

GRUPPE 4

E2PRJ2 - Dokumentation

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

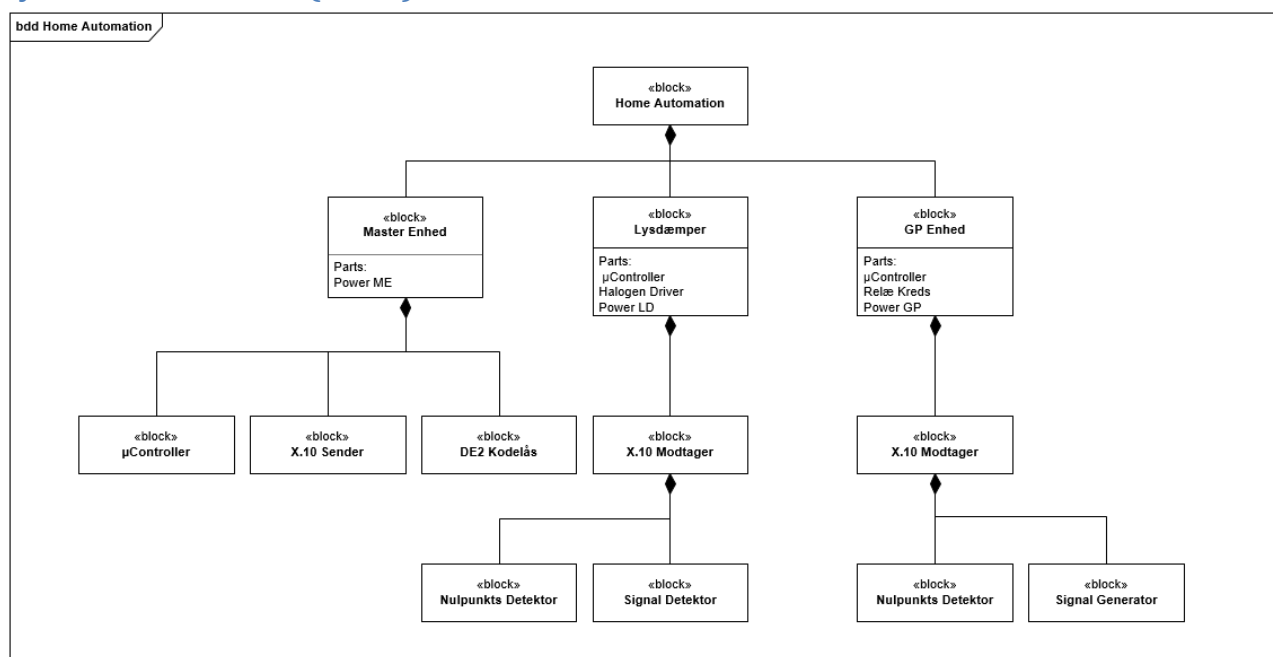
Indhold

1. Kravspecifikation (ALLE)	3
1.1. Systembeskrivelse (ALLE)	3
1.2. Krav (ALLE)	5
1.2.1. Funktionelle krav (ALLE)	5
1.2.2. Use Cases (ALLE)	8
1.2.3. Ikke-funktionelle krav (ALLE)	23
1.3. Accepttest (ALLE)	23
1.3.1. Accepttest for funktionelle krav (ALLE)	23
1.3.2. Accepttest for Use cases (ALLE)	24
1.3.3. Accepttest for ikke-funktionelle krav (ALLE)	38
2. Systemarkitektur (ALLE)	39
3. Strukturering (HW) (ALLE)	39
3.1. Masterenhed (ALLE)	39
3.2. Lysdæmper (ALLE)	39
3.3. GP-enhed (ALLE)	39
3.4. Manuel overwrite (ALLE)	39
3.5. Signalbeskrivelse (ALLE)	40
3.6. Overordnet BDD for Home Automation. (ALLE)	42
3.7. IBD System Overblik (ALLE)	43
3.8. IBD Masterenhed (ALLE)	44
3.9. IBD X.10 Sender (ALLE)	45
3.10. IBD GP-Enhed (ALLE)	46
3.11. IBD Lysdæmper (ALLE)	47
3.12. IBD X.10 Modtager (ALLE)	48
4. Strukturering/design (SW)	49
4.1. Applikationsmodel for Masterenheden (ALLE)	49
4.1.1. Domæne model (ALLE)	49
4.1.2. Sekvens- og state-machinediagrammer (ALLE)	50
4.1.3. Klassediagram (ALLE)	53
4.2. Applikationsmodel for Lysdæmper (ALLE)	54
4.2.1. Domæne model (ALLE)	54
4.2.2. State machine diagram (ALLE)	55

4.2.3.	Klassediagram (ALLE).....	56
4.3.	Header filer lysdæmper (ALLE).....	57
4.4.	Applikationsmodel for GP-Enhed (ALLE)	58
4.4.1.	Domæne model (ALLE)	58
4.4.2.	Sekvens diagram (ALLE).....	59
4.4.3.	Klasse applikations model (ALLE)	60
4.4.4.	Klasse diagram (ALLE)	60
4.5.	Header filer GP-enhed (ALLE)	61
5.	Design/Implementering (HW)	61
5.1.	X10 sender og modtager (Felix).....	61
5.1.1.	Zero-cross detektor (Felix).....	61
5.1.2.	Modtager (Felix)	64
5.1.3.	Sender (Felix)	67
5.2.	Masterenhed (ALLE)	68
5.3.	Lysdæmper (Martin, Anders)	70
5.4.	GP-enhed (Ruben, Niels og Freddie)	72
6.	Implementering(SW)	73
6.1.	DE2 Kodelås (ALLE)	73
1.1.	Sender/ Modtager (Felix)	76
1.2.	Lysdæmper (Martin, Anders)	76
7.	Test (HW/SW).....	76
7.1.	Modultest lysdæmper (Martin, Anders).....	76
7.2.	Modultest sender/modtager (Felix)	77
8.	Bilags Oversigt	79

1. Kravspecifikation (ALLE)

1.1. Systembeskrivelse (ALLE)



Figur 1 - Overordnet overblik

Dette system giver mulighed for at automatisere forskellige elektriske enheder i hjemmet via lysnettet, f.eks med henblik på indbrudsforebyggelse. Forebyggelsen består i en scenariestyreret 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 2.

Hovedmenu:

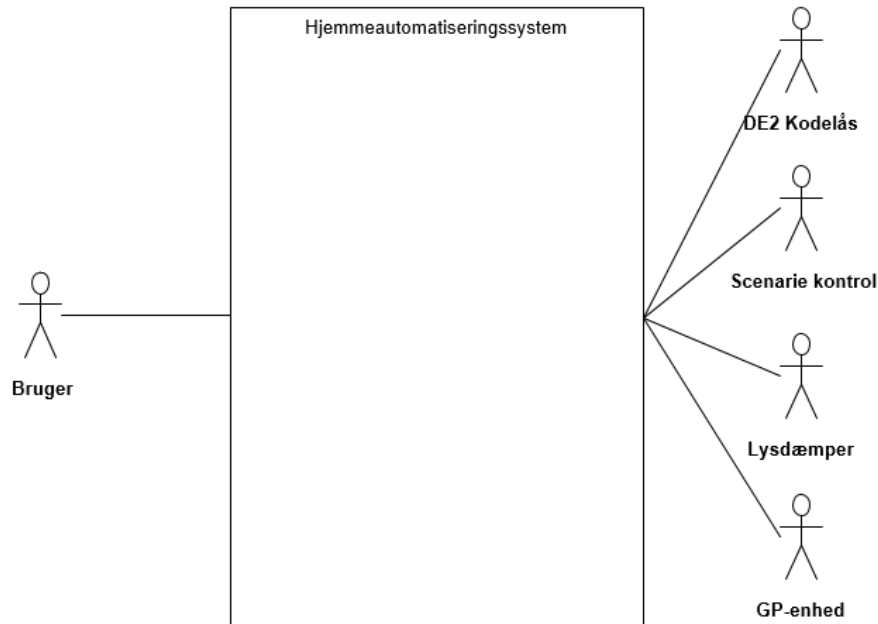
1. Opret scenarie
2. Slet scenarie
(Scenarier)
3. Rediger scenarie
(Scenarier)
4. Aktiver scenarie
(Scenarier)
5. Deaktiver scenarie
(Scenarier)

Figur 2 - Eksempel på hovedmenu

1.2. Krav (ALLE)

1.2.1. Funktionelle krav (ALLE)

1.2.1.1. Aktør beskrivelse og -diagram



Figur 3 - Aktør diagram

Aktørnavn:	Bruger
Alternativ reference:	
Type:	Primær
Beskrivelse:	Brugeren opretter/styrer scenarierne efter eget ønske gennem konsollen.

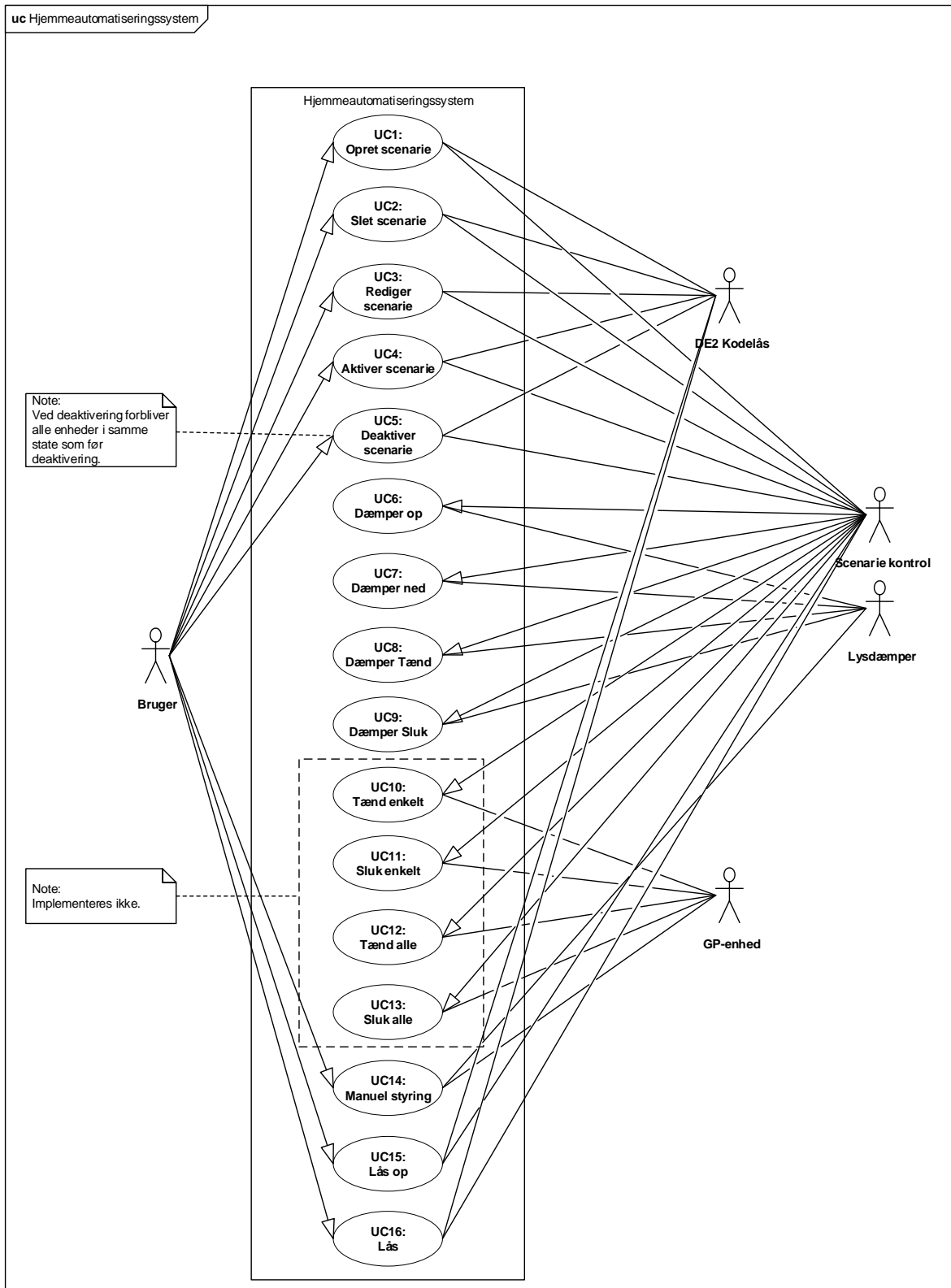
Aktørnavn:	DE2 Kodelås
Alternativ reference:	
Type:	Sekundær
Beskrivelse:	DE2 kodelåsen sikrer, at kun godkendte personer må foretage ændringer i scenarierne gennem konsollen.

Aktørnavn:	Scenarie kontrol
Alternativ reference:	Masterenhed
Type:	Sekundær
Beskrivelse:	Scenarie kontrollen er vores masterenhed, der administrerer og validerer de forskellige scenarier.

Aktørnavn:	Lysdæmper
Alternativ reference:	Dæmper
Type:	Sekundær
Beskrivelse:	Lysdæmperen er styret af scenarie kontrol i henhold til de aktive scenarier (Use case 6-9).

Aktørnavn:	General purpose
Alternativ reference:	GP-enheden
Type:	Sekundær
Beskrivelse:	GP-enheden er styret af scenarie-kontrol i henhold til de aktive scenarier (Use case 10-13).

1.2.1.2. Usecase diagram



Figur 4 - Use case diagram

1.2.2. Use Cases (ALLE)

1.2.2.1. Generelt for UC 1-5

Under hvilket som helst punkt i hovedforløbet i disse UCs kan brugeren vælge at klikke 'esc', som vil bringe brugeren tilbage til hovedmenuen.

1.2.2.2. UC1

Scenarie for use case 1:	Scenariet kontrol
Navn:	Opret scenarie
Use case ID:	1
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariet kontrol
Initialisere:	Oprette et scenarie forskelligt fra default scenarie
Forudsætninger:	Validering fra DE2 kodelås
Resultat:	Nyt scenarie bliver oprettet
Reference:	UC15 – Lås op

Hovedforløb:

1. Brugeren åbner konsollen, som starter hovedmenuen.
2. Brugeren vælger "Opret scenarie" i hovedmenuen.
3. Kommandoer bliver indtastet - Navn, Beskrivelse.
3-1. Scenariet kontrol værdier bliver indtastet og valideret.
[Undtagelse 3-1a.] De indtastede værdier er ugyldige
4. Scenariet gemmes i scenarielisten(kontrol).
5. Hovedmenuen vises igen.

Undtagelser:

- 3-1a. Der udskrives en fejlbesked til brugeren, og punkt 3-1 afvikles.

1.2.2.3. UC2

Scenarie for use case 2:	Scenariet kontrol
Navn:	Slet scenarie
Use case ID:	2
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariet kontrol
Initialisere:	Slette et specifikt scenarie
Forudsætninger:	Validering fra DE2 kodelås; Der er oprettet et scenarie
Resultat:	Det specifikke scenarie bliver slettet
Reference:	UC15 – Lås op, UC1 – Opret scenarie

Hovedforløb:

1. Brugeren åbner konsollen, som starter hovedmenuen.
2. Brugeren vælger "Slet scenarie" i hovedmenuen.
3. Brugeren vælger hvilket af de oprettede scenarier der skal slettes. (DEFAULT scenarie vises ikke på listen)
4. Brugeren bliver spurgt om det er det korrekte scenarie der skal slettes.
5. Det valgte scenarie bliver fjernet fra listen over mulige aktive scenarier.
6. Hovedmenuen vises igen.

1.2.2.4. UC3

Scenarie for use case 3:	Scenariet kontrol
Navn:	Rediger scenarie
Use case ID:	3
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariet kontrol
Initialisere:	Redigerer det valgte scenarie
Forudsætninger:	Validering fra DE2 kodelås; Der skal mindst være oprettet et scenarie
Resultat:	Det valgte scenarie bliver valgt
Reference:	UC15 – Lås op; UC1 – Opret Scenarie

Hovedforløb:

1. Brugeren åbner konsollen, som starter hovedmenuen.
2. Brugeren vælger "Rediger scenarie" i hovedmenuen.
3. Brugeren vælger hvilket scenarie der skal redigeres. (DEFAULT scenarie vises ikke på listen)
4. Brugeren vælger hvilket element af scenariet der skal redigeres (navn, beskrivelse, enhedsværdier).
 - 4-1. Kommandoer bliver indtastet - Navn, Beskrivelse.
 - 4-2. Enhedsværdier bliver indtastet og valideret.
[Undtagelse 4-2a.] De indtastede værdier er ugyldige.
5. Redigeringerne bliver gemt i Scenariet kontrol.
6. Hovedmenuen vises igen.

Undtagelser:

- 4-2a. Der skrives en fejlbesked til brugeren, og punkt 4-2 afvikles.

1.2.2.5. UC4

Scenarie for use case 4:	Scenariet kontrol
Navn:	Aktiver scenarie
Use case ID:	4
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariet kontrol
Initialisere:	Værdierne tilhørende det specifikke scenarie sendes til modtager
Forudsætninger:	Validering fra DE2 kodelås
Resultat:	Modtagerenheden aktiveres
Reference:	UC15 – Lås op

Hovedforløb:

1. Brugeren åbner konsollen, som starter hovedmenuen.
2. Brugeren vælger "Aktiver scenarie" i hovedmenuen.
3. Brugeren vælger hvilket scenarie der skal aktiveres.
4. Terminalen spørger brugeren om bekræftelse af scenarie valg.
5. Brugeren bekræfter.
[Undtagelse 5a.] Brugeren afviser.
6. Værdierne tilhørende det valgte scenarie sendes til Scenariet kontrol.
7. Hovedmenuen vises igen.

Undtagelse:

- 5a. Brugeren sendes tilbage til punkt 3.

1.2.2.6. UC5

Scenarie for use case 5:	Scenariet kontrol
Navn:	Deaktiver scenarie
Use case ID:	5
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariet kontrol
Initialisere:	Deaktiver det aktuelle scenarie
Forudsætninger:	Validering fra DE2 kodelås; Der er et aktivt scenarie.
Resultat:	Det aktuelle scenarie bliver deaktiveret
Reference:	UC15 – Lås op; UC4 – Aktiver scenarie.

Hovedforløb:

1. Brugeren åbner konsollen, som starter hovedmenuen.
2. Brugeren vælger "Deaktiver scenarie" i hovedmenuen.
3. Terminalen spørger brugeren om bekræftelse af scenarie valg.
4. Brugeren bekræfter.
[Undtagelse 4a.] Brugeren afviser.
5. Det aktuelle scenarie bliver deaktiveret.
6. Hovedmenuen vises igen.

Undtagelse:

- 4a. Brugeren sendes tilbage til punkt 2.

1.2.2.7. UC6

Scenarie for use case 6:	Lysdæmper
Navn:	Dæmper op
Use case ID:	6
Samtidige forekomster:	1
Primær aktør:	Scenarietkontrol
Sekundær aktør:	Lysdæmper
Initialiseres:	Signal fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Lysintensiteten stiger.
Reference:	Ingen

Hovedforløb:

1. Lysdæmperen modtager scenariekode med X.10 kommando "brighten" fra scenarietkontrol.
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden..
2. Lysdæmperen justerer lysintensiteten et trin op.
[Undtagelse 2a.] Lysintensiteten er på højeste niveau.

Undtagelser:

- 1a. Afslutter use case.
- 2a. Afslutter use case.

