

# Remote Ischemic Conditioning System Arkitektur

Simon Vammen Grønbæk  
Karl-Johan Schmidt  
Aarhus University  
Aarhus School of Engineering  
Efteråret 2015

**Titel:**

System Arkitektur

**Godkendelse:****Projekt:**

Remote Ischemic Conditioning

---

Karl-Johan Schmidt**Projektperiode:**

Juli 2015 - December 2015

**Projektgruppe:**

15155

---

Simon Vammen Grønbæk**Deltagere:**Simon Vammen Grønbæk  
Karl-Johan Schmidt

---

Peter Johansen**Vejledere:**

Peter Johansen

**Projektudbyder:**

Rolf Blauenfeldt

---

Rolf Blauenfeldt**Oplagstal: 10****Sidetall: ??****Afsluttet 18-12-2014**

# Indholdsfortegnelse

<b>Kapitel 1</b>	<b>Indledning</b>	<b>5</b>
1.1	Formål . . . . .	5
1.2	Projektreferencer . . . . .	5
1.3	Læsevejledning og dokumentstruktur . . . . .	5
1.4	Definitioner og forkortelser . . . . .	5
<b>Kapitel 2</b>	<b>Systemets dele</b>	<b>7</b>
2.1	Microcontroller . . . . .	7
2.2	Manchetten . . . . .	7
2.3	User interface, knapper og displays . . . . .	7
2.4	Power system . . . . .	7
2.5	Pumpe . . . . .	8
2.6	Ventil . . . . .	8
2.7	Tryksensor . . . . .	8
2.8	SD kort . . . . .	8
2.9	Pulsoximeter . . . . .	8
<b>Kapitel 3</b>	<b>Arkitektur</b>	<b>9</b>
3.1	4+1 view architecture . . . . .	9
3.2	Logic . . . . .	11
3.2.1	Block definition diagram . . . . .	11
3.2.2	Domænemodel . . . . .	11
3.2.3	State machine diagram . . . . .	12
3.2.3.1	Boot . . . . .	12
3.2.3.2	Konditionering . . . . .	13
3.2.3.3	Okklusion . . . . .	14
3.2.3.4	Setup . . . . .	15
3.3	Process . . . . .	16
3.3.1	Sekvensdiagrammer . . . . .	16
3.3.2	Konditionering - UC1 . . . . .	16
3.3.3	Initialiser blodtryksmåling - UC2 . . . . .	17
3.3.4	Mål blodtryk - UC3 . . . . .	18
3.3.5	Overfør data - UC4 . . . . .	19
3.3.6	Sikkerhedskontrol med pulsoximeter - UC5 . . . . .	20
3.3.7	Okklusionstræning - UC6 . . . . .	20
3.3.8	Afbryd - UC7 . . . . .	21
3.3.9	Setup - UC8 . . . . .	22
3.4	Implementation . . . . .	23

3.4.1	Hardware . . . . .	23
3.4.1.1	Beskrivelse af hardware . . . . .	23
3.4.1.2	Arduino Mega og Motor Shield . . . . .	23
3.4.2	Software . . . . .	23
3.4.2.1	Class diagram . . . . .	24
3.4.2.2	Sprog . . . . .	24
3.4.2.3	3-lags modellen . . . . .	24
3.4.2.4	Udviklingsværktøjer . . . . .	24
3.5	Deployment . . . . .	25

# 1 | Indledning

Arkitektur beskrivelsen giver en formel præsentation og forklaring af systemet. Her beskrives hvordan systemet er organiseret, hvilke strukturelle elementer der indgår og hvordan elementer interagerer med hinanden. Der lægges både vægt på software og hardware, samt deres grænseflade. System arkitekturen beskriver hvordan Konditioneringsapparatet skal forstås og hvilke undersystemet det består af.

## 1.1 Formål

System arkitekturen har til formål at beskrive og give forståelse for systemet. Dokumentet fastlægger overordnede softwarekomponentet og hardwarekomponentet, samt strukturen og grænsefladerne mellem disse. Dokumentet udgør en slags plan for, hvordan systemet skal udvikles og hvilke undersystemer det skal bestå af.

## 1.2 Projektreferencer

- Reference til kravspecifikation
- Reference til accepttest

## 1.3 Læsevejledning og dokumentstruktur

Dokumentet ligger sig tæt op af kravspecifikation, da disse krav ligger til grunde for hvad systemet skal kunne. For at give en struktureret gennemgang af system arkitekturen gøres der brug af modellen “*4+1 view architecture*”, der beskriver systemet fra flere forskellige vinkler. Forklaring af modellen kan læses nedenfor. Der gøres som udgangspunkt brugt af SysML til at beskrive systemet. Alt SysML udvikles og skrives på engelsk

## 1.4 Definitioner og forkortelser

Udtryk / Forkortelse	Forklaring
UML	Unified Modeling Language, sprog til forklaring af software arkitektur
SysML	System Modeling Language, sprog til forklaring af system arkitektur
PWM	Pulse-width modulation

Modeswitch	Knap til at styre hvilket program Konditioneringsapparatet skal køre
------------	--

## 2 | Systemets dele

Dette afsnit beskriver systemet, *Konditioneringsapparats*, fysiske dele og deres funktionalitet

### 2.1 Microcontroller

Styring af alle systemets dele. Her processerer brugeren interagering med *Konditioneringsapparat* og handlingen eksekveres. Microcontrolleren er en AtMega32 og styringen af chippen skrives i C++.

### 2.2 Manchetten

Trykmanchet til at skabe okklusion af armen. Manchetten skal kunne holde trykket, som skabes af pumpen. Manchetten kobles til apparatet via en lufttæt slange.

### 2.3 User interface, knapper og displays

Brugerfladen består af et display hvor blodtryk, antal okklusioner, resterende tid og mm. vises. Displayet skal bruges til at give brugeren feedback og fx. informere det medicinske personale hvor lang tid der er indtil konditioneringen er færdig.

På *Konditioneringsapparatet* er der to knapper [Start/Stop] og [Mål blodtryk]. Disse knapper bruges til at initierer konditioneringsbehandling, blodtryksmålinger og okklusionstræning. På bagsiden af apparatet sidder desuden en Modeswitch, hvor brugeren kan skifte mellem *Okklusionstræning*, *Konditionering*, eller *Setup*.

