

# 1 Systemarkitektur

## 1.1 Version

Dato	Version	Initialer	Ændring
29. oktober	1	LS	Første udkast af dokumentet
19. november	2	KT	Mange mindre rettelser efter review
02. december	3	HBJ	Rettelser af lampe signaler

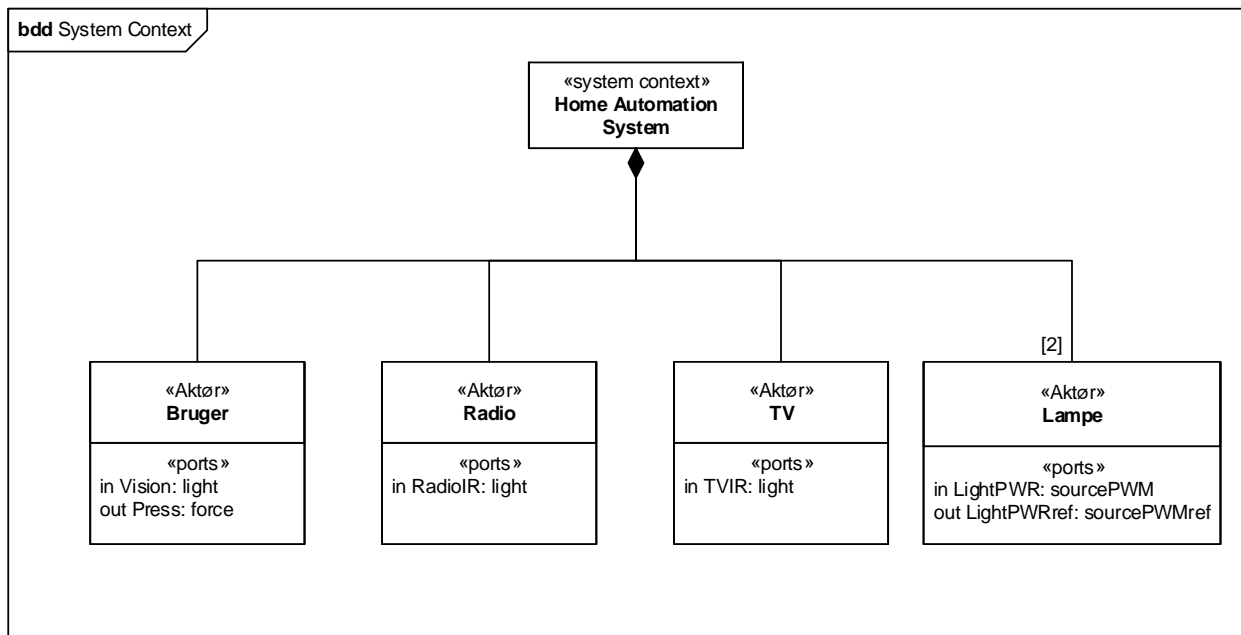
## 1.2 SysML diagrammer

Beskrivelse af samtlige signaler findes i Tabel 1 på side 8.

### 1.2.1 BDD for system kontekst

I Figur 1 vises konteksten for systemet, som består af de elektriske enheder samt brugeren af systemet. Yderligere vises porte på aktørerne, som agerer med systemet. Bruger blokken beskriver den person der interagerer med systemet. Radio blokken er en radio med mulighed for styring via en IR fjernbetjening, TV blokken er tilsvarende. De to lampe blokke er to 12V lyskilder, som kan dimmes via pulsbreddemodulation.

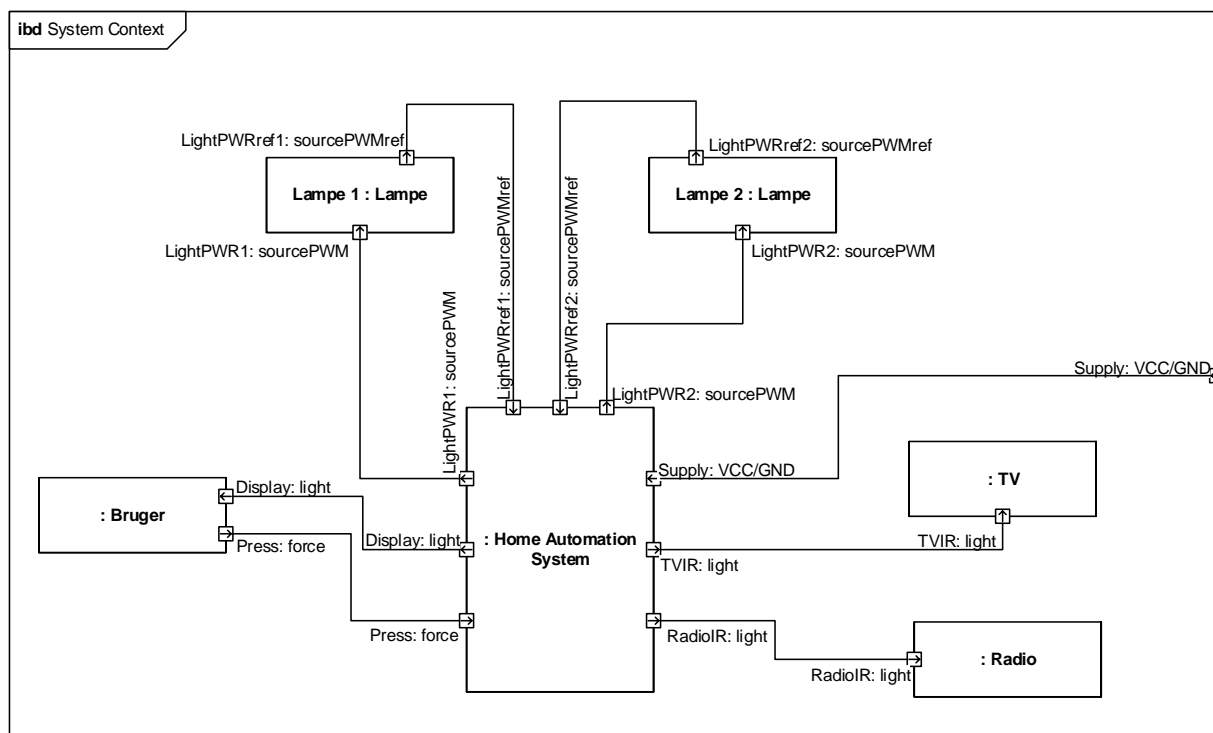
Bemærk at ingen blokke eller enheder forsynes fra 18V AC nettet, da dette kun bruges til X.10 kommunikation. Der benyttes derimod eksterne spændingsforsyninger.



Figur 1: BDD diagram for system konteksten

## 1.2.2 IBD for system kontekst

I Figur 2 vises systemets eksterne forbindelser til de øvrige blokke omkring systemet.



Figur 2: IBD diagram for system konteksten

Det antages at der til TV samt Radio allerede er tilført spændingsforsyninger, da disse enheder ikke er en del af systemet.