1.2.2.8. UC7

Scenarie for use case 7:	Lysdæmper
Navn:	Dæmper ned
Use case ID:	7
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	Lysdæmper
Initialiseres:	Signal fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Lysintensiteten ændres
Reference:	Ingen

Hovedforløb:

1. Lysdæmperen modtager scenariekode med X.10 kommando "dim" fra scenariekontrol.
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden
2. Lysintensiteten dæmpes et trin.
[Undtagelse 2a.] Lysintensiteten er på laveste trin.
[Undtagelse 2b.] Glødepæren er slukket.

Undtagelser:

- 1a. Afslutter use case.
- 2a Afslutter use case.
- 2c Afslutter use case.

1.2.2.9. UC8

Scenarie for use case 8:	Lysdæmper
Navn:	Dæmper tænd
Use case ID:	8
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	Lysdæmper
Initialiseres:	Signal fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Lys tændes med max intensitet
Reference:	Ingen

Hovedforløb:

1. Modtager x.10 kommando "on" fra Scenarie kontrol
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden
2. Lys tændes med max intensitet.
[Undtagelse 2a.] Lyset har allerede max intensitet

Undtagelser:

- 1a. Afslutter use case.
- 2a. Afslutter use case.

1.2.2.10. UC9

Scenarie for use case 9:	Lysdæmper
Navn:	Dæmper sluk
Use case ID:	9
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	Lysdæmper
Initialiseres:	Signal fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Lyset slukker
Reference:	Ingen

Hovedforløb:

1. Lysdæmperen modtager scenariekode med X.10 kommando "off" fra scenariekontrol.
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.
2. Lysdæmperen slukker Glødepæren.
[Undtagelse 2a.] Glødepæren er slukket.

Undtagelser:

- 1a. Afslutter use case.
- 2a. Afslutter use case.

1.2.2.11. UC10

Scenarie for use case 10 :	General Purpose Enhed
Navn:	Tænd enkelt
Use case ID:	10
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	GP-enheden
Initialisere:	"Tænd enkelt" signal modtages fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Den ønskede GP-enheden stikkontakt tændes
Reference:	Ingen

Hovedforløb:

1. GP-enheden modtager et "Tænd enkelt" signal fra scenariet kontrol
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.
2. GP-enheden tændes
 - a. Stikkontakt 1
 - b. Stikkontakt 2
 - c. Stikkontakt 3
 - d. Stikkontakt 4

Undtagelser:

- 1a. Afslutter use case.

1.2.2.12. UC11

Scenarie for use case 11 :	General Purpose Enhed
Navn:	Sluk enkelt
Use case ID:	11
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	GP-enheden
Initialisere:	"Sluk enkelt" signal modtages fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Den ønskede GP-enheden stikkontakt slukkes
Reference:	Ingen

Hovedforløb:

1. GP-enheden modtager et "Sluk enkelt" signal fra scenariet kontrol
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.
2. GP-enheden slukkes
 - a. Stikkontakt 1
 - b. Stikkontakt 2
 - c. Stikkontakt 3
 - d. Stikkontakt 4

Undtagelser:

- 1a. Afslutter use case.

1.2.2.13. UC12

Scenarie for use case 12 :	General Purpose Enhed
Navn:	Tænd alle
Use case ID:	12
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	GP-enheden
Initialisere:	"Tænd alle" signal modtages fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Alle stikkontakterne i GP-enheden tændes
Reference:	Ingen

Hovedforløb:

1. GP-enheden modtager et "Tænd alle" signal fra scenarietkontrol
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.
2. Alle stikkontakterne i GP-enheden tændes

Undtagelser:

- 1a. Afslutter use case.

1.2.2.14. UC13

Scenarie for use case 13 :	General Purpose Enhed
Navn:	Sluk alle
Use case ID:	13
Samtidige forekomster:	1
Primær aktør:	Scenarie kontrol
Sekundær aktør:	GP-enheden
Initialisere:	"Sluk alle" signal modtages fra scenarie kontrol
Forudsætninger:	Ingen
Resultat:	Alle stikkontakterne i GP-enheden slukkes
Reference:	Ingen

Hovedforløb:

1. GP-enheden modtager et "Sluk alle" signal fra scenarietkontrol
[Undtagelse 1a.] Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.
2. Alle stikkontakterne i GP-enheden slukkes

Undtagelser:

1a. Afslutter use case.

1.2.2.15. UC14

Scenarie for use case 14:	Ekstern
Navn:	Manuel styring
Use case ID:	14
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Lysdæmper, GP-enheden
Initialisere:	Bruger trykker på kontakten (switch)
Forudsætninger:	Validering fra DE2 kodelås
Resultat:	Brugeren skifter tilstand (tændt/slukket) på den ønskede genstand
Reference:	UC15 – Lås op

Hovedforløb:

1. Brugeren trykker på kontakten til den ønskede brugsgenstand.
 - a. Lysdæmper
 - b. GP-enheden
2. Den valgte brugsgenstand skifter tilstand (tændes/slukkes).

Undtagelser:

Ingen.

1.2.2.16. UC15

Scenarie for use case 15:	Kodelås
Navn:	Lås op
Use case ID:	15
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenarietkontrol
Initialisere:	Låse op for brugen af scenarietkontrol
Forudsætninger:	Kodelåsen er låst
Resultat:	Scenarietkontrol bliver låst op
Reference:	UC16 - Lås

Hovedforløb:

1. Brugeren indtaster kode 1 og trykker Enter. Koden valideres af kodelåsen.
[Undtagelse 1a.] Koden er forkert.
2. Brugeren indtaster kode 2 og trykker Enter. Koden valideres af kodelåsen.
[Undtagelse 2a.] Koden er forkert.
3. Brugeren indtaster kode 3 og trykker Enter. Koden valideres af kodelåsen.
[Undtagelse 3a.] Koden er forkert.
4. Kodelåsen sender et lavt signal til scenarietkontrol.

Undtagelser:

- 1a. Afvikler punkt 1 igen. Ved tre samlede forkerte forsøg, går kodelåsen i et permanent låst mode og skal resettes.
- 2a. Afvikler punkt 2 igen. Ved tre samlede forkerte forsøg går kodelåsen i et permanent låst mode og skal resettes.
- 3a. Afvikler punkt 3 igen. Ved tre samlede forkerte forsøg går kodelåsen i et permanent låst mode og skal resettes.

1.2.2.17. UC16

Scenarie for use case 16:	Kodelås
Navn:	Lås
Use case ID:	16
Samtidige forekomster:	1
Primær aktør:	Bruger
Sekundær aktør:	Kodelås, Scenariekontrol
Initialisere:	Låse for brugen af scenariekontrol
Forudsætninger:	Kodelåsen er låst op
Resultat:	Scenariekontrol bliver låst op
Reference:	UC15 – Lås op

Hovedforløb:

1. Brugeren trykker på Enter og kodelåsen sender et højt signal til scenariekontrol.

1.2.3. Ikke-funktionelle krav (ALLE)

1. Scenariet kontrol firmware skal kunne køre på en ATmega32 µController.
2. Outputtet fra modtagerenhederne har en max spænding på 12 V med en tolerance på $\pm 10\%$.
3. Lysdæmperen skal kunne dæmpe lyset fra max intensitet til slukket i 10 trin. De 10 trin skal regulere den leverede spænding i lige store trin, med samme tolerance som den generelle output tolerance (ikke funk. Krav 2).
4. GP-enheden leverer 0V eller max spænding afhængig af om den er slukket eller tændt.
5. Da strømforsyningerne i lab ikke kan levere mere en 2Ampere, skal hver af enhederne (Masterenhed, GP og lysdæmper) køre med en input spænding på 30V, og maksimalt trække 54 W (90% af max output effekten for forsyningerne givet ved $2A \cdot 30V$). Inden for disse rammer maksimeres output effekten for GP-enheden.

1.3. Accepttest (ALLE)

1.3.1. Accepttest for funktionelle krav (ALLE)

1.3.1.1. Test Firmware.

I accepttesten af UC 6 – 13 benyttes en firmware på modtagerenhederne designet til at teste modtagne signaler. Denne firmware benytter de samme drivere til at modtage signaler, som den færdige firmware kommer til at bruge, men implementerer ingen specifikke kommandoer for de forskellige enheder. I stedet benyttes en lysdiode til at indikere, om der succesfuldt blev modtaget en kommando, og i så fald hvilken. Dette tillader, at man kan teste sender-modtager systemet før modtagernes specifikke funktioner er implementeret, samt en mere specifik test af signal modtagelse, uafhængigt af resten af systemet. For aflæsning af lysdiodens signal benyttes Tabel 1. Ved modtagelse af et af de listede signaler i Tabel 1, blinker lysdioden først tre blink á 0,5sekund efterfulgt af en pause på tre sekunder, hvorefter lysdiodens antal blink aflæses og bestemmes til en kommando ud fra Tabel 1.

Kommando	Lysdiodens antal blink [stk.]	Tid pr. blink [sek.]
Korrekt kommando modtaget	3	0,5
Kommandoen er ikke en gyldig X.10 kommando eller er ikke implementeret i enheden.	1	10
UC6 "brighten"	1	2
UC7 "dim"	2	2
UC8 "on"	3	2
UC9 "off"	4	2
UC10 "Tænd enkelt"	5	2
UC11 "Sluk enkelt"	6	2
UC12 "Tænd alle"	7	2
UC13 "Sluk alle"	8	2

Tabel 1 - Afkodning af debug lysdiode

1.3.2. Accepttest for Use cases (ALLE)

1.3.2.1. Accepttest for UC 1

Use case under test		Use case 1 – Opret scenarie		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Validering fra DE2 kodelås		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren åbner konsollen, som starter hovedmenuen	Visuel test: Hovedmenuen kommer frem.		
2	Brugeren vælger "Opret scenarie" i hovedmenuen.	Visuel test: Opret scenarie menuen kommer frem.		
3	Kommandoer bliver indtastet – navn, beskrivelse.	Visuel test: Scenariet får et navn og en beskrivelse som kan aflæses på scenarielisten.		
3-1	Scenariekontrol værdier bliver indtastet og valideret.	Visuel test: Konsol bekræftelse og validering af værdier.		
4	Scenariet gemmes i scenarielisten.	Visuel test: Scenariet bliver gemt på næste frie plads på scenarielisten. Konsollen printer en bekræftelse. Evt. debugger.		
5	Hovedmenuen vises igen.	Visuel test: Hovedmenuen kommer frem igen.		
Undtagelse 3-1a.	Der udskrives en fejlbesked til brugeren. Gå til punkt 3-1.	Visuel test: Konsollen printer fejlmeddelelse og viser gyldige værdier jf. X10 protokollen. Afvikler punkt 3-1.		

1.3.2.2. Accepttest for UC 2

Use case under test		Use case 2 – Slet scenarie		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Validering fra DE2 kodelås; Der er oprettet et scenarie		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren åbner konsollen, som starter hovedmenuen	Visuel test: Hovedmenuen kommer frem.		
2	Brugeren vælger "Slet scenarie" i hovedmenuen.	Visuel test: Slet scenarie menuen kommer frem.		
3	Brugeren vælger hvilket af de oprettede scenarier der skal slettes.	Visuel test: Konsollen bekræfter scenarie valg, ved kald af navn og beskrivelse.		
4	Brugeren bliver spurgt om det er det korrekte scenarie der skal slettes.	Visuel test: Valideringstekst bliver vist - navn og beskrivelse vises fortsat.		
5	Det valgte scenarie bliver fjernet fra listen over mulige scenarier.	Visuel test: Konsol Bekræftelse/Print opdateret scenarieliste.		
6	Hovedmenuen vises igen.	Visuel test: Hovedmenuen kommer frem igen.		

1.3.2.3. Accepttest for UC 3

Use case under test		Use case 3 – Rediger scenarie		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Validering fra DE2 kodelås; Der skal mindst være oprettet et scenarie		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren åbner konsollen, som starter hovedmenuen	Visuel test: Hovedmenuen kommer frem.		
2	Brugeren vælger "rediger scenarie" i hovedmenuen.	Visuel test: "Rediger scenarie" menuen kommer frem. Scenarietkontrollisten bliver vist.		
3	Brugeren vælger hvilket scenarie der skal redigeres.	Visuel test: Konsollen bekræfter valg ved navn og beskrivelse af scenariet.		
4	Brugeren vælger hvilket element af scenariet der skal redigeres(navn, beskrivelse, enhedsværdier).	Visuel test: Scenariets nuværende element printes.		
4-1	Kommandoer bliver indtastet(navn, beskrivelse.)	Visuel test: Konsollen printer ændringerne i elementet.		
4-2	Enhedsværdier bliver indtastet og valideret.	Visuel test: Konsollen printer ændringerne i elementet.		
5	Redigeringerne bliver gemt i scenariet på Scenarietkontrol.	Visuel test: Samtlige elementer printes for det redigerede scenarie.		
6	Hovedmenuen vises igen.	Visuel test: Hovedmenuen kommer frem.		
Undtagelse 4-2a.	De indtastede enhedsværdier er ugyldige.	Visuel test: Konsollen printer fejlbesked. Afvikler punkt 4-2 på ny.		

1.3.2.4. Accepttest for UC 4

Use case under test		Use case 4 – Aktiver scenarie		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Validering fra DE2 kodelås		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren åbner konsollen, som starter hovedmenuen	Visuel test: Hovedmenuen kommer frem.		
2	Brugeren vælger "Aktiver scenarie" i hovedmenuen.	Visuel test "Aktiver scenarie" menuen kommer frem. Scenariekontrollisten bliver vist.		
3	Brugeren vælger hvilket scenarie der skal aktiveres.	Visuel test: Konsollen bekræfter valg ved navn og beskrivelse af scenariet.		
4	Terminalen spørger brugeren om bekræftelse af valg.	Visuel test: Bekræftelse af valg vises på skærmen.		
5	Brugeren bekræfter.	Visuel test: Konsollen printer en bekræftelsesbesked.		
6	Værdierne tilhørende det valgte scenarie sendes til Scenariekontrol.	Visuel test: Konsollen printer det aktiverede scenaries værdier. Evt. debugger.		
7	Hovedmenuen vises igen.	Visuel test: Hovedmenuen kommer frem.		
Undtagelse 5a.	Brugeren afviser.	Visuel test: Konsollen afvikler punkt 3.		

1.3.2.5. Accepttest for UC 5

Use case under test		Use case 5 – Deaktiver scenarie		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Validering fra DE2 kodelås; Der er et aktivt scenarie.		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren åbner konsollen, som starter hovedmenuen.	Visuel test: Hovedmenuen kommer frem.		
2	Brugeren vælger "Deaktiver scenarie" i hovedmenuen.	Visuel test: Konsollen printer bekræftelse af valg.		
3	Terminalen spørger brugeren om bekræftelse af valg.	Visuel test: Konsollen printer valideringsbesked.		
4	Brugeren bekræfter.	Visuel test: Konsollen printer en bekræftelsesbesked.		
5	Det aktuelle scenarie bliver deaktiveret.	Visuel test: Konsollen printer bekræftelsesbesked om at scenariet er blevet deaktiveret.		
6	Hovedmenuen vises igen.	Visuel test: Hovedmenuen kommer frem.		
Undtagelse 4a	Brugeren afviser.	Visuel test: Konsollen afvikler punkt 2.		

1.3.2.6. Accepttest for UC 6

Use case under test		Use case 6 – Dæmper op		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Ingen		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når lysdæmperen modtager signalet "brighthen" fra scenariekontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Glødepæren observeres når der sendes en "brighten" kommando.	Visuel test: Lysintensiteten går et trin op.	Lysintensiteten går et trin op.	OK
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
Undtagelse 2a	Glødepæren observeres når der sendes en "brighten" kommando. glødepæren er på max niveau.	Visuel test: Der sker ingen handling.	Der sker ingen handling.	OK

1.3.2.7. Accepttest for UC 7

Use case under test		Use case 7 - Dæmper ned		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Ingen		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med en debug lysdiode når lysdæmperen modtager signalet "dim" fra scenariekontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Glødepæren observeres når der sendes en "dim" kommando. Glødepæren er i forvejen tændt.	Visuel test: Lysintensiteten går et trin ned.	Lysintensiteten går et trin ned.	OK
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
Undtagelse 2a	Lysintensiteten sættes på laveste niveau, og glødepæren observeres når der sendes kommandoen "dim".	Visuel test: Lysintensiteten ændres ikke.	Lysintensiteten ændres ikke.	OK
Undtagelse 2b	Glødepæren slukkes, og lampen observeres når der sendes kommandoen "dim"	Visuel test: Lysintensiteten ændres ikke.	Lysintensiteten ændres ikke.	OK

1.3.2.8. Accepttest for UC 8

Use case under test		Use case 8 - Dæmper tænd		
Scenarie		Hovedforløb		
Forudsætninger		Ingen		
Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når lysdæmperen modtager signalet "on" fra scenariekontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Glødepæren observeres når der sendes en "tænd" kommando.	Visuel test: Lyset tændes med max intensitet.	Lyset tændes med max intensitet.	OK
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
Undtagelse 2a	Hvis lyset allerede har max intensitet.	Visuel test: Lysintensiteten ændres ikke.	Lysintensiteten ændres ikke.	OK

1.3.2.9. Accepttest for UC 9

Use case under test		Use case 9 – Dæmper sluk		
Scenarie		Hovedscenarie/undertagelse		
Forudsætninger		Ingen		
Step	Handling/Test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når lysdæmperen modtager signalet "off" fra scenariekontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Glødepæren observeres når der sendes en "off" kommando.	Visuel test: Glødepæren slukker.	Glødepæren slukker.	OK
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
Undtagelse 2a	Glødepæren observeres når der sendes en "off" kommando. Lampen er i forvejen slukket.	Visuel test: Ingen handling.	Ingen handling.	OK

1.3.2.10. Accepttest for UC 10

Use case under test		Use case 10 – Tænd enkelt		
Scenarie		Hovedforløb		
Forudsætninger		Ingen		
Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når GP-enheden modtager signalet "Tænd enkelt" fra scenariekontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Den ønskede stikkontakt tændes.	Måling: Der måles med multimeter om den ønskede stikkontakt er ON (max spænding [V]).		
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		

1.3.2.11. Accepttest for UC 11

Use case under test		Use case 11 - Sluk enkelt		
Scenarie		Hovedforløb		
Forudsætninger		Ingen		
Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når GP-enheden modtager signalet "sluk enkelt" fra scenarietkontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Den ønskede stikkontakt slukkes	Måling: Der måles med multimeter om den ønskede stikkontakt er OFF (0V).		
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		

1.3.2.12. Accepttest for UC 12

Use case under test		Use case 12 – Tænd alle		
Scenarie		Hovedforløb		
Forudsætninger		Ingen		
Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når GP-enheden modtager signalet "Tænd alle" fra scenarietkontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Alle stikkontakterne i GP-enheden tændes	Måling: Der måles med multimeter om alle stikkontakterne i GP-enheden er ON (max spænding [V])		
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		

1.3.2.13. Accepttest for UC 13

Use case under test		Use case 13 – Sluk alle		
Scenarie		Hovedforløb		
Forudsætninger				
Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Test firmwaren sættes til at blinke med debug lysdioden når GP-enheden modtager signalet "Sluk alle" fra scenarietkontrol.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		
2	Alle stikkontakterne i GP-enheden slukkes.	Måling: Der måles med multimeter om alle stikkontakterne i GP-enheden er OFF (0V).		
Undtagelse 1a	Test firmwaren får debug lysdioden til at vise fejlsignal, når kommandoen ikke er en gyldig X.10 kommando eller ikke er implementeret i enheden.	Visuel test: Debug lysdioden blinker som defineret i Tabel 1.		