### 2.4 Power system

Forsyning af systemet foregår med 8 batterier af typen AAA for at opnå en spænding på 12V. Systemet af forsynes med et batteri løsning for at gøre det mere mobilt.

Foruden at forsyne apparatet, er power system også bestående af et motor shield. Når microcontroller fx ønsker at starte pumpen, sørge motorshieldet for at levere det korrekte spænding.

## 2.5 Pumpe

Består en motor og en luftindtag. Pumpe kan både bruges til at skabe tryk og vakuum. Pumpe skal bruges til at inflatere manchetten til måling af blodtryk og til okklusion af armen, både under konditionering og under træning. Pumpen skal forsynes med 12 V og hastigheden kan styres med PWM.

## 2.6 Ventil

Ventil indgår i systemet til at nedregulere trykket i manchetten. Ventilen er “Normally closed”, det vil sige at ventilen først åbnes når den påtrykkes en spænding. Funktionen af ventilen under en blodtryksmåling er gradvis at lukke trykket ud, så det er muligt at registrere oscillationerne og det aktuelle tryk. Under okklusion har ventilen en anden funktion, her indgår ventilen i reguleringen.

## 2.7 Tryksensor

En 12 V tryksensor der bruges til registrering af trykket i manchetten og til efter regulering. Tryksensor skal også registrere oscillationerne der skabes i manchetten når trykket er omkring systolisk niveau og ved middeltrykket. Ved okklusionstræning skal tryksensor bruges til at holde trykket konstant omkring 100 mmHg

## 2.8 SD kort

Apparatet udstyres med ekstern hukommelse, for at det er muligt for *Konditioneringsapparatet* at gemme information omkring behandlingsforløbet. Der er valgt et SD kort, fordi når behandlingen er færdig, er det muligt at skifte SD kortet ud, og på den måde have backup af information og det er nemmere at overføre informationen.

## 2.9 Pulsoximeter

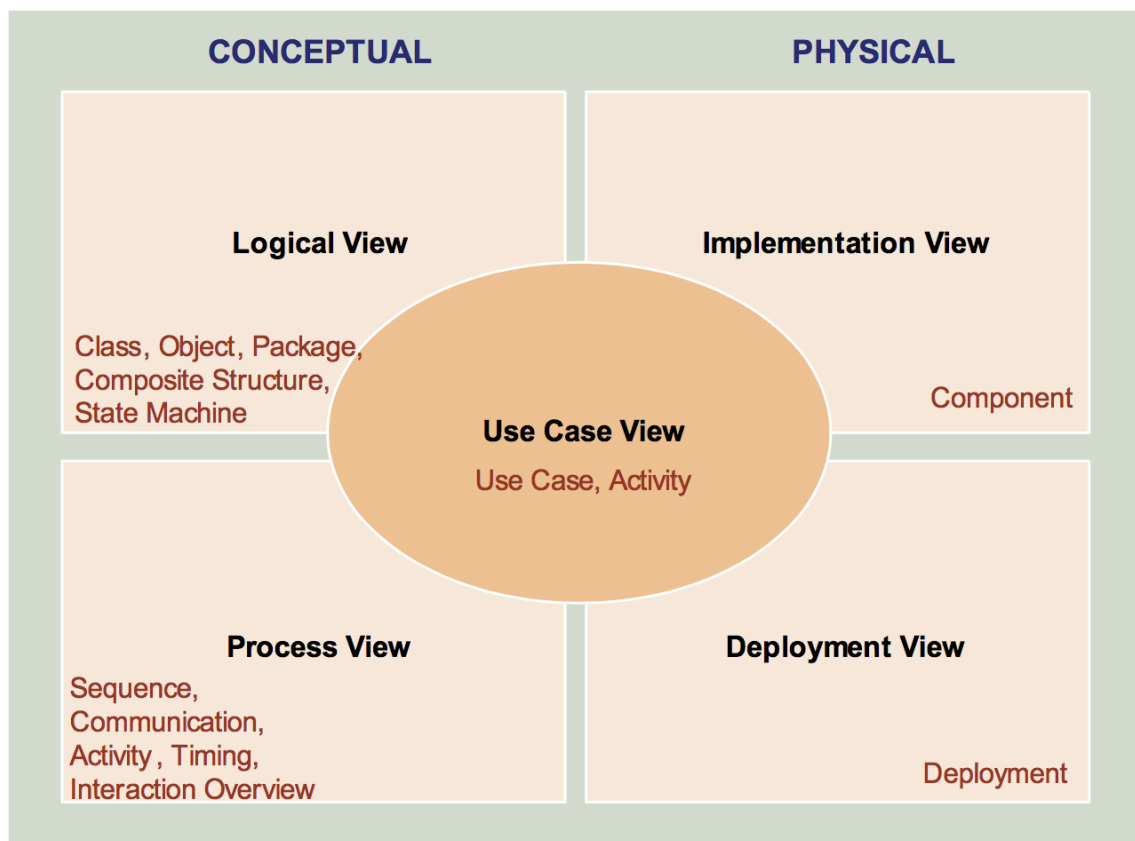
Som undersystemet i *Konditioneringsapparatet* indgår et pulsoximeter, der skal bruges til overvågning af patientens tilstand under konditioneringsbehandling. Pulsoximeteret leverer en saturation efter hver endt okklusion og den saturation er med til at bestemme om patientens kredsløb kan tåle behandlingen.



## 3 | Arkitektur

### 3.1 4+1 view architecture

Denne model beskriver arkitekturen af software baserede systemer. For at skabe en fyldestgørende gennemgang af systemet gøres brug af fire forskellige synsvinkler. Disse synsvinkler til for at tilfredsstille alle interessenter og sørge for at alle parter forstår systemet. Eksempler på parter kunne være kunden, projektleder eller udviklere. Med udgangspunkt i use cases består modellen af følgende punkter:



*Figur 3.1.* 4+1 view architecture model

*Logical view:* Denne synsvinkel beskriver systemets funktionalitet via centrale elementer, mekanismer og stadier.

*Process view:* Beskæftiger sig med den ikke funktionelle del af systemet, og hvordan de centrale elementer fra logical view interagerer med hinanden.