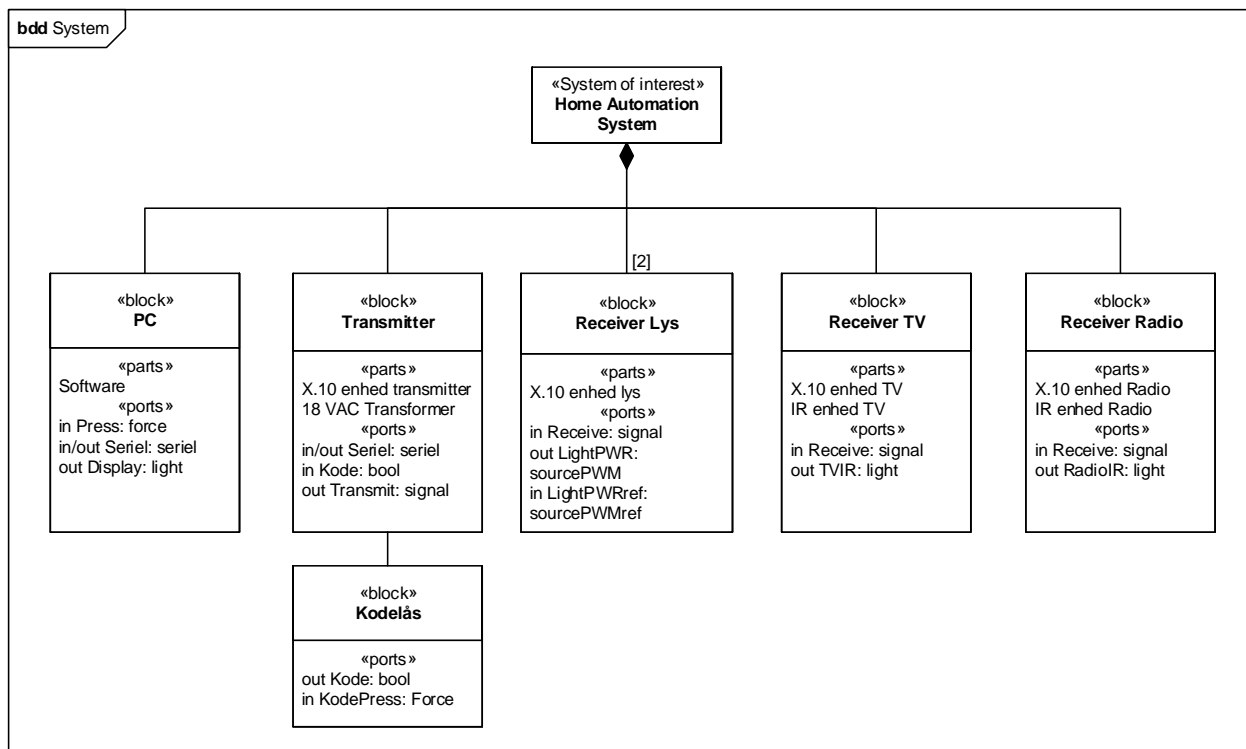
### 1.2.3 BDD for systemet

I Figur 3 vises et BDD over systemet i sin helhed. Diagrammet viser overordnede blokke i systemet, samt deres ports og parts.

PC blokken er brugerens grænseflade til systemet. Transmitterblokken modtager information fra PC softwaren, såfremt kodelåsen er aktiveret, og sender kommandoer til Receiver-blokkene.

Receiver-blokkene består kun af receiveere, dvs. selve TV'et, Radioen og lamperne ikke er en del af systemet.

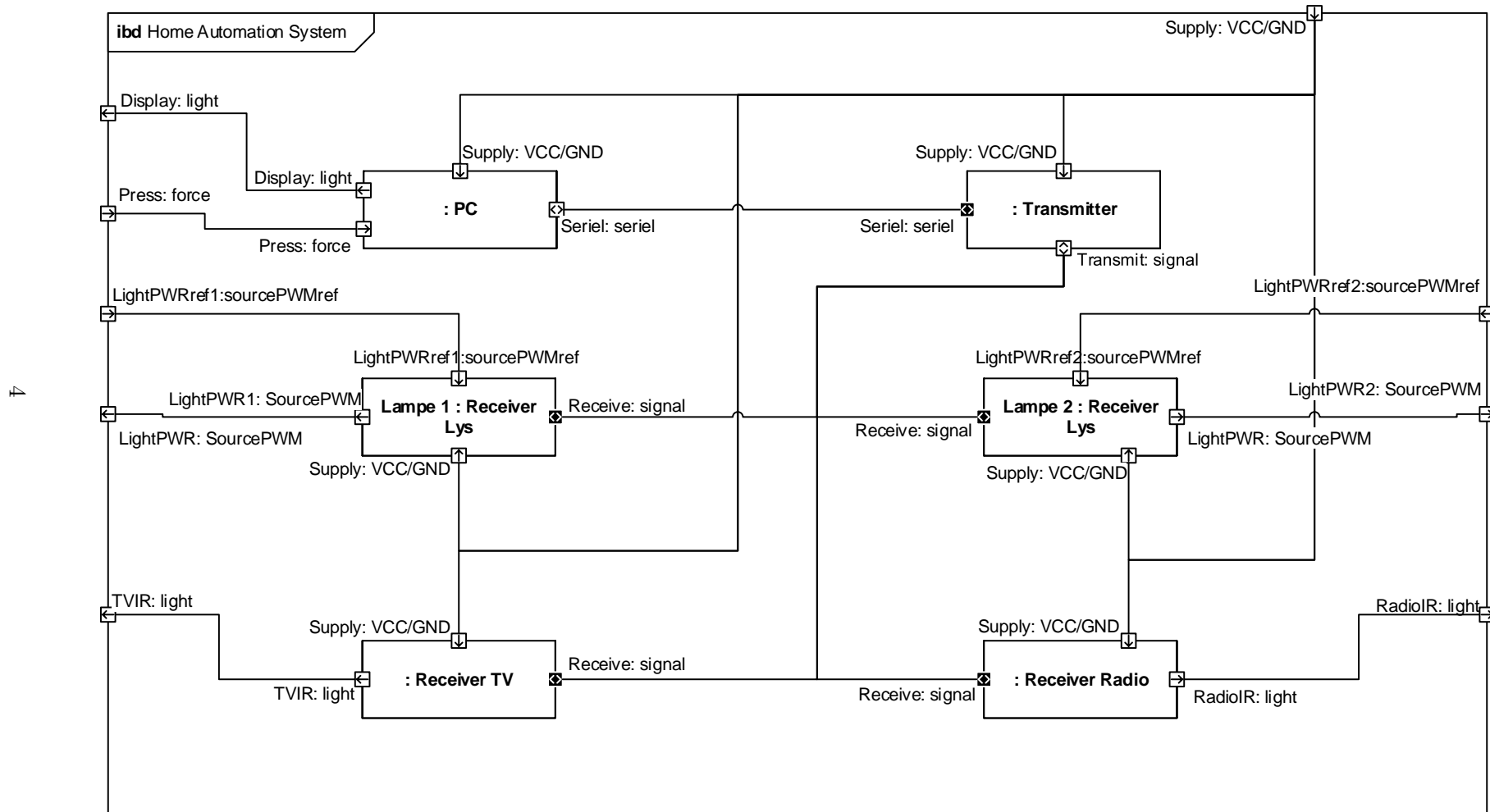
Der er for overskuelighedens skyld valgt ikke at indskrive systemets porte i «System of interest» blokken. Disse fremgår af Figur 2.



Figur 3: BDD diagram for systemet

### 1.2.4 IBD for systemet

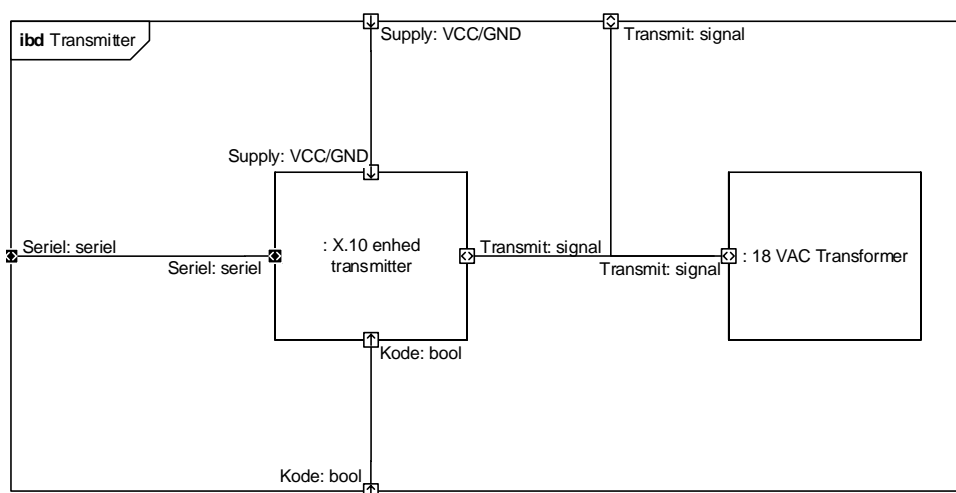
I Figur 4 vises de interne forbindelser mellem blokke i systemet. Selve lampen får sin forsyning gennem *LightPWR: SourcePWM* og har en reference fra *LightPWRref: SourcePWMref*. Signaltypen *signal* har sin reference via *VCC/GND*.



