FORSIDE

FORORD

[1 Kravspecifikation 5](#_Toc453108534)

[1.1 Aktører 5](#_Toc453108535)

[1.2 Terminologiliste 6](#_Toc453108536)

[1.3 Use Cases 7](#_Toc453108538)

[1.4 Yderlige tekniske krav 16](#_Toc453108539)

[1.5 Grafisk Brugerflade 17](#_Toc453108540)

[2 Systemarkitektur 18](#_Toc453108541)

[2.1 Overordnet Arkitektur 18](#_Toc453108542)

[2.2 Styreboks 22](#_Toc453108543)

[2.3 Enhed 27](#_Toc453108544)

[2.4 Protokolbeskrivelse 30](#_Toc453108545)

[2.5 Domæneanalyse 34](#_Toc453108546)

[3 Softwarearkitektur 35](#_Toc453108547)

[3.1 Blokbeskrivelse: Domænemodel 35](#_Toc453108548)

[3.2 Identifikation af control klasser 35](#_Toc453108549)

INDHOLD

# Kravspecifikation

## Aktører

Dette afsnit indeholder en oversigt og beskrivelse af aktørerne i systemet. På PLACEHOLDER ses et aktør diagram der viser hvilken rolle aktørerne har i forvindelse med brugen af systemet.



Figur Aktørdiagram

### Bruger

|  |  |
| --- | --- |
| Aktørnavn | Bruger |
| Type | Primær |
| Beskrivelse | Brugeren er den aktør der ønsker at benytte systemet. Brugeren har kendskab til koden til kodelåsen der kræves for konfiguration og betjening af systemet, og er den aktør der står for at konfigurere enhederne samt tidsplanen som simuleringen foregår ud fra. |

### Lampe

|  |  |  |
| --- | --- | --- |
| Aktørnavn | Lampe | |
| Type | Sekundær | |
| Beskrivelse | Lampen er forbundet til systemets enhed(er), og bliver styret af systemet. |

## Terminologiliste

### PC Software

PC Softwaren betegner hele systemets PC-del. Dette inkluderer den grafiske brugerflade(GUI), såvel som den bagvedliggende funktionalitet.

### Styreboks

Systemets styreboks indeholder en liste over enheder og handlinger. Styreboksen har til formål at simulere brugerens ønskede konfigurering, ved at sammenligne det nuværende klokkeslet med et klokkeslet tilknyttet en specifik handling.

Styreboksen indeholder derved også kredsløb til at sende og modtage X10.1 kommunikation.

### Enhed

En enhed defineres i systemet som en boks indeholdende sender og modtagerkredsløb, hvor et givent stykke forbrugerelektronik kan tilsluttes. Enhedens funktionalitet er begrænset til at kunne tænde/slukke for den tilsluttede elektronik, ved brug af et relæ der kan åbne eller lukke for strømmen, på samme måde som en almindelig stikkontakt.

### X10.1

X10.1 er en navngivning givet til den protokol der bruges til kommunikation mellem styreboks og enheder. Protokollen er baseret på X10, med meget få ændringer. Navngivningen bruges for at synliggøre at det ikke er den nøjagtige X10 protokol der benyttes.

### Kodelås

Kodelås er et eksternt interface tilkoblet styreboksen. Kodelåsen indeholder funktionalitet til at godkende/afvise en indtastet kode, og sikre at systemets indstillinger ikke kan ændres uden den korrekte kode.

### Tidsplan

En tidsplan er tilknyttet en given enhed. Én tidsplan indeholder information om hvilken dag den gælder, hvilket klokkeslet den er aktiv, samt hvilken handling der skal udføres(tænde eller slukke).



## Use Cases

På PLACEHOLDER vises et Use Case diagram for X10 Home Automation systemet. Diagrammet har til formål at illustrere hvilke aktører der er tilknyttet til hvilke use-cases.



Figur Use Case diagram for X10 Home Automation

### Use Case 1: Opstart af System

|  |
| --- |
| Navn: Opstart af System |
| Mål: At få startet systemet op og få adgang til software |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: Styreboks og PC er forbundet korrekt. |
| Postkondition: Software starter op og viser grafisk brugerflade. |
| Hovedscenarie:   1. Bruger starter PC software.   [Udvidelse 1: Software kører allerede.]   1. Software anmoder om indastning af kode. 2. Bruger indtaster kode på kodelås og trykker på godkend.   [Udvidelse 2: Forkert kode indtastet.]   1. PC Software skifter til forsiden i grafisk bruger flade. |
| Udvidelser  [Udvidelse 1: Software kører allerede]   1. Intet nyt software vindue åbnes.   [Udvidelse 2: Forkert kode indtastes.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK.” 3. Fortsæt fra punkt 2. |

### Use Case 2: Status Forespørgsel

|  |
| --- |
| Navn: Status Forespørgsel |
| Mål: At vise en oversigt over tilsluttede enheder og deres status på PC Softwaren |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. |
| Postkondition: PC Software viser status oversigt. |
| Hovedscenarie:   1. Bruger vælger ”Opdater Enhedsstatus” 2. PC Software viser en opdateret status på alle tilsluttede enheder. |

### Use Case 3: Tilføjelse af enhed

|  |
| --- |
| Navn: Tilføjelse af enhed |
| Mål: At brugeren tilføjer en enhed til systemet. |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. |
| Postkondition: Den givne enhed bliver tilføjet til systemet. |
| Hovedscenarie:   1. Brugeren tilslutter enheden til lysnettet. 2. Brugeren vælger tilføj enhed. 3. Brugeren indtaster enhedens adresse. 4. Brugeren indtaster addressen til nehedens rum.   [Udvidelse 1: Brugeren tildeler ikke et rum.]   1. Brugeren trykker ”OK”.   [Udvidelse 2: Brugeren annullerer indtastningen.]   1. PC Software viser godkendt.   [Udvidelse 3: Adressen er allerede registreret.]   1. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Brugeren tildeler ikkke et rum.]   1. Fortsæt fra punkt 5.   [Udvidelse 2: Brugeren annullerere indtastningen.]   1. Skærm på PC viser hovedmenu.   [Udvidelse 3: Adressen er allerede registreret.]   1. Skærm på PC viser fejlmeddelelse. 2. Brugeren trykker OK. 3. Fortsæt fra punkt 7. |