*Implementation view:* Den vinkel involvere udviklerens perspektiv og beskæftiger sig med hvordan software implementeres

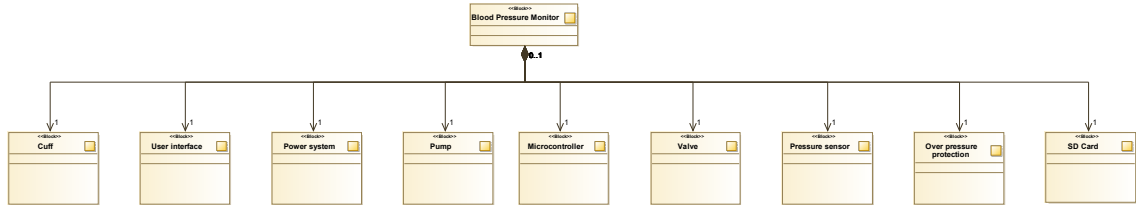
*Deployment view:* Beskriver systemet fra en fysisk synsvinkel, hvordan eksekveres softwares på de brugtes devices, hvordan systemets fysisk setup ser ud

Modellen “*4+1 view architecture*” er beregnes primært til software baserede udviklingsprojekter og derfor bruges den som en retningslinje og inspiration til systemet arkitekturen. Da Konditioneringsapparatet er en prototype som involvere både hardware og software er modellen blevet tilpasset dertil. Endvidere vil systemet blive præsenteret og gennemgået ved hjælp af SysML standarden, selvom modellen er lavet til UML

## 3.2 Logic

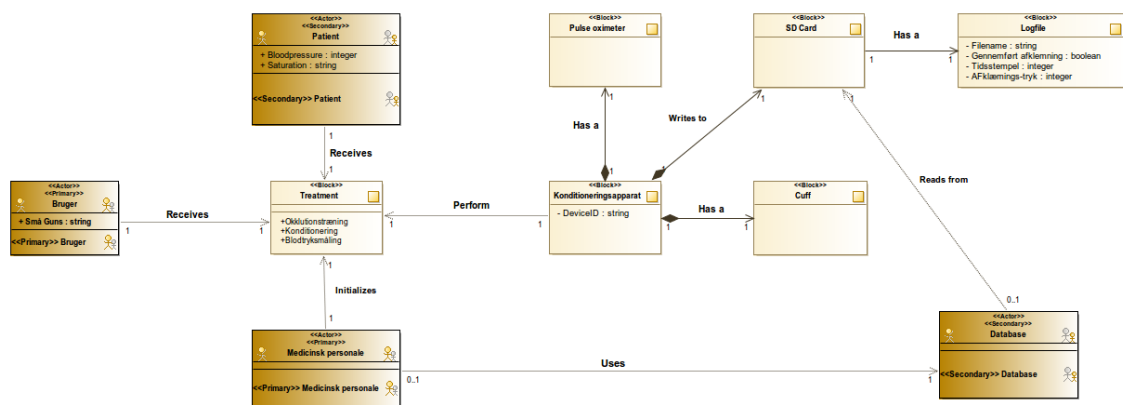
### 3.2.1 Block definition diagram

Blokdiagrammer giver et indblik på den overordnede strukturen af *Konditioneringsapparatet*. Hver kasse skal ses som en del der indgår i systemet



### 3.2.2 Domænemodel

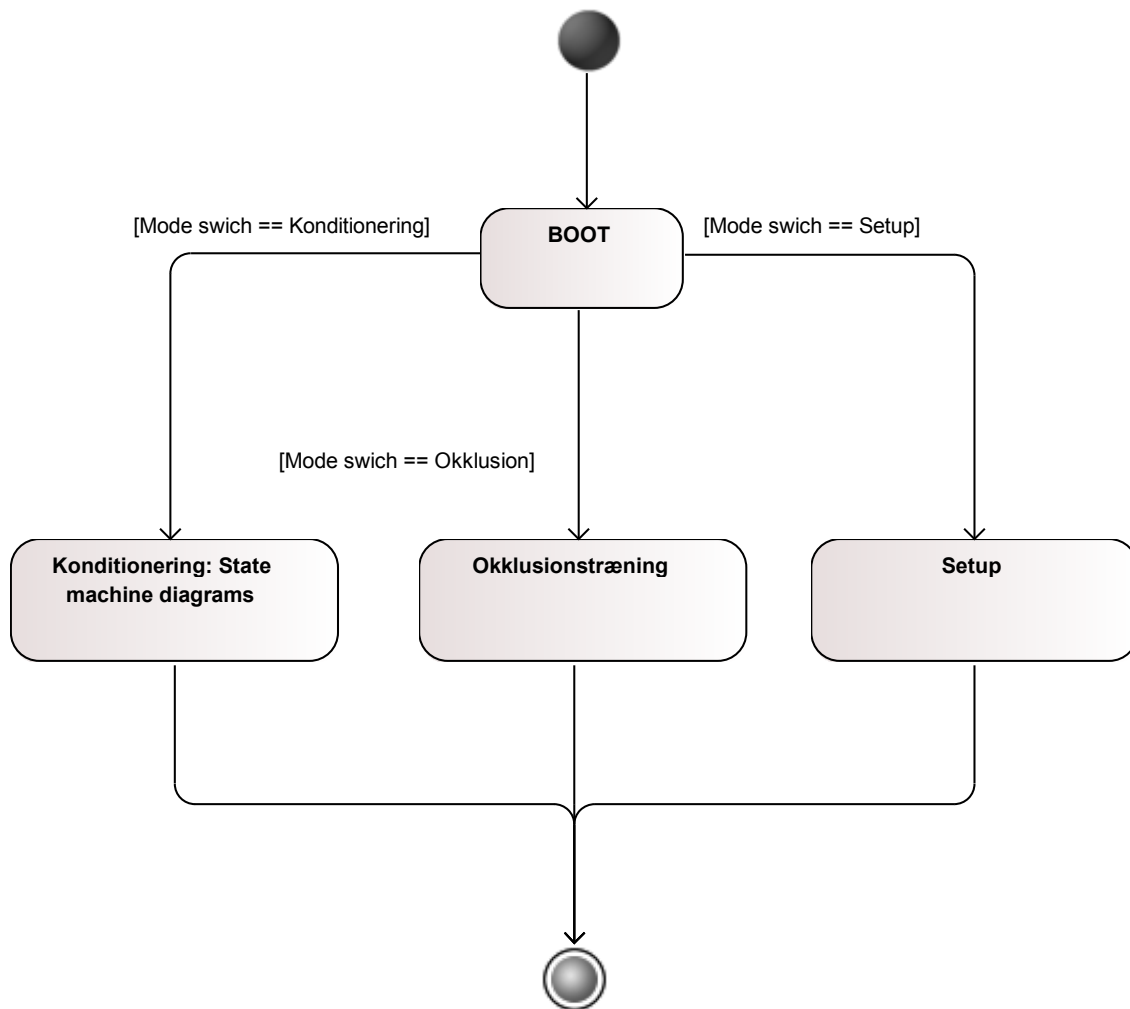
Diagrammer beskriver det systemet som helhed. Ved gennemgang af alle use cases findes væsentlig navneord og disse oprettet som konceptuelle klasser. Det konceptuelle klasser er derefter oversat til engelsk