Figur 4: IBD diagram for Home Automation systemet

### 1.2.5 IBD for Transmitter-blokken

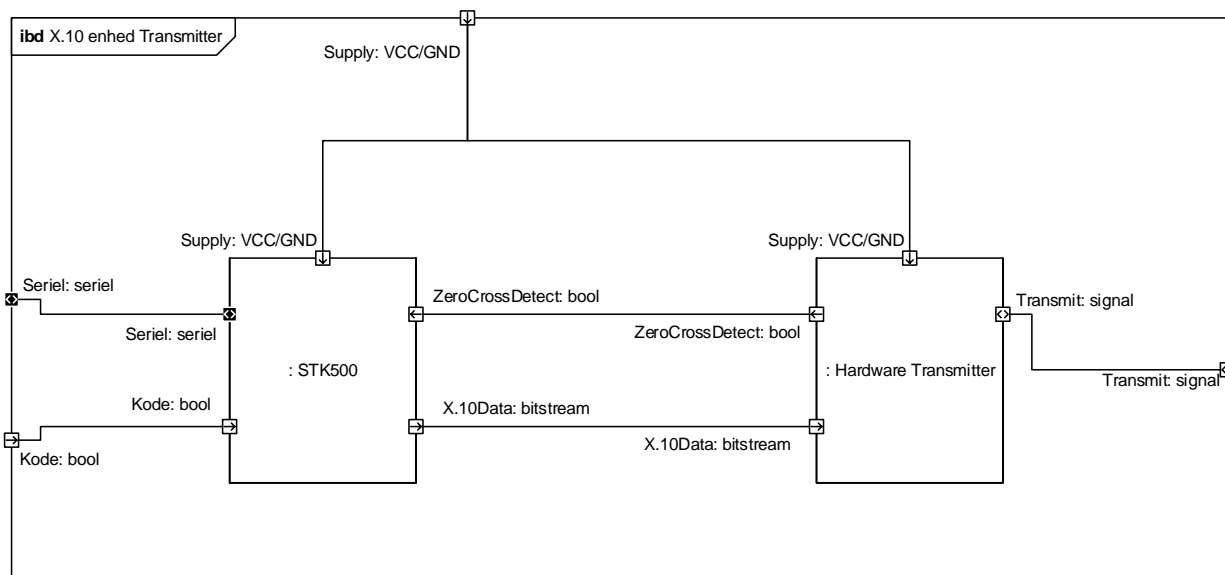
I Figur 5 vises de interne forbindelser mellem parts i Transmitter-blokken.



Figur 5: IBD diagram for Transmitter

### 1.2.6 IBD for X10-enhed i transmitter-blokken

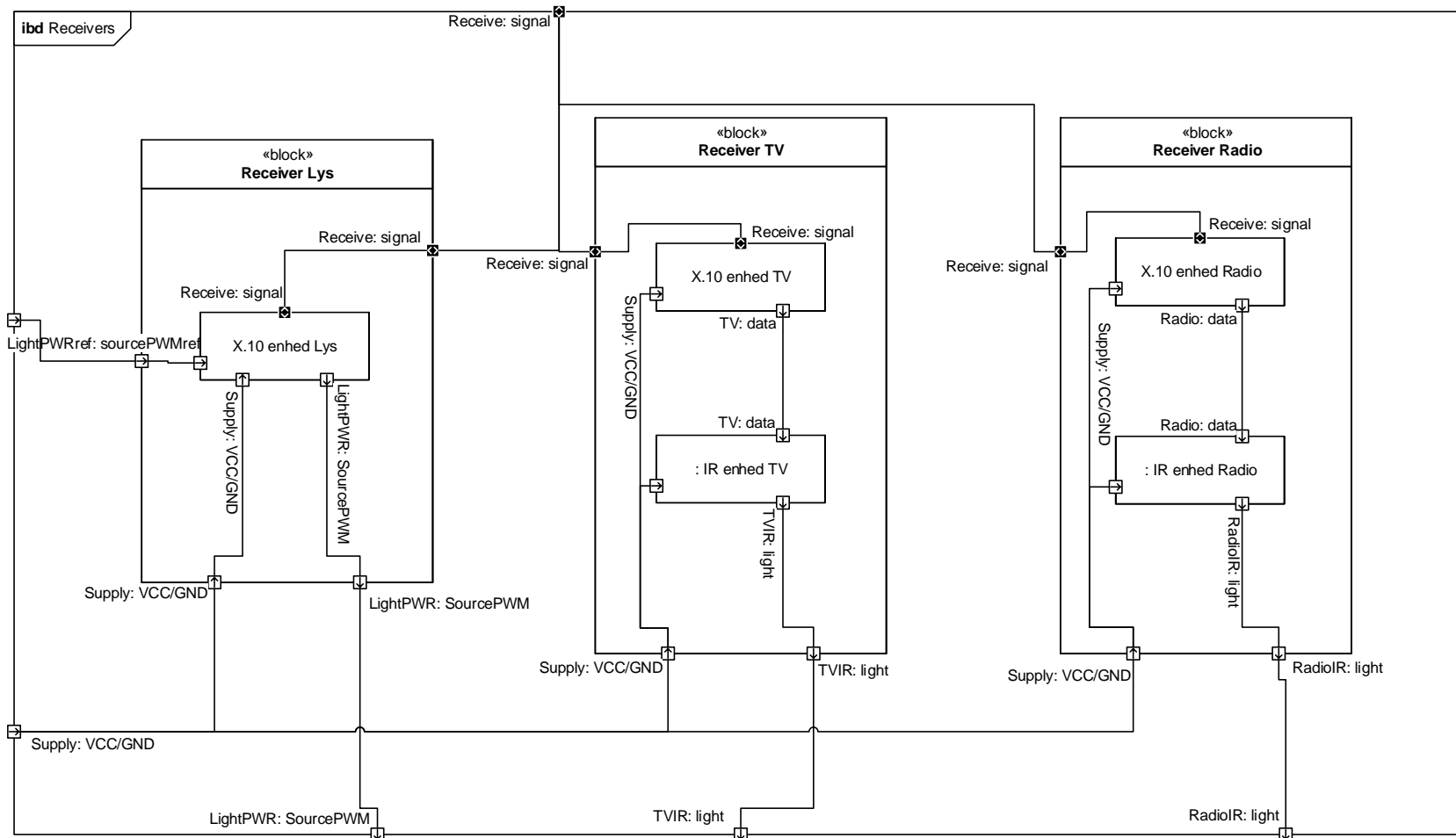
I Figur 6 vises forbindelser mellem STK500 og øvrig hardware i blokken, og dermed grænsefladen mellem software og hardware.



Figur 6: IBD diagram for X.10 enheden i transmitteren

### 1.2.7 IDB for receivers

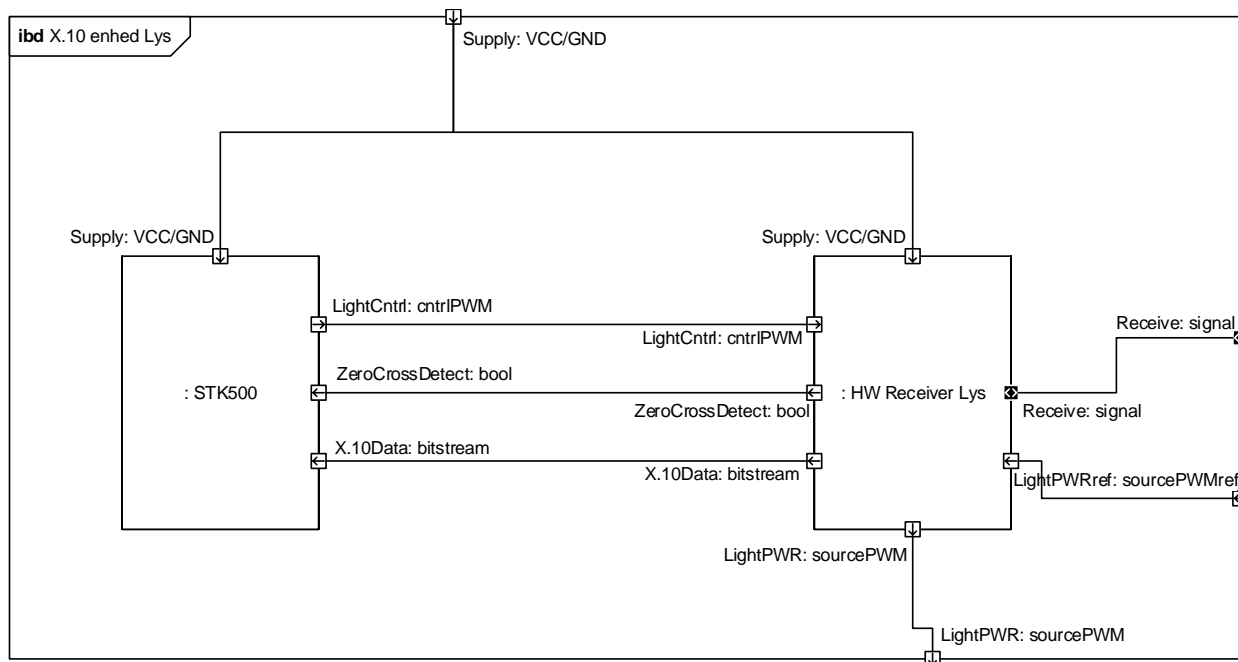
I Figur 7 vises samtlige typer af receivers i systemet. Af hensyn til overskuelighed vises der kun én instans af Receiver Lys, selvom systemet reelt indeholder to, jf multiplicitet i Figur 3 side 3.