1.3.2.14. Accepttest for UC 14

Use case under test		Use case 14 – Manuel styring		
Scenarie		Hovedscenarie		
Forudsætninger		Validering fra DE2 kodelås		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren trykker på kontakten til den ønskede brugsgenstand.	Visuel test: Kontakten til den ønskede brugsgenstand bliver aktiveret.		
2	Den valgte brugsgenstand skifter tilstand (tænd/sluk)	Måling: Der måles med multimeter, om den ønskede brugsgenstand skifter tilstand (0V og max spænding [V])		

1.3.2.15. Accepttest for UC 15

Use case under test		Use case 15 – Lås op		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Kodelåsen er låst		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Brugeren indtaster kode 1 og trykker Enter	Visuel test: Test dioderne på DE2-boardet skifter fra diode 1 til diode 2.		
2	Brugeren indtaster kode 1 og trykker Enter	Visuel test: Test dioderne på DE2-boardet skifter fra diode 2 til diode 3.		
3	Brugeren indtaster kode 1 og trykker Enter	Visuel test: Test dioderne på DE2-boardet skifter fra diode 3 til diode 4.		
4	Kodelåsen sender et lavt signal til scenariekontrol	Visuel test: Test diode 5 på DE2-boardet slukker. Visuel test: Hovedmenuen fremstår på PC'en		

1.3.2.16. Accepttest for UC 16

Use case under test		Use case 16 – Lås		
Scenarie		Hovedscenarie/undtagelse		
Forudsætninger		Kodelåsen er låst op		
Step	Handling/test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Kodelåsen er låst op og brugeren trykker på Enter.	Visuel test: Test diode 5 på DE2-boardet tænder. Visuel test: Hovedmenuen forsvinder fra PC'en		

1.3.3. Accepttest for ikke-funktionelle krav (ALLE)

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/kommentar
1	Scenariet kontrol firmware skal kunne køre på en ATmega32 µController.	Den færdige firmware compiles i Atmel studio, og konsol outputtet fra kompileringen studeres.	Visuel test: Det i outputtet specificerede programhukommelsesforbrug skal være under 32 KB, hvilket ses i Atmel Studio.		
2	Outputtet fra modtagerenhederne har en max spænding på 12 V med en tolerance på +-10%.	Der måles med et multimeter, om outputtene overholder den angivne spænding med dertilhørende tolerance	Måling: Outputs spænding overholder den angivne spænding inden for toleranceområdet.	Output spænding målt til: 11.7V.	Godkendt!
3	Lysdæmperen skal kunne dæmpe lyset fra slukket til max intensitet i 10 trin. De 10 trin skal regulere den outputtede spænding i lige store trin +-10%.	Når "brigthen" kommandoen er sendt 9 gange, kan glødepærens lysintensitet ikke blive mere intens. For hvert trin måles der med et multimeter, om outputtene overholder den angivne spænding med dertilhørende tolerance	Visuel test: Lysintensiteten bliver ikke højere efter trin 10. Måling: Alle lysdæmperens trin overholder den forudbestemte spænding inden for toleranceområdet.	Visuel test: Lysintensiteten bliver ikke højere efter trin 10. Måling: Der anvendes PWM strømstyring af lampen i stedet.	Godkendt! Delvist godkendt.
4	GP-enheden outputter 0V eller max spænding afhængig af om den er slukket eller tændt.	Der måles med et multimeter, om outputtene overholder den angivne spænding med dertilhørende tolerance	Måling: Outputs spænding overholder den angivne spænding inden for toleranceområdet.		

5	Hver af de tre enheder (Masterenhed, GP og lysdæmper) skal kunne køre med en maksimal input effekt på 54W hver ved 30V.	Der måles med multimeter den strøm der løber ind i enheden fra dens spændingskilde. Imens dette måles sættes den givne enhed til at køre sine forskellige funktioner.	Måling: Den største strømmåling for enheden benyttes sammen med den påtrykte spænding til at udregne effekten. Den udregnede effekt skal være mindre eller lig med den angivne max effekt.	Indgangsspænding: 30V Lampens maks. strøm: 0.8A $P_{\text{maks}}: 24W$	Godkendt.
---	---	---	--	--	-----------

2. Systemarkitektur (ALLE)

3. Strukturering (HW) (ALLE)

Som beskrevet i systembeskrivelsen, består systemet af tre separate moduler. For en oversigt over modulernes indbyrdes kommunikation henvises der til Figur 5, som viser forbindelserne mellem disse.

3.1. Masterenhed (ALLE)

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.

3.2. Lysdæmper (ALLE)

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.

3.3. GP-enhed (ALLE)

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.

3.4. Manuel overwrite (ALLE)

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.

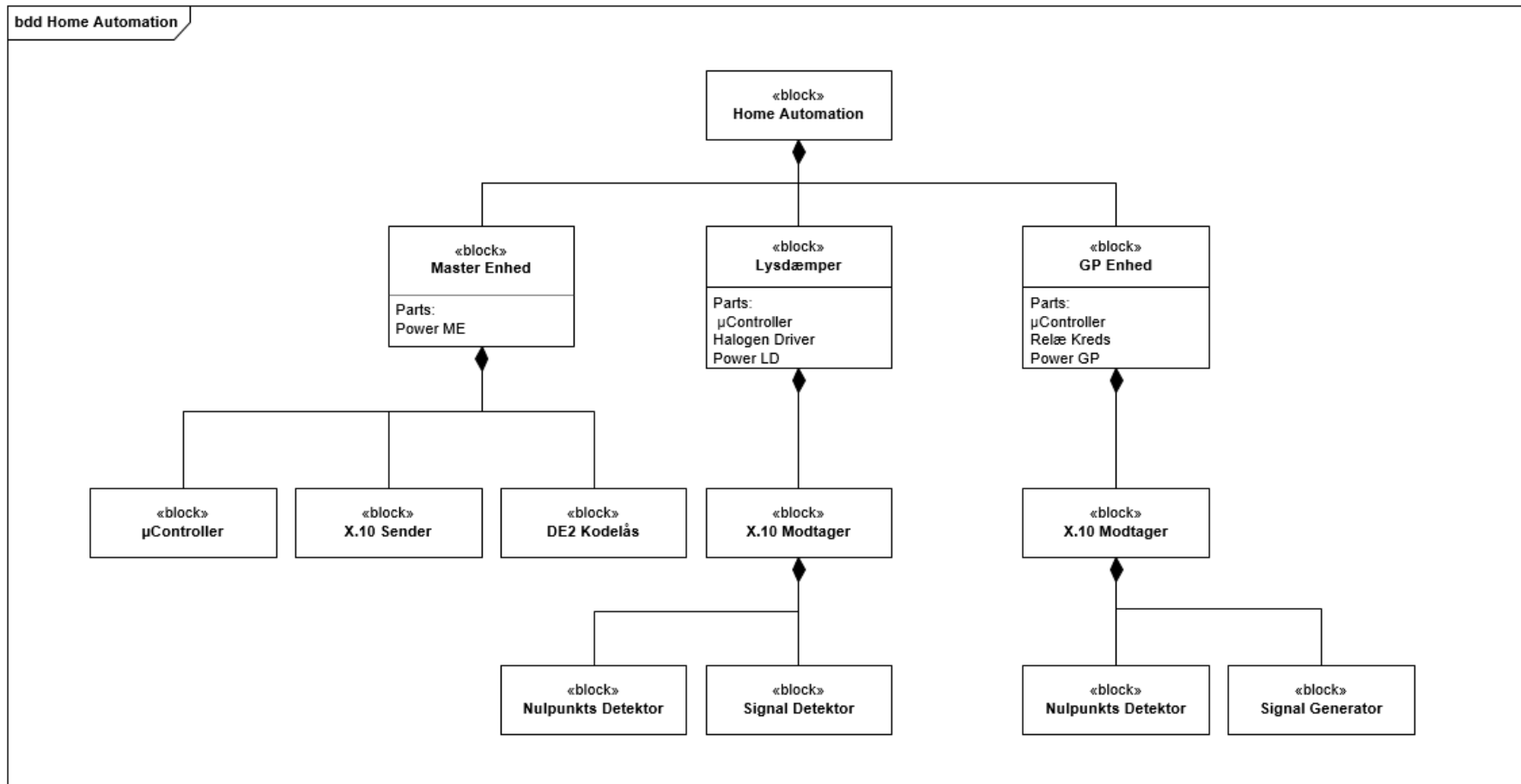
3.5. Signalbeskrivelse (ALLE)

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	Seriell 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	Seriell 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	Seriell 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 tryk	Manuel overwrite kontakt til lysdæmperen
kon_in1	Digitalt	0V eller 5V digitalt	5V= Aktiveret

		spænding	
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	

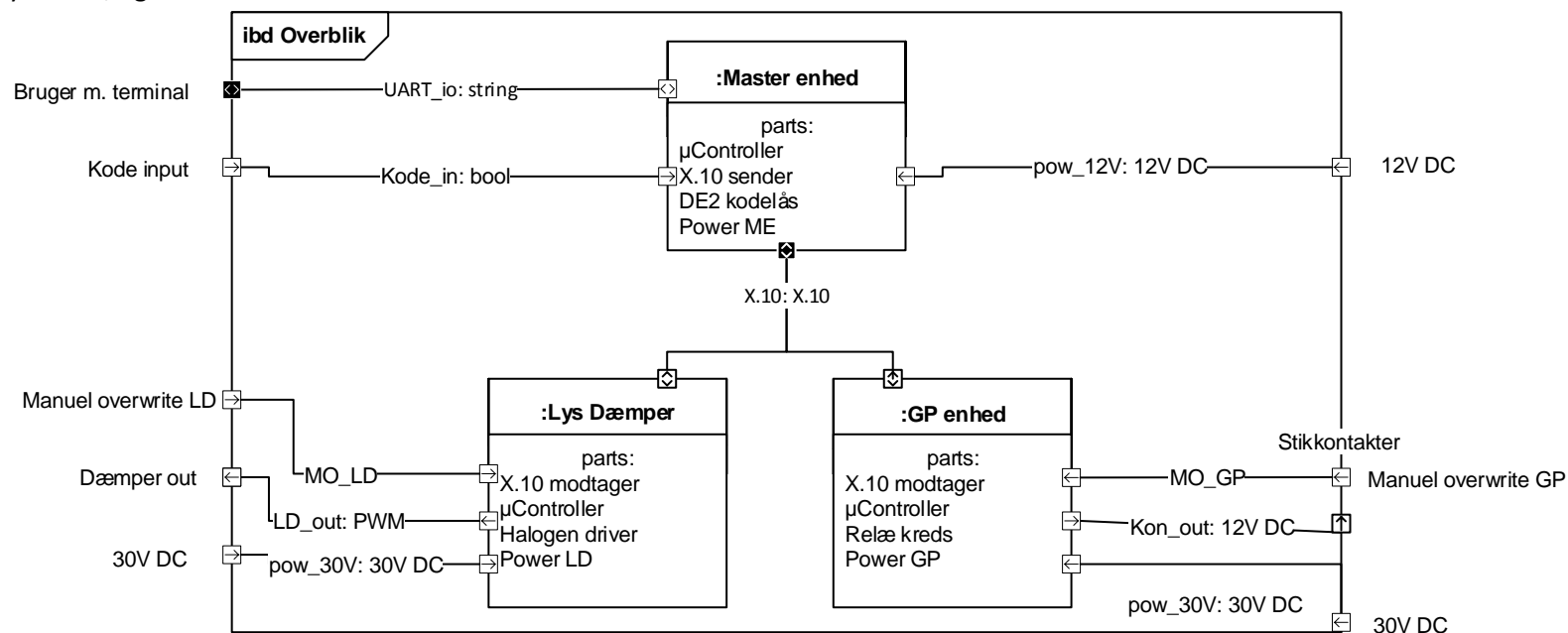
3.6. Overordnet BDD for Home Automation. (ALLE)



Figur 5 - Overordnet BDD diagram

3.7. IBD System Overblik (ALLE)

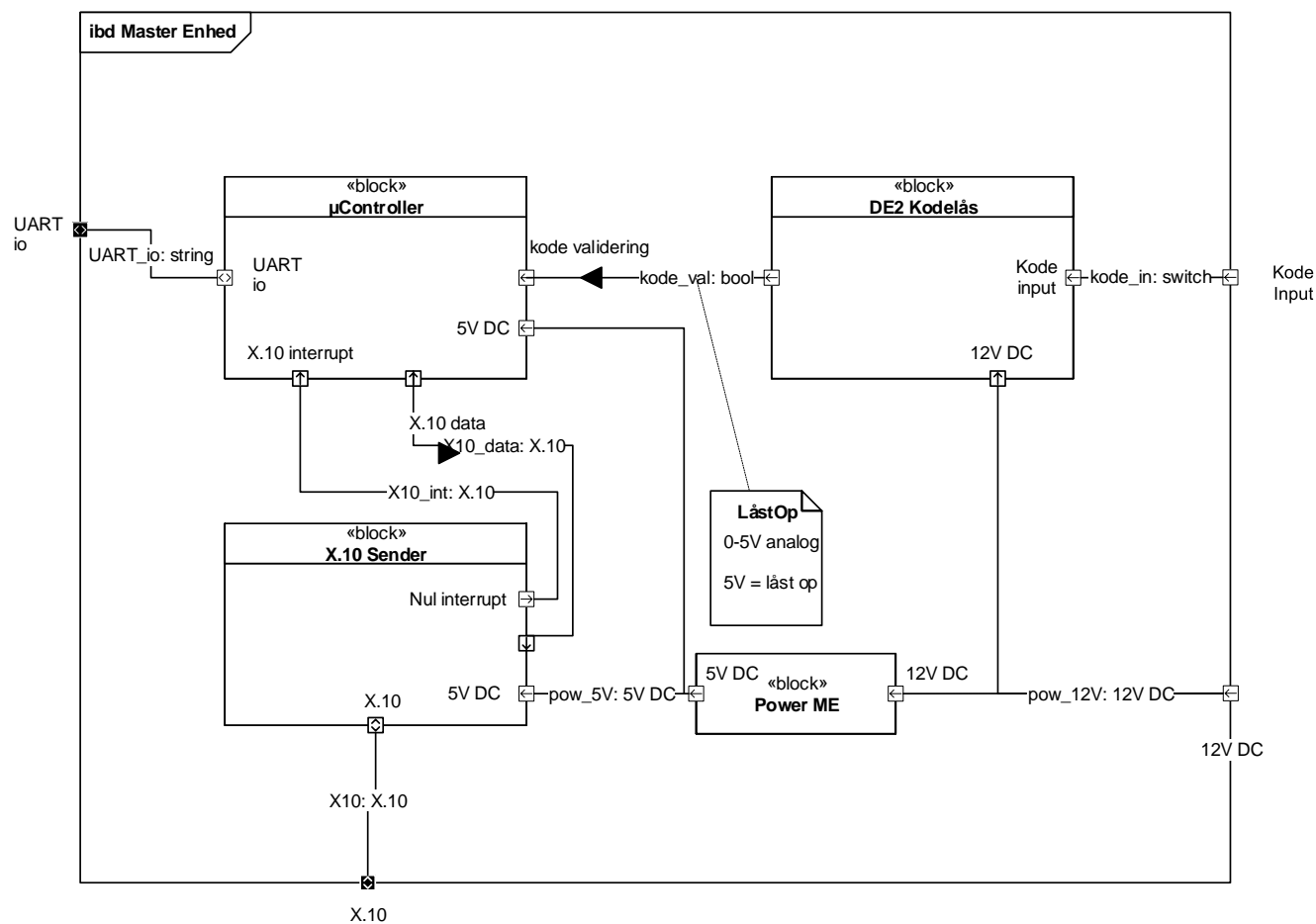
På Figur 6 vises home automation systemets overordnede IBD diagram. Diagrammet beskriver hvordan de overordnede blokke, er forbundet internt i systemet, og deres forbindelser til omverdenen.



Figur 6 - Overordnet IBD diagram

3.8. IBD Masterenhed (ALLE)

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



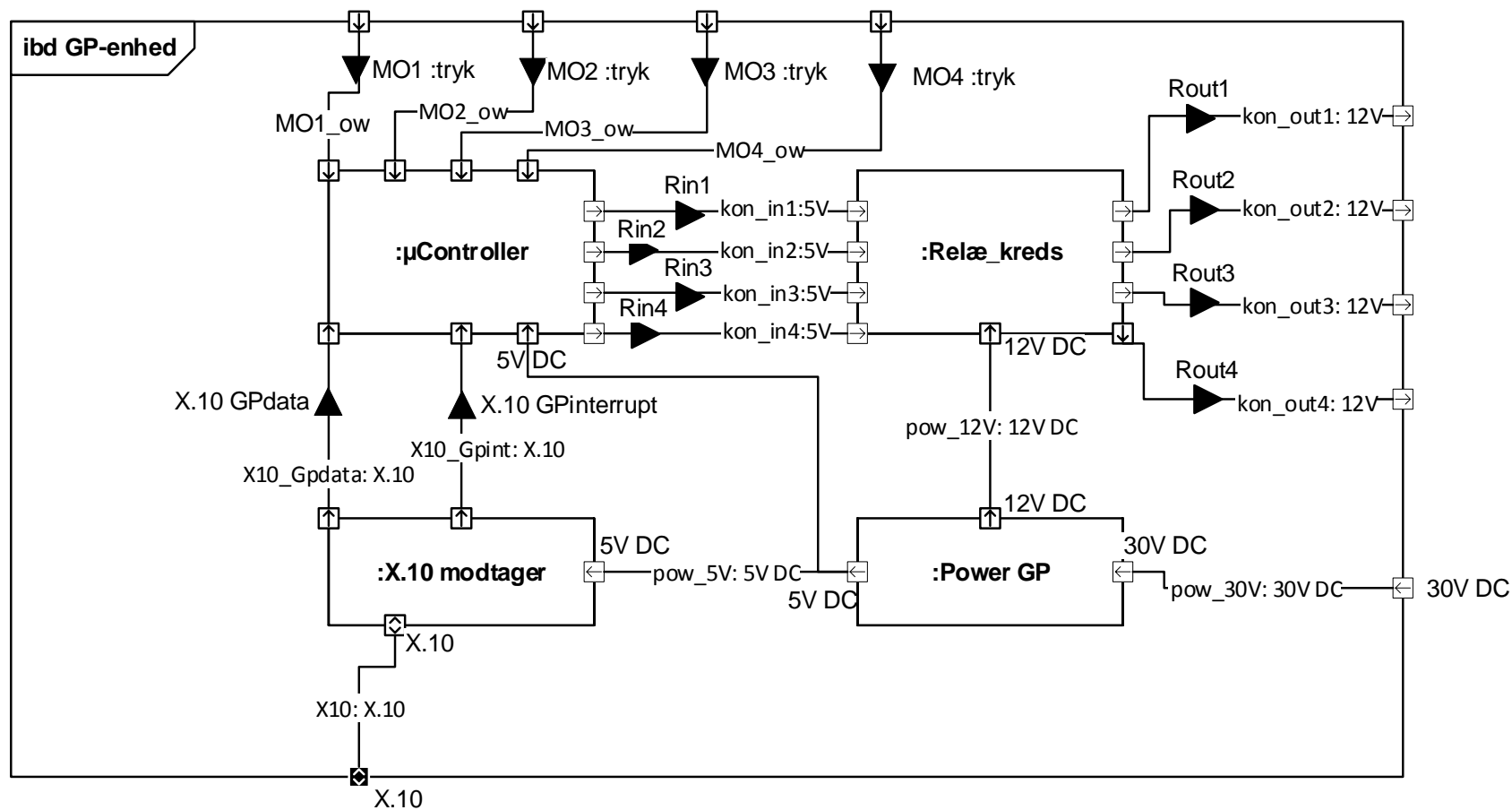
Figur 7 - IBD for Masterenheden

Figur 8 - IBD for X.10 senderen 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 μ Controlleren i masterenheden.