Figur 3.2. Block definition diagram over *Konditioneringsapparatet*

### 3.2.3 State machine diagram

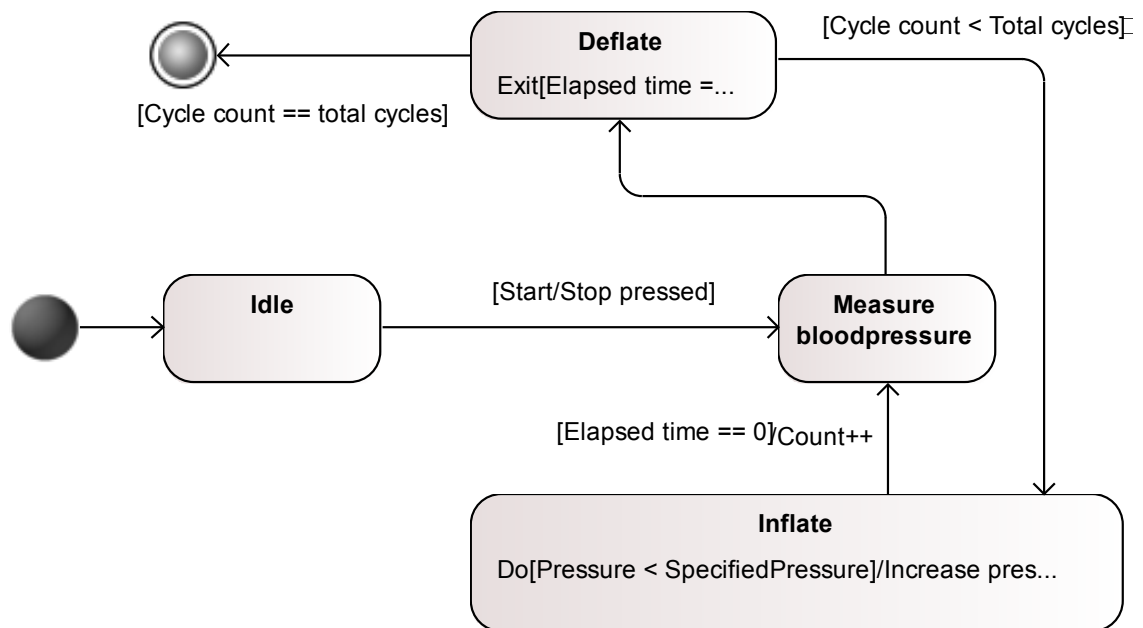
#### 3.2.3.1 Boot



*Figur 3.3.*

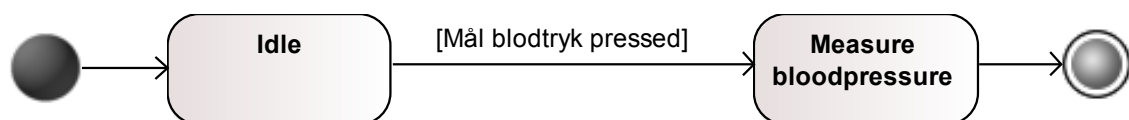
### 3.2.3.2 Konditionering

Ved knap tryk på [Start/Stop]



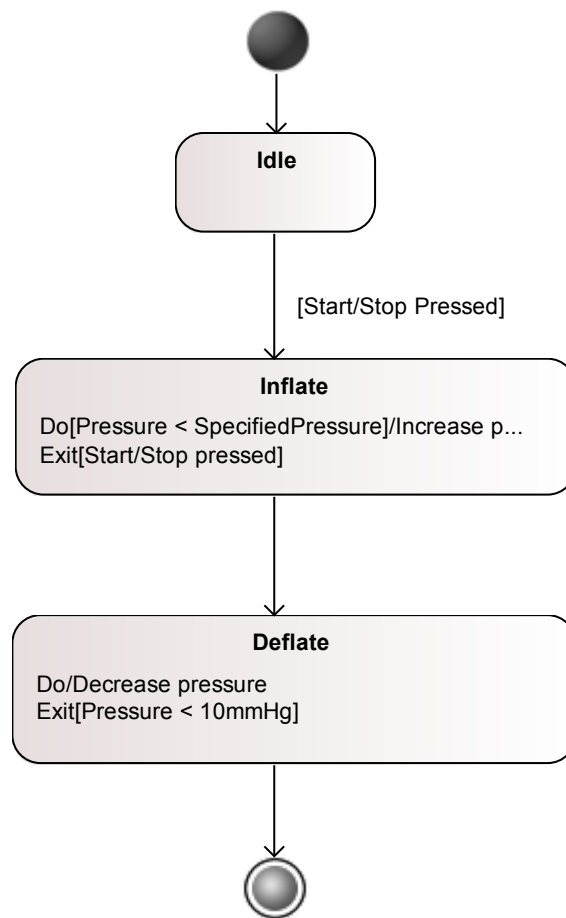
**Figur 3.4.** State machine diagram over Konditioneringsforløb

Ved knap tryk på [Mål blodtryk]



