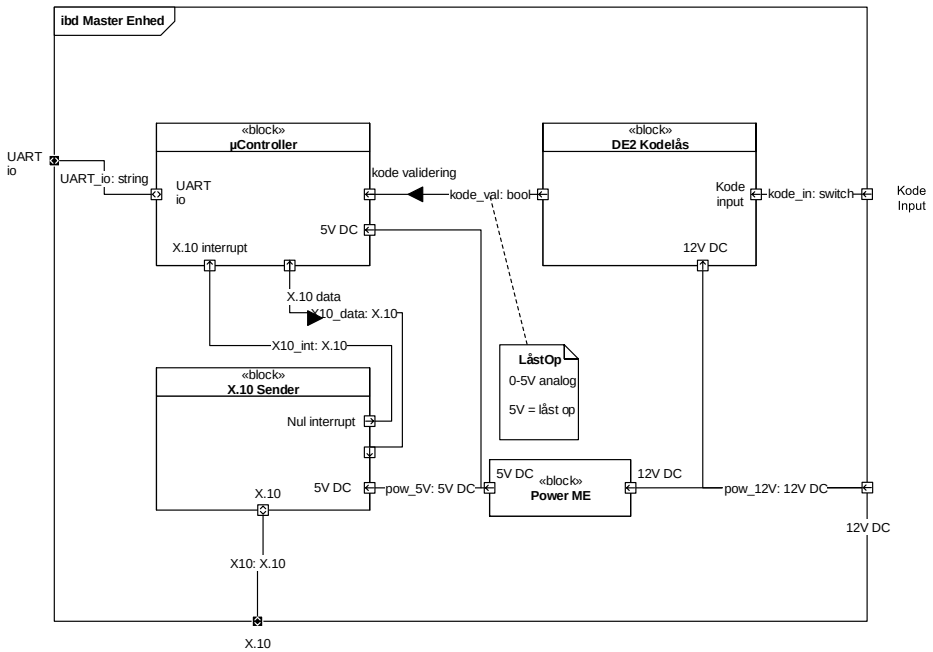
På Figur 6 vises home automation systemets overordnede IBD diagram. Diagrammet beskriver hvordan de overordnede blokke, er forbundet internt i systemet, og deres forbindelser til omverdenen.



Figur 6 - Overordnet IBD diagram

6.3.3. IBD Masterenhed

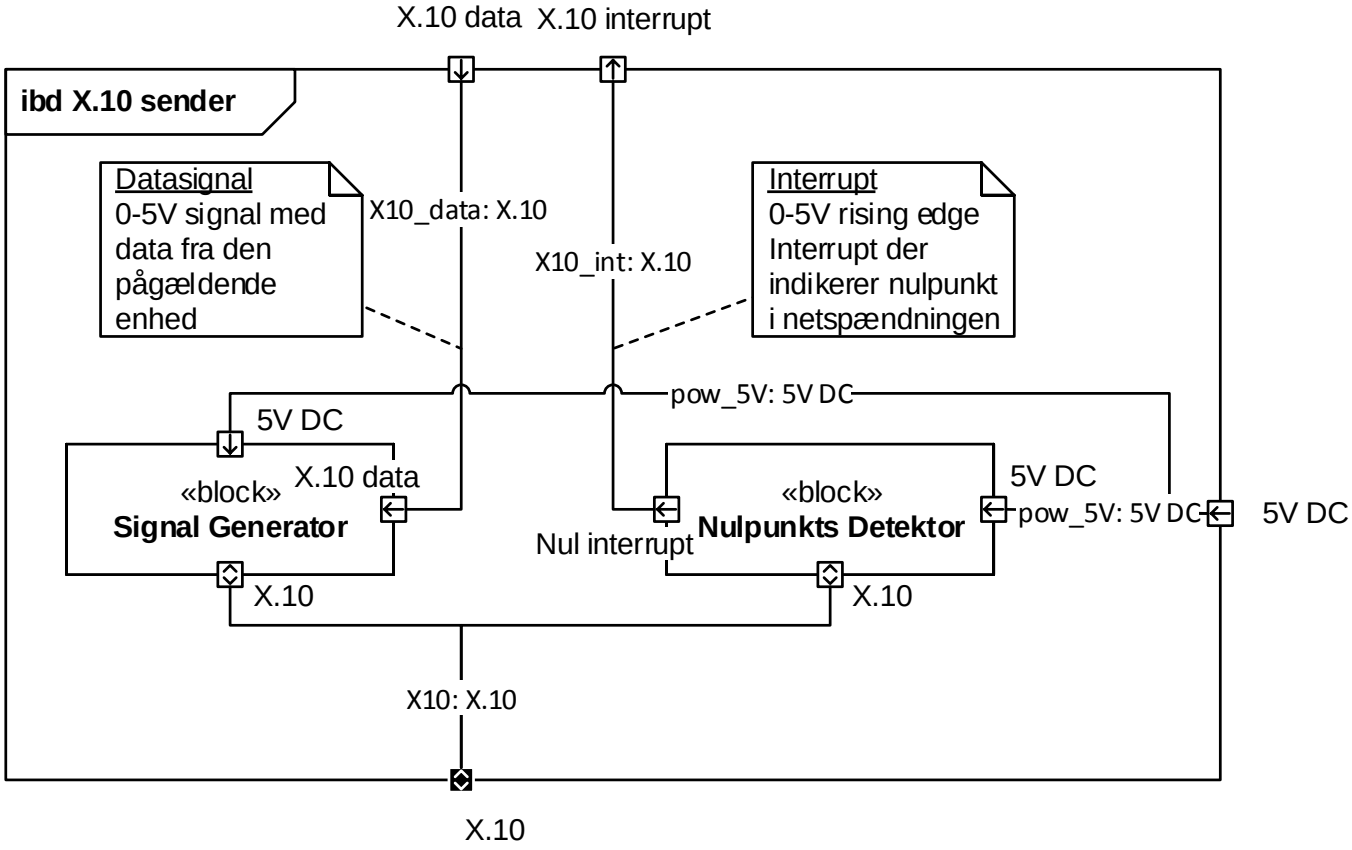
Figur 7 viser et IBD diagram over systemets masterenhed. På diagrammet vises hvilke forbindelser der eksisterer internt i blokken masterenhed, og hvordan kommunikationen fungere i mellem masterenheden, brugerkonsollen, kodelåsen og X.10 lysnettet.