Figur 7: IBD diagram for receivers

### 1.2.8 IDB for X.10 enhed i lys-blokken

I Figur 8 vises forbindelser mellem STK500 og øvrig hardware i blokken, og dermed grænsefladen mellem software og hardware.



Figur 8: IBD diagram for X.10 enheden i transmitteren

Bemærk at projektdokumentationen indtil dette punkt har indeholdt det komplette system, men fra dette punkt behandles TV- og Radio-delen af systemet ikke. Dvs. at en del af signalerne ikke er beskrevet samt at videre dokumentationer og design ikke omfatter TV- og Radio-delen.

## 1.3 Signalbeskrivelser

### 1.3.1 Signaltyper

Signaltype	Funktion	Område	Kommentar
bitstream	Serie af 1'er og 0'er	HIGH: $3.1 - 5.4V$ , LOW: $-0.4 - 0.9V$	1 = Tilstedeværelse af $120kHz$ signal, som går fra LOW til HIGH. 0 = LOW
cntrlPWM	Firkantformet PWM signal på 1 kHz til regulering af lysstyrke	$0.0 - 5.0V$ maks $10mA$ .	PWM duty cycle kan variere i området $5\% - 95\%$ i trin af $10\% \pm 1\%$ .
force	Brugerens input på PC		
light	Output på PC'ens skærm		
seriel	Kommunikation mellem PC og transmitter	jf. RS232	
signal	Sammensætning af $18VAC$ og X.10	$18VAC \pm \text{Lasse}\%$	$18VAC$ leveres fra transformeren og X.10 ( $120kHz$ i nulgen-nemgange) leveres fra Transmitter-blokken.
sourcePWM	Firkantformet PWM spændingsforsyning på 1 kHz til at drive lys	$0.0V - VCC$	Samme pulsbredde som <i>cntrlPWM</i> .
sourcePWMref	Reference til lampe	$0.0V - (2.9V \pm 0.5V)$ med samme frekvens som SourcePWM	Signalet kommer til at ligne sourcePWM blot med en lavere spænding
VCC/GND	DC spændingsforsyning og reference	VCC: $11.8 - 12.2V$ , GND: $0.0V$	Signalet er en sammensætning af to forbindelser, VCC og stel.
bool	Kan være enten HIGH (1) eller LOW (0)	HIGH: $3.1 - 5.4V$ , LOW: $-0.4 - 0.9V$	1 = HIGH, 0 = LOW

Tabel 1: Beskrivelse af samtlige signaler.

### 1.3.2 Grænseflader

- *ZeroCrossDetected: bool*  
Skal toggle hver gang der registreres en nulgen-nemgang på *Transmit: Signal*.
- *Receive: Signal & Transmit: Signal*  
Består af et  $18VAC$  ved  $50Hz$ , som for hver nulgen-nemgang til  $1ms$  efter kan indeholde



120KHz signaler. Hvis der forefindes et 120KHz signal i en nulgennemgang, betragtes det som HIGH. Hvis der ikke forefindes et 120KHz signal, betragtes det som LOW.

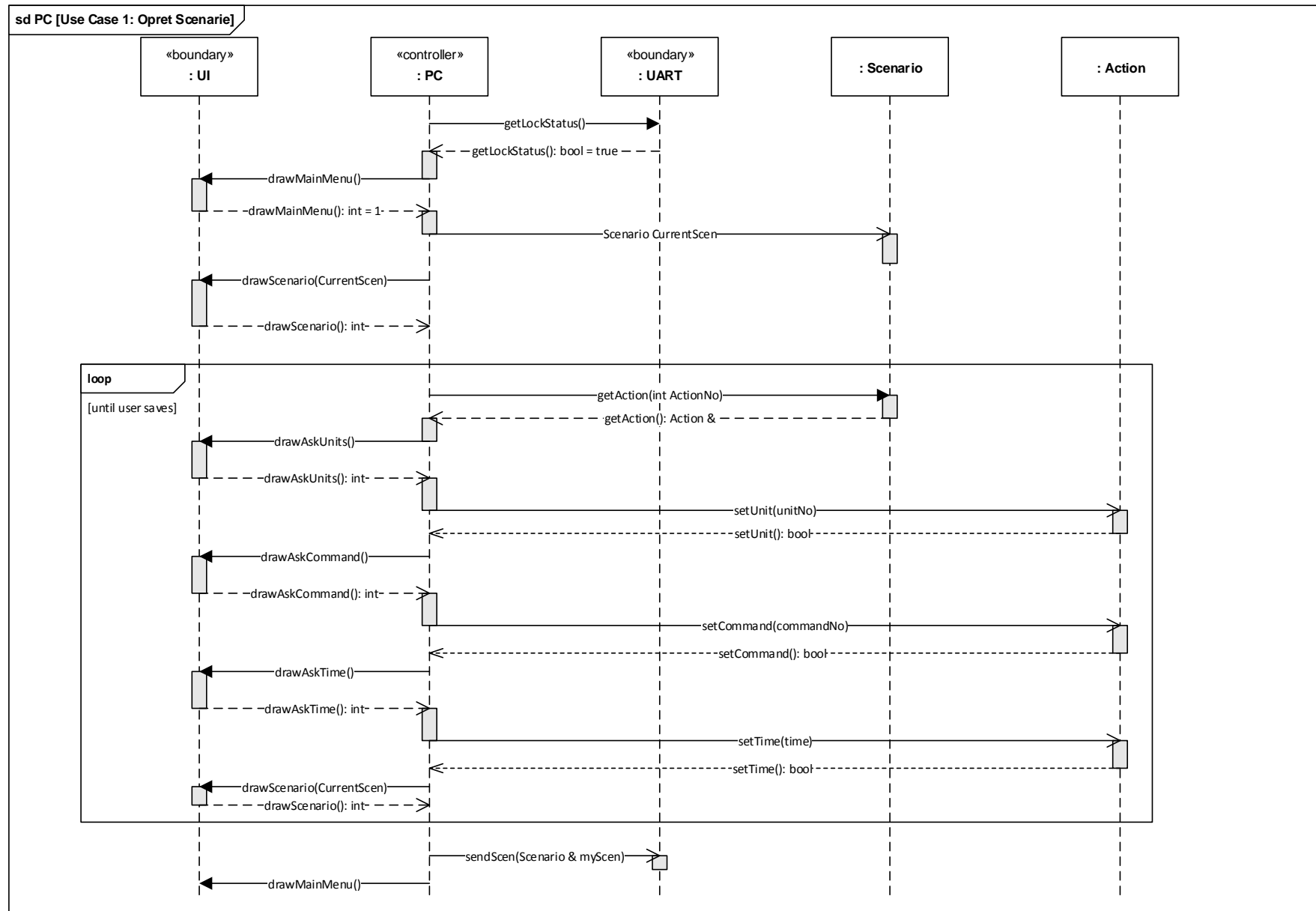
- *X.10Data: bitstream*  
Signalet er grænsefladen mellem STK500 og hhv. transmitter og receiver. HIGH på signalet svarer til at der forefindes 120kHz på *signal*. Når signalet bruges til at sende med skal det holdes HIGH i 1ms fra nulgennemgang.
- *Kode: bool*  
Er LOW hvis koden er indtastet korrekt og HIGH hvis den ikke er indtastet korrekt.
- *Seriel: seriel*  
BAUD rate på 9600, 8 bit, 1 startbit, 2 stopbit og ingen paritet.
- *LightCntrl: CntrlPWM*  
Firkantformet PWM styresignal til regulering af *LightPWR: SourcePWM*.
- *LightPWR(1 & 2): SourcePWM*  
Firkantformet PWM forsyningssignal til at drive lampen. Duty cycle samt frekvens for *LightPWR: SourcePWM* og *LightCntrl: CntrlPWM* skal være ens.
- *LightPWRref(1 & 2): SourcePWMref*  
Reference til *LightPWR: SourcePWM*. Samme frekvens og duty cycle som *LightPWR: SourcePWM*.