Figur 8 - IBD for X.10 senderen

3.10. IBD GP-Enhed (ALLE)

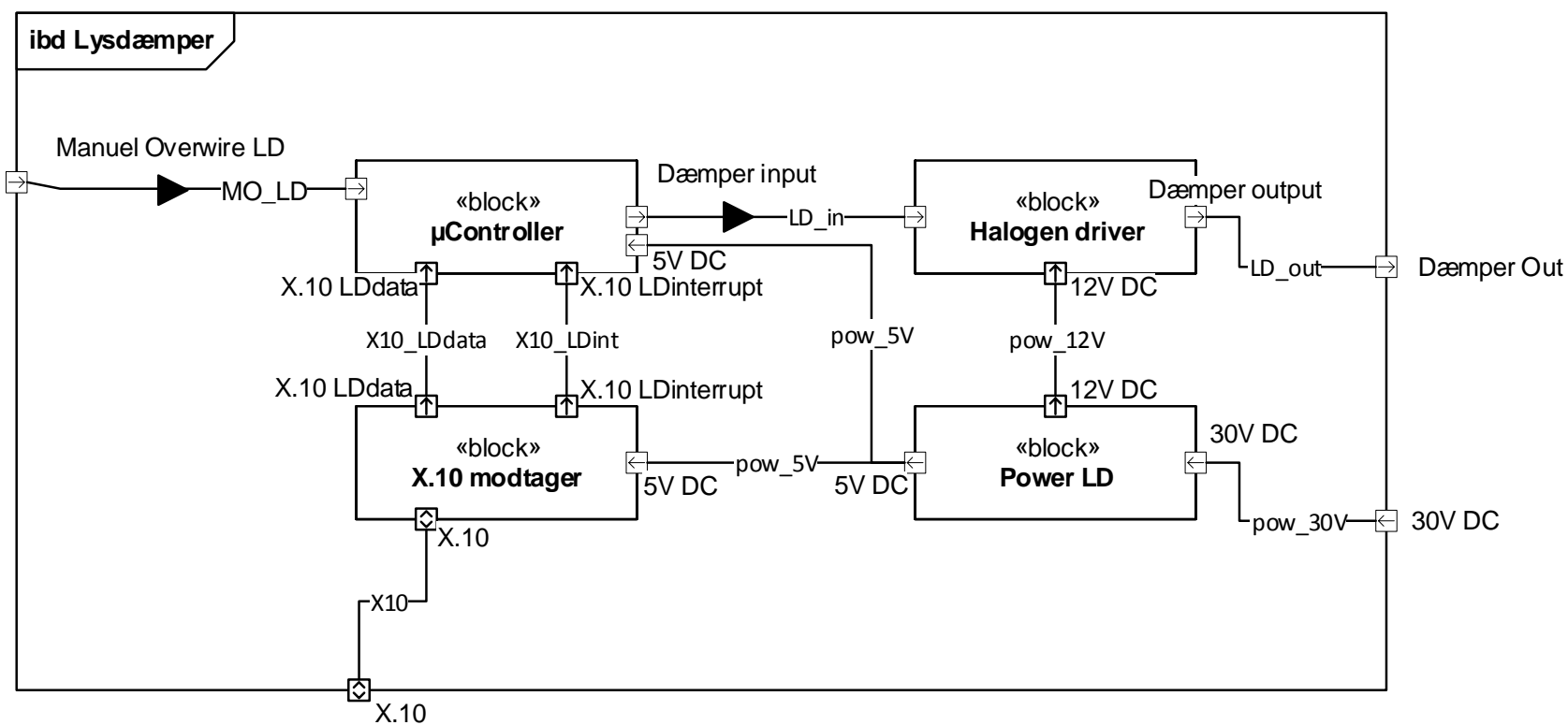
Figur 9 viser et IBD diagram over GP-enheden. Diagrammet fortæller hvilke forbindelser der optræder internt i blokken GP-enheden, og hvordan GP-enheden kommunikerer med x.10, stikkontakterne og trykknapperne fra manuel overwrite.



Figur 9 - IBD for GP-enheden

3.11. IBD Lyisdæmper (ALLE)

Figur 10 viser et IBD diagram over lyisdæmperen. Diagrammet beskriver hvilke forbindelser der forekommer i blokken lyisdæmper, og hvordan lyisdæmperen er forbundet med X.10 lysnettet, pæren og trykknapperne fra manuel overwrite.



Figur 10 - IBD for lyisdæmperen

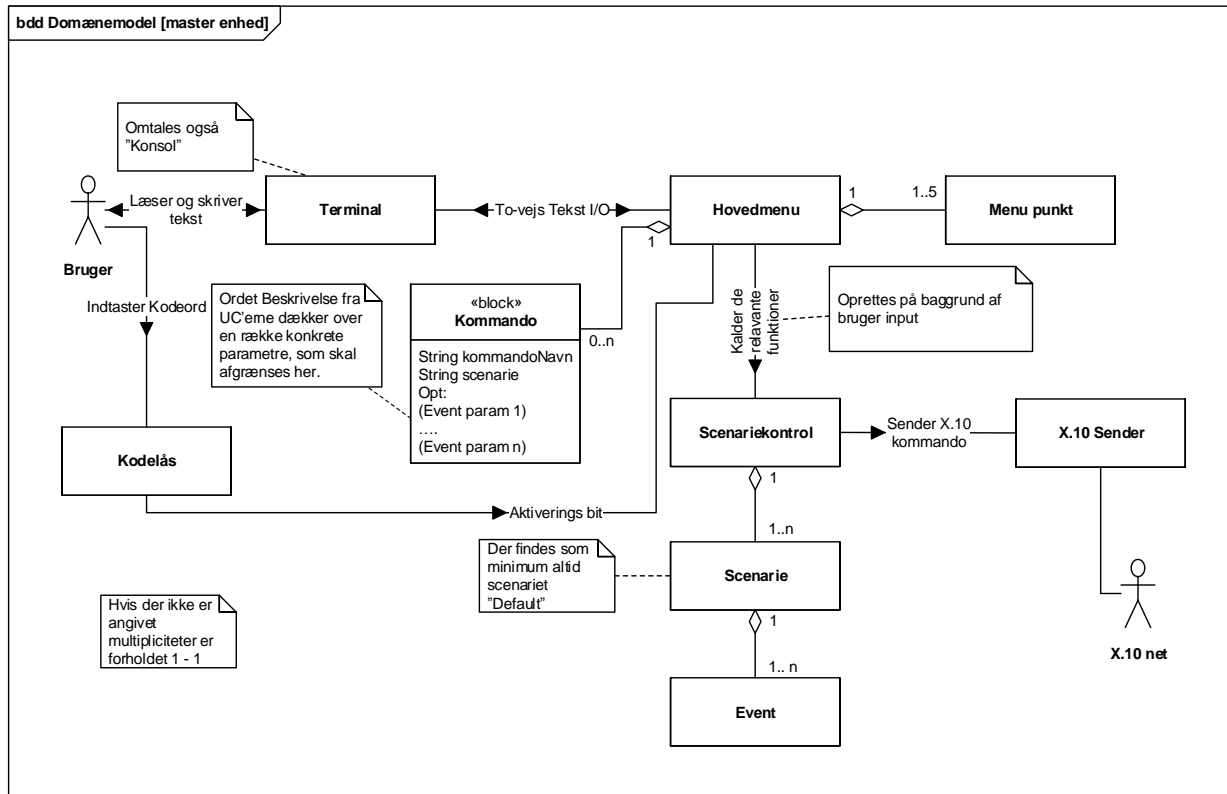
Figur 11 viser et IBD diagram over systemets X.10 modtager, som er placeret i GP-enheden og lysdæmperen. Diagrammet beskriver kommunikationsvejene i modtagerblokken, og modtagerens forbindelser ud til X.10 lysnettet og til μ Controller i henholdsvis GP-enheden og lysdæmperen.

Figur 11 - IBD for modtageren

4. Strukturering/design (SW)

4.1. Applikationsmodel for Masterenheden (ALLE)

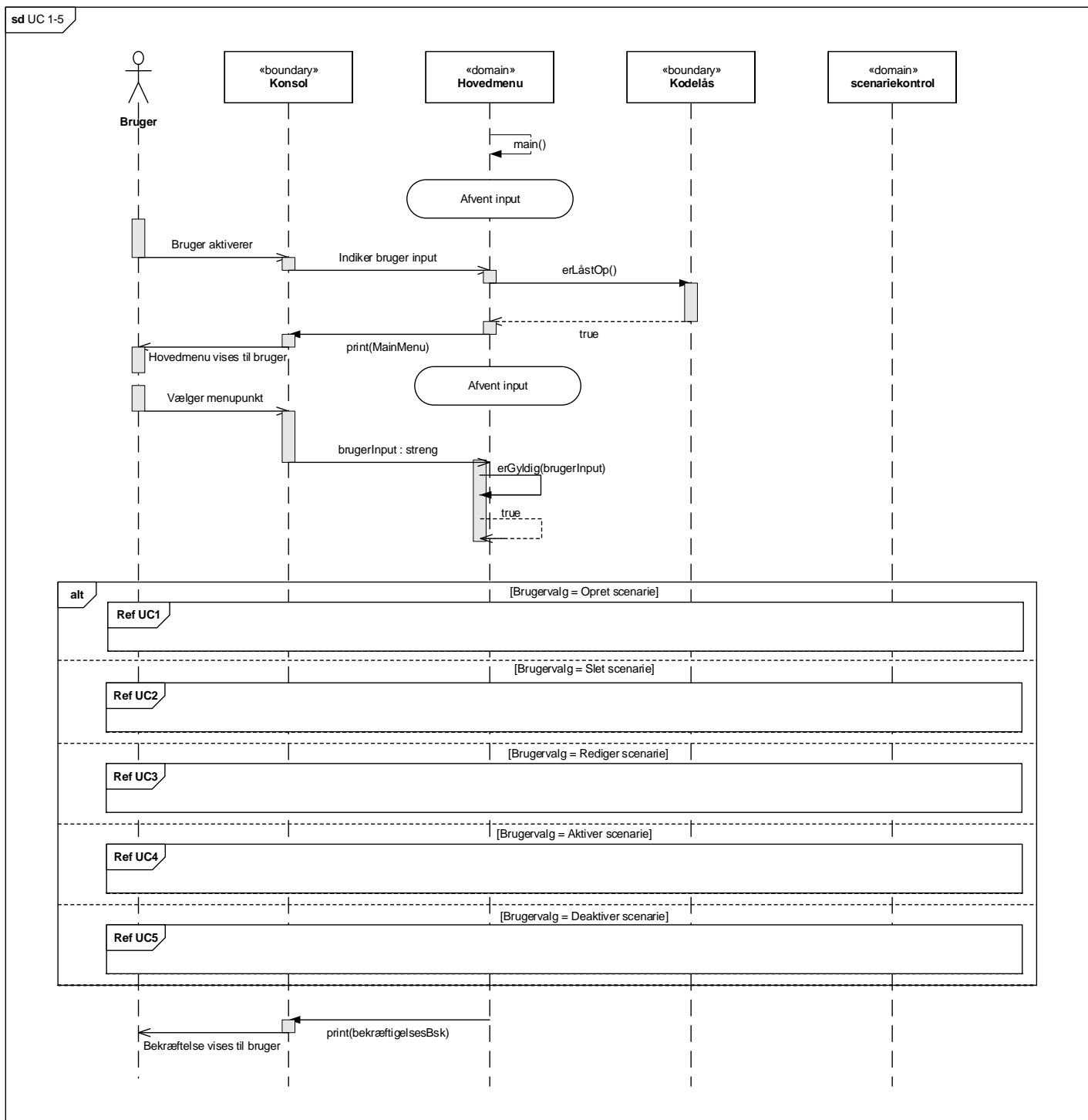
4.1.1. Domæne model (ALLE)



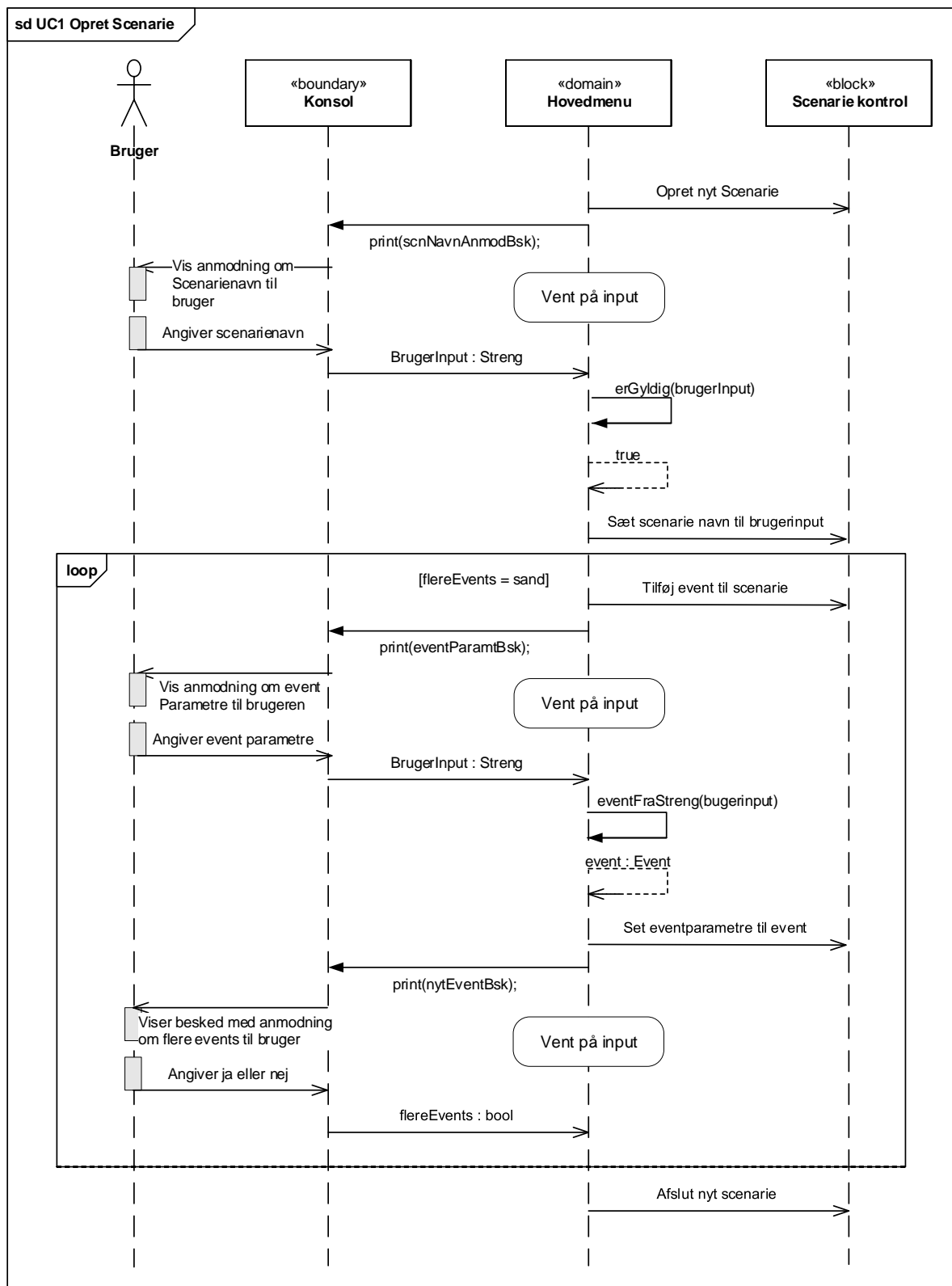
Figur 12 - Domænemodel for masterenheden

4.1.2. Sekvens- og state-machinediagrammer (ALLE)

Her er vist et sekvens diagram for det fælles forløb der starter UC 1-5, samt en udspecificering af UC 1. UC 2-5 er ikke vist i detaljer da UC 1 – 5 alle har lignende karakter.

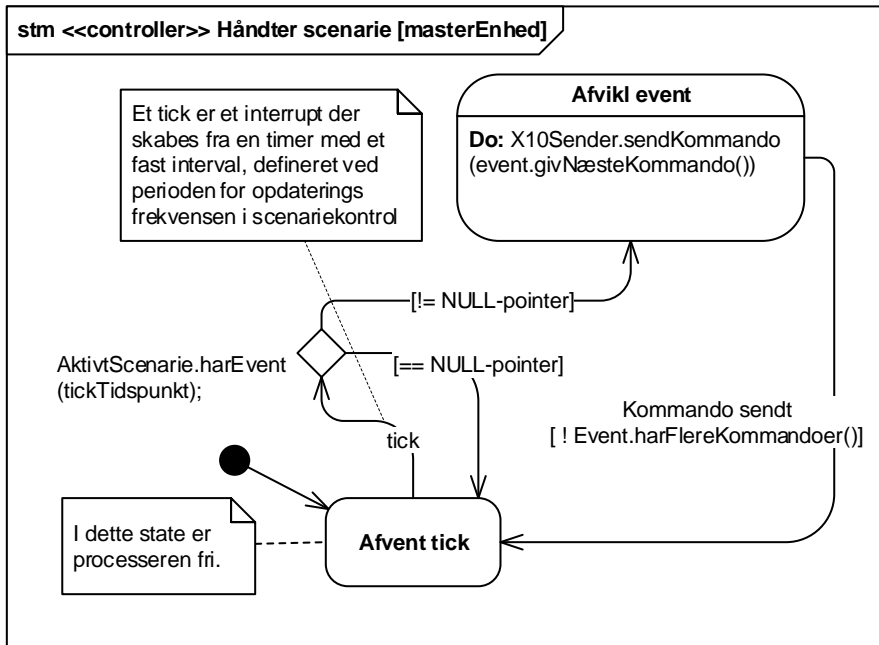


Figur 13 - sd for indledende forløb af UC 1-5



Figur 14- sd for UC1 som den forløber efter der refereres fra Figur 13

Dette statemachine diagram beskriver funktionaliteten af den interrupt rutine i scenariekontrol som står for at eksekverer de relevante events i det aktive scenarie. Dette knytter sig ikke til nogen specifik UC.