### Use Case 4: Fjernelse af enhed

|  |
| --- |
| Navn: Fjernelse af enhed |
| Mål: At fjerne en enhed fra systemet. |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført og mindst en enhed er registreret i systemet. |
| Postkondition: Den valgte enhed fjernes fra systemet. |
| Hovedscenarie:   1. Brugeren vælger fjern enhed 2. Brugeren vælger hvilken enhed der skal fjernes. 3. Brugeren trykker OK.   [Udvidelse 1: Brugeren annullerer.]  [Udvidelse 2: Brugeren har ikke valgt en enhed.]   1. PC Software viser Godkendt. 2. Bruger trykker ”OK”. 3. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Brugeren annullerer]   1. Skærm på PC viser hovedmenu.   [Udvidelse 2: Brugeren har ikke valgt en enhed.]   1. Skærm på PC viser fejlmeddelelse. 2. Brugeren trykker ”OK.” 3. Fortsæt fra punkt 1. |

### Use Case 5:Rediger Enhed

|  |
| --- |
| Navn: Rediger Enhed |
| Mål: At rette oplysninger for en given enhed |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. Mindst én enhed er registreret i systemet. |
| Postkondition: Oplysningerne for den valgt enhed ændres. |
| Hovedscenarie:   1. Brugeren vælger rediger enhed. 2. Brugeren vælger hvilken enhed der skal redigeres. 3. Brugeren indtaster et nyt enheds ID.   [Udvidelse 1: Brugeren indtaster ikke et nyt enheds ID.]   1. Bruger indtaster nyt Rum ID.   [Udvidelse 2: Brugeren indtaster ikke et nyt Rum ID.]   1. Bruger trykker Gem Ændringer.   [Udvidelse 3: Brugeren annullerer.]  [Udvidelse 4: Bruger har ikke valg en enhed til ændring.]  [Udvidelse 5: Bruger har indtastet et nyt ID der allerede findes i systemet.]   1. PC Software viser Godkendt. 2. Brugeren trykker ”OK”. 3. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Brugeren indtaster ikke et nyt enheds ID.]]   1. Fortsæt til punkt 4.   [Udvidelse 2: Brugeren indtaster ikke et nyt Rum ID.]   1. Fortsæt til punkt 5.   [Udvidelse 3: Brugeren annullerer.]   1. Fortsæt til punkt 8.   [Udvidelse 4: Bruger har ikke valgt en enhed til ændring.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 2.   [Udvidelse 5: Bruger har indtastet et nyt ID der allerede findes i systemet.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 2. |

### Use Case 6: Ændring af tidsplan

|  |
| --- |
| Navn: Ændring af tidsplan |
| Mål: At ændre tidsplanen for en given enhed. |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. Der er minimum én enhed, med minimum én tidsplan tilføjet til systemet. |
| Postkondition: Den valgte tidsplan ændres. |
| Hovedscenarie:   1. Bruger vælger Tidsplan. 2. Bruger vælger ønsket enhed at ændre tidsplan for.   [Udvidelse 1: Brugeren vælger ikke en enhed at ændre tidsplan for.]  [Udvidelse 2: Den valgte enhed har ingen tidsplaner at ændre.]   1. Brugeren vælger tidsplanen der skal ændres. 2. Brugeren vælger hvilken dag tidsplanen skal være aktiv. 3. Brugeren indtaster starttidspunkt. 4. Brugeren indtaster sluttidspunkt. 5. Bruger trykker Gem Ændringer.   [Udvidelse 3: Brugeren anullerer.]  [Udvidelse 4: Bruger har indtastet sluttidspunkt før starttidspunkt.]   1. PC Software viser Godkendt. 2. Bruger trykker ”OK”. 3. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Brugeren vælger ikke en enhed at ændre tidsplan for.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 2.   [Udvidelse 2: Den valgte enhed har ingen tidsplaner at ændre.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 2.   [Udvidelse 3: Brugeren annullerer.]   1. Fortsæt fra punkt 10.   [Udvidelse 4: Bruger har indtastet sluttidspunkt før starttidspunkt.]   1. Skærm på PC viser fejlbesked. 2. Brugeren trykker ”OK.” 3. Fortsæt fra punkt 4. |

### Use Case 7: Kør simulering

|  |
| --- |
| Navn: Kør simulering |
| Mål: Den gemte tidsplan kører som simulering |
| Initiering: Initieres af styreboks. |
| Aktører: Lampe(sekundær) |
| Antal samtidige forekomster: 1 |
| Prækondition: |
| Postkondition: |
| Hovedscenarie: |
| Udvidelser |

### Use Case 8: Fjernelse af tidsplan.

|  |
| --- |
| Navn: Fjernelse af tidsplan |
| Mål: At fjerne en tidsplan fra en enhed. |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. Minimum én enhed, med minimum én tidsplan er tilføjet til systemet. |
| Postkondition: Den valgte tidsplan fjernes fra systemet. |
| Hovedscenarie:   1. Brugeren vælger tidsplan. 2. Skærm på PC viser oversigt over enheder og antal tidsplaner. 3. Brugeren vælger enhed der skal arbejdes på.   [Udvidelse 1: Enheden har ingen tidsplan.]   1. Brugeren vælger Fjern Tidsplan.   [Udvidelse 2: Brugeren har ingen enhed valgt.]  [Udvidelse 3: Brugeren annullerer.]   1. Skærm på PC viser oversigt med tidsplaner for valgte enhed. 2. Brugeren vælger en tidsplan som ønskes fjernet.   [Udvidelse 4: Brugeren ønsker at slette tidsplaner for en hel dag.]   1. Brugeren vælger Fjern Tidsplan.   [Udvidelse 3: Brugeren annullerer.]   1. PC Software viser godkendt. 2. Brugeren trykker ”OK”. 3. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Enheden har ingen tidsplan.]   1. Skærm på PC viser fejlbesked. 2. Fortsæt fra punkt 3.   [Udvidelse 2: Brugeren har ingen enhed valgt.]   1. Skærm på PC viser fejlbesked. 2. Fortsæt fra punkt 3.   [Udvidelse 3: Brugeren annullerer.]   1. Fortsæt fra punkt 10.   [Udvidelse 4: Brugeren ønsker at slette tidsplaner for en hel dag.]   1. Brugeren vælger hvilken dag som skal ryddes. 2. Brugeren trykker Fjern tidsplan for valgte dag. 3. Fortsæt fra punkt 8. |