## 1.4 Software arkitektur

For samtlige diagrammer gælder det at prækonditioner til de enkelte UC er medtaget.

### 1.4.1 Sekvensdiagram for PC [Use Case 1: Opret Scenarie]

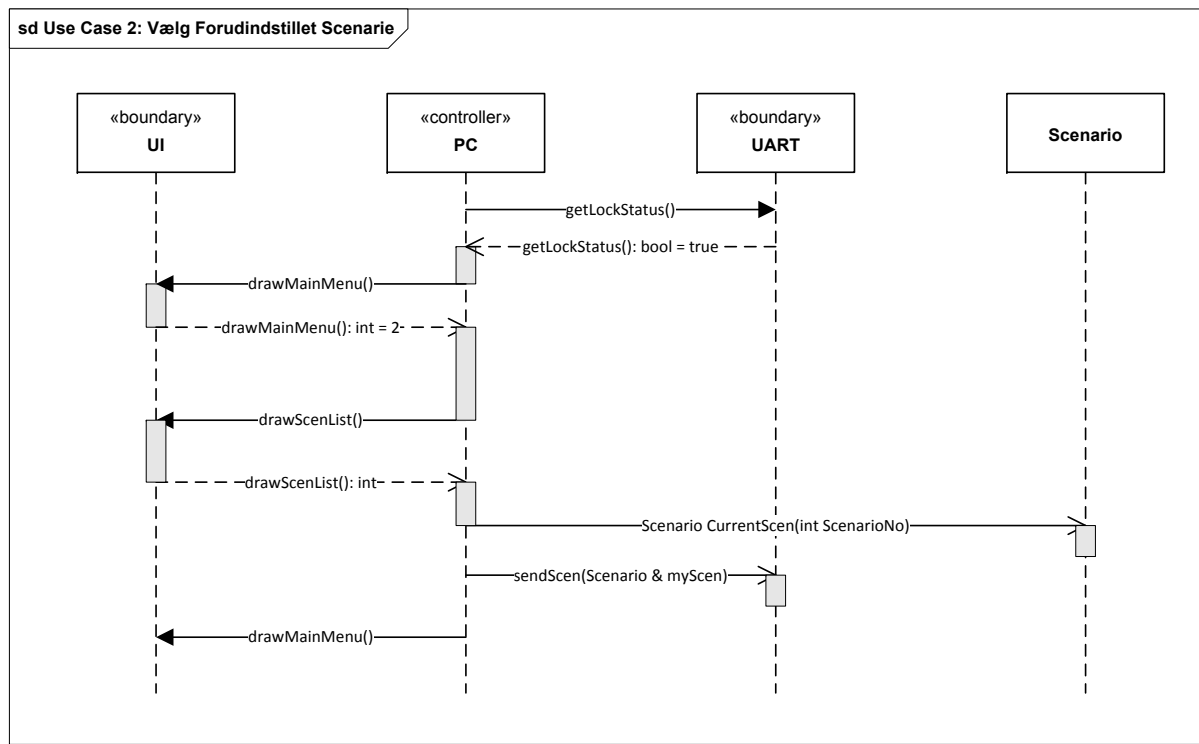
Diagrammet i Figur 9 viser sekvenser for PC'en ved gennemgang af hovedscenariet i UC1. Når der oprettes et objekt af klassen Scenario, oprettes det automatisk med 20 tomme aktioner.



Figur 9: Sekvensdiagram for PC [Use Case 1: Opret Scenarie]

### 1.4.2 Sekvensdiagram for PC [Use Case 2: Vælg Scenarie]

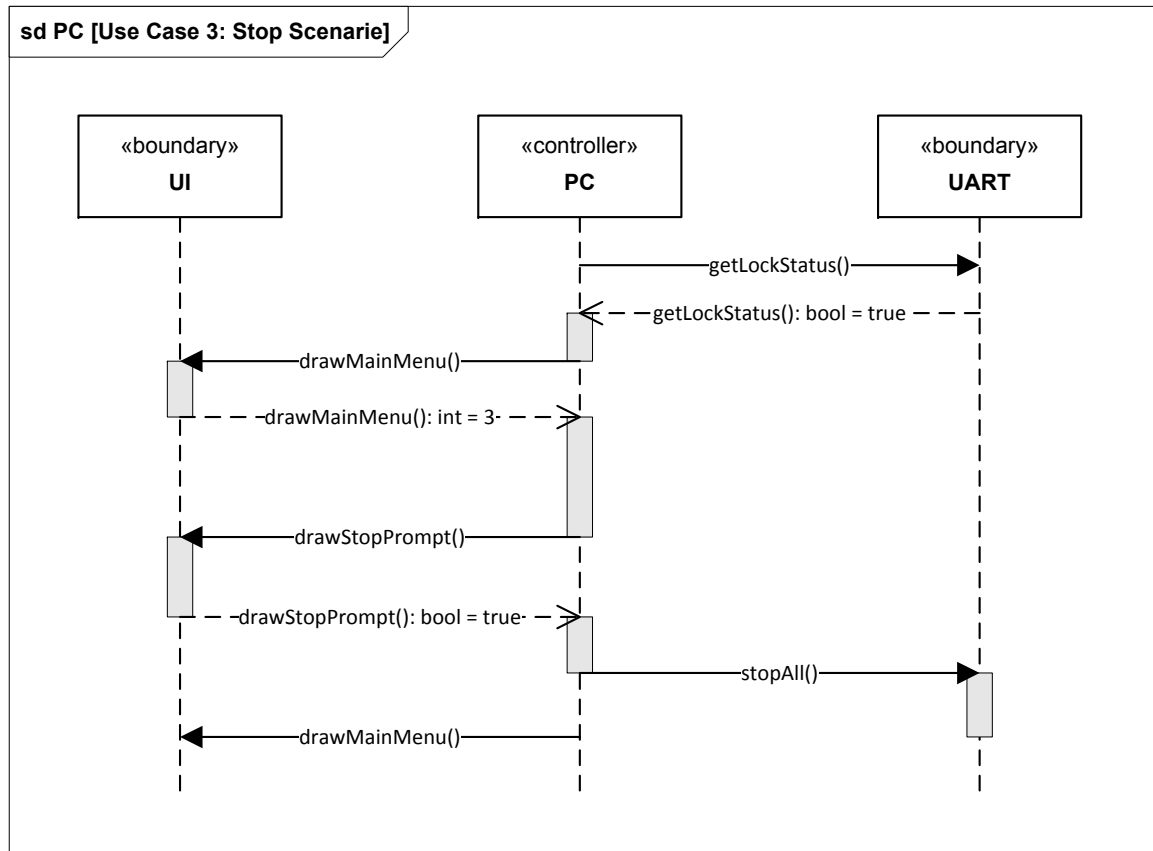
Diagrammet viser sekvenser for PC'en ved gennemgang af hovedscenariet i UC2.