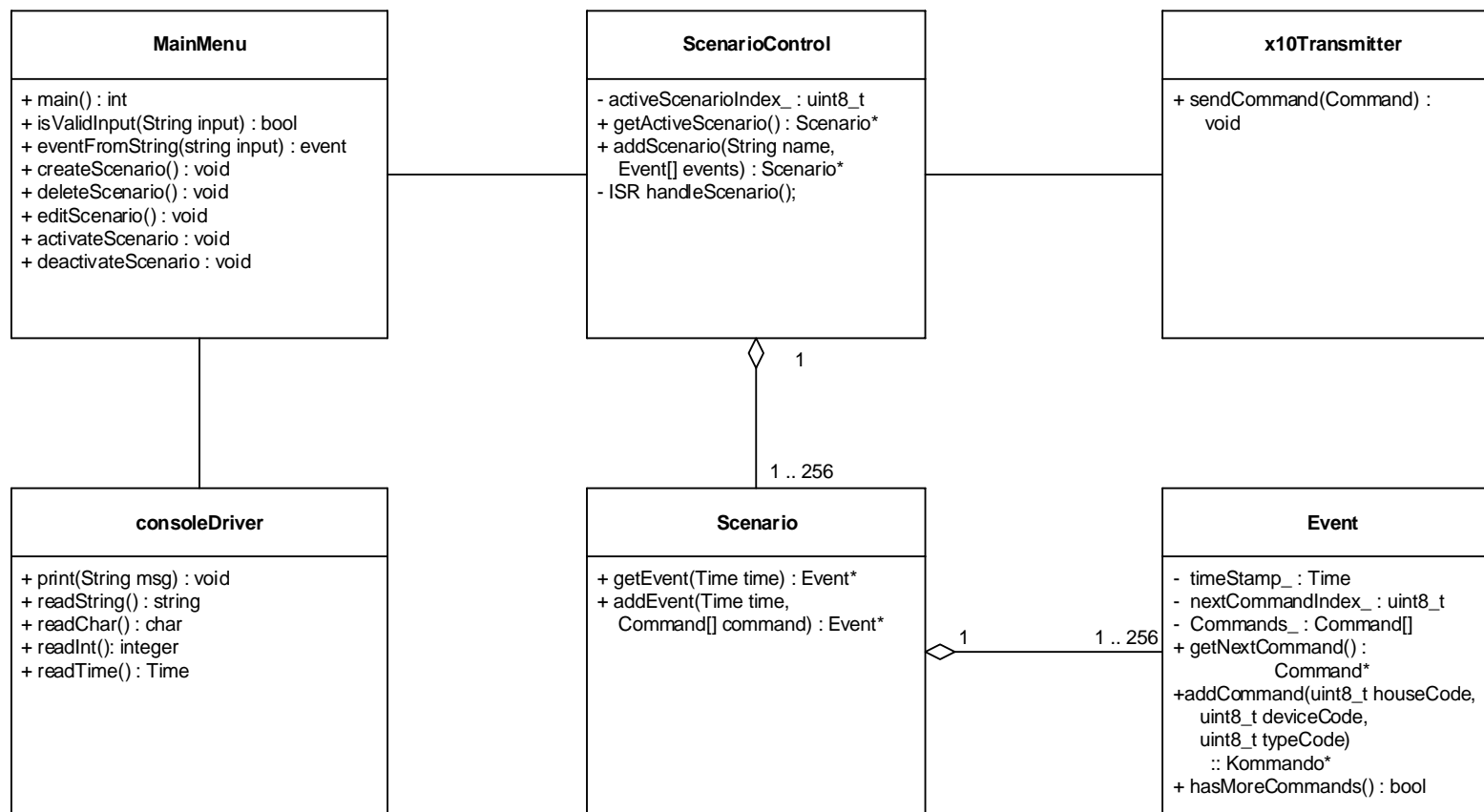


Figur 15 - stm for håndtering af aktivt scenarie.

4.1.3. Klassediagram (ALLE)

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).

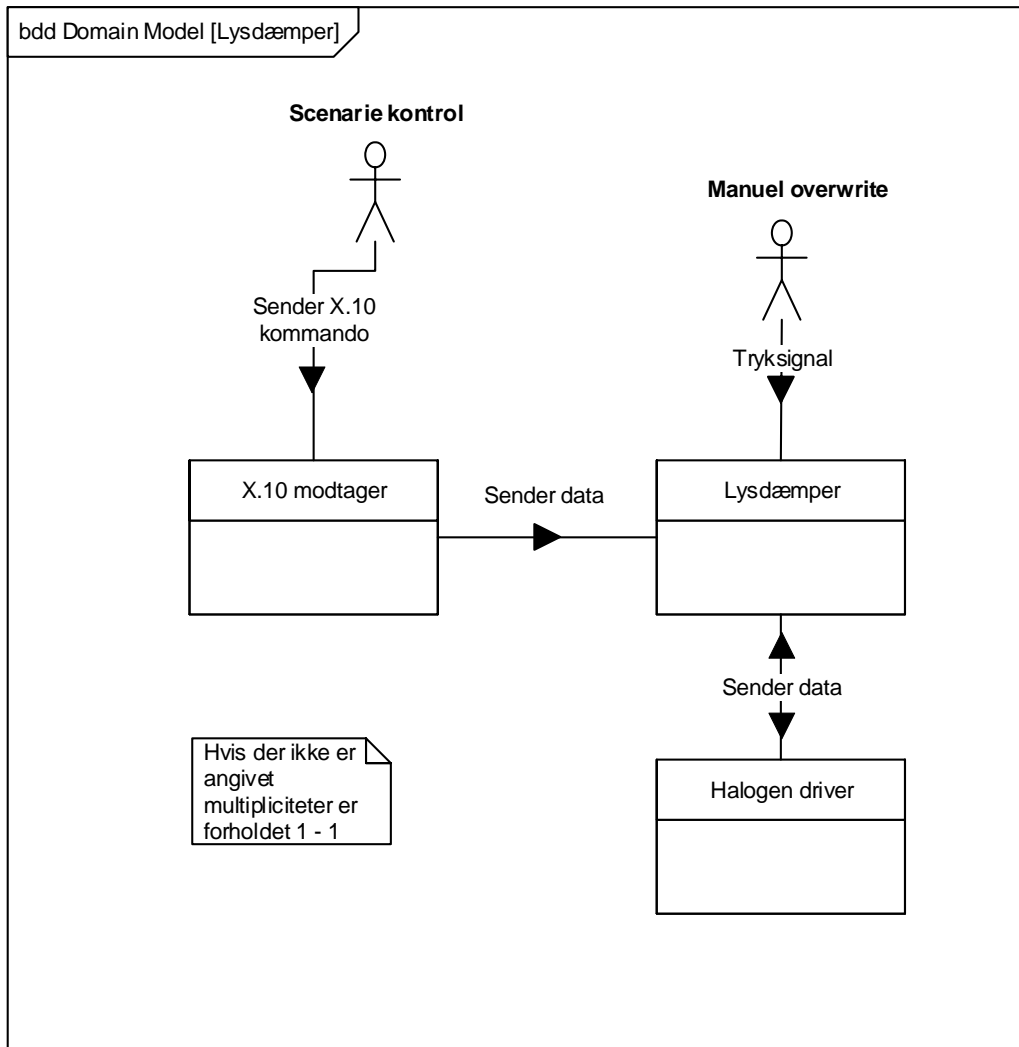
Class Diagram [Master Unit]



Figur 16 - UML klassediagram for masterenheden

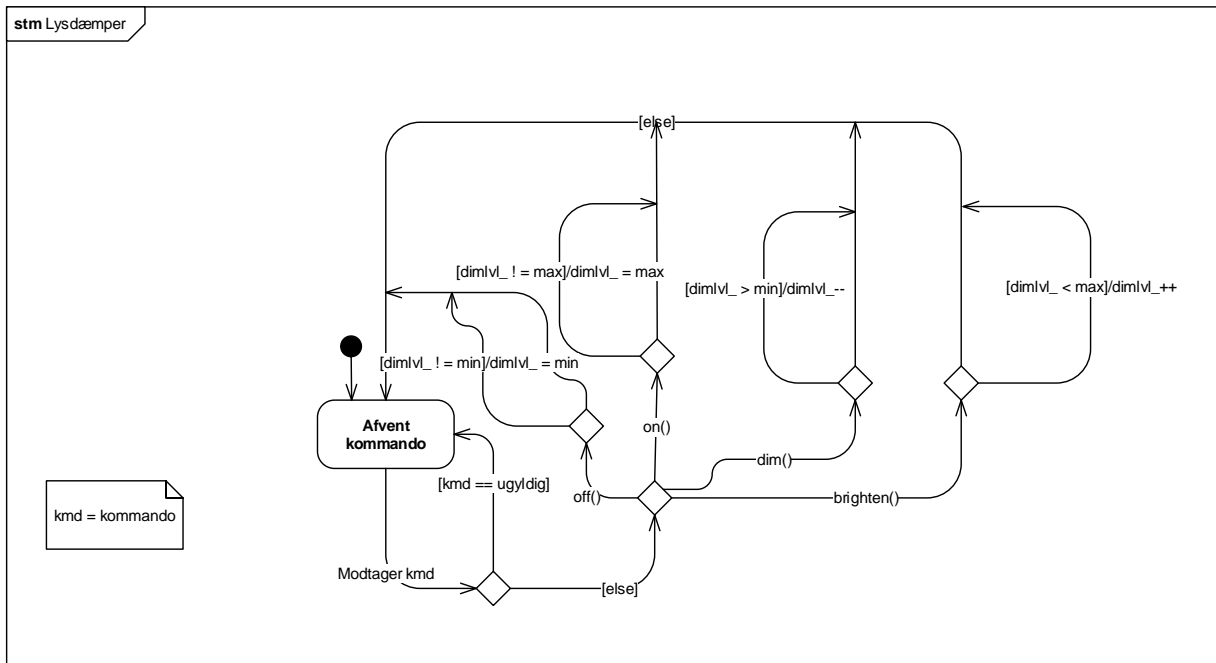
4.2. Applikationsmodel for Lysdæmper (ALLE)

4.2.1. Domæne model (ALLE)



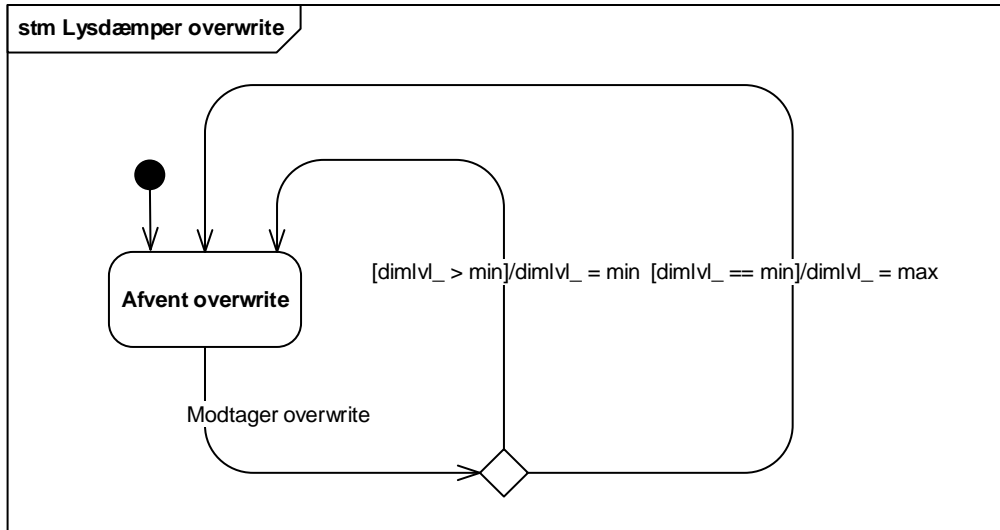
Figur 17 Domæne model for lysdæmper

4.2.2. State machine diagram (ALLE)



Figur 18 STM diagram for lysdæmper. Dimlvl er lysdæmperens nuværende lysintensitet.

I "Afvent kommando" staten bliver funktionen "checkForCommand()" kørt i loop.

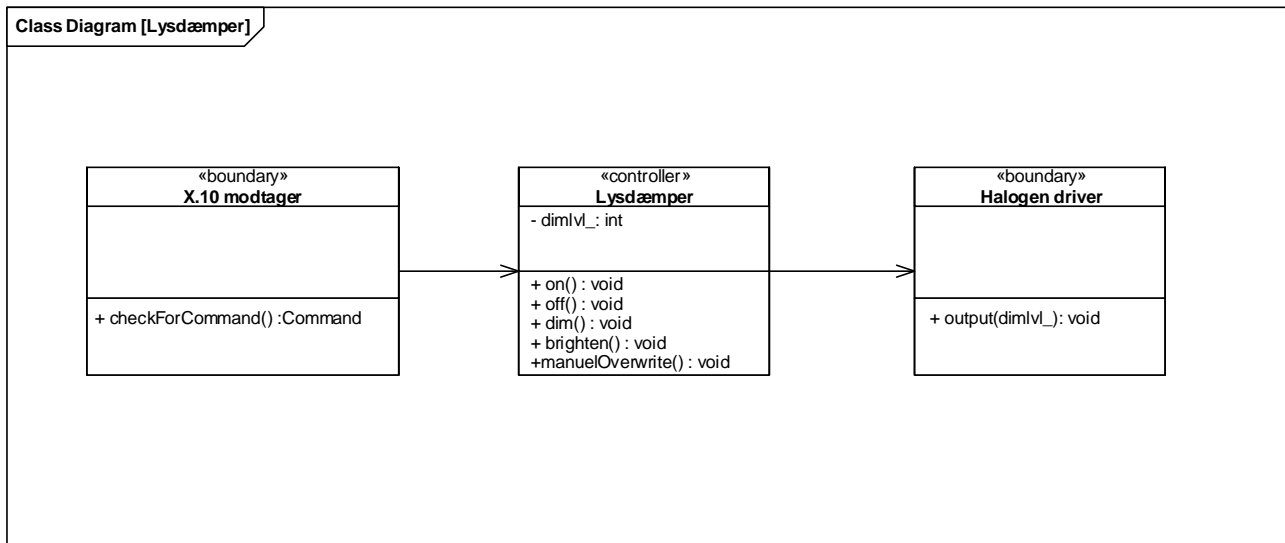


Figur 19 STM diagram for lysdæmper overwrite.

Overwritefunktionen består af en fysisk knap som slukker lampen hvis den er tændt, og tænder hvis den er slukket.

4.2.3. Klassediagram (ALLE)

Metoderne er ændret til engelsk for at der ikke bliver skabt forvirring ift. metodenavnene. Hermed er IHA kodestandarden for C/C++ overholdt.



Figur 20 Klassediagram for lysdæmper

"checkForCommand()" tjekker om der er modtaget en kommando.

"dimlvl_" beskriver lysintensiteten i 10 trin fra 0-9.

"output(dimlvl_)" er Halogen driverens output til lampen.

"manuelOverwrite()" er funktionen der bliver kaldt, hvis der bliver trykket på knappen.

4.3. Header filer lysdæmper (ALLE)

På Figur 21 ses header filerne til lysdæmperen, som er lavet ud fra klasse diagrammet på Figur 21.

```
#pragma once
using namespace std;

class Relay_Circuit
{
public:
    void singleOn(int);
    void singleOff(int);
    void allOn();
    void alloff();
private:
    int stikNr_;
};

#pragma once
#include <iostream>
using namespace std;
const int MAX = 9;
const int MIN = 0;

class Lysdæmper
{
public:
    Lysdæmper( int dimlvl = 0);
    void on();
    void off();
    void dim();
    void brighten();
    void manuelOverwrite();
    void print_dimlvl() const;
    int getdimlvl() const;

private:
    int dimlvl_;
};

#pragma once
using namespace std;

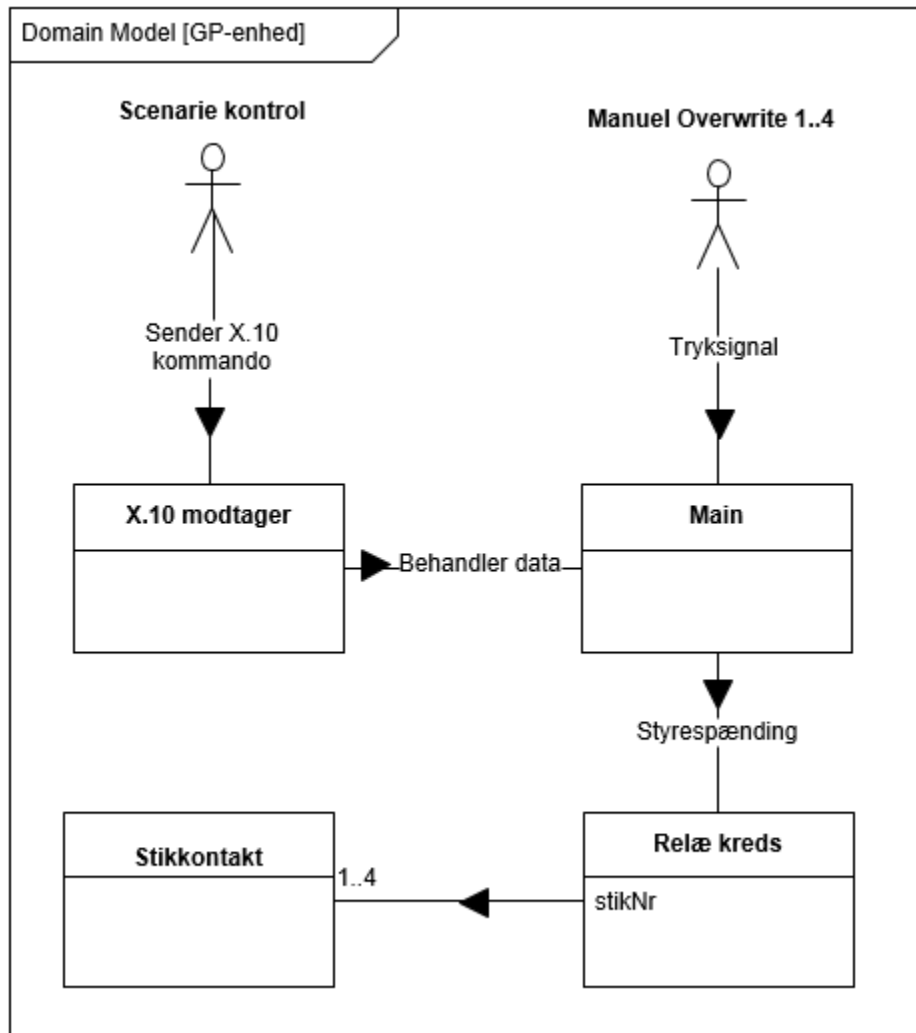
class X10_Receiver
{
public:
    Command chechForCommand();
};
```

Figur 21 - Header filerne til lysdæmperen

4.4. Applikationsmodel for GP-Enhed (ALLE)

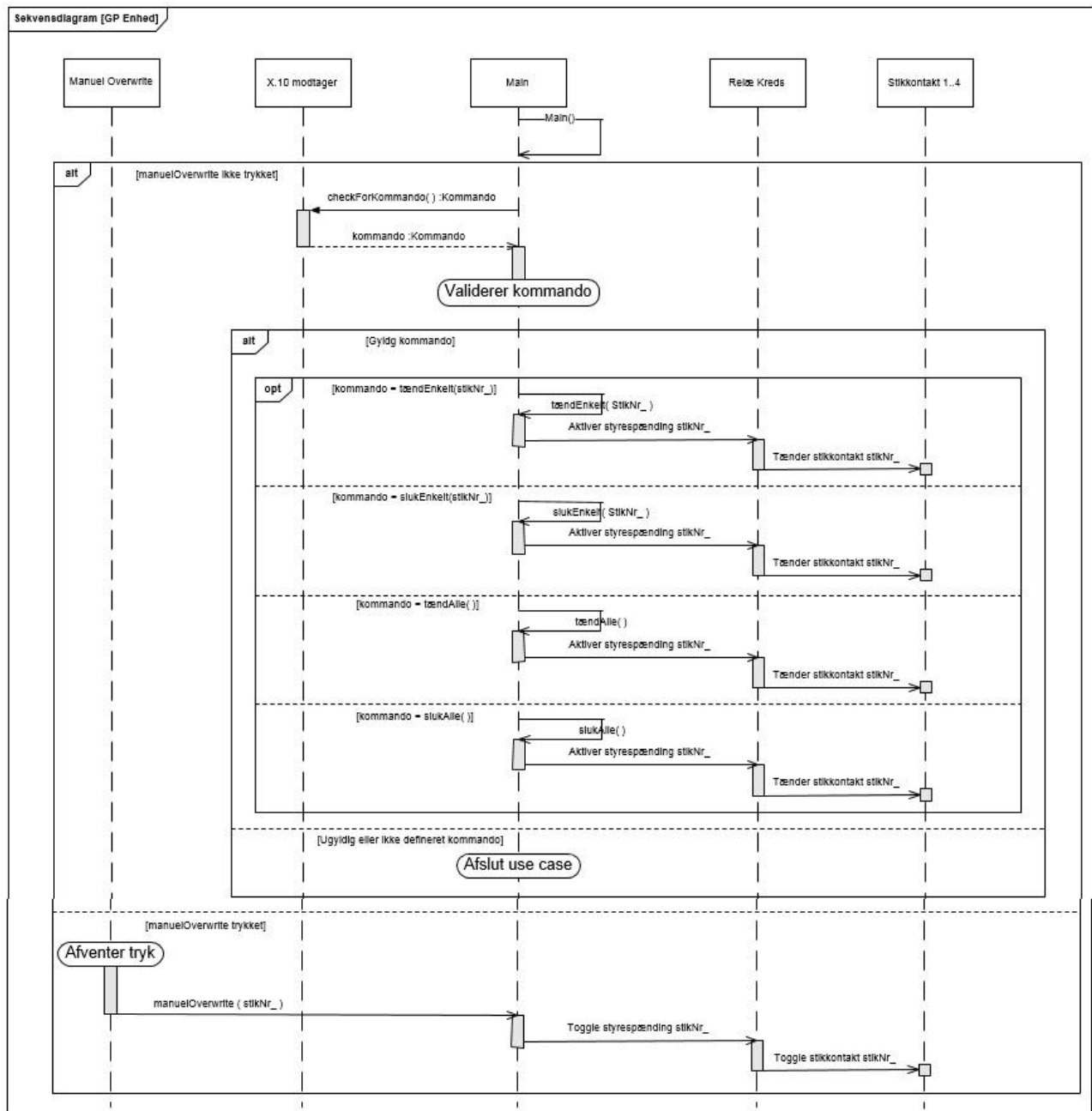
På Figur 22 ses domæne modellen for GP-enheden. "Main" er alle vores fire use cases, der omhandler GP-enheden (UC10-14).