### Use Case 9: Tilføjelse af tidsplan.

|  |
| --- |
| Navn: Tilføjelse af tidsplan |
| Mål: At tilføje en ny tidsplan til en eksisterende enhed. |
| Initiering: Startes af bruger |
| Aktører: Bruger (primær) |
| Antal samtidige forekomster: 1 |
| Prækondition: UC1 er udført. Minimum én enhed, med minimum én tidsplan er tilføjet til systemet. |
| Postkondition: En tidsplan tilføjes til den valgt enhed. |
| Hovedscenarie:   1. Bruger vælger Tidsplan. 2. Bruger vælger en enhed fra tabel over tilføjede enheder. 3. Bruger trykker Tilføj Tidsplan.   [Udvidelse 1: Brugeren har ikke valgt en enhed at tilføje tidsplan til.]   1. Bruger vælger ønsket tidsplanen skal være aktiv. 2. Bruger indtaster starttidspunkt. 3. Bruger indtaster sluttidspunkt. 4. Bruger trykker Tilføj Tidsplan.   [Udvidelse 2: Brugeren har indtastet sluttiddspunkt før starttidspunkt.]  [Udvidelse 3: Brugeren annullerer.]   1. PC Software viser Godkendt. 2. Bruger trykker ”OK”. 3. Skærm på PC viser hovedmenu. |
| Udvidelser  [Udvidelse 1: Brugeren har ikke valgt en enhed at tilføje tidsplan til.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 2.   [Udvidelse 2: Brugeren har indtastet sluttidspunkt før starttidspunkt.]   1. Skærm på PC viser fejlbesked. 2. Bruger trykker ”OK”. 3. Fortsæt fra punkt 10.   [Udvidelse 3: Brugeren annullerer.]   1. Fortsæt fra punkt 10. |

## Yderlige tekniske krav

### Enhed

* 1. Skal kunne tilsluttes dansk stikkontakt

### Styreboks

* 1. Skal kunne tilsluttes dansk stikkontakt
  2. Skal have en LCD skærm.
  3. Skal have en LED indikator når data sendes på lysnettet.
  4. Skal have en LED indikator når enheden er tændt.
  5. Skal kunne genstarte i tilfælde af kritiske systemfejl.

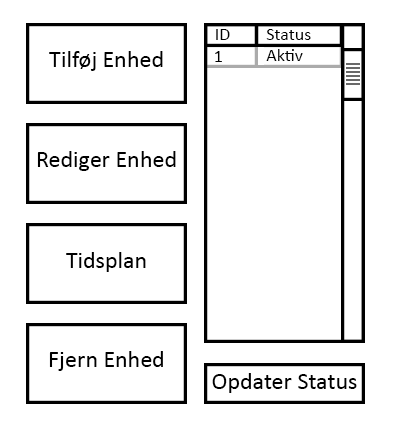
### Ikke funktionelle krav

* 1. Der skal være en gui med 5 knapper.
  2. Systemet skal have en mean time between failure på 95%
  3. Systemet bør kommunikere med op til 60 bit/s.
  4. Systemet skal have en svartid på maksimalt 2 minutter.
  5. Systemet skal kunne fungere ved tilslutning til lysnettet.
  6. Skal kunne håndtere op til 255 enheder.
  7. LED indikator når data sendes på lysnettet skal være gul.
  8. LED indikator for tændt enhed skal være grøn.

## Grafisk Brugerflade

På PLACEHOLDER vises første skitse til PC Softwarens hovedmenu. Hovedmenuen indeholder en liste over de enheder der er tilføjet til systemet, samt hvorvidt de enkelte enheder er tændte eller slukkede. Der er desuden mulighed for at navigere videre til diverse undermenuer afhængig af hvad brugeren ønsker at gøre.

Knapperne relatere sig til de enkelte use cases, hvor UC6, 8 og 9 findes under ”Tidsplan”.



Figur Udkast til hovedmenu for PC Software

# Systemarkitektur

I dette afsnit beskrives arkitekturen for X10 Home Automation. Arkitekturen dannus ud fra systemet som beskrevet i projektformuleringen og specifieret i kravspecifikationen.

Formålet med systemarkitekturen er at identificere de blokke systemet sammensættes af, samt grænseflader imellem disse blokke. Desuden bruges systemarkitekturen som et led i projektorganisationen, da den skaber overblik over arbejdsopgaver.

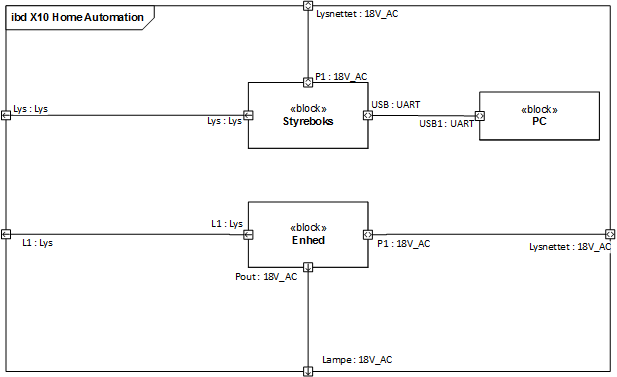
## Overordnet Arkitektur

Systemet der bygges er et distribueret system, som er opdelt som vist på PLACEHOLDER. Der forefindes ikke hardware diagrammer for PC, da dette er et system leveret af et eksternt parti.



Figur Overodnet system diagram for X10 Home Automation

På PLACEHOLDER ses et IBD for det overordnede system.



### Grænseflader for det overordnede system

|  |  |  |
| --- | --- | --- |
| Grænseflade | Porte | Stiktype |
| P1 – Enhed & styreboks | Input 18V AC Hz  Input X10.1  Output 18 V AC 50 Hz  Output X10.1  GND | Stik af dansk lovkrav |
| USB1 – Styreboks | Input 5V DC  Input max 0,5 A  Input D-  Input D+  GND | USB-stik type B hun (4-pin) |
| USB1 – PC | Input 5V DC  Input D-  Input D+  GND | USB-stik type A hun (4-pin) |
| Pout – Enhed | Output 18V AC 50 Hz  GND | Stik efter dansk lovkrav |

### Signaler for overordnet IBD

|  |  |  |
| --- | --- | --- |
| Signaltype | Definition | Beskrivelse |
| 18V\_AC | 18V AC 50 Hz signal kombineret med et 100kHz X10.1 signal | 18V AC 50 Hz signal fra forsyningsnettet der også indeholder kommunikationen via X10.1 protokollen der udvikles specifikt til dette produkt, se protokol afsnit for yderligere information. |
| Lys | Lys i det synlige spektrum | Lys i 3 farver afhængig af hvilket LED indikator der lyser. |
| Uart | Kommunikation følger UART-standarden | Protokollen udvikles specifikt til dette produkt. Se protokol afsnit for yderligere information. |

### Beskrivelser

**USB: UART** er en kommunikation der står for at sende konfigurationsdata fra pc til styreboks samt at sende informationer om systemets nuværende status og konfiguration fra styreboks til pc.