Figur 10: Sekvensdiagram for PC [Use Case 2: Vælg Scenarie]

### 1.4.3 Sekvensdiagram for PC [Use Case 3: Stop Scenarie]

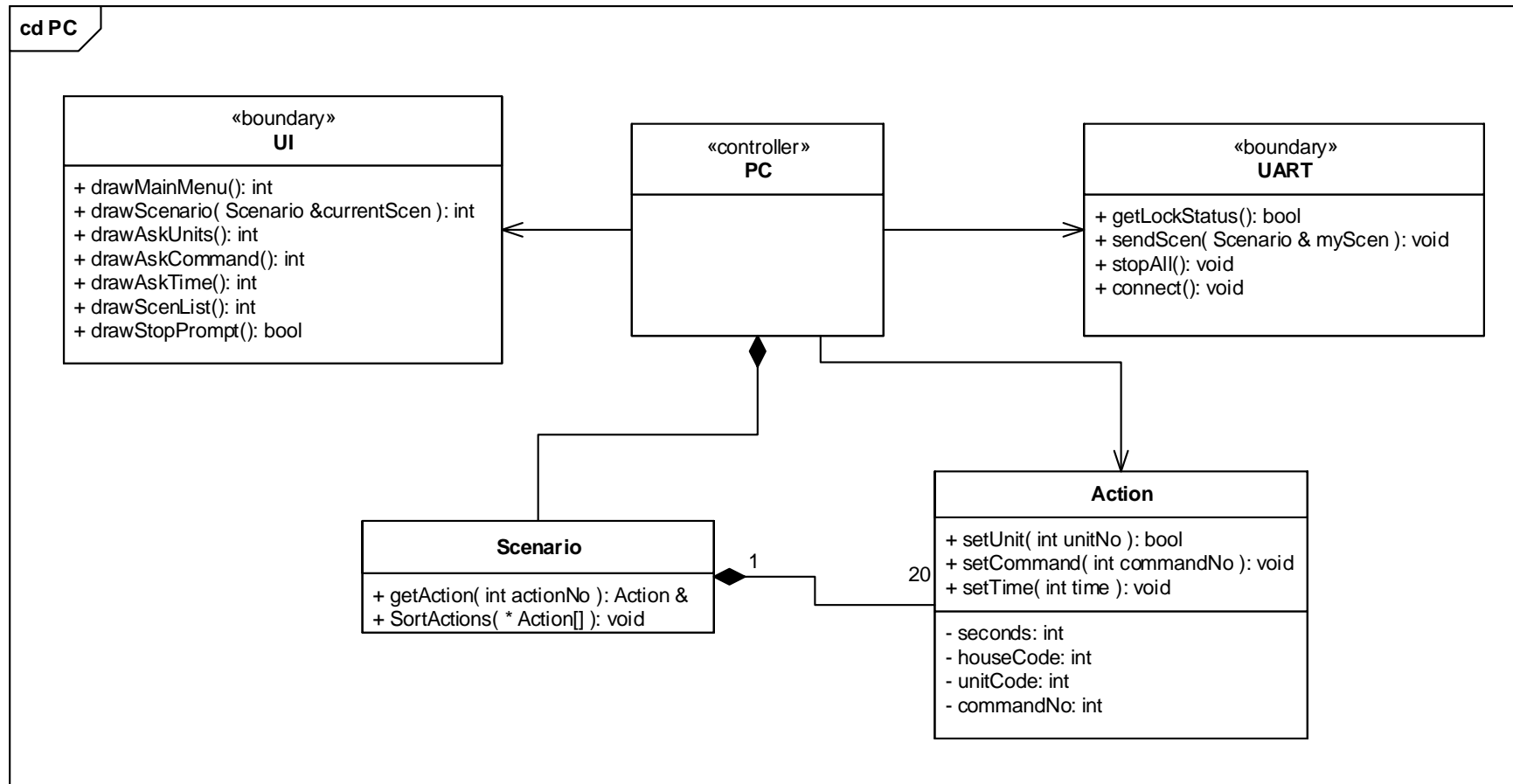
Diagrammet viser sekvenser for PC'en ved gennemgang af hovedscenariet i UC3.



Figur 11: Sekvensdiagram for PC [Use Case 3: Stop Scenarie]

#### 1.4.4 Klassediagram for PC

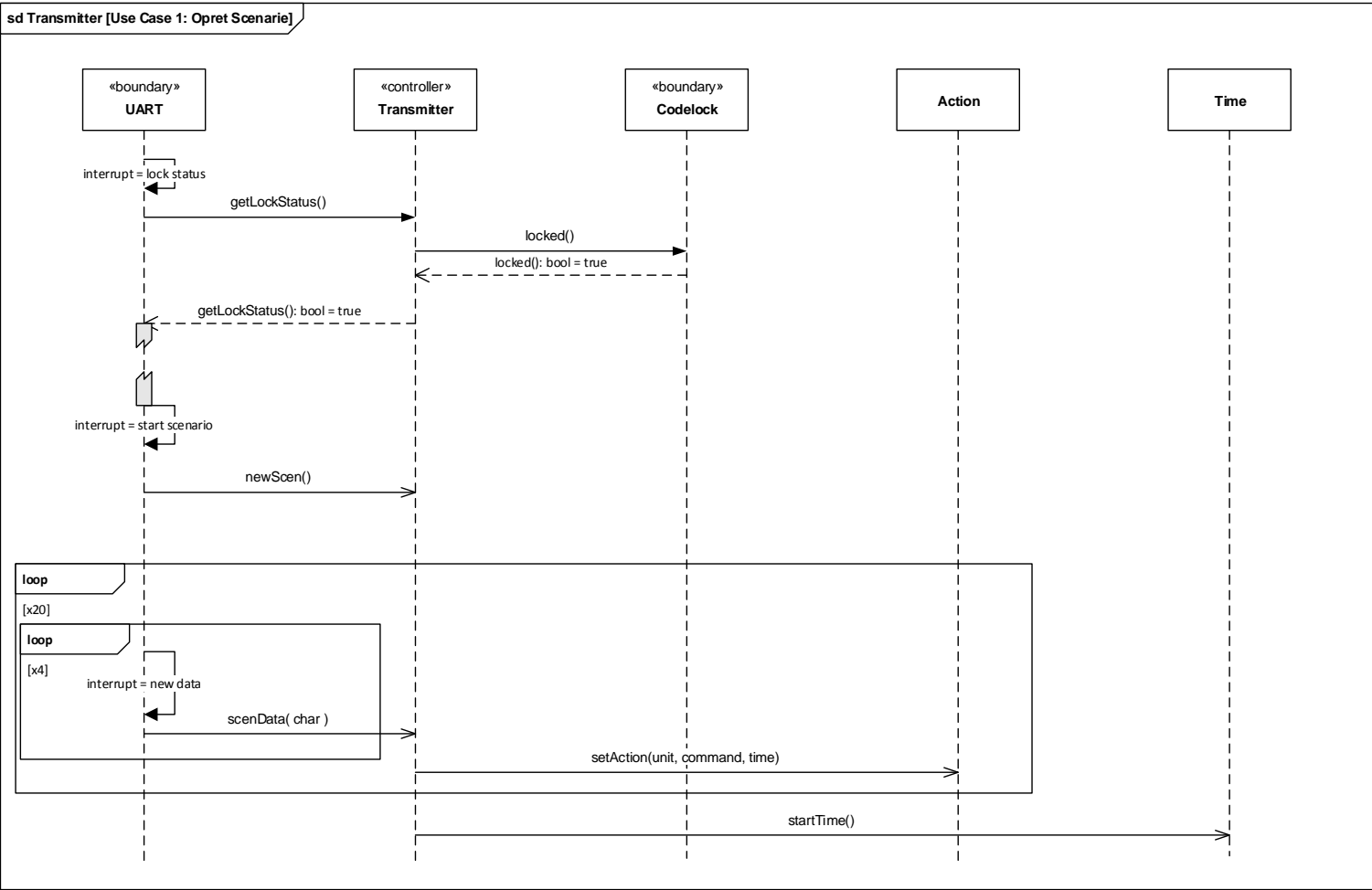
Diagrammet viser klassediagram for den software, der ligger på PC'en med domæne-, controller- og boundary klasser. Boundaryklassen UI har til formål at formidle kommunikation mellem bruger og controller klasssen PC. Boundary klassen UART har ansvar for at kommunikere mellem controller klassen PC og transmitterblokken. Domæneklassen Scenario indeholder op til 20 objekter af domæneklassen Action, der indeholder informationer om aktionen.