4.4.1. Domæne model (ALLE)



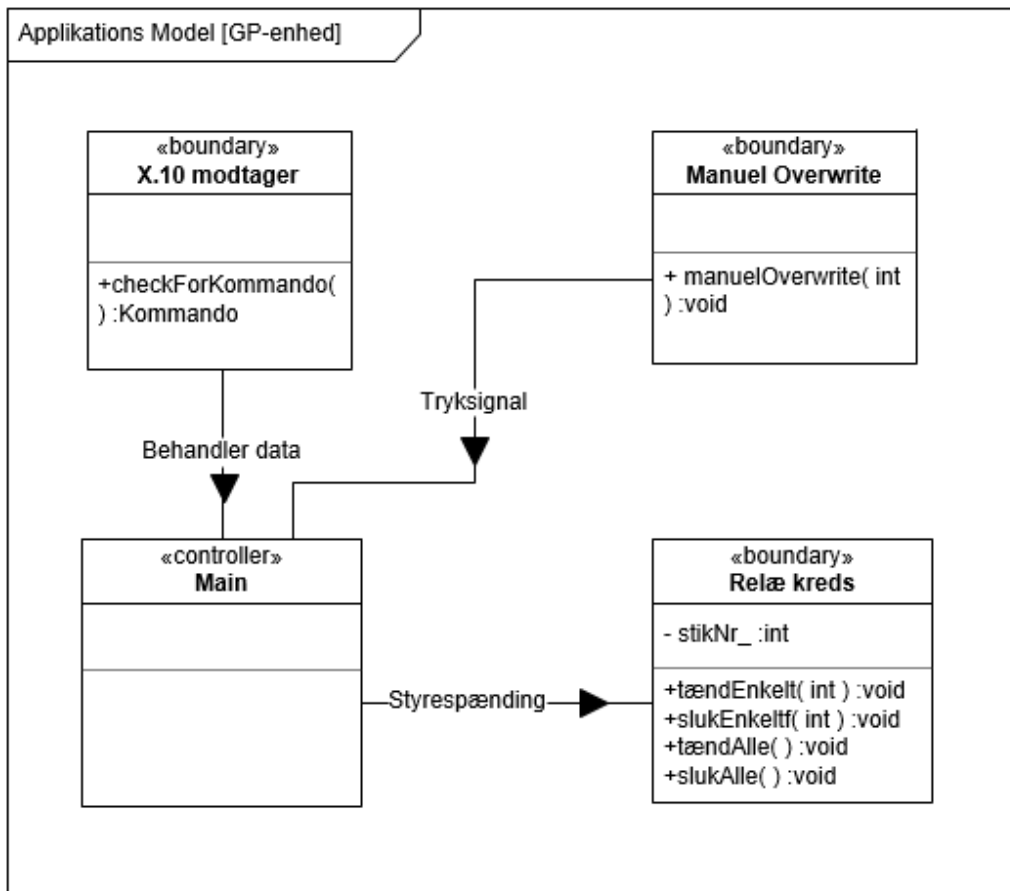
Figur 22 - Domæne model for GP-enheden

4.4.2. Sekvens diagram (ALLE)



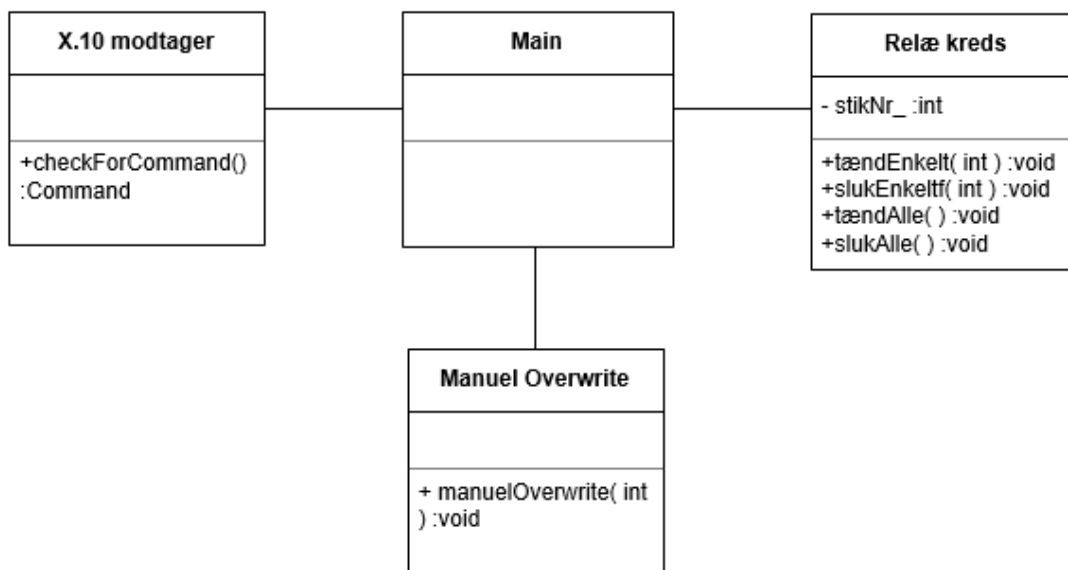
Figur 23 - Sekvensdiagram for GP-enheden

4.4.3. Klasse applikations model (ALLE)



Figur 24 - Klasse applikationsmodellen for GP-enheden

4.4.4. Klasse diagram (ALLE)



Figur 25 - Det færdige klasse diagram for GP-enheden

4.5. Header filer GP-enhed (ALLE)

På Figur 26 ses header filerne til GP-enheden, som er lavet ud fra klasse diagrammet på Figur 25.

```
#pragma once
using namespace std;

class Relay_Circuit
{
public:
    void singleOn(int);
    void singleOff(int);
    void allOn();
    void alloff();
private:
    int stikNr_;
};

#pragma once
using namespace std;

class Manuel_Overwrite_GP
{
public:
    void manuelOverwrite(int);
};

#pragma once
using namespace std;

class X10_Receiver
{
public:
    Command checkForCommand();
};
```

Figur 26 - Header file til GP-enhed

5. Design/Implementering (HW)

5.1. X10 sender og modtager (Felix)

Da hardwaren til håndtering af X.10 kommunikation er ens for alle sendere og modtagere, er designet af dette her beskrevet selvstændigt. Følgende afsnit gælder altså for alle tre enheder.

Ved designet af X.10 hardwaren blev designet i AN236 først gransket for at se om en modificeret udgave af dette kunne være egnet. De steder hvor design elementer er taget fra denne note er dette angivet.

5.1.1. Zero-cross detektor (Felix)

Ved design af zero-cross detektoren blev den simple løsning fra AN236 først taget i betragtning. Her benyttes en modstand til strømbegrænsning, og de interne spændingsbegrænser dioder kunne således benyttes som spændingsklammer. På denne måde kan µcontrollerens eksterne interrupt benyttes direkte på signalet når den er konfigureret til "any logical change". Dette giver dog det problem at der ved en rising edge genereres et interrupt når spændingen når V_{HL} som er ca. 4.2 V. Når signalet er 230Vrms, som i noten, er dette ikke noget problem, idet 4.2V indtræffer meget kort tid efter det egentlige 0 punkt:

$$f := x \rightarrow 230 \cdot \text{sqrt}(2) \cdot \sin(50 \cdot 2 \pi x);$$

$$\text{solve}(f(x) = 4.2, x);$$

$$x \rightarrow 230 \sqrt{2} \sin(100 \pi x)$$

$$0.00004110253926$$

I vores projekt arbejder vi til gengæld på 18Vrms i stedet, og tiden bliver derfor væsentligt større, og de detekterede nulpunkter bliver stærkt forskudte:

$f := x \rightarrow 18 \cdot \text{sqrt}(2) \cdot \sin(50 \cdot 2 \text{ Pi } x);$

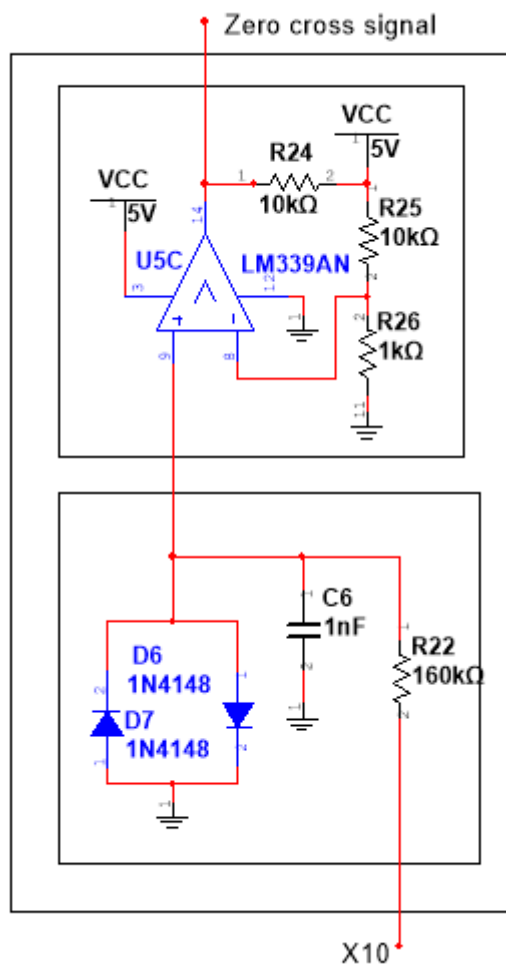
$$x \rightarrow 18 \sqrt{2} \sin(100 \pi x)$$

$\text{solve}(f(x) = 4.2, x);$

0.0005275969704

Da der ifølge protokollen skal være en 120kHz puls, som er et ms langt startende ved nulkrydsningen, er dette en alt for stor fejl.

Ud fra denne betragtning blev der derfor i stedet designet følgende kreds:



Her benyttes først et lavpasfilter (C6 og R22) til at fjerne 120kHz komponenten i signalet, samtidig med at de to dioder D6 og D7 benyttes til at begrænse spændingen. Det resulterende signal ud af den første kasse, er 50Hz komponenten, med alle spændinger udenfor ± 0.7 V området skåret af.

Herefter benyttes en LM339 komparator til at sammenligne dette signal med en referencespænding på 0,5 V. Når inputtet her er lavere end referencen trækker IC'en outputtet til 0V, og når det er højere lader den outputtet flyde, og pull-up modstanden R24 holder her signalet ved 5V. Det resulterende signal bliver et 0-5V signal der skifter logisk niveau hver gang 50Hz komponenten krydser 0,5 V, hvilket giver en udmærket tilnærmelse af nulkrydsningen:

$$f := x \rightarrow 18 \cdot \text{sqrt}(2) \cdot \sin(50 \cdot 2 \cdot \text{Pi} \cdot x);$$

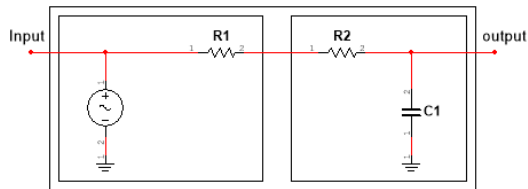
$$\text{solve}(f(x) = 0.5, x);$$



$$x \rightarrow 230 \sqrt{2} \sin(100 \pi x)$$

$$0.000004893025384$$

Til dimensioneringen af lavpasfilteret blev der udført følgende beregninger:



Overføringsfunktion og cutoff frekvens:

$$T_v = \frac{\frac{1}{C1 \cdot s}}{R1 + R2 + \frac{1}{C1 \cdot s}} = \frac{1}{(R1 + R2) \cdot C1 \cdot s + 1} = \frac{\frac{1}{C1 \cdot (R1 + R2)}}{s + \frac{1}{C1 \cdot (R1 + R2)}};$$

$$T_v := \frac{\frac{1}{C1 \cdot (R1 + R2)}}{s + \frac{1}{C1 \cdot (R1 + R2)}};$$

$$\omega_c := \frac{1}{C1 \cdot (R1 + R2)};$$

Dimensionering:

$$R1 := 5.71;$$

$$C1 := 1 \cdot 10^{-9};$$

$$\text{solve}(\omega_c = 1000 \cdot 2 \cdot \text{Pi}, R2);$$

$$1.591492331 \cdot 10^5$$

(3.1)

$$R2 := (3.1);$$

Individuel forsøgning for de to signal komponenter:

$$T_v;$$

$$\frac{6283.185307}{s + 6283.185307}$$

(3.2)

$$T_{vs} := s \rightarrow (3.2);$$

$$\text{Gain}_{125\text{kHz}} = \text{evalf}(20 \cdot \log_{10}(\text{abs}(T_{vs}(1 \cdot 120000 \cdot 2 \cdot \text{Pi}))));$$

$$\text{Gain}_{125\text{kHz}} = -41.58392650$$

(3.3)

$$\text{Gain}_{50\text{Hz}} = \text{evalf}(20 \cdot \log_{10}(\text{abs}(T_{vs}(1 \cdot 50 \cdot 2 \cdot \text{Pi}))));$$

$$\text{Gain}_{50\text{Hz}} = -0.01084381099$$

(3.4)

lækspænding fra 125 kHz signalet:

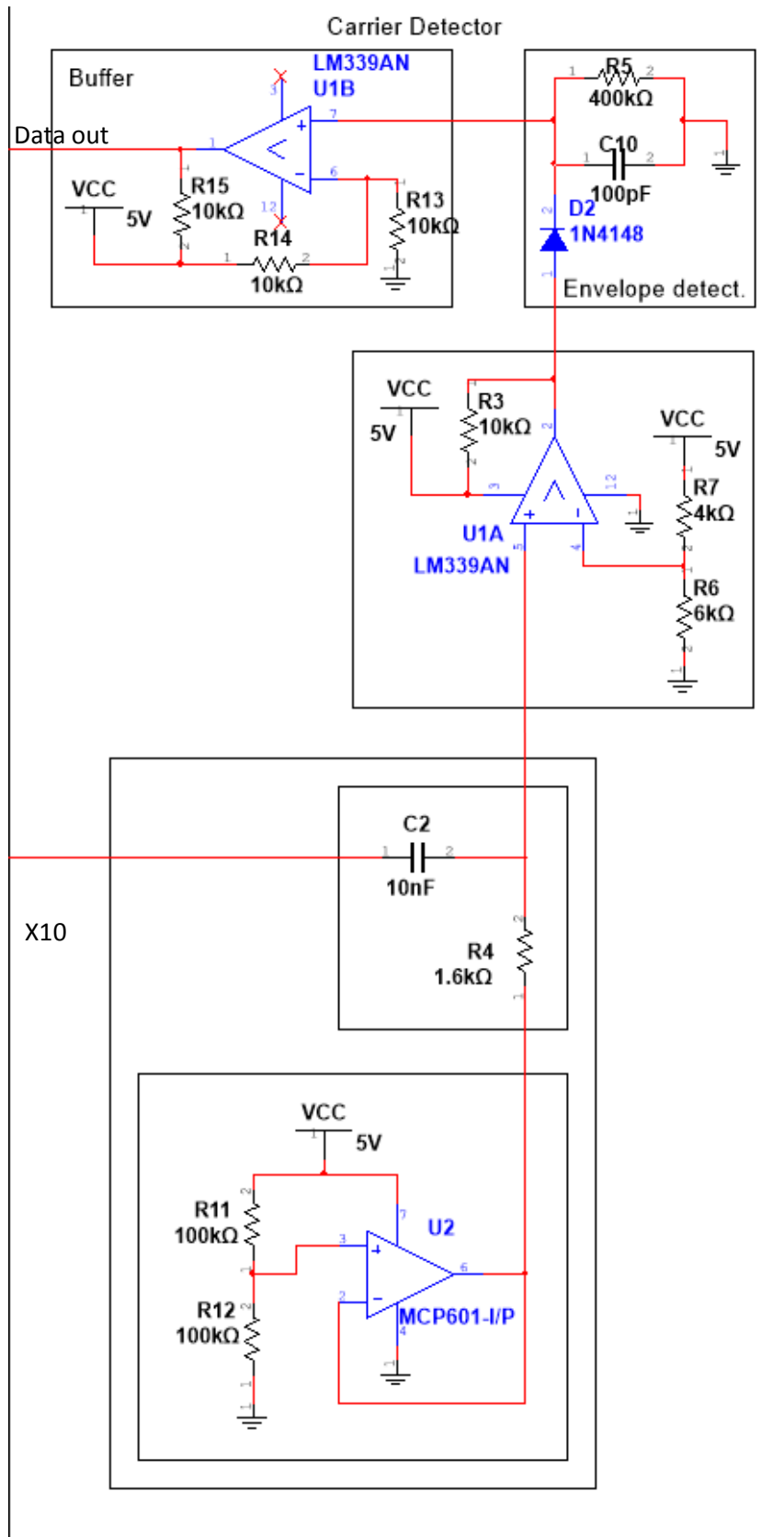
$$V_{125\text{kHz}} = \text{evalf}\left(5 \cdot 10^{\frac{\text{rhs}((3.3))}{20}}\right);$$

$$V_{125\text{kHz}} = 0.04166522000$$

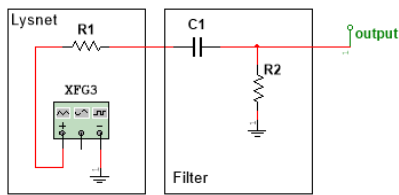
(3.5)