Figur 7 - IBD for Masterenheden

6.3.4. IBD X.10 Sender

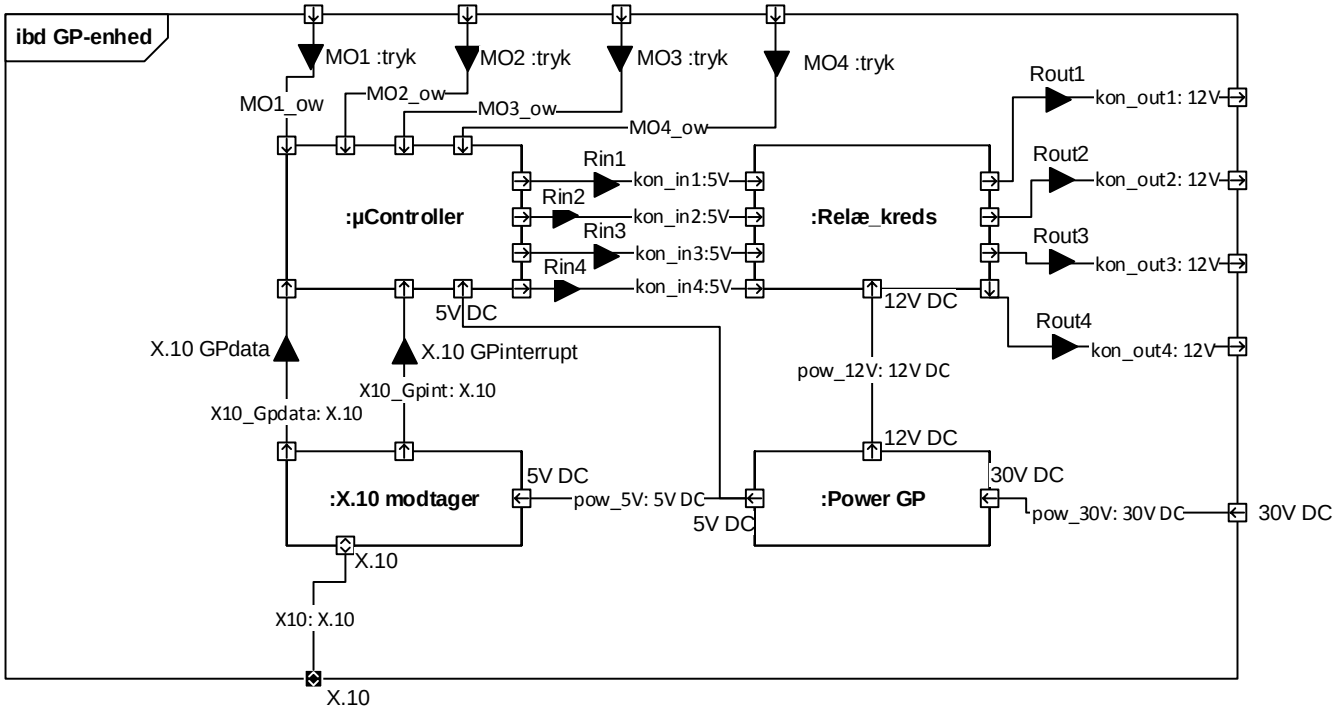
Figur 8 viser et IBD diagram over systemets X.10 sender, som er placeret i masterenheden. Diagrammet beskriver kommunikationsvejene i senderblokken, og senderens forbindelser ud til X.10 lysnettet og til µControlleren i masterenheden.



Figur 8 - IBD for X.10 senderen

6.3.5. IBD GP-Enhed

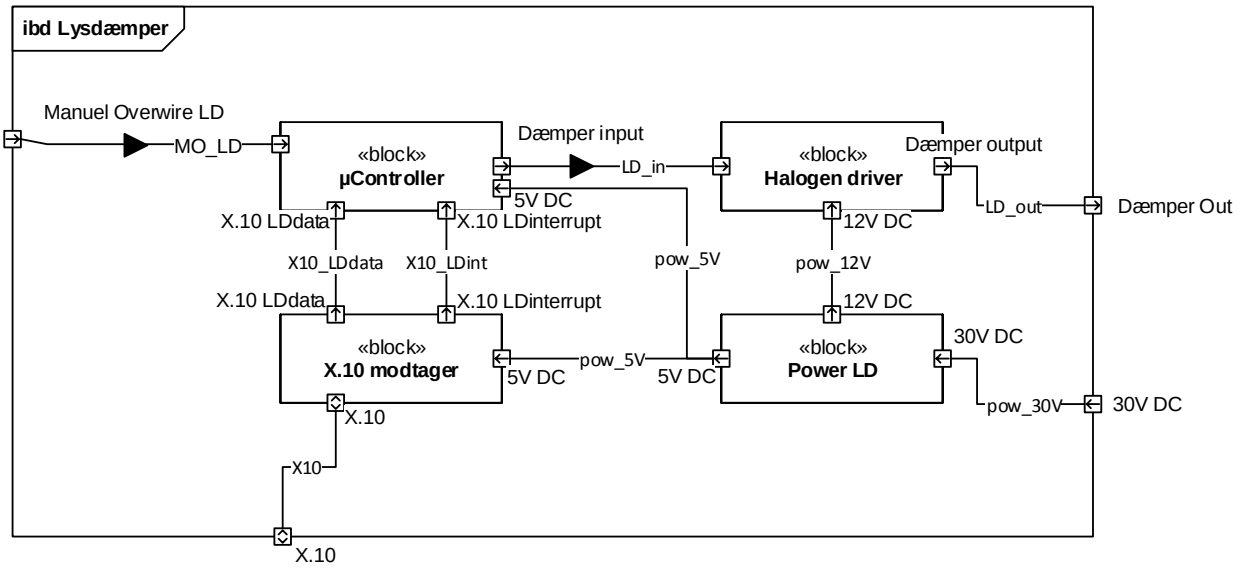
Figur 9 viser et IBD diagram over GP-enheden. Diagrammet fortæller hvilke forbindelser der optræder internt i blokken GP-enheden, og hvordan GP-enheden kommunikerer med x.10, stikkontakterne og trykknapperne fra manuel overwrite.