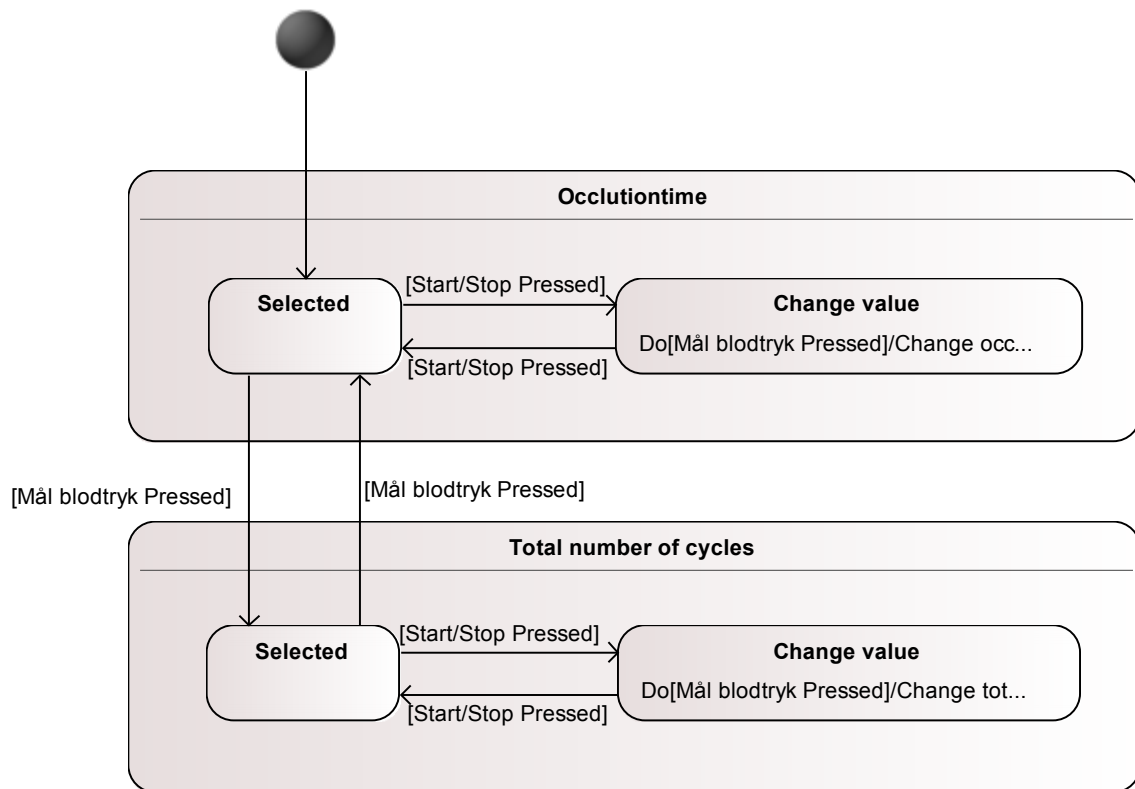
**Figur 3.5.** State machine diagram over blodtryksmåling

### 3.2.3.3 Okklusion



*Figur 3.6.* State machine diagram over okklusionsforløb

## 3.2.3.4 Setup

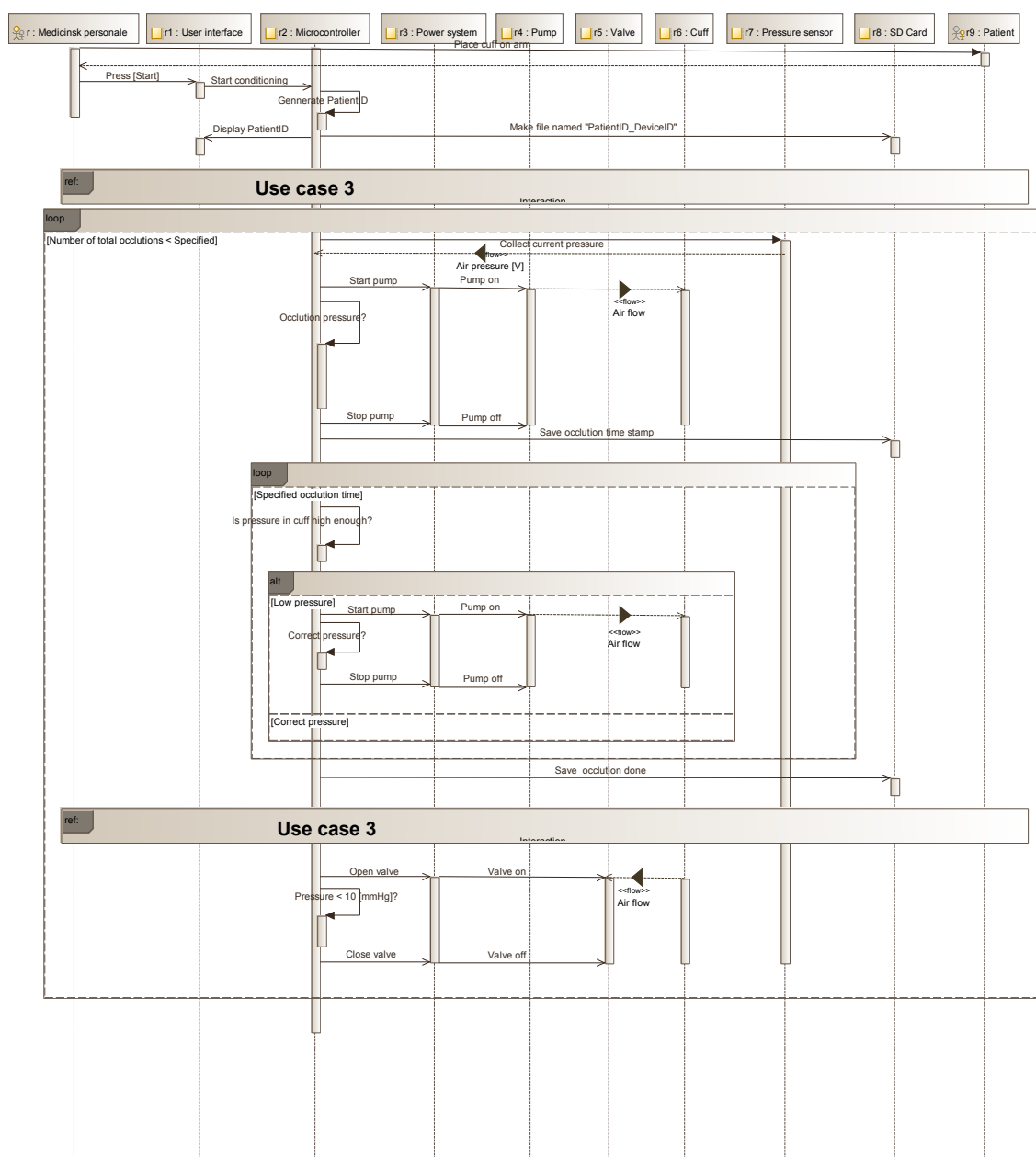
*Figur 3.7.* State machine diagram over setup forløbet