5.1.2. Modtager (Felix)

Til X10 modtageren er følgende kreds designet fra bunden:



Her benyttes der først et højpasfilter til at isolere 120kHz komponenten. Filteret består af R4 og C2 og er dimensioneret ud fra følgende udregninger:



Overføringsfunktion:

$$T_v = \frac{R_2}{\frac{1}{C_1 \cdot s} + R_1 + R_2} = \frac{s \cdot R_2}{s \cdot (R_1 + R_2) + \frac{1}{C_1}} = \frac{\frac{s \cdot R_2}{R_1 + R_2}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} = \frac{1}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} \cdot \frac{s \cdot R_2}{R_1 + R_2}$$

$$= \frac{\frac{1}{C_1 \cdot (R_1 + R_2)}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} \cdot \frac{s \cdot R_2 \cdot C_1 \cdot (R_1 + R_2)}{R_1 + R_2} = \frac{\frac{1}{C_1 \cdot (R_1 + R_2)}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} \cdot s \cdot R_2 \cdot C_1 = \frac{\frac{1}{C_1 \cdot (R_1 + R_2)}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} \cdot \frac{s}{\frac{1}{R_2 \cdot C_1}} \quad (1.1)$$

$$T_v := \frac{\frac{1}{C_1 \cdot (R_1 + R_2)}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}} \cdot \frac{s}{\frac{1}{R_2 \cdot C_1}}$$

$$T_{v1} := \frac{\frac{1}{C_1 \cdot (R_1 + R_2)}}{s + \frac{1}{C_1 \cdot (R_1 + R_2)}}$$

$$T_{v2} := \frac{s}{\frac{1}{R_2 \cdot C_1}}$$

Cutoff frekvenserne for de to led:

$$\omega_1 := \frac{1}{C_1 \cdot (R_1 + R_2)}$$

$$\omega_2 := \frac{1}{R_2 \cdot C_1}$$

Dimensionering:

$$R_1 := 5.71$$

$$C_1 := 10 \cdot 10^{-9}$$

$$\text{solve}(\omega_1 = 10000 \cdot 2 \cdot \pi, R_2);$$

$$1585.839431$$

(1.2)

$$R_2 := (1.2)$$

Reelle cutoff frekvenser:

$$\omega_{1-Hz} = \text{evalf}\left(\frac{\omega_1}{2 \cdot \pi}\right);$$

$$\omega_{1-Hz} = 9999.999999$$

(1.3)

$$\omega_{2-Hz} = \text{evalf}\left(\frac{\omega_2}{2 \cdot \pi}\right);$$

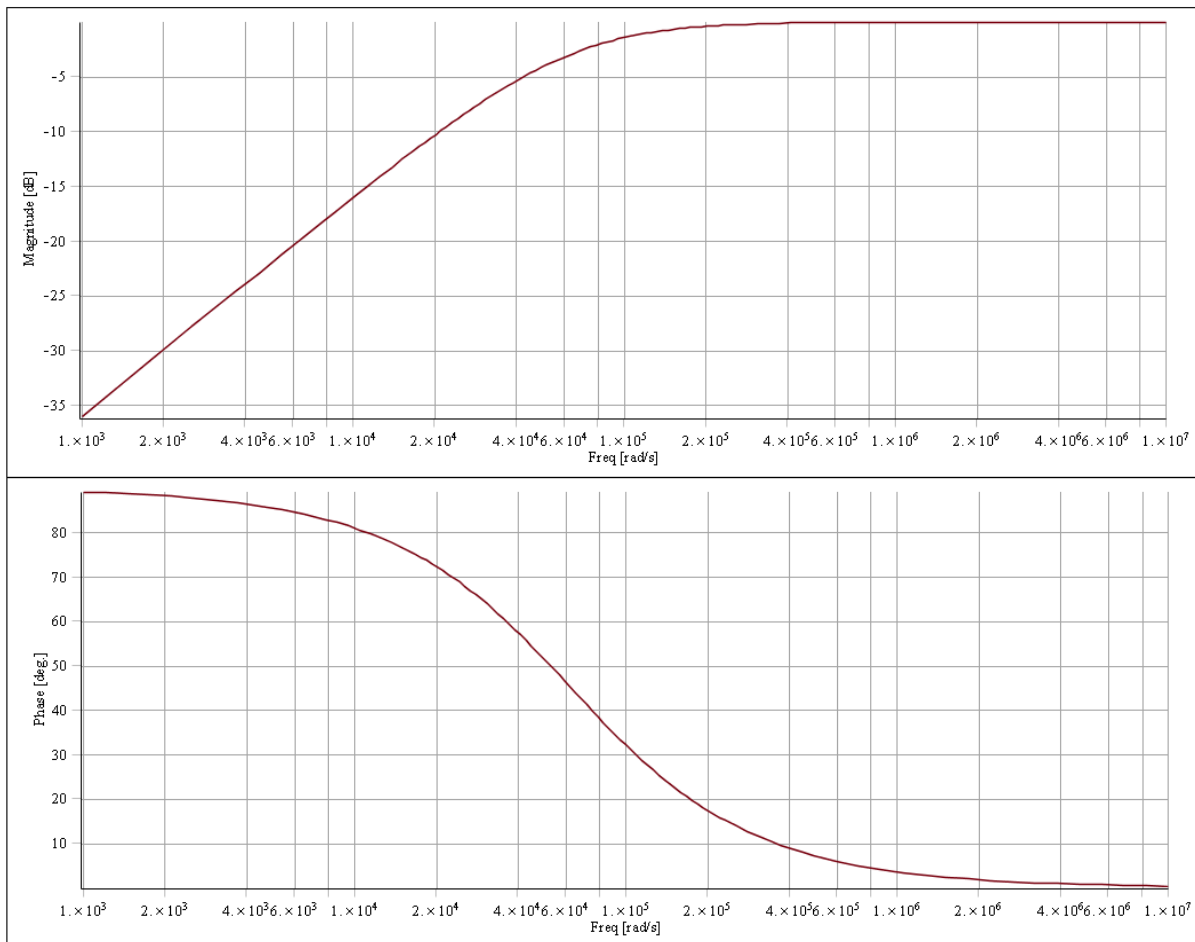
$$\omega_{2-Hz} = 10036.00616$$

(1.4)

Bodeplot:

with(DynamicSystems) :

BodePlot(TransferFunction(Tv));



T_v ;

$$\frac{0.9964123012 s}{s + 62831.85307} \quad (1.5)$$

$T_{vs} := s \rightarrow (1.5)$:

Forstærkning for 125 kHz signalet (dB):

$$Gain_{125kHz} = evalf(20 \cdot \log_{10}(\text{abs}(T_{vs}(I \cdot 120000 \cdot 2 \cdot \text{Pi}))));$$

$$Gain_{125kHz} = -0.06127349120 \quad (1.6)$$

Forstærkning for 50 Hz signalet (dB):

$$Gain_{50Hz} = evalf(20 \cdot \log_{10}(\text{abs}(T_{vs}(I \cdot 50 \cdot 2 \cdot \text{Pi}))));$$

$$Gain_{50Hz} = -46.05192688 \quad (1.7)$$

Netspænding der går igennem filteret ved hhv. 230V og 18V rms AC:

$$V_{230V} = evalf\left(230 \cdot \sqrt{2} \cdot 10^{\frac{rhs(1.7)}{20}}\right);$$

$$V_{230V} = 1.620490502 \quad (1.8)$$

$$V_{18V} = evalf\left(18 \cdot \sqrt{2} \cdot 10^{\frac{rhs(1.7)}{20}}\right);$$

$$V_{18V} = 0.1268209958 \quad (1.9)$$

Filteret er her dimensioneret efter en ønsket cutoff frekvens på 10kHz, da denne er tættest på 120kHz komponenten (For maksimal dæmpning af 50 Hz) samtidig med at der stadig er en stor sikkerhedsmargin op til de 120 kHz.

Fra udregningerne ses det, at filteret selv ved 230Vrms kun lukker 1,6Vrms igennem fra 50Hz komponenten, og denne kreds burde således godt kunne benyttes på det rigtige X.10 lysnet. I vores tilfælde er der dog kun tale om 18Vrms, og spændingen fra 50 Hz komponenten er forsvindende lille.

I den designede kreds er der tilføjet en halv Vcc(2,5V) reference, således at filterets output får et dc offset på 2,5 V, hvormed 5Vpp signalet fra 120kHz komponenten kommer til at ligge fra 0 til 5 V. Denne reference spænding er lavet med en opamp og en spændingsdeler med 2 lige store modstande. Idet denne kreds har store begrænsninger i hht. at skulle drive filteret, er der valgt en lille kondensator (giver en tilsvarende stor modstand for samme cutoff frekvens), så den strøm reference generatoren skal levere og sluge bliver minimeret.

Efter filteret køres signalet ind i en komparator med en reference på 3V, hvilket giver et pænere 0-5V firkant signal, idet der kun er 2 output states. Komparatoren har samtidig en buffer effekt, som afkobler filteret fra resten af kredsen.

Til sidst benyttes en envelope detektor bestående af R5, C10 og D2 til at generere et signal, der forbliver højt så længe, at der er et 0-5V 120kHz signal, således at µcontrolleren kan læse inputtet på et vilkårligt tidspunkt i løbet af det første millisekund fra en nul krydsning. Denne fungerer ved, at der løber strøm gennem dioden når signalet er højt, hvormed kondensatoren oplades, og der kommer til at stå ca. 4,3V efter dioden, og når signalet så går lavt aflades kondensatoren gennem modstanden. RC leddet dimensioneres så således, at kondensatoren ikke kan nå at aflades på den tid signalet er lavt (halv periode for 120kHz):

$$\text{evalf}\left(\frac{1}{120000 \cdot 2}\right);$$

0.000004166666667 (3)

$$\tau := R \cdot C;$$

$$R := 400000;$$

$$C := 100 \cdot 10^{-12};$$

$$\text{evalf}(\tau);$$

0.00004000000000 (4)

Det sidste led i data detektoren er en komparator som agerer buffer. Komparatoren sammenligner med en 2.5V reference, og sikrer at µControlleren ser ind i et rent 0-5V logisk signal.

5.1.3. Sender (Felix)

Til X.10 Senderen er der benyttet en omdimensioneret udgave af sender kredsen fra AN236.

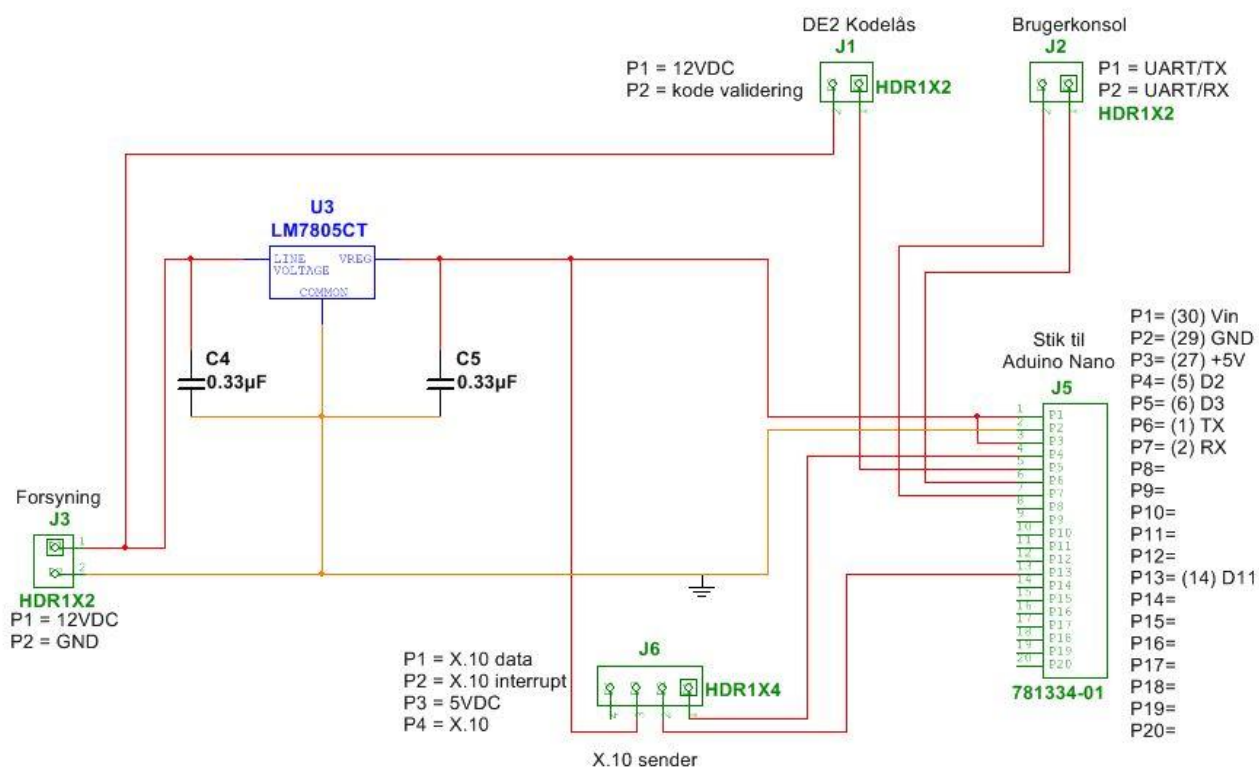
Størrelserne på komponenterne i filteret er baseret på samme udregninger, som filteret i modtageren, bare med en faktor 100 større kondensator, og en tilsvarende mindre modstand, for at gøre modstanden så lille som mulig.

Her er benyttet samme størrelser på modstandene som i modtageren, og udregningerne er således også de samme.

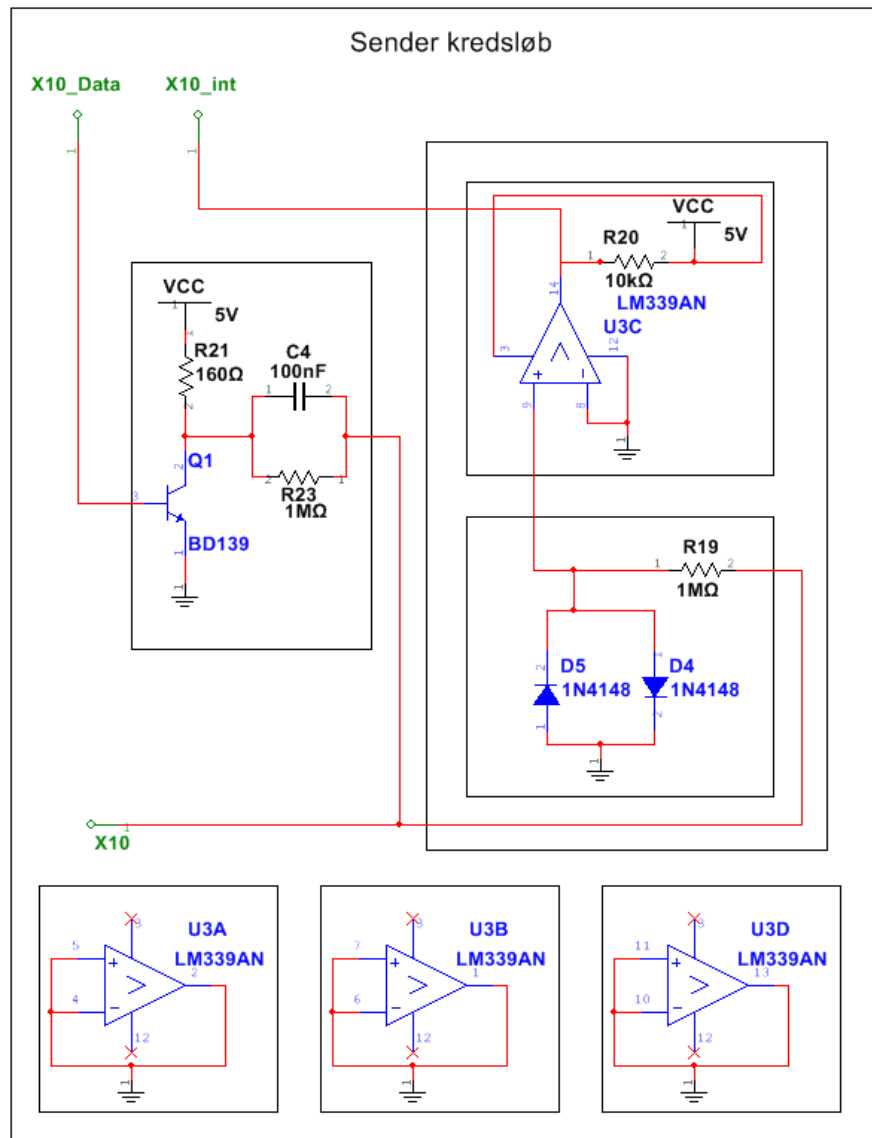
Kredsen ser ud som følger:

5.2. Masterenhed (ALLE)

Diagrammet over masterenheden 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 27 - Kredsløbsdesign til Masterenheden

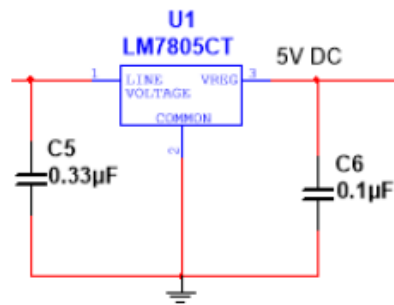


Figur 28 - Kredsløbsdesign for X10 senderen

Lysdæmper HW udregninger:

5V spændingsregulator:

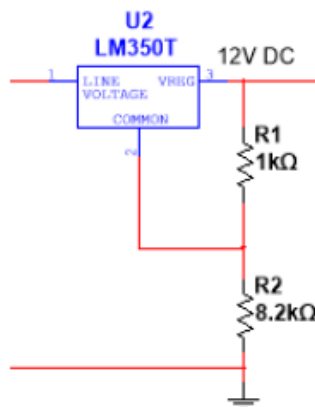
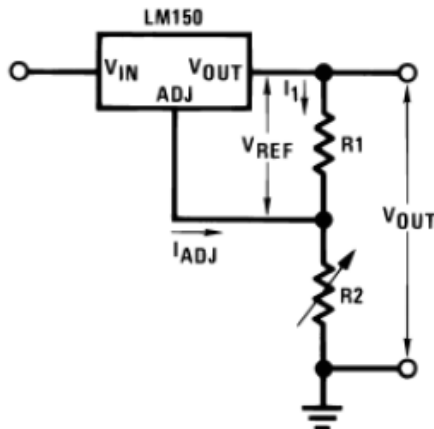
Ifølge databladets "Typical Application" anvendes to kondensatorer på henholdsvis 0.33μF og 0.1μF.



12V spændingsregulator:

Ifølge databladets "Application Hints" anvendes følgende formel for regulering af output spænding:

$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} R_2.$$



$$I_{adj} := 50 \cdot 10^{-6} \text{ A}$$

$$V_{ref} := 1.25 \text{ V}$$

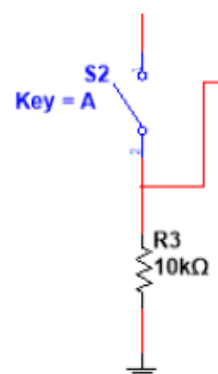
$$R_1 := 1000 \Omega$$

$$V_{out} := 12 \text{ V}$$

$$V_{out} = V_{ref} \left(1 + \frac{R_2}{R_1} \right) + I_{adj} \cdot R_2 \text{ solve } R_2 \rightarrow \frac{215000.0 \text{ V} \cdot \Omega}{25.0 \text{ V} + \text{A} \cdot \Omega} = 8.269 \text{ k}\Omega$$

ManuelOverwrite knap:

Tommelfingerregel at der anvendes 10kΩ til pull-down.