Figur 9 - IBD for GP-enheden

6.3.6. IBD Lysdæmper

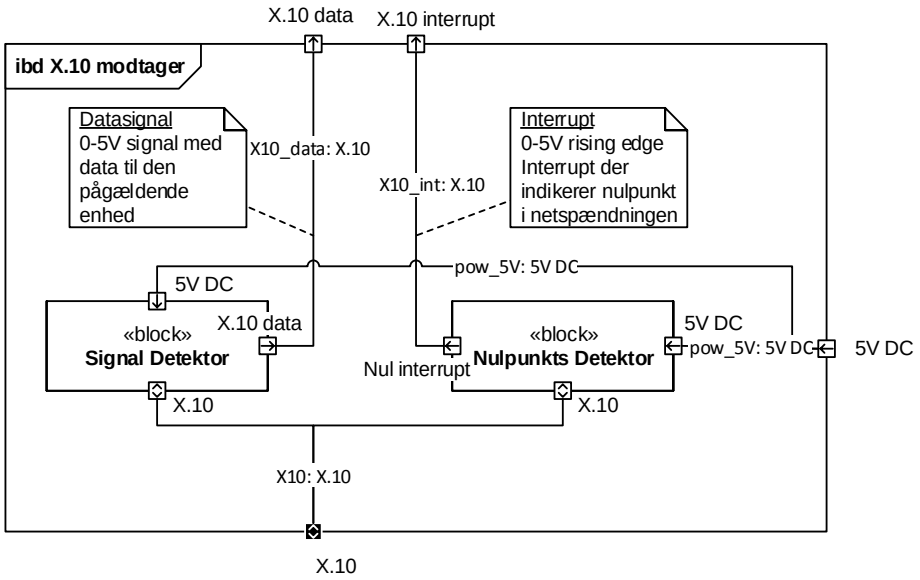
Fejl! Henvisningskilde ikke fundet. viser et IBD diagram over lysdæmperen. Diagrammet beskriver hvilke forbindelser der forekommer i blokken lysdæmper, og hvordan lysdæmperen er forbundet med X.10 lysnettet, pæren og trykknapperne fra manuel overwrite.



Figur 10 - IBD for lysdæmperen

6.3.7. IBD X.10 Modtager

Figur 11 viser et IBD diagram over systemets X.10 modtager, som er placeret i GP-enheden og lysdæmperen. Diagrammet beskriver kommunikationsvejene i modtagerblokken, og modtagerens forbindelser ud til X.10 lysnettet og til µController i henholdsvis GP-enheden og lysdæmperen.



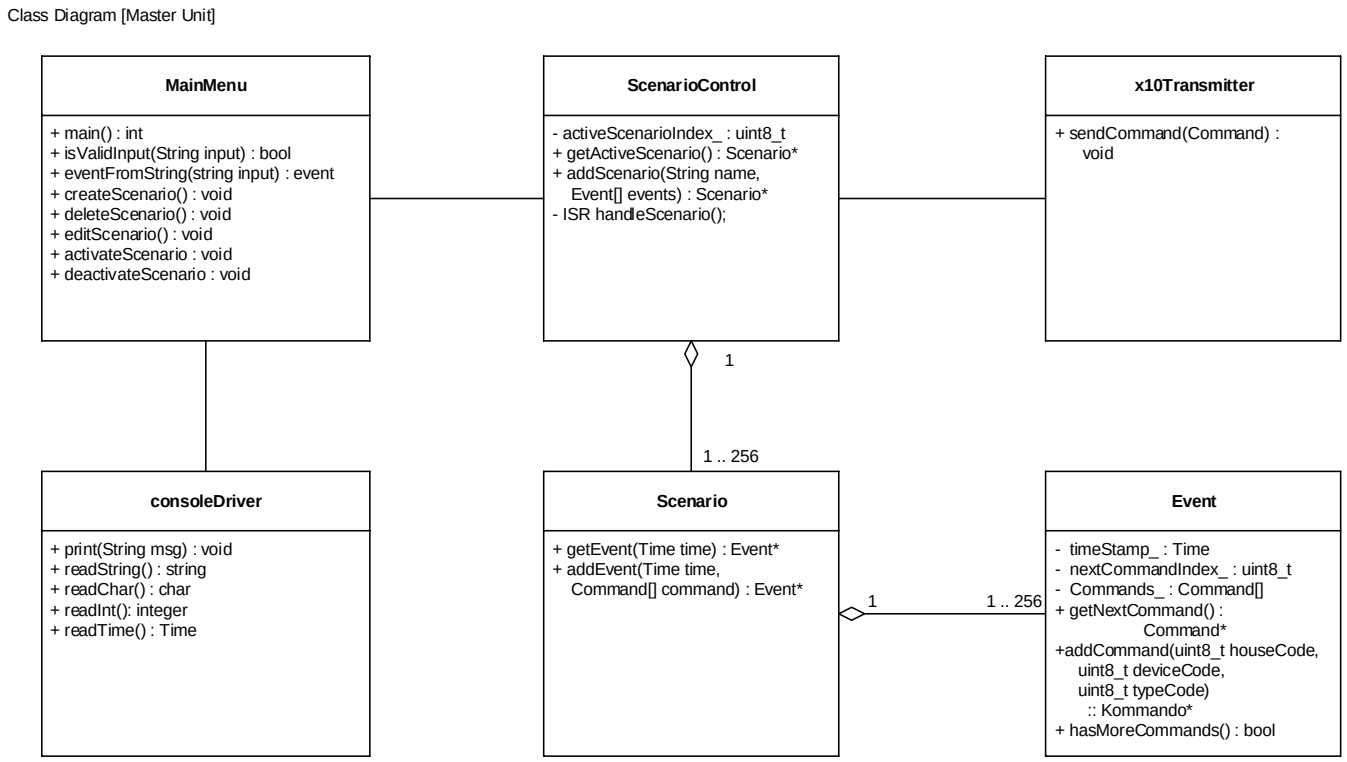
Figur 11 - IBD for X.10 Modtageren

6.4. Software strukturering og design

Nedenfor vises de færdige klassediagrammer for de tre delsystemer. For yderligere information herom henvises til dokumentationen.

6.4.1. Klassediagram Masterenhed

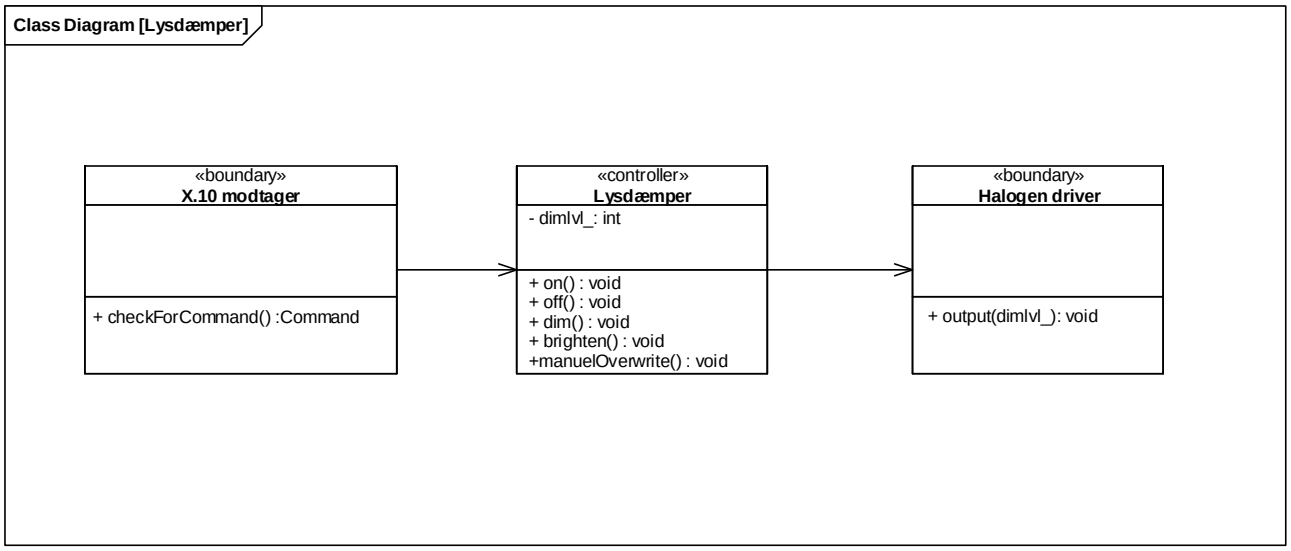
Klassediagrammet for denne applikationsmodel er skrevet på engelsk for at der ikke opstår tvivl om klassenavne når koden skal implementeres. (ifølge IHA's kodestandard for C/C++ skal alle navne være på engelsk). Klassediagrammet er blevet udarbejdet fra en applikationsmodel, der er opbygget fra en domænemodel og et sekvensdiagram.