Figur 12: Klassediagram for PC.

1.4.5 Sekvensdiagram for transmitter [Use Case 1: Opret Scenarie]

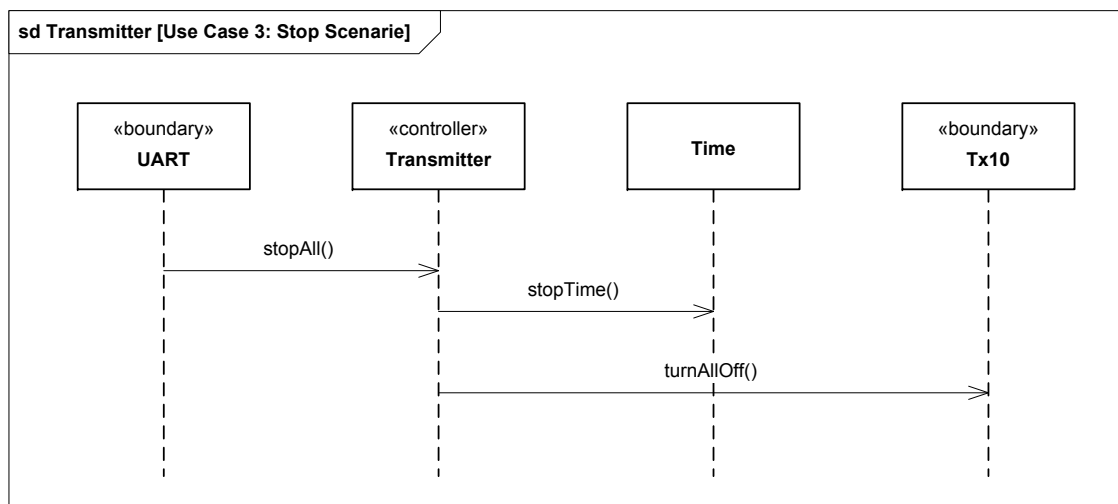
Diagrammet viser sekvenser for transmitteren ved gennemgang af hovedscenariet i UC1.



Figur 13: Sekvensdiagram for transmitter [Use Case 1: Opret Scenarie]

#### 1.4.6 Sekvensdiagram for transmitter [Use Case 3: Stop Scenarie]

Diagrammet viser sekvenser for transmitteren ved gennemgang af hovedscenariet i UC3.

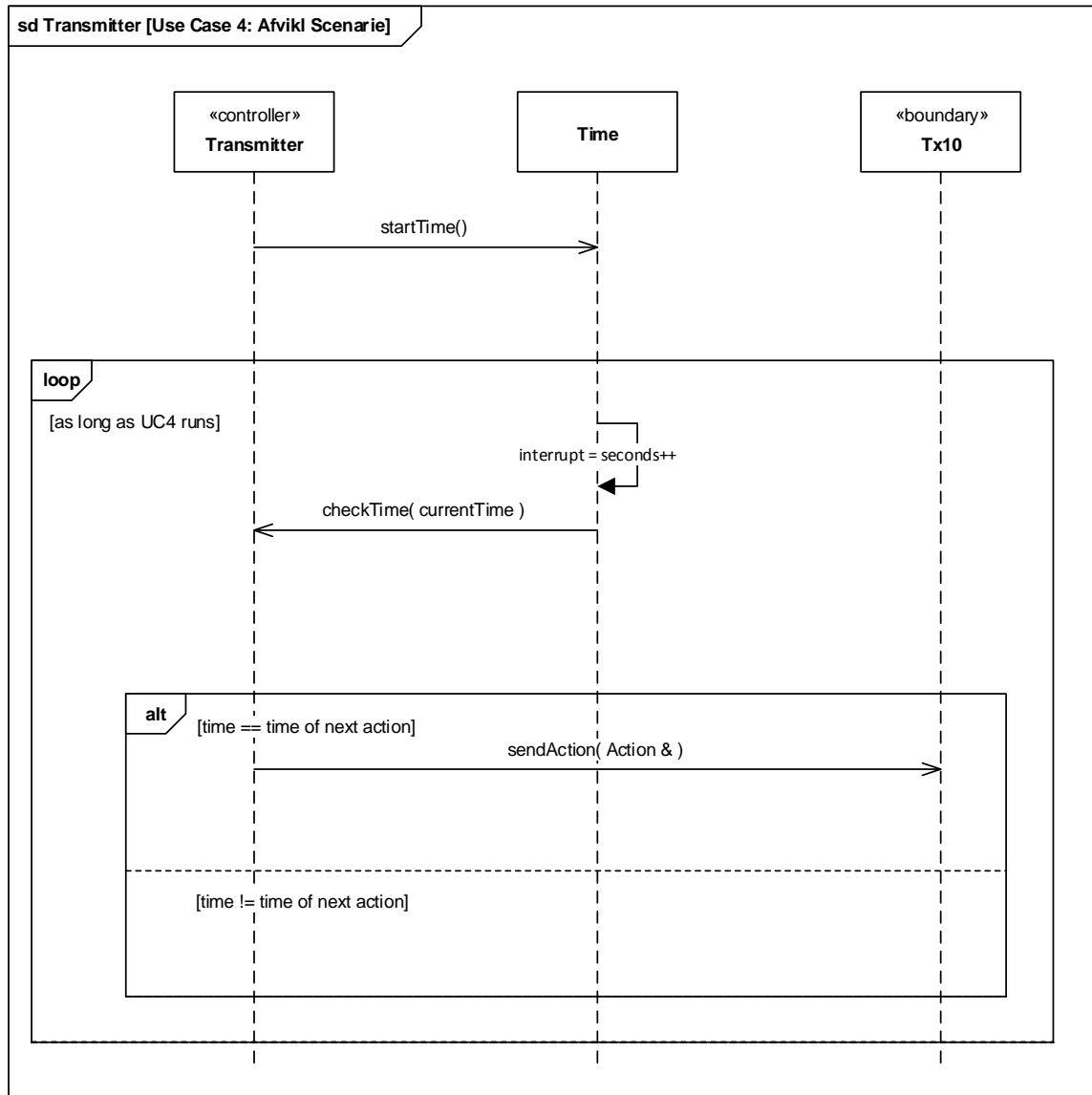


Figur 14: Sekvensdiagram for transmitter [Use Case 3: Stop Scenarie]