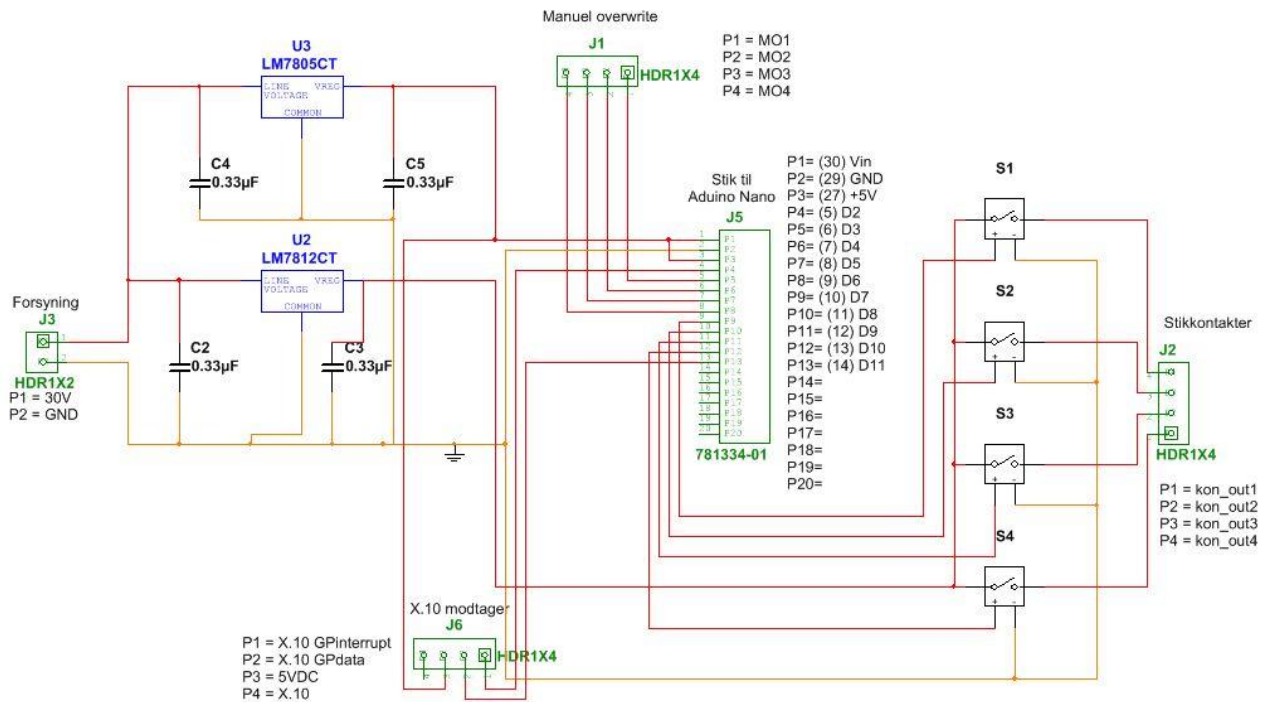
Figur 30 - Beregninger til lysdæmperen

5.4. GP-enhed (Ruben, Niels og Freddie)

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

På Figur 31 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 μ Controlleren. X10 modtageren er identisk med modtageren i lysdæmperen på **Fejl!** **Henvisningskilde ikke fundet.**

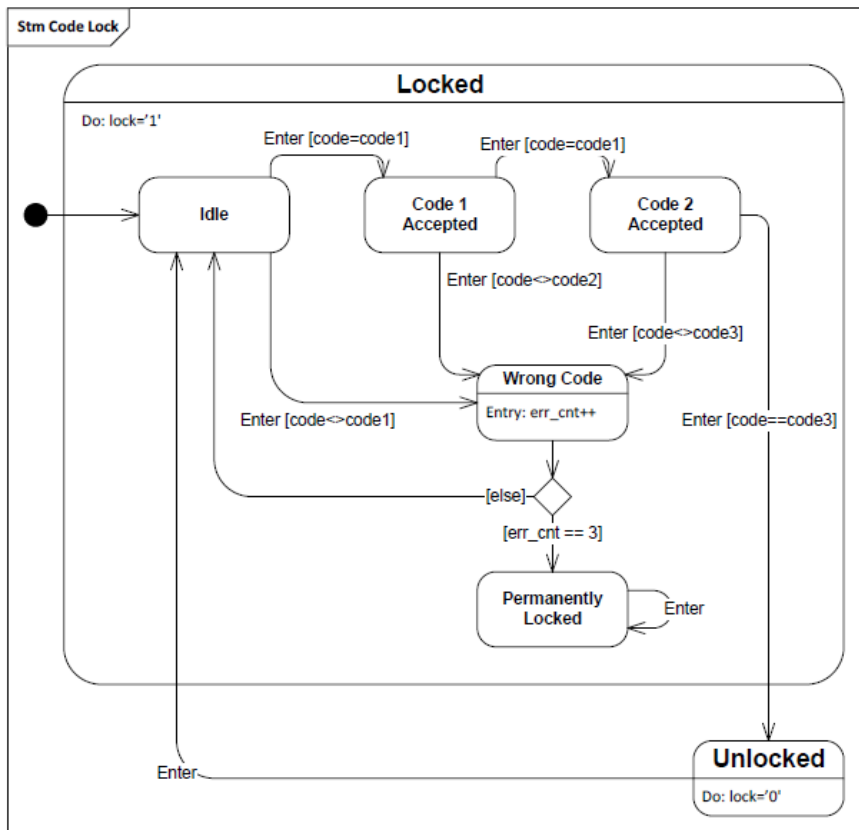
OBS! Stiknavnenes numre i parentes på Aduino Nanoen refererer til benene på selve Aduinoen og ikke til selve μ Controlleren i denne.



Figur 31 - Kredsløbsdesign til GP-enheden

6. Implementering(SW)

6.1. DE2 Kodelås (ALLE)



Figur 32 - State machine til kodelåsen

På de 2 efterfølgende sider ses koden for implementeringen af kodelåsen. Denne er implementeret ud fra det givne state machine diagram på Figur 32. Koden har et par tilføjelser i forhold til de angivne krav. Ud over det angivne moore output lock, som indikerer om låsen er låst eller ej, er der tilføjet status outputs for hvert state, således at systemet blev lidt nemmere at debugge. Til sidst er der tilføjet et vector output for at indikere antallet af mislykkede forsøg.

Designet er testet på DE2 boardet og det er konstateret, at der er korrekt overgang fra hvert af de tre kode states til wrong code state, samt korrekt overgang fra wrong code til permanently locked eller idle afhængig af antallet af fejl. Det er derudover konstateret, at de korrekte koder i korrekt rækkefølge tillader at boarded låses op, indtil der igen trykkes enter og systemet går tilbage til idle state. Der er ikke vedlagt billeder fra dette design, da det ville kræve mange billeder at vise den testede adfærd.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  -- code 1 -> 7 (0111)
6  -- code 2 -> 9 (1001)
7  -- code 3 -> 13 (1101)
8
9  entity code_lock is
10   port (clk, reset, enter : in std_logic;
11         code : in std_logic_vector (3 downto 0);
12         lock, idle_out, c1_out, c2_out, wrong_out, perm_out : out std_logic;
13         wrongs : out std_logic_vector (3 downto 0));
14   end code_lock;
15
16   architecture state_machine of code_lock is
17     -- Build an enumerated type for the state machine
18     type state_type is (idle, wrong_code, perm_locked, c1_accepted, c2_accepted, unlocked);
19
20     -- Register to hold the next- and current state
21     signal nxt_state, state : state_type;
22     signal nxt_counter, counter : unsigned (3 downto 0);
23     signal enter_event, enter_last : std_logic;
24
25   begin
26     -- Logic to advance to the next state
27     p_state_reg : process (clk, reset)
28     begin
29       if reset = '0' then
30         state <= idle;
31         counter <= "0000";
32       elsif (rising_edge(clk)) then
33         case state is
34           when idle =>
35             if enter_event = '1' then
36               if code = "0111" then
37                 state <= c1_accepted;
38               else
39                 state <= wrong_code;
40               end if;
41             end if;
42           when c1_accepted =>
43             if enter_event = '1' then
44               if code = "1001" then
45                 state <= c2_accepted;
46               else
47                 state <= wrong_code;
48               end if;
49             end if;
50           when c2_accepted =>
51             if enter_event = '1' then
52               if code = "1101" then
53                 state <= unlocked;
54               else
55                 state <= wrong_code;
56               end if;
57             end if;
58           when unlocked =>
59             if enter_event = '1' then
60               state <= idle;
61             end if;
62           when perm_locked =>
63             state <= perm_locked;
64           when wrong_code =>
65             counter <= counter + 1;
66             if counter >= 2 then

```

```

67         state <= perm_locked;
68     else
69         state <= idle;
70     end if;
71 end case;
72 end if;
73 end process p_state_reg;
74
75 p_enter_event : process (clk, reset)
76 begin
77     if reset = '1' then
78         enter_event <= '0';
79         enter_last <= '1';
80     elsif clk'event and clk = '1' then
81         enter_last <= enter;
82         enter_event <= '0';
83         if enter_last = '1' and enter = '0' then
84             enter_event <= '1';
85         end if;
86     end if;
87 end process p_enter_event;
88
89 p_moo_outputs : process (state)
90 begin
91     wrongs <= std_logic_vector(counter);
92     case state is
93         when unlocked =>
94             lock <= '0';
95             c1_out <= '0';
96             c2_out <= '0';
97             wrong_out <= '0';
98             perm_out <= '0';
99             idle_out <= '0';
100        when idle =>
101            lock <= '1';
102            c1_out <= '0';
103            c2_out <= '0';
104            wrong_out <= '0';
105            perm_out <= '0';
106            idle_out <= '1';
107        when c1_accepted =>
108            lock <= '1';
109            c1_out <= '1';
110            c2_out <= '0';
111            wrong_out <= '0';
112            perm_out <= '0';
113            idle_out <= '0';
114        when c2_accepted =>
115            lock <= '1';
116            c1_out <= '0';
117            c2_out <= '1';
118            wrong_out <= '0';
119            perm_out <= '0';
120            idle_out <= '0';
121        when wrong_code =>
122            lock <= '1';
123            c1_out <= '0';
124            c2_out <= '0';
125            wrong_out <= '1';
126            perm_out <= '0';
127            idle_out <= '0';
128        when perm_locked =>
129            lock <= '1';
130            c1_out <= '0';
131            c2_out <= '0';
132            wrong_out <= '0';
133            perm_out <= '1';
134            idle_out <= '0';
135    end case;
136 end process p_moo_outputs;
137
138 end state_machine;
  
```

Figur 33 - Kode fra implementeringen af kodelås

1.1. Sender/ Modtager (Felix)

For overblik over koden henvises til bilag 32.

1.2. Lysdæmper (Martin, Anders)

For overblik over koden henvises til bilag 31.

7. Test (HW/SW)

7.1. Modultest lysdæmper (Martin, Anders)

Lysdæmperen er testet ved at lave et test program, der justerer lysintensiteten op og ned i en løkke.

```
while(1)
{
    output(getDimlvl_());
    _delay_ms(300);
    dim();
    if(getDimlvl_()==MIN)
        while (getDimlvl_()!=MAX)
        {
            _delay_ms(300);
            brighten();
            output(getDimlvl_());
        }
}
```

Figur 34 - Test program

Undervejs i programmet kan vi vha. en fysisk knap, tænde eller slukke for lampen, alt efter om den er tændt eller slukket.

```
void manuelOverwrite()
{
    if(dimlvl_ > MIN)
        dimlvl_ = MIN;
    else
        dimlvl_ = MAX;
}
```

Figur 35 - ManuelOverwrite knap

Knappen anvender et interrupt, til at bryde ind i løkken.

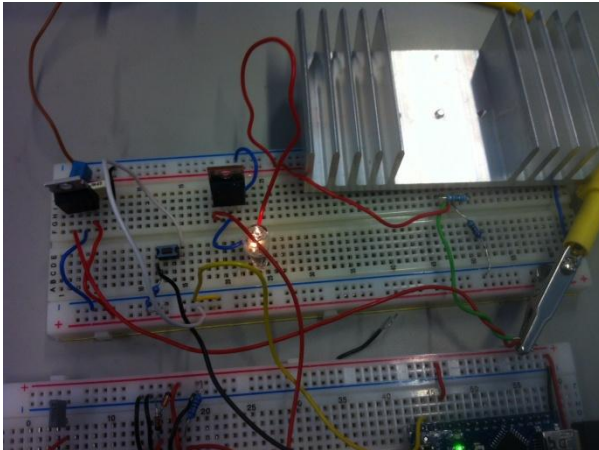
Funktionen "output" beregner en OCR2A-værdi ud fra værdien af dimlvl_, som genererer et PWM signal.

```
OCR2A = (unsigned char)(255-((85*dimlvl_)/3));
```

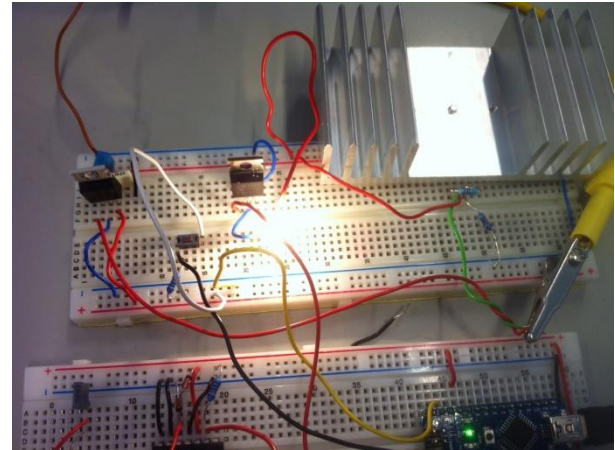
Tabel 2 - Teoretisk PWM output.

Trin	PWM
------	-----

0	0%	6	66.6%
1	11.1%	7	77.7%
2	22.2%	8	88.8%
3	33.3%	9	100%
4	44.4%		
5	55.5%		



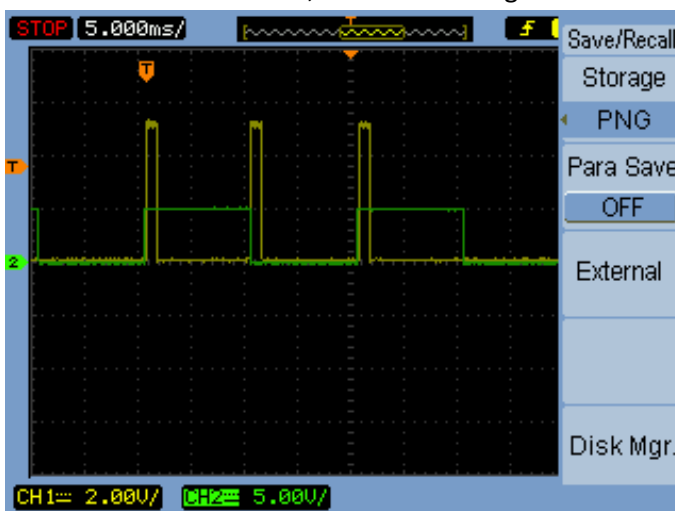
Figur 37 - Trin 1 fra tabel 2



Figur 36 - Trin 9 fra tabel 2

7.2. Modultest sender/modtager (Felix)

X10 sender softwaren er testet ved at μ Controlleren 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 38 ses en start bit som den ses på benene fra μ Controlleren.



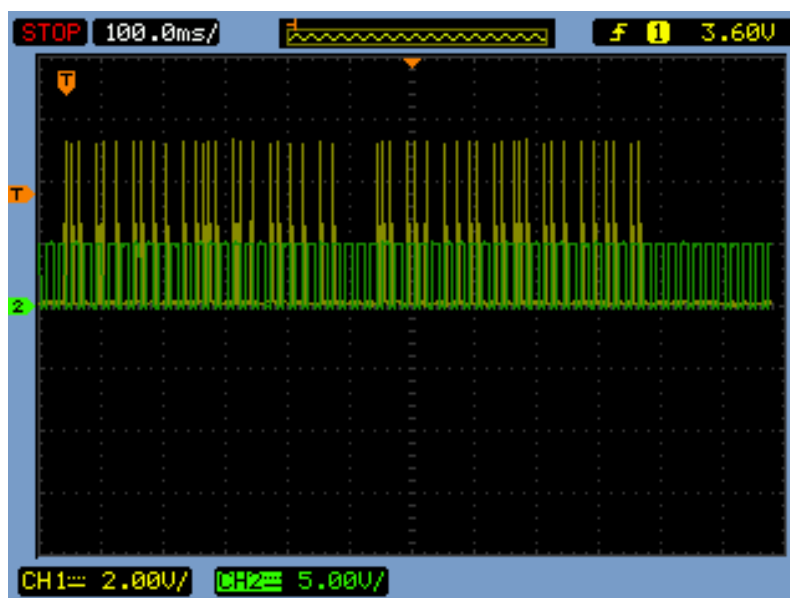
Figur 38 - Startbit fra softwaretest

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 39.



Figur 39 - 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 μ Controlleren kiggede ind i. Udlæsningen af dette ses på Figur 40, hvor der sendes kommandoen: hus A, unit 2, ON.



Figur 40 - Output fra modtagerhardwaren

8. Bilags Oversigt

- Bilag 01**-Datasheet diode 1n4148_4448
- Bilag 02**-Datasheet spændingsregulator 78xx 1A motorola
- Bilag 03**-Datasheet Microcontroller ATmega328
- Bilag 04**-Datasheet Transistor BD139
- Bilag 05**-Datasheet Mosfet IRLZ44
- Bilag 06**-Datasheet Quad amplifier LM124
- Bilag 07**-Datasheet Voltage Quad comparator LM339
- Bilag 08**-Datasheet CMOS OpAmp MCP601
- Bilag 09**-Danmarks statistik Indbrud
- Bilag 10**-Ofte indbrud i Danmark end i resten af Norden _ Nyhederne.tv2
- Bilag 11**-GP-enhed komponentliste
- Bilag 12**-Lysdæmper komponentliste
- Bilag 13**-Masterenhed Komponentliste
- Bilag 14**-120_kHz_elnet
- Bilag 15**-an236
- Bilag 16**-Development Processes
- Bilag 17**-2014-02-24 vejl.møde -Referat
- Bilag 18**-2014-02-27 grup.møde -Referat
- Bilag 19**-2014-03-04 grup.møde -Referat
- Bilag 20**-2014-03-05 vejl.møde -referat
- Bilag 21**-2014-03-10 grup.møde -Referat
- Bilag 22**-2014-03-21 Referat af Reviewmøde
- Bilag 23**-2014-03-26 vejl.møde -Referat
- Bilag 24**-2014-04-25 Referat af Reviewmøde 2
- Bilag 25**-2014-04-30 vejl.møde -Referat
- Bilag 26**-2014-05-07 vejl.møde -Referat
- Bilag 27**-2014-02-27 Tidsplan Version 1
- Bilag 28**-2014-03-27 Tidsplan Version 2
- Bilag 29**-2014-04-04 Tidsplan Version 3
- Bilag 30**-2014-04-28 Tidsplan Version 4
- Bilag 31**- LightDimmer
- Bilag 32**- Sender/Modtager