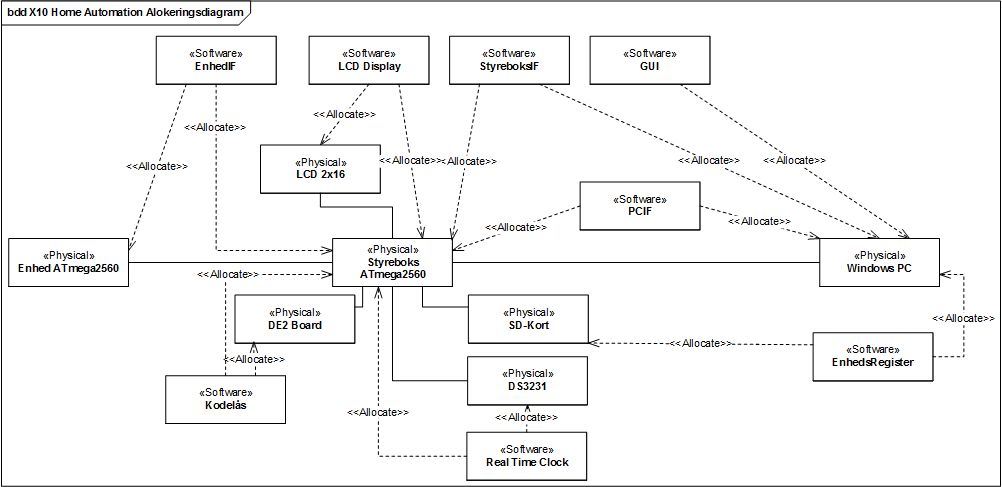
**LYS:LYS** er et synligt lyssignal fra LED indikatorerne på styreboksen der viser om styreboksen er korrekt tilsluttet lysnettet og om der er igangværende kommunikation på X10.1 eller Uart til Pc’en.

**L1:LYS** er et synligt lyssignal fra LED indikatorerne på enheden der viser om enheden er tilsluttet korrekt til lysnettet samt om der er igangværende kommunikation på X10.1.

**Pout: 18V\_AC** er et 18V AC 50 Hz signal der fungere som spændingsforsyning til en lampe, dette signal anvendes til at styre om lampen er tændt eller slukket.

**P1: 18V\_AC** er et 18V AC 50 Hz signal som samtidig indeholder et kommunikationssignal der følger X10.1 protokollen. Signalet fungere som spændingsforsyning til systemets blokke samt som kommunikationssignal mellem styreboksen og enhederne via X10.1 protokollen.

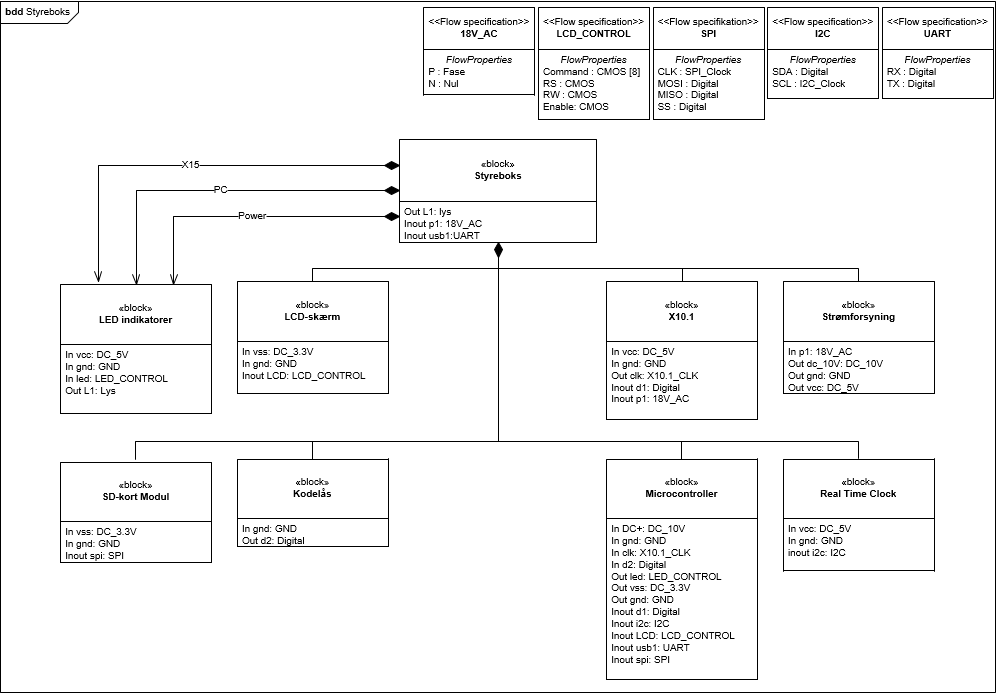
### Allokeringsdiagram



Figur Allokeringsdiagram for X10 Home Automation

## Styreboks

Analyse af styreboksen, resultere i følgende BDD vist på PLACEHOLDER. Styreboksen vil indeholde følgende blokke: **LED Indikatore,** en **LCD-skærm**, en **Microcontroller**, en **Strømforsyning**, en **Real Time Clock**, et **SD-kort Modul**, en **Kodelås**, samt en **X10.1 Sender/Modtager**. Disse blokke beskrives yderligere i det kommende afsnit.



Figur BDD og flow specifikationer for styreboks

### Blokbeskrivelser: Styreboks

Funktionaliteten af de enkelte blokke i styreboksen beskrives i dette afsnit.

**Kodelås** har til formål at forhindre ændringer i systemets indstillinger af uautoriserede brugere. Kodelåsen sender et højt eller lavt signal til **microcontrolleren** på styreboksen, når den er henholdsvist låst eller åben.

Styreboksens **strømforsyning** bruges til at omdanne en 18V AC spænding til 5V og 10V DC spænding. De 5V DC er forsyningsspænding til **Real Time Clock**, **X10.1** og **LED-indikatorer**. De 10V DC er spændingsforsyning til **microcontrolleren.**

**Microcontrolleren** bruges som en central computer, der håndterer kommunikation mellem blokkene internt i styreboksen, samt til eksterne moduler som PC og Enheder. **Microcontrolleren** er desuden ansvarlig for at læse nuværende tid fra **Real Time Clock**, og sammenligne det med eventuelle handlinger til systemets enheder.