### 3.3 Process

#### 3.3.1 Sekvensdiagrammer

Der er udarbejdet et sekvensdiagram for hver use case. Et sekvensdiagram viser hvordan systemets dele og aktører interagerer med hinanden, og hvilke processer der sker ved disse interaction. Det er beskrevet som sekventiel process og der illustreret diagrammet også hvilke rækkefølge processerne skal eksekveres i. Fordi at simplificeret store sekvensdiagrammer gør nogle af dem brug af andre use case, dette ses fx. i sekvensdiagrammet for use case 1.

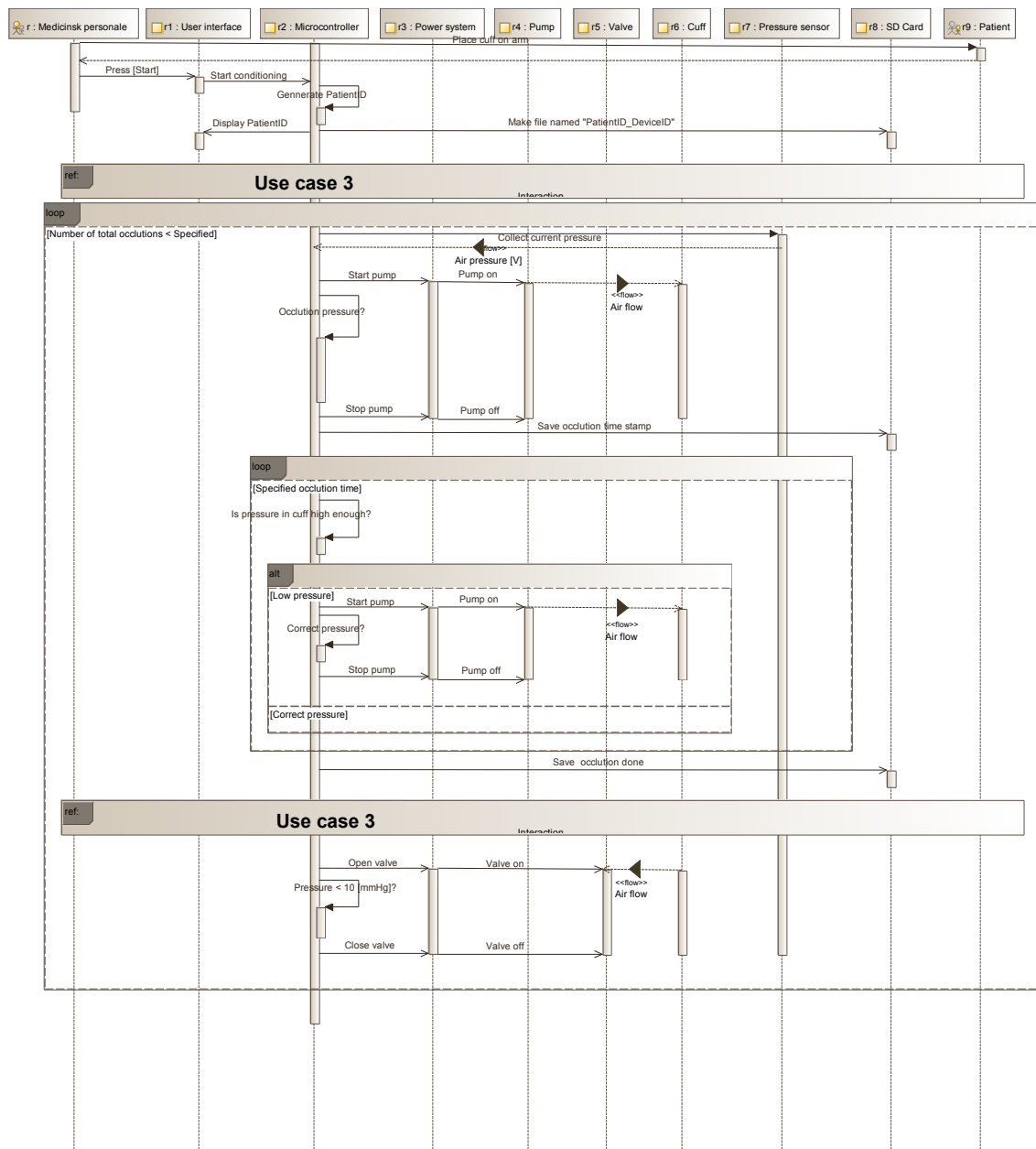
#### 3.3.2 Konditionering - UC1



Figur 3.8. Sekvens diagram over forløb i use case 1

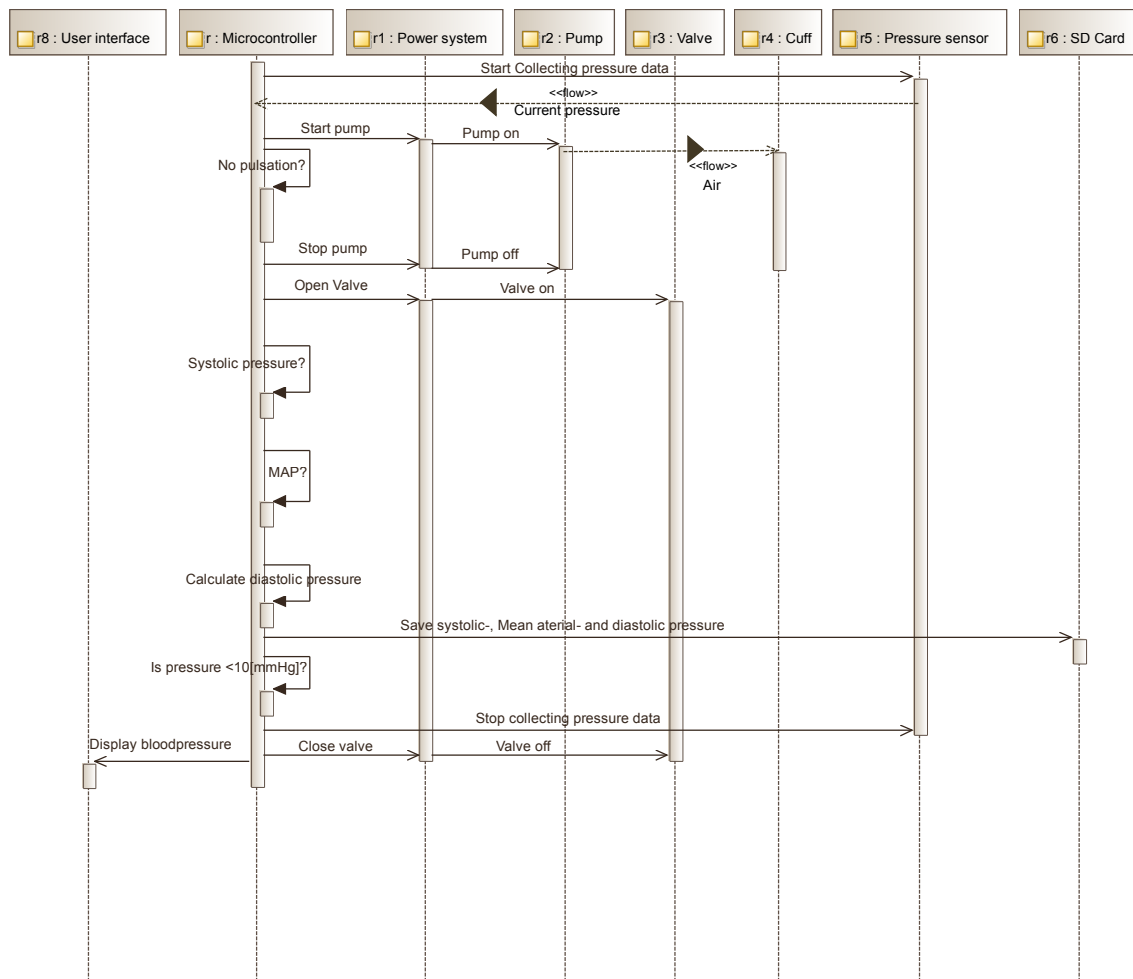


## 3.3.3 Initialiser blodtryksmåling - UC2



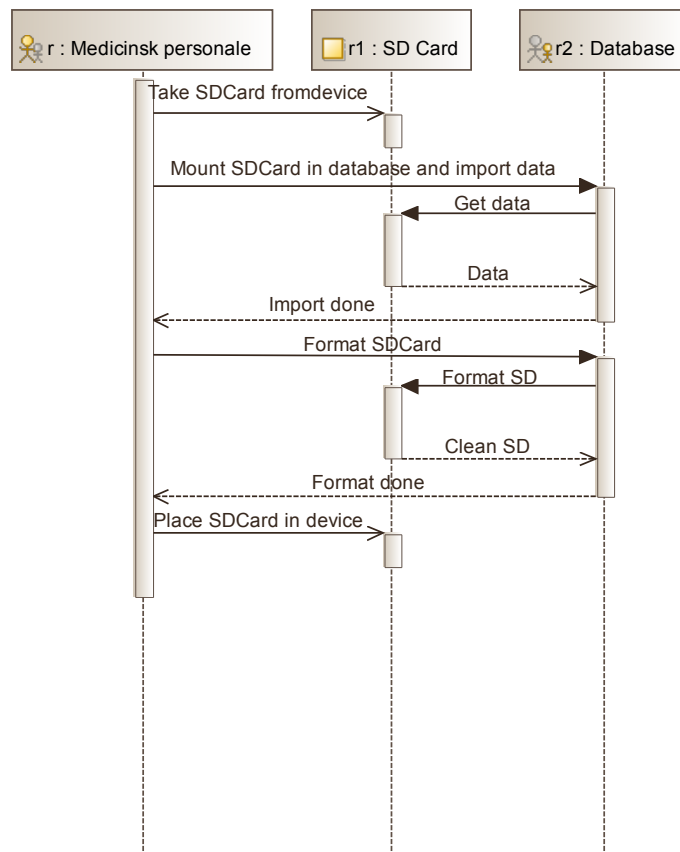
Figur 3.9. Sekvens diagram over forløb i use case 2