Figur 12 - Klassediagram for Masterenheden

6.4.2. Klassediagram Lysdæmper

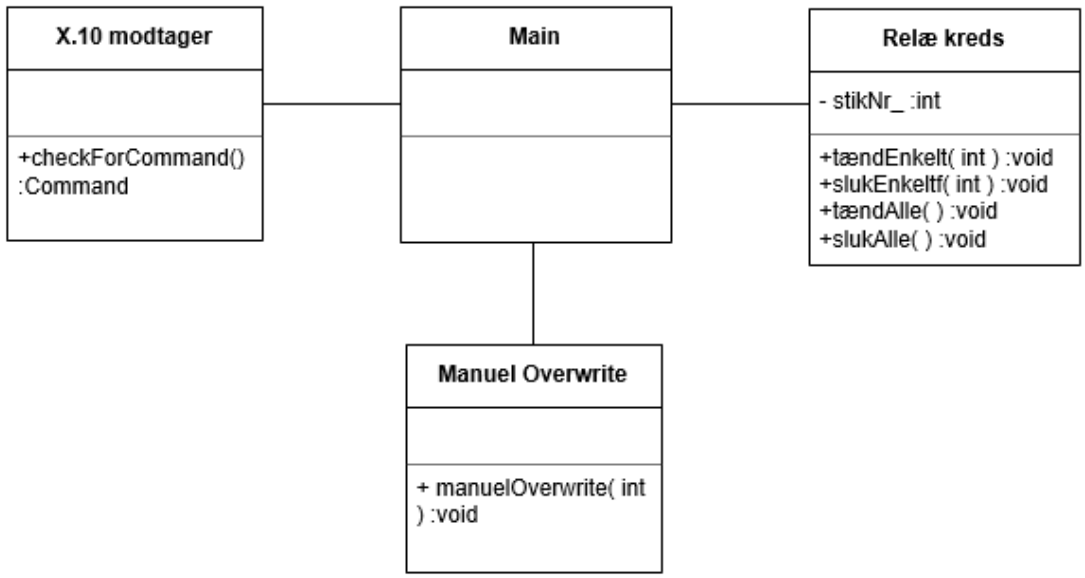
Metoderne er ændret til engelsk for at der ikke bliver skabt forvirring ift. metodenavnene. Hermed er IHA kodestandarden for C/C++ overholdt. Klassediagrammet er opbygget ud fra en applikatinosmodel, der er opbygget fra en domænemodel og et state machine diagam. Grunden til et state machine diagram er benyttet frem for et sekvensdiagram i denne delenhed er, at det giver et bedre overblik programmeringsmæssigt end et sekvensdiagram ville have gjort.



Figur 13 Klassediagram for lysdæmper

6.4.3. Klassediagram GP-enhed

Klassediagrammet for GP-enheden er lavet ved hjælp af en applikationsmodel, som består af en domænemodel og et sekvens diagram. Domænemodellen er opbygget af blokke, som beskriver de fysiske elementer i GP-enheden, der er lavet ud fra use casene, der vedrører GP-enheden. Disse blokke bliver herefter indført i sekvensdiagrammet, hvor beskederne i mellem blokkene repræsenterer trinene i use casene. Beskederne bliver efterfølgende til funktioner i klassediagrammet for GP-enheden.



Figur 14 - Klassediagram for GP-enheden

6.5. Hardware design/implementering

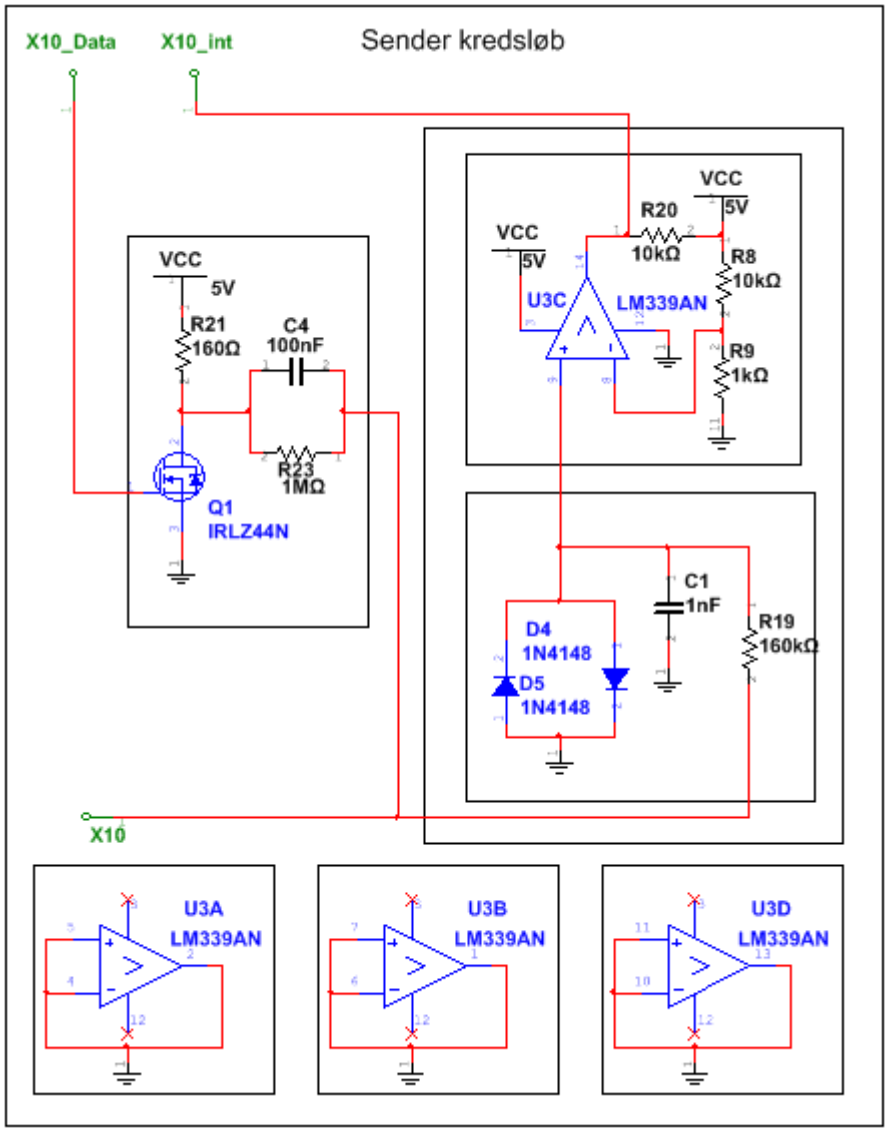
I de efterfølgende afsnit er der udviklet et design resulterende i en efterfølgende implementering. For yderligere information såsom beregninger henvises til dokumentationen.