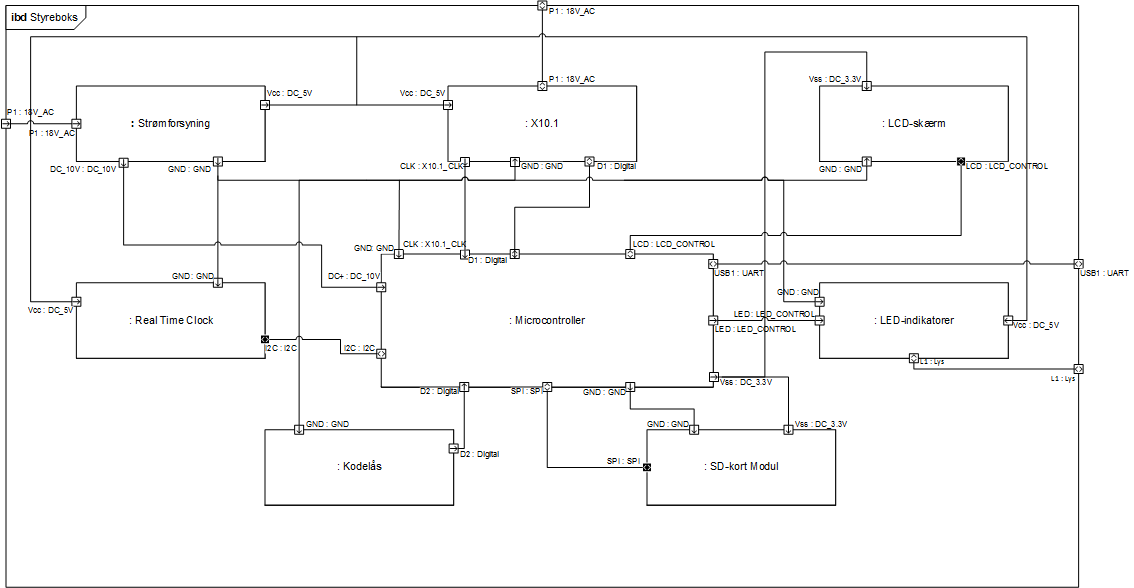
Systemet lagre alle informationer vedrørende enheder og tilhørende handlinger på **SD-Kort Modulet**. Dette gøres da **SD-Kortet** er non-volatil hukommelse, og vil derfor forblive gemt ved evt. Strømafbrydelse eller frakobling af systemet. Et **SD-Kort** benyttes da det maksimale antal enheder er sat til 255, hvilket resultere i en stor mængde potentielle handligner der skal lagres.

**LCD-Skærmen** benyttes til at give et overblik over eventuelle fejl i systemet og andre kritiske meddelelser, uden brugeren skal forbinde computeren.

**X10.1** blokken består af et sender og modtager kredsløb. Dette kredsløb benyttes til at sende kommandoer til enheder, samt at modtage svar og status for individuelle enheder.

På styreboksen forefindes tre **LED-indikatorer** der viser: Om der transmitteres data via X10.1 modulet, om der transmitteres data via UART til PC og om styreboksen er tændt.

Det logiske flow mellem ovennævnte blokke vises på PLACEHOLDER, der viser et IBD for Styreboksen. Som udgangssignaler findes kommunikation mellem PC og styrebox som UART kommunikation, samt ind og udgangssignaler til X10.1 moduleet.



Figur IBD for Styreboks

### Evt mere dekomponering.

### Signalbeskrivelse: Styreboks

Herunder findes beskrivelser af grænseflader og signaler vist på BDD og IBD for styreboksen

#### Grænseflader

|  |  |  |
| --- | --- | --- |
| Grænseflade | Porte | Stiktype |
| P1 – X10.1 | Input/Output 18 V AC 50 Hz  Input/Output X10.1 |  |
| P1 – Strømforsyning | Input 18 V AC 50 Hz  Output 18 V AC 50 Hz | Stik efter dansk lovkrav |
| LED – LED-indikatorer | Input L0 til L2 min 0 - 5 V DC |  |
| I2C – Real Time Clock | Input/Output SCA 0 – 5 V DC  Input/Output SCL 0 – 5 V DC |  |
| SPI – SD-modul | Input CLK 1 MHz  Input SS 0 – 3.3 V DC  Input/Output MOSI 0 – 3.3 V DC |  |
| LCD – LCD-skræm | Input D4 til D7 0 – 3.3 V DC  Input RS: 0 – 3.3 V DC  Input EN: 0 – 3.3 V DC  Input RW: 0 – 3.3 V DC |  |
| I2C – Microcontroller | Input/Output SCA 0 – 5 V DC  Input/Output SCL 0 – 5 V DC |  |
| LED - Microcontroller | Output L0 til L2 0 – 5 V DC |  |
| SPI – Microcontroller | Output CLK 1 MHz  Output SS 0 – 3.3 V DC  Input/Output MOSI 0 – 3.3 V DC |  |

#### Signaler

|  |  |  |
| --- | --- | --- |
| Signaltype | Definition | Beskrivelse |
| 18V\_AC | 18V AC 50 Hz signal kombineret med et 100kHz X10.1 signal | 18V AC 50 Hz signal fra forsyningsnettet der også indeholder X10.1 kommunikation. |
| Lys | Lys i det synlige spektrum | Lys i 3 farver afhængig af hvilket LED indikator der lyser. |
| DC\_10V | +10V DC signal | Forsyningsspænding til Arduino mega2560 microcontroller der har egen dc converter monteret.  Signalet er et 10V DC signal +/- 0,5V |
| DC\_5V | +5V DC signal | Forsyningsspænding til moduler der har behov for 5V DC spænding.  Signalet er et 5V DC signal +/- 0,5V |
| DC\_3.3V | +3.3V DC signal | Forsyningsspænding til moduler der har behov for 3.3V DC spænding  Signalet er et 3.3V DC signal +/- 0,2V |
| GND | 0V DC signal | Reference spænding til DC forsyningsspændinger. |
| Digital | 0-5V DC signal | Digitalt signal  3V til 5V = 1 (høj)  0V til 1,5V= 0 (lav) |
| X10.1\_CLK | 0-5V timing signal | 0-5V timing signal der er genereret af zero-cross detektoren til at bestemme hvornår der skal aflæses kommunikation ud fra X10.1 |
| LED\_CONTROL | 0-5V digitale signaler til styring af LED indikatorer | To 0-5V digitale signaler der styrer power indikator LED samt X10.1 kommunikationsindikator LED der begge er active high.  3V til 5V = 1 (høj)  0V til 1,5V= 0 (lav) |
| LCD\_CONTROL | 0-3.3V digitale signaler til styring af LCD display | Kommunikation til LCD display som følger kommunikationsprotokollen for LCD display, se protokol afsnit for yderligere information. |
| I2C | Kommunikation der følger I2C standarden | Kommunikation via I2C standarden til Real Time Clock. Se protokol afsnit for yderligere information. |
| SPI | Kommunikation der følger SPI protokollen | Kommunikation via SPI til styring af SD-kort. Se protokol afsnit for yderligere information. |

#### Beskrivelse af Signaler

**L1:LYS** er et synligt lyssignal fra LED indikatorerne på enheden der viser om enheden er tilsluttet korrekt til lysnettet samt om der er igangværende kommunikation på X10.1 eller UART.