### 3.3.4 Mål blodtryk - UC3



**Figur 3.10.** Sekvens diagram over forløb i use case 3

## 3.3.5 Overfør data - UC4

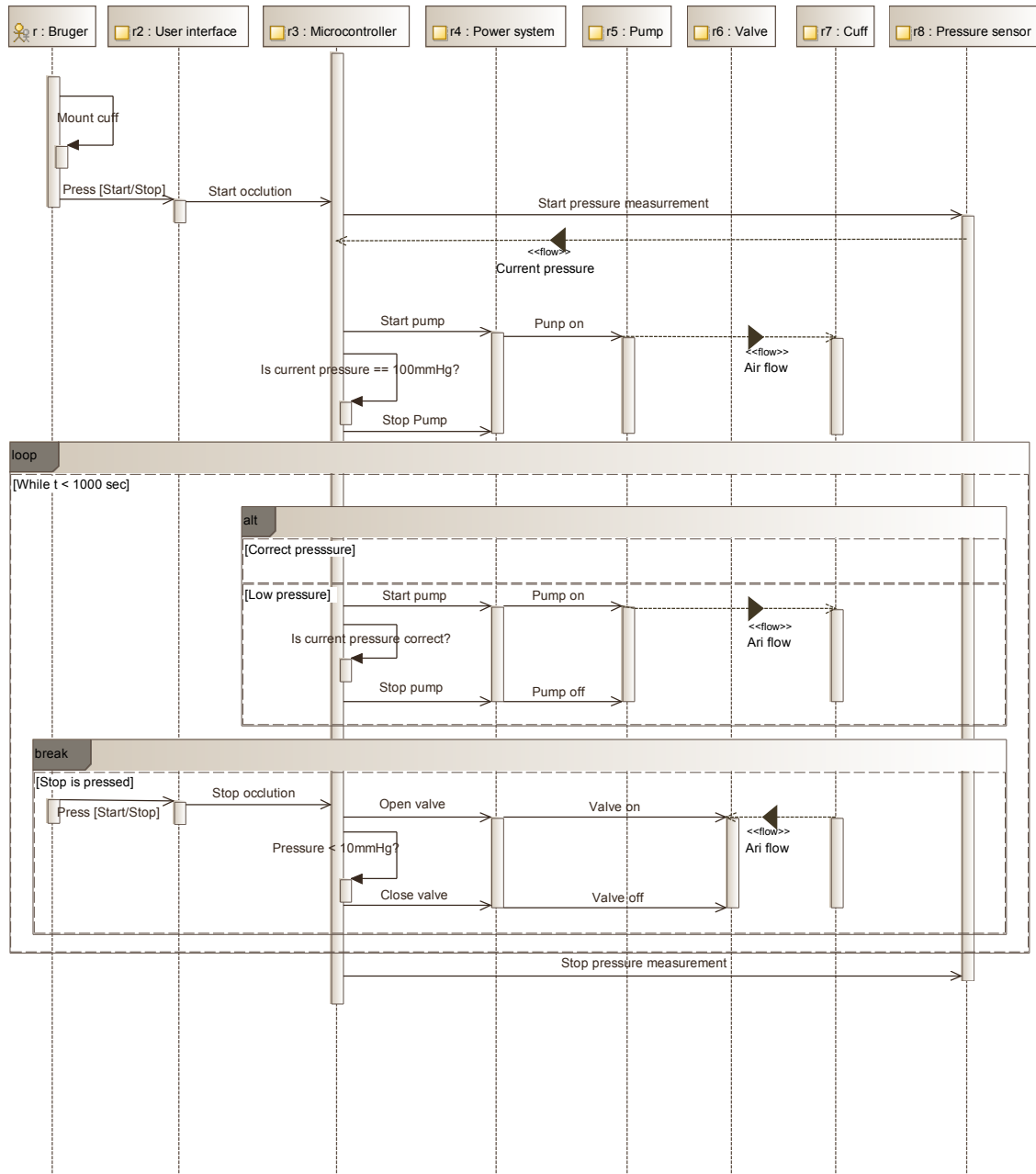


*Figur 3.11.* Sekvens diagram over forløb i use case 4

### 3.3.6 Sikkerhedskontrol med pulsoximeter - UC5

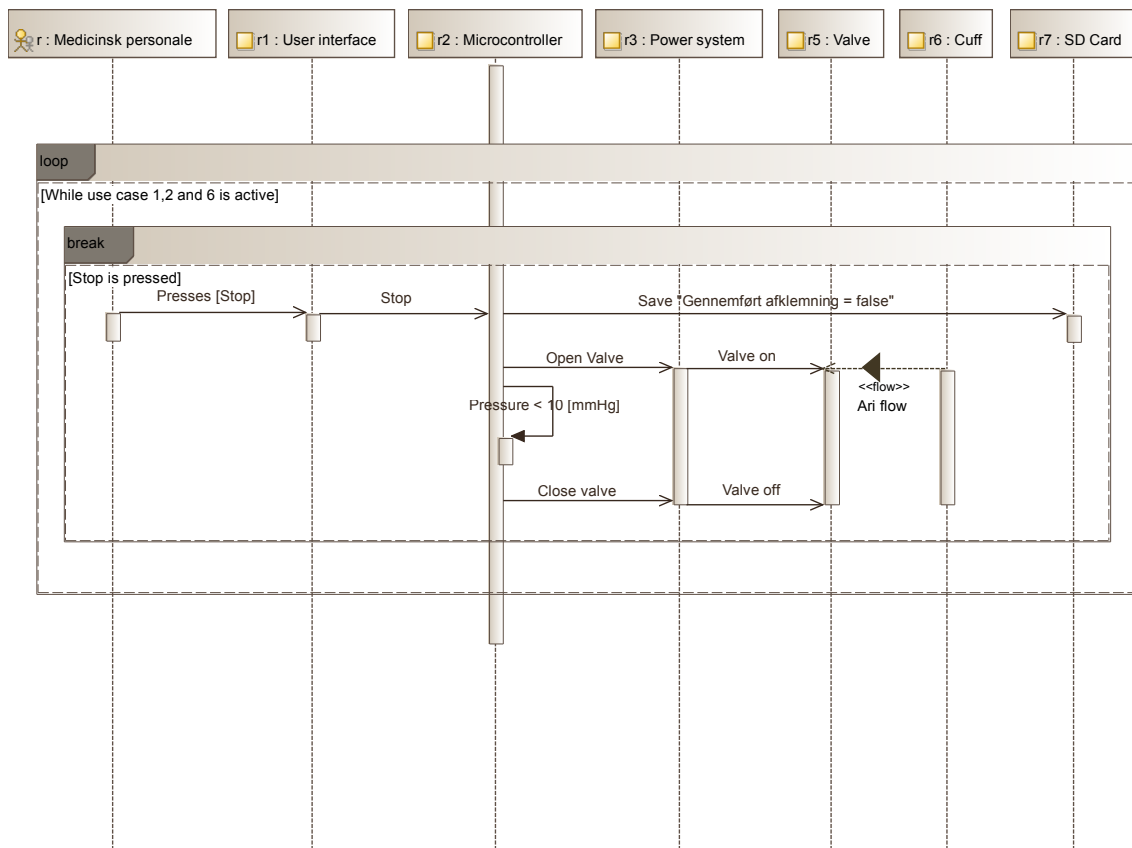
Mangler stadig...

### 3.3.7 Okklusionstræning - UC6