6.5.1. X.10 sender og modtager

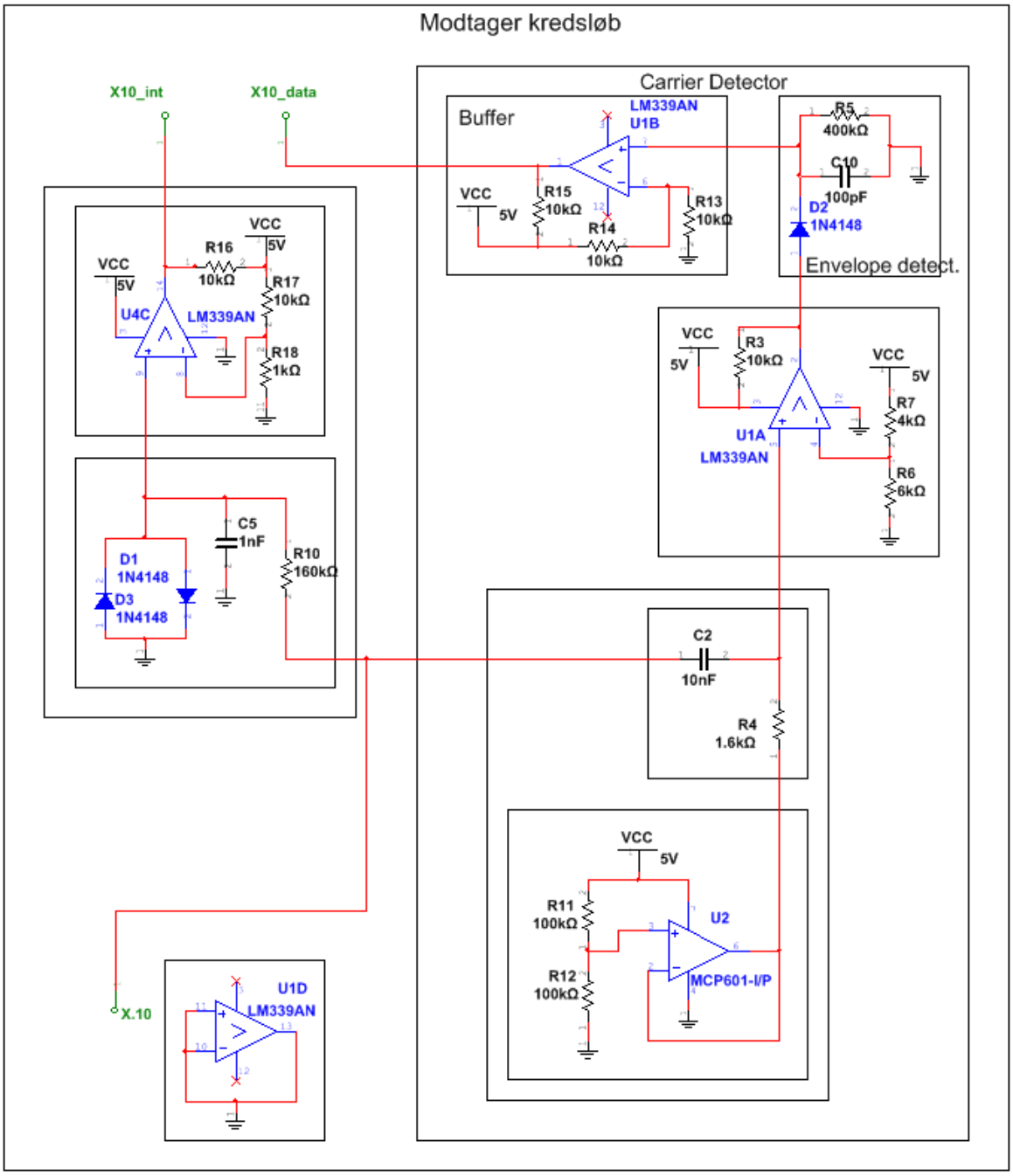
Da X.10 sender og modtager HW har en større samhørighed med hinanden end med de moduler de er inkorporeret i beskrives de her samlet.

X.10 sender og modtager deler det samme design for deres zero-cross detektorer. Dette design består af et lavpasfilter for at undgå forstyrrelser fra 120kHz komponenten, en spændingsbegrænser der fjerner spændinger som ligger for langt fra 0V til hvad komponenterne kan tåle, og en komparator som giver et 5V output når 50Hz komponenten er positiv, og 0V når 50Hz komponenten er negativ.

Dette signal kan læses med µControllerens eksterne interrupt funktion, med interrupt på alle logiske skift.



Figur 15 - Kredsløbsdesign for X.10 senderen



Figur 16 - Kredsløbsdesign af modtageren

X.10 senderen består hovedsageligt af et højpasfilter, som tillader at 120kHz signalet fra µControlleren kan føres ud på lysnettet, men at den høje spænding på lysnettet ikke kommer ind til resten af sender kredsen, som ville blive ødelagt.

Senderen benytter en CMOS MOSFET transistor til at drive 120kHz signalet fra µControlleren ud i filteret.

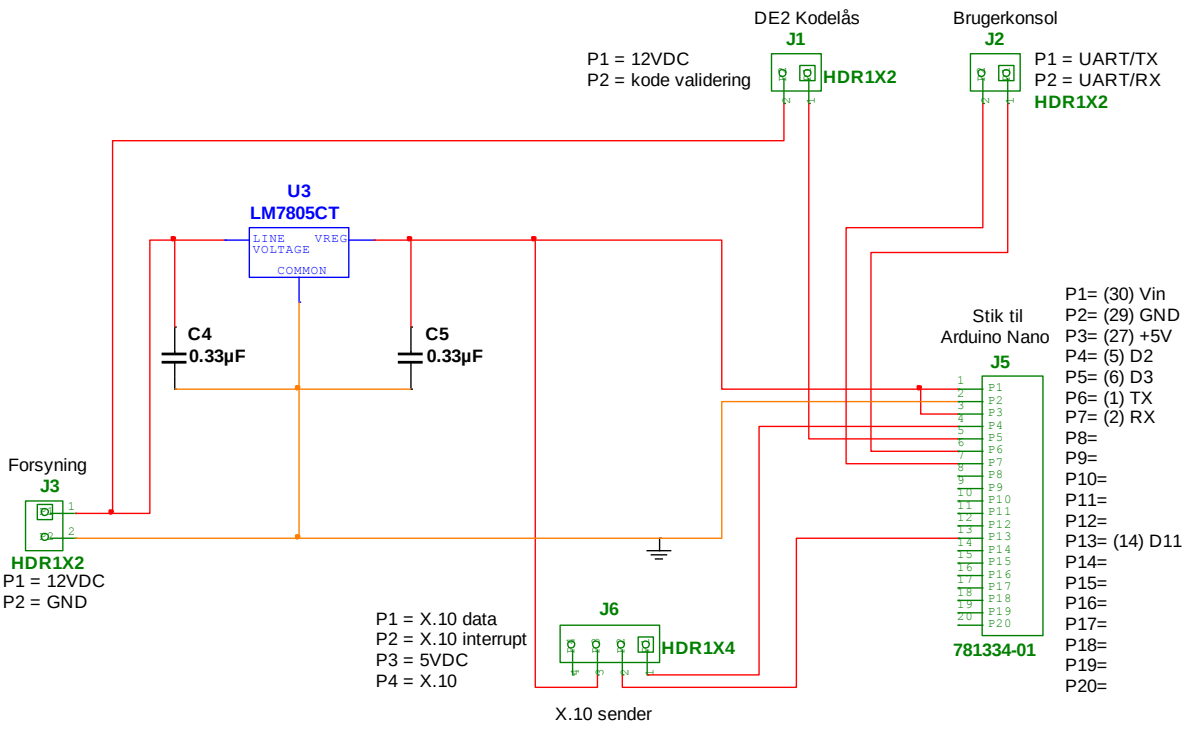
X.10 Modtageren består ligeledes af et højpasfilter, men dette er lavet med en reference til 2.5V i stedet for ground, således at outputtet fra filteret ligger i 0-5V området. Dette tillader brug af single supply

komparatorer, hvormed systemet ikke behøves at have en negativ spændingsforsyning, og kan køre på en enkelt 5V forsyning sammen med µControlleren.

Efter signalet har været gennem filteret benyttes der en diode og et RC led til at udglatte signalet, således at der bliver et højt signal hvis der er en høj frekvens komponent på lysnettet og et lavt hvis der ikke er. Dette tillader SW nemt at konstatere om der er et signal når nulkryds interruptet modtages.

6.5.2. Masterenhed

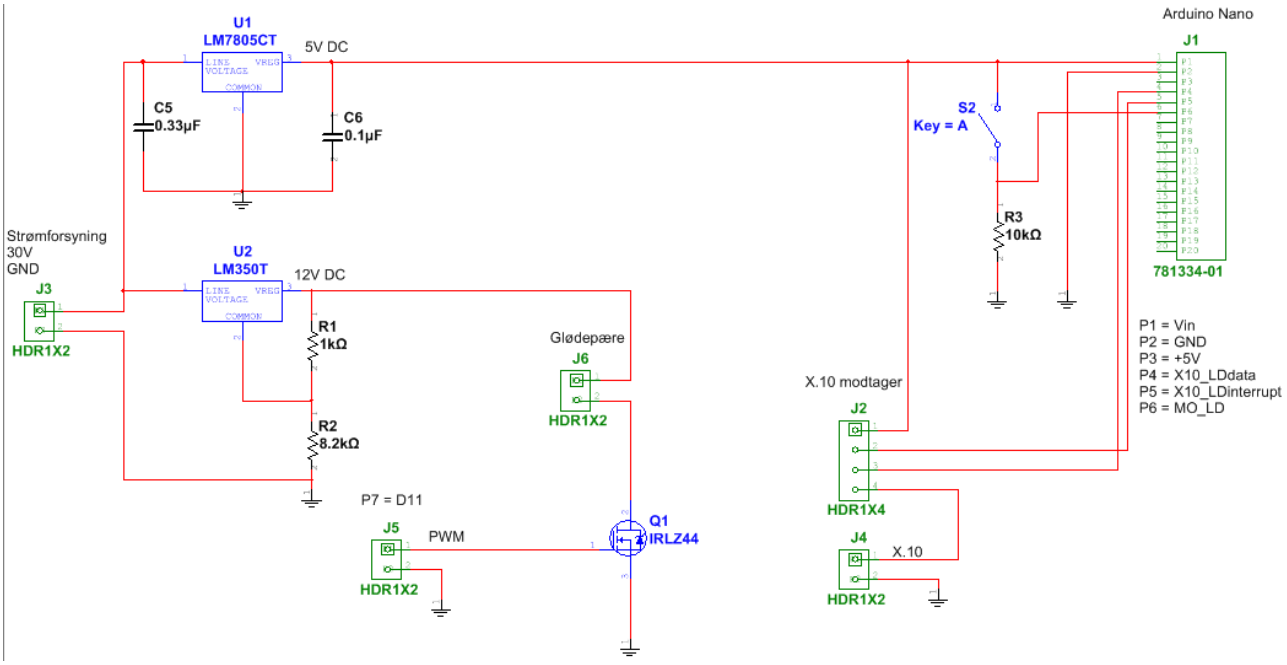
På Figur 17 ses et diagram over masterenheden som indeholder en spændingsregulator, der nedregulerer den eksterne forsyningsspænding på 12VDC til 5VDC. Denne 5VDC spænding bliver brugt til at forsyne Arduino nano boardet og X.10 senderen. Herudover er stikkene til DE2 kodelåsen, brugerkonsollen, Arduino nano boardet og X.10 senderen påtegnet, for at vise de interne forbindelser hertil.



Figur 17 - kredsløbsdesign af masterenhed

6.5.3. Lysdæmper

På Figur 18 ses kredsløbsdesignet for lysdæmperen. Dette er lavet ud fra IBD diagrammet for lysdæmperen (Figur 10). Blokken Power-LD består af de to spændingsregulatorer (U1-U2). Halogen driver blokken er stikkene J5 og J6 og mosfet transistoren(Q1). Til de to resterende blokke µController og X.10 modtager er der lavet stik ud til disse. Modtager designet ses på Figur 16.



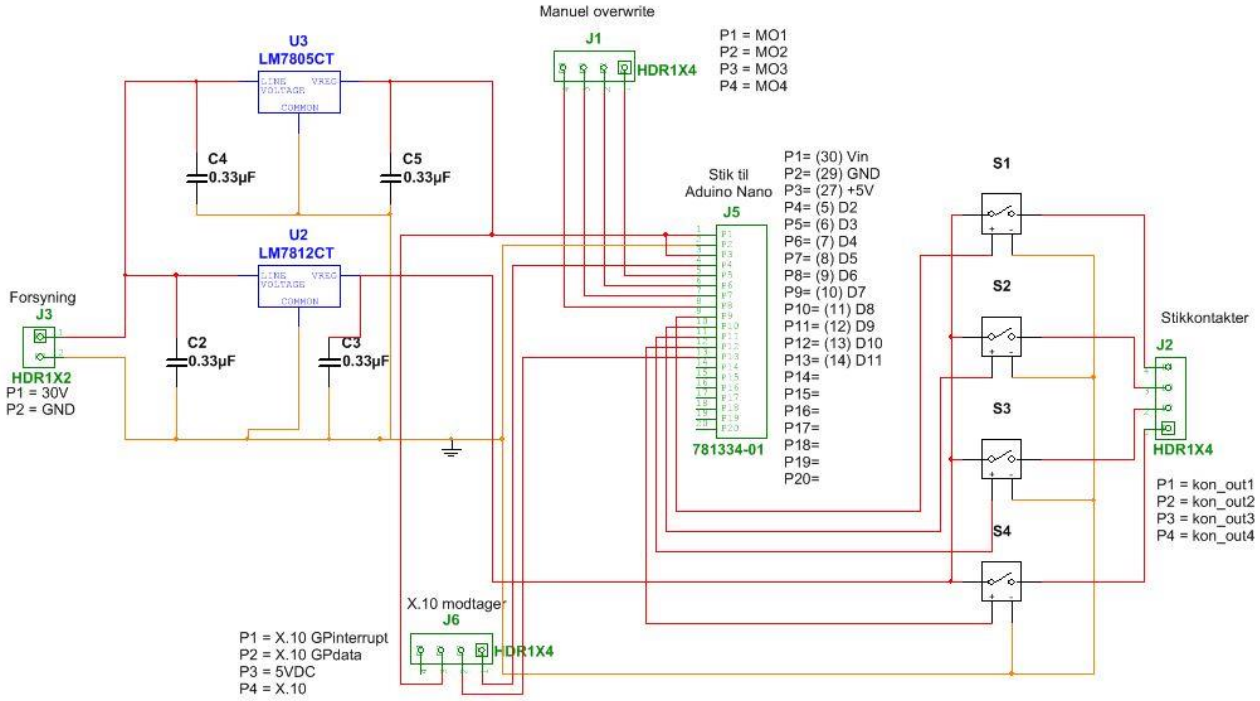
Figur 18 - Kredsløbsdesign for lysdæmperen

6.5.4. GP-enhed

Selve GP-enheden vil kun blive designet med kredsløbsdesign samt headers og vil derfor ikke blive implementeret.

På Figur 19 ses kredsløbsdesignet for GP-enheden. U2 og U3 er vores spændingsregulatorer i blokken "Power GP" på IBD-diagrammet på Figur 9. Disse nedsætter spændingen til henholdsvis 5V og 12V. De valgte kondensatorer (C2-C5) er tilføjet indgangen og udgangen på regulatoren for at sikre en stabil spænding jf. databladet. Stikket J1 er forbindelserne ud til vores Manuel Overwrite kontakter, mens J2 er stikket til stikkontakterne i GP-enheden. Relækredsen på IBD-diagrammet består af de fire relæer (S1-S4), der modtager styresignal fra µControlleren. Selve modtageren i GP-enheden er identisk med modtageren i lysdæmperen på Figur 16.

OBS! Stiknavnenes numre i parentes på Arduino Nanoen refererer til benene på selve Arduinoen og ikke til selve µControleren i denne.



Figur 19 - Kredsløbsdesign til GP-enheden

7. Software implementering

7.1. X.10 sender og modtager

Da X.10 sender og modtager SW, har en større samhørighed med hinanden end med de moduler de er inkorporeret i beskrives på samme måde som HW samlet.

Både sender og modtager er implementeret i C da de begge er interrupt baserede, og AVR interrupt rutinen er statisk, og dermed svær at bruge sammen med et objekt orienteret modul.

Senderen fungerer ved at modulet har 3 variable med hhv. House-, Unit- og Functioncode, som sættes når der laves et kald til x10SendCommand(), hvorefter interrupt rutinen ud fra et index der inkrementeres ved hvert interrupt vurderer om der skal sendes et 120kHz signal ved den givne nulkrydsning.

Senderen har også funktionen x10Ready() som kan bruges til at vente på at senderen er færdig med at sende, og dermed klar til at sende en ny kommando.

Modtageren er implementeret som en statemachine som har forskellige states til at repræsentere hvilken del af en kommando den er ved at modtage. Bliver der på noget tidspunkt fundet en fejl (forskellige hus koder eller lign.) sættes et fejl flag, og modtageren returnerer til startbit state hvor den venter på en start bit.

Modtageren har en cirkulær buffer med plads til 10 kommandoer, hvilket vurderes til at være rigeligt, idet en kommando tager næsten et helt sekund at sende.

7.2. Masterenhed

Grundet tidspres er dette punkt udgået.

7.3. Lysdæmper

Lysdæmperkoden tager udgangspunkt i brug af et interrupt til manuelstyring af tænd/sluk funktion til lampen. Herudover bliver der genereret et PWM signal til styring af lysintensitet, som konstant bliver opdateret med det nuværende trin af dimlevel. Dimlevel er en global variabel i programmet som funktionerne ændrer på.

Se dokumentation/bilag for kode eksempler og hele koden.

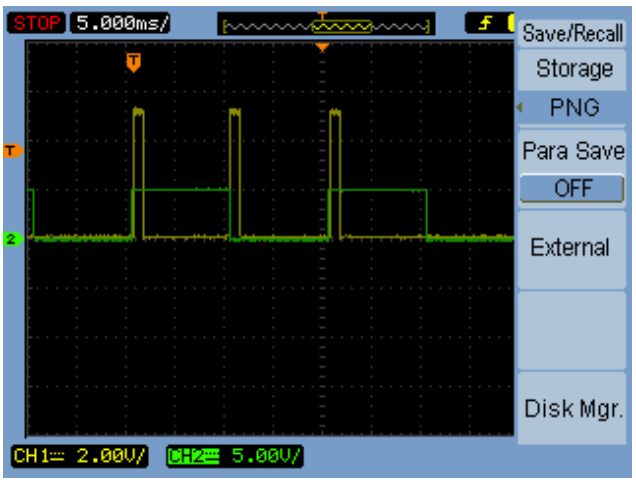
8. Resultater og test

Selve produktet endte langtfra som et færdigt produkt. Selve brugerinterfacet med brugermenuen blev aldrig implementeret, hvilket var altafgørende for at kunne opfylde vores problemformulering. Af samme årsag var det ikke muligt at udfylde vores accepttest, da testene krævede et fuldt fungerende system med valg gennem hovedmenuen. Dog kunne vi udføre vores modultests for kodelåsen, lysdæmperen, senderen samt hardware delen i modtageren. Ud fra modultestene kunne der så konkluderes, om de enkelte enheder virkede hver for sig. For yderligere dokumentation af testene henvises til dokumentationen.

Kodelåsen blev designet og implementeret særskilt i faget E2DSD med testdioder, der simulerede hvilket state kodelåsen var i. Kodelåsen gjorde det forventede, så denne virker som en selvstændig enhed.

Lysdæmperen blev modultestet uden modtagerenheden og fik derfor en kommando udenom denne. Dette resulterede i at glødepæren dæmpede op og ned som ønsket.

X10 sender softwaren er testet ved at µControlleren blev givet et kunstigt zero-cross signal fra en funktionsgenerator, hvorefter outputtet på timer benet blev studeret på Oscilloscop. På denne måde kunne bitsekvensen kontrolleres, bit for bit. På Figur 20 ses en start bit som den ses på benene fra µControlleren.

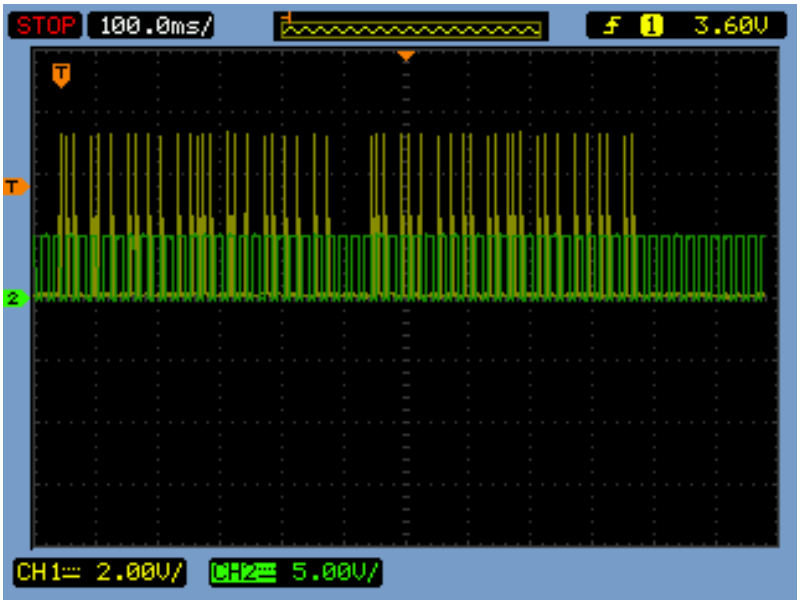


Figur 20 - Startbit fra softwaretest

Efter at det var blevet kontrolleret at softwaren genererede den korrekte bitsekvens med den korrekte timing blev hele HW kredsen sat op og forbundet til 18V transformatoren, og det blev kontrolleret på oscilloscopet at 120 kHz pulserne blev flettet korrekt ind i 18V ac signalet. Resultatet af dette ses på Figur 21.

Figur 21 - Close up af det afsendte data på X.10 lysnettet

Da modtagerens SW ikke nåede at blive færdig implementeret kunne denne ikke testes, men der blev udført en modultest af modtager HW'en. Dette blev gjort ved at sende en kommando med den i forvejen testede sender, og så bruge oscilloscopet til at teste hvad modtager µControlleren kiggede ind i. Udlæsningen af dette ses på Figur 22, hvor der sendes kommandoen: hus A, unit 2, ON.



Figur 22 - Output fra modtagerhardwaren

9. Konklusion

Igennem hele projektet er vores viden og kompetencer blevet udfordret og det har været en meget lærerig proces. At starte projektet op helt fra bunden ud fra en given opgavebeskrivelse, har givet et større indblik i, hvor meget man skal tage hensyn til under hele processen. Særligt har det vist sig, hvordan en god struktur letter arbejdet og hele tiden er med til at holde projektet overskueligt. Allerede fra starten havde vi en klar idé om, hvordan vi hurtigt ønskede at uddelegere arbejdet for at komme godt i gang med projektet tidligt. Dette viste sig at være en omvej, da vores use cases endte med at have forskellig syntax, sprogbrug og udformning. Et effektivt møde samlede overblikket og samlet fik vi lavet nogle faste rammer og et overordnet use case diagram. Dette endte i en del use cases (16 stk.) og disse kunne være reduceret, når vi ser tilbage på projektet. Selve udformningen af en kravspecifikation samt hvordan der arbejdes videre ud fra denne har givet et godt indblik i, hvordan de forskellige analyser og modeller benyttes samt hvilke tanker man skal gøre sig, når man udformer førnævnte dokumenter.

Før projektet havde vi en klar idé om, hvad vi ville implementere. Planen var at overholde minimumskravet om en lysdæmper, der kunne kommunikere over et X.10 lysnet. Samtidig ønskede vi at udvide systemet med en enhed, der ikke var fastlåst til et specifikt apparat, men i stedet fremstod som en fleksibilitet for brugeren. Efter færdiggørelsen af projektet "Home Automation System" kan vi overordnet se tilbage på et godkendt resultat. Selve dokumentationen af kravspecifikationen og struktureringen har taget længere tid end forventet i vores tidsplan, hvilket har resulteret i, at vi blev presset på implementeringen til sidst. Dette problem kunne måske være løst, ved at køre en form for Scrum med kravspec., struktureringen osv. som sprints, hvor få personer tog ansvaret for de rettelser, der måtte forekomme og køre deres egen sprint på disse herefter. Grundet tidspresset er vores produkt ikke blevet fuldt færdigudviklet, da flere ting ikke indgår som ønsket. Vi var nødsaget til, at måtte droppe implementeringen af GP-enheden, samt opbygningen af hele vores konsol med den ønskede brugermenu. Lysdæmperen er blevet færdig implementeret, og virker efter hensigten. Der er blevet udført en modultest på lysdæmperen, som efterfølgende er blevet dokumenteret. Den hardwaremæssige del af masterenheden, sender og modtager er blevet opbygget, men softwaren til masterenheden og modtageren mangler at blive implementeret. Der er derfor kun blevet foretaget en modultest på senderen som er beskrevet i dokumentationen.