**P1: 18V\_AC** er et 18V ac 50 Hz signal som samtidig indeholder et kommunikationssignal der følger X10.1 protokollen. Signalet fungere som spændingsforsyning til systemets blokke samt som kommunikationssignal mellem styreboksen og enhederne via X10.1 protokollen.

**LED:LED\_CONTROL** er 3 digitale signaler 0-5V der styre de 3 LED’er der indikere om enheden modtager forsyningsspænding fra forsyningsnettet samt om der kommunikeres på lysnettet via X10.1 protokollen.

**LCD:LCD\_CONTROL** er en digital kommunikationsbus til styring af LCD displayet.

**D1: Digital** er et digital 0-5V serielt signal til datatransmission mellem X10.1 og microcontrolleren hvor 3V til 5V = 1 og 0V til 1,5V = 0.

**D2: Digital** er et digitalt 0-5V signal der er active-high som fortæller microcontrolleren om koden er korrekt indtastet hvor de digitale signaler er repræsenteret ved følgende spændinger: 3V til 5V = 1 og 0V til 1,5V= 0.

**CLK: X15\_CLK** er et timingsignal der fortæller microcontrolleren hvornår D1 kan aflæses eller skrives til i forbindelse med kommunikation via X10.1 protokollen.

**Vcc: DC\_5V** er et +5V (+/- 0,5V) dc signal der forsyner 5V modulerne med spænding.

**Vss: DC\_3.3V** er et +3.3V (+/- 0,2 V) dc signal der forsyner SD-kort modulet med 3.3V dc.

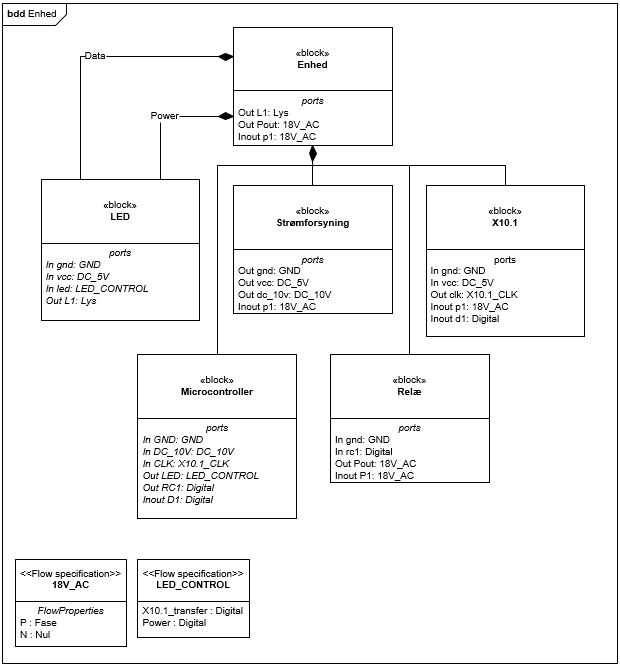
**DC\_10V:DC\_10V** er et +10V (+/- 0,5 V) dc signal der fungere som spændingsforsyning for microcontrolleren.

**GND: GND** er 0V reference spændingen til DC forsyningsspændingerne på 5V og 10V.

**USB1:UART** er et kommunikationssignal til kommunikation med pc-softwaren.

## Enhed

Den ønskede funktionalitet fra en Enhed er blevet analyseret, og illustreret i det følgende afsnit. På PLACEHOLDER ses BDD for enheden. Enheden består af en **strømforsyning**, et **X10.1** sender/modtager modul, et **relæ**, en **microcontroller** samt en/flere LED’er. For beskrivelser af **strømforsyning** samt **X10.1** modul henvises til afsnit 2.2.1.



Figur BDD for Enhed

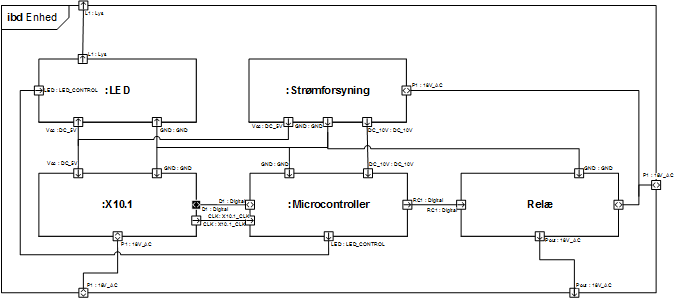
### Blokbeskrivelser: Enhed

Enheden indeholder en **microcontroller** der er ansvarlig for at tolke kommunikationen der modtages og sendes vha. **X10.1** modulet. **Microcontrolleren** håndterer også tænd og sluk af relæ modulet.

**Relæet** på enheden fungerer som switch til det tilsluttede apparat, og vil tænde og slukke baseret på input fra **microcontroller.**

**LED**

Ud fra BDD og blokbeskrivelserne udarbejdes et IBD der ses på PLACEHOLDER. Herpå vises de interne forbindelser for enheden, samt de signaler der sendes imellem.

****

Figur IBD for enhed

### Signalbeskrivelser: Enhed

Herunder findes beskrivelserne af signalerne der ses på BDD og IBD for enheden.

#### Grænseflader

|  |  |  |
| --- | --- | --- |
| Grænseflade | Porte | Stiktype |
| P1 – strømforsyning | Input 18 V AC 50 Hz  Output 18 V AC 50 Hz | Stik efter dansk lovkrav |
| LED – LED | Input 0 – 5 V DC |  |
| P1 – X15 | Input 18 V­ AC 50 Hz  Input X10.1  Output 18 V AC 50 Hz  Output X10.1 |  |
| P1 – Relæ | Input 18 V AC 50 Hz  Output 18 V AC 50 Hz |  |
| 18V\_AC – Relæ | Output 18 V AC 50 Hz |  |

#### Signaler

|  |  |  |
| --- | --- | --- |
| Signaltype | Definition | Beskrivelse |
| 18V\_AC | 18V AC 50 Hz signal kombineret med et 100kHz X10.1 signal | 18V AC 50 Hz signal fra forsyningsnettet der også indeholder kommunikationen via X10.1 protokollen der udvikles specifikt til dette produkt, se protokol afsnit for yderligere information. |
| Lys | Lys i det synlige spektrum | Lys i 2 farver afhængig af hvilket LED indikator der lyser. |
| DC\_10V | +10V DC signal | Forsyningsspænding til vores arduino mega2560 microcontroller der har egen dc converter monteret.  Signalet er et DC 10V signal +/- 0.5V |
| DC\_5V | +5V DC signal | Forsyningsspænding til moduler der har behov for 5V dc spænding.  Signalet er et DC signal på 5V +/- 0.5V |
| GND | 0V DC signal | Reference spænding til DC forsyningsspændinger samt til digitale signaler. |
| Digital | 0-5V DC signal | Digitalt signal  3V til 5V = 1  0V til 1.5V = 0 |
| X10.1\_CLK | 0-5V timing signal | 0-5V timing signal der er genereret af zero-cross detektoren til at bestemme hvornår der skal aflæses kommunikation ud fra X10.1 protokollen. Se protokol afsnit for yderligere information. |
| LED\_CONTROL | 0-5V digitale signaler til styring af LED indikatorer | 2 0-5V digitale signaler der styre power indikator LED samt X10.1 kommunikationsindikator LED der begge er active high. |

#### Beskrivelse af signaler

**L1:LYS** er et synligt lyssignal fra LED indikatorerne på enheden der viser om enheden er tilsluttet korrekt til lysnettet samt om der er igangværende kommunikation på X10.1.

**Pout: 18V\_AC** er et 18V AC 50 Hz signal der fungere som spændingsforsyning til en lampe, dette signal anvendes til at styre om lampen er tændt eller slukket.

**P1: 18V\_AC** er et 18V AC 50 Hz signal som samtidig indeholder et kommunikationssignal der følger X10.1 protokollen. Signalet fungere som spændingsforsyning til systemets blokke samt som kommunikationssignal mellem styreboksen og enhederne via X10.1 protokollen.

**LED:LED\_CONTROL** er 2 digitale signaler 0-5V hvor 3V til 5V = 1 og 0V til 1,5V= 0 som styrer de 2 LED’er der indikere om enheden modtager forsyningsspænding fra forsyningsnettet samt om der kommunikeres på lysnettet via X10.1 protokollen.

**D1: Digital** er et digital 0-5V serielt signal til datatransmission mellem X10.1 og microcontrolleren hvor de digitale 1 og 0 er repræsenteret som spændingerne 3V til 5V = 1 og 0V til 1,5V= 0.

**CLK: X15\_CLK** er et timingsignal der fortæller microcontrolleren hvornår D1 kan aflæses eller skrives til i forbindelse med kommunikation via X10.1 protokollen.

**Vcc: DC\_5V** er et +5V dc (+/- 0,5V ) signal der forsyner 5V modulerne med spænding.

**DC\_10V:DC\_10V** er et +10V (+/- 0,5V ) dc signal der fungere som spændingsforsyning for microcontrolleren.

**GND: GND** er 0V reference spændingen til DC forsyningsspændingerne på 5V og 10V.

**RC1:Digital** er et digitalt 0-5V signal der driver relæet der styre om der er et 18V AC 50 Hz udgangssignal til den tilsluttede lampe hvor 3V til 5V = 1 og 0V til 1,5V= 0

## Protokolbeskrivelse

I følgende afsnit beskrives protokollerne der bruges til kommunikation mellem systemets blokke.

### X10.1

Til at kommunikere mellem Styreboks og enhed benyttes X10.1 kommunikation.

X10.1 kommunikation sendes 1 bit ad gangen hvorfor en kommando kan opdeles i flere pakker. Fast for hver kommando er 4 startbits, 4 huskode bits, 8 adressebits, 3 typebits, nul eller flere databits, en paritetsbit og 6 slutbits.

Startbits (STX): Fortæller at en besked er på vej. Forekommer altid som: '1110'  
Huskode: Indeholder koden for det hus hvor enheden befinder sig  
Adresse: Indeholder adressen på enheden  
Typebits: Definerer antal databit i beskeden og kommandoen/svaret i datapakken. Se afsnittet *typeliste* for uddybning.   
Databits: Indeholder bools eller fejlkoder  
Paritetsbits: Validerer om besked er modtaget intakt. Der benyttes even parity (se terminologi liste for uddybning). Forekommer altid som: '0' (true) eller '1' (false)  
Slutbits (ETX): Fortæller at en besked er slut. Forekommer altid som: '000000'

I X10.1 protokollen er alle lige typenumre hvor Styreboks tilgår Enhed, oftest kommandoer. Alle ulige typenumre er hvor Enhed tilgår Styreboks, alle svar.

I oversigten over protokollerne er startbits, paritetsbit og slutbits ikke synlige eftersom de altid er en del af beskeden, og ikke defineret af brugeren. Eks:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX | Huskode | Adresse | Type | Data | Paritet | ETX |

Alle beskeder bliver altid bekræftet med en godkendelsesbesked eller en fejlmeddelelse.

#### Tabeloversigt over X10.1 kommunikation.

Nedenstående tabeller beskriver de forskellige datapakker der sendes mellem Styreboks og Enhed.  
Tabellerne er opdelt efter følgende eksempler:

|  |  |  |
| --- | --- | --- |
| Use case der tages udgangspunkt i | | |
| Metode/funktion | | |
| Beskrivelse af type og metode/funktion | | |
| Retning på dataflow: Fra > Til | | |
| Huskode: | '4-cifret binært tal' | |
| Adresse: | '8-cifret binært tal' | |
| Type: | '3-cifret binært tal' | |
| Data: | Bitnummer: data | |
| UC7: Kør Simulering (**eksempel**) | | | |
| switchState(unitID) | | | |
| Sender ny status | | | |
| Styreboks > Enhed | | | |
| Huskode: | | '0001' | |
| Adresse: | | '0001-0001' | |
| Type: | | '010' | |
| Data: | | b0: true | |

”UC7: Kør Simulering (**eksempel**)” som datapakke i bits:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX | Huskode | Adresse | Type | Data | Paritet | ETX |
| '1110' | '0001' | '0001-0001' | '010' | '1' | '0' | '000000' |

#### Specifikke kommandoer

|  |  |
| --- | --- |
| **Kommando** | **Svar** |
| |  |  | | --- | --- | | UC2: Statusforespørgsel | | | getUnitStatus(unitID) | | | Anmoder om status på enheden | | | Styreboks > Enhed | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '000' | | Data: | - | | |  |  | | --- | --- | | UC2: Statusforespørgsel | | | status | | | Sender status på enheden | | | Enhed > Styreboks | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '001' | | Data: | b0: bool | |
| |  |  | | --- | --- | | UC7: Kør Simulering | | | switchState(unitID) | | | Sender ny status | | | Styreboks > Enhed | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '010' | | Data: | b0: bool | | |  |  | | --- | --- | | UC7: Kør Simulering | | | Return true | | | Sætter enhed i ny tilstand | | | Enhed > Styreboks | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '011' | | Data: | b0: bool | |

#### Generelle Svar

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Validated | | | Besked modtaget og forstået | | | Styreboks > Enhed | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '100' | | Data: | - | | |  |  | | --- | --- | | Error | | | Besked modtaget men ikke forstået | | | Styreboks > Enhed | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '110' | | Data: | b0-b1: Fejlkode | |
| |  |  | | --- | --- | | Validated | | | Besked modtaget og forstået | | | Enhed > Styreboks | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '101' | | Data: | - | | |  |  | | --- | --- | | Error | | | Besked modtaget men ikke forstået | | | Enhed > Styreboks | | | Huskode: | 'XXXX' | | Adresse: | 'XXXX-XXXX' | | Type: | '111' | | Data: | b0-b1: Fejlkode | |

### PC Styreboks Forbindelse(PSF)

Bla bla beskrivelse

For PSF bruges UART med paritetsbit, noget noget.

Kommandoer sendt fra PC sendes efter følgende skabelon:

|  |  |  |  |
| --- | --- | --- | --- |
| Start | CMD | Data | Stop |
| 0xF0F0 | 0x\*\* | Antal data bytes afhænger af kommando | 0x0F0F |

Som standardsvar fra styreboks bruges 0x0F som godkendt(ACK) kommando.

Nedenfor ses en oversigt over kommandoer der sendes fra PC til styreboks, svar repræsenterer det der sendes tilbage fra styreboksen. Efterfølgende findes mere detaljerede beskrivelser af kommandoerne:

* Hent enheder
* Send tidsplaner

|  |  |  |  |
| --- | --- | --- | --- |
| CMD | Beskrivelse | Antal Data | Svar |
| 0x01 | PC tilsluttet | 0 bytes | ACK |
| 0x02 | PC frakoblet | 0 bytes | ACK |
| 0x03 | Validate Pin | 0 bytes | 1 byte ( 7 don’t care, 1 true false) |
| 0x05 | Anmod om enheds status | 1 byte: enheds ID | 0x01 hvis enhed er tændt.  0x00 hvis enhed er slukket. |
| 0x06 | Hent enheder | 0 bytes | 3584 Bytes.  Enheds ID  Rum ID  Eventuelle tidsplaner |
| 0x07 | Send enhed | 2 bytes: Enheds ID og Rum ID | ACK |
| 0x08 | Slet Enhed | 1 byte: Enheds ID | ACK |
| 0x09 | Ret Enhed | 3 bytes: Nuværende Enheds ID, nyt Enheds ID, nyt Rum ID. | ACK |
| 0x0A | Send tidsplaner | Ved Ingen tidsplaner:  4 Bytes.  Ellers 7-83 Bytes | ACK for hver modtaget byte. |
| 0x0B | Set nuværende tid | 7 bytes | ACK |

#### Hent Enheder

Ved hent enheder sendes hent enhed kommandoen, hvorefter styreboks svarer tilbage med 3584 bytes hvis der er en enhed at sende, eller FEJL hvis der ikke er nogen enhed at sende.

Hver enhed består af en 512 byte blok for hver dag. Hver blok indeholder enhedens ID, enhedens rum ID, hvilken dag tidsplanerne i den enkelte blok tidsplanerne hører til, tidsplanens ID, tidsplanens starttidspunkt, tidsplanens sluttidspunkt og tidsplanens ønskede handling.

Alle bytes for hver blok sendes. Hvis der ikke findes nogen/flere tidsplaner fyldes de resterende bytes med 0x00.

#### Send Tidsplaner

Ved tilføjelse, fjernelse eller ændring af tidsplan bruges Send Tidsplan kommandoen. Kommandoen sender enhedens ID, enhedens rum ID og hvilken dag de sendte tidsplaner tilhører. Hvis der ingen tidsplaner er at sende sendes 0x00 som den 4. byte, hvorefter der ikke sendes mere.

Hvis der er tidsplaner at sende, bliver data bytes enhedens ID, enhedens rum ID, hvilken dag tidsplanerne hører til. Disse bliver efterfulgt af gentagende Tidsplans ID, time, minut, handling. Disse gentages indtil der ikke er flere tidsplaner at sende.

## Domæneanalyse

Domæneanalysen for X10 Home Automation bruges som indledende identifikation af domain og boundary klasser. Ud fra modellen vist på PLACEHOLDER er følgende boundaries identificeret:

* Mellem PC og bruger
* Mellem PC og styreboks
* Mellem styreboks og enhed



Figur Domæneanalyse for X10 Home Automation

# Softwarearkitektur

I følgende afsnit beskrives softwaren for X10 Home Automation systemet. Som før beskrevet er det et distribueret system, og software arkitekturen er derfor delt op mellem de tre processorer: PC, Styreboks og Enhed.

## Blokbeskrivelse: Domænemodel

Herunder følger en kort beskrivelse af de enkelte blokke på domænemodellen, vist på PLACEHOLDER.

### PC

PC bliver anvendt af brugeren vha. En grafisk brugerflade. På PC findes software til at behandle brugerens input, samt at sende ændringer og tilføjelser af enheder/tidsplaner til styreboksen. PC’en læser fra styreboksen for at hente informationer vedr. Status for kodelåsen, status på tilføjede enheder eller for at hente informationer om gemte enheder.

PC’en skriver til styreboksen når der skal tilføjes eller fjernes nye enheder og tidsplaner.

## Identifikation af control klasser

Som led i at identificere hvilke control klasser der skal bruges for hver CPU skabes en matrice over systemets processorere og hvilke use cases de er relevante i. Dette vises i PLACEHOLDER, hvori et X repræsenterer at CPU’en er inkluderet i den givne use case

|  |  |  |  |
| --- | --- | --- | --- |
| UseCase/cpu | PC Software | Styreboks | Enhed |
| UC1 | X | X |  |
| UC2 | X | X | X |
| UC3 | X | X |  |
| UC4 | X | X |  |
| UC5 | X | X |  |
| UC6 | X | X |  |
| UC7 |  | X | X |
| UC8 | X | X |  |
| UC9 | X | X |  |

Tabel Use Case / CPU matrix

## Overordnet Funktionalitet : Sekvensdiagrammer

I følgende afsnit findes overordnede diagrammer for de forskellige use cases. Diagrammerne inkludere de tre CPU’er inkluderet i systemet. Disse benyttes til at skabe et overblik over hvordan de enkelte blokke skal fungere. Efterfølgende laves en uddybning af funktionaliteten for hver use case til hver CPU.