### 1.4.7 Sekvensdiagram for transmitter [Use Case 4: Afvikl Scenarie]

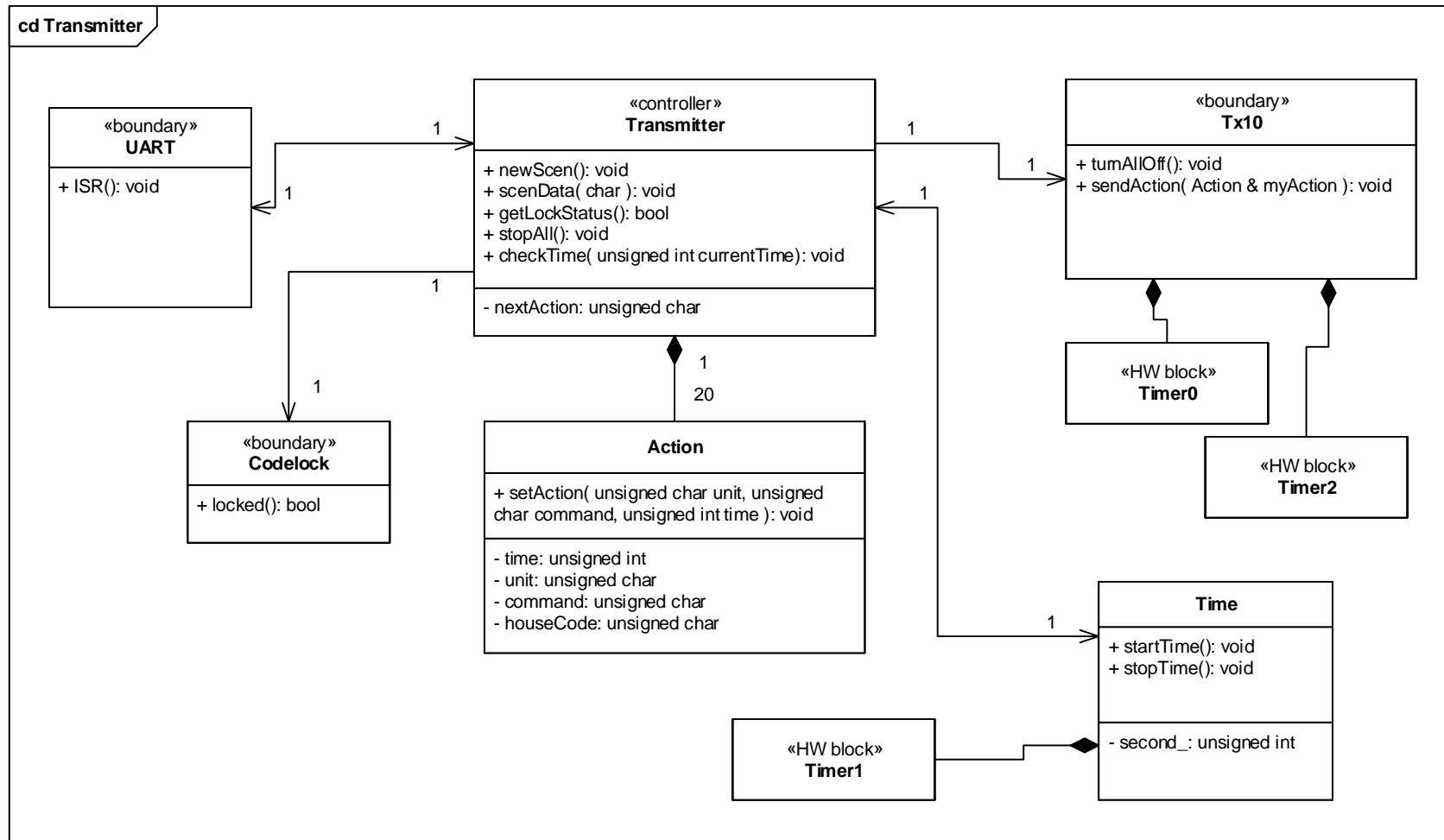
Diagrammet viser sekvenser for transmitteren ved gennemgang af hovedscenariet i UC4.



Figur 15: Sekvensdiagram for transmitter [Use Case 4: Afvikl Scenarie]

### 1.4.8 Klassediagram for transmitter

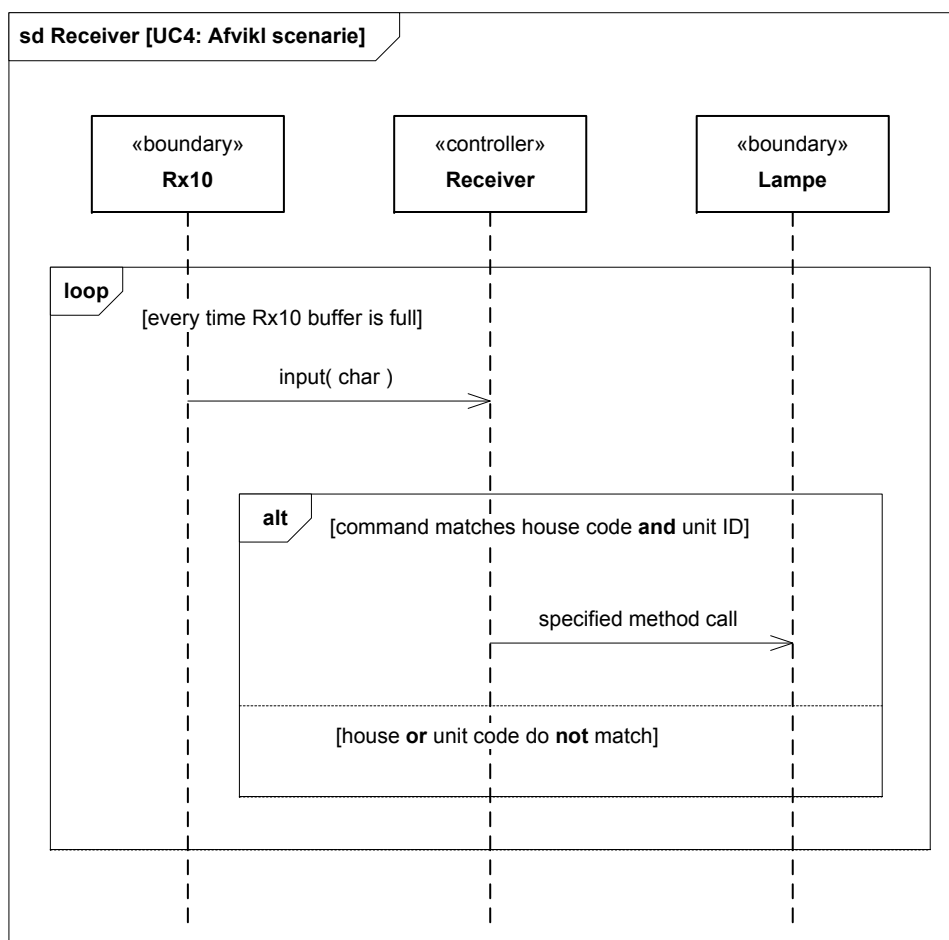
Klassediagram for transmitter.



Figur 16: Klassediagram for transmitter

#### 1.4.9 Sekvensdiagram for receiver [Use Case 4: Afvikl Scenarie]

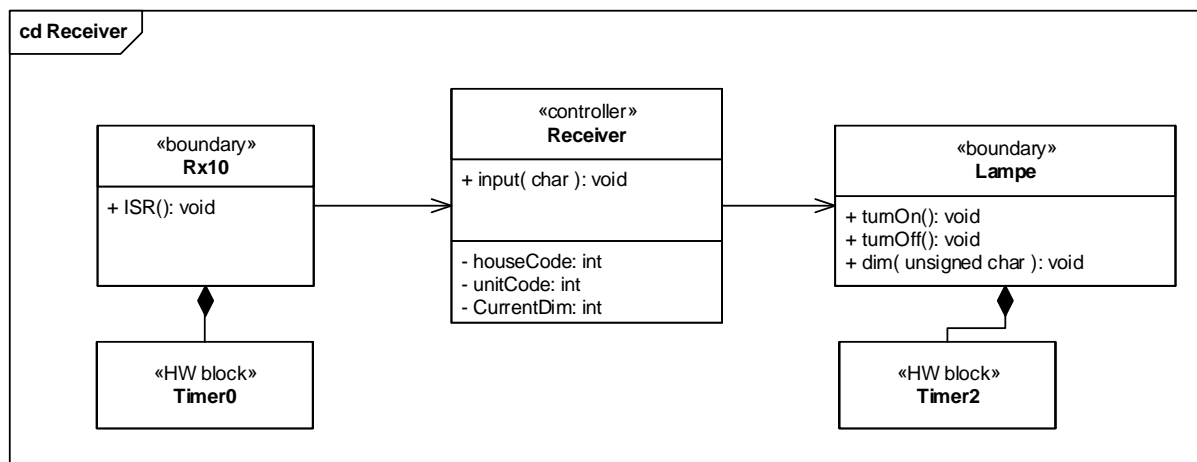
Diagrammet viser sekvenser for receiveren ved gennemgang af hovedscenariet i UC4. Rx10 bufferen kan forklares ved en buffer, der gemmer de seneste 8 nulgennemgange fra X.10. *specified method call* er et kald af en af metoderne i klassen **Lampe**, se Figur 18.



Figur 17: Sekvensdiagram for transmitter [Use Case 4: Afvikl Scenarie]

#### 1.4.10 Klassediagram for receiver

Diagrammet viser klassediagram for den software, der ligger på receiveren med controller- og boundary klasse(r). Boundary klassen Rx10 har til formål at fortolke information fra hardware receiver blokken og gør det tilgængeligt for controller klassen Receiver. Boundary klassen Lampe har til formål at formidle kommunikation mellem controller klassen Receiver og hardware receiver blokken.



Figur 18: Klasse diagram for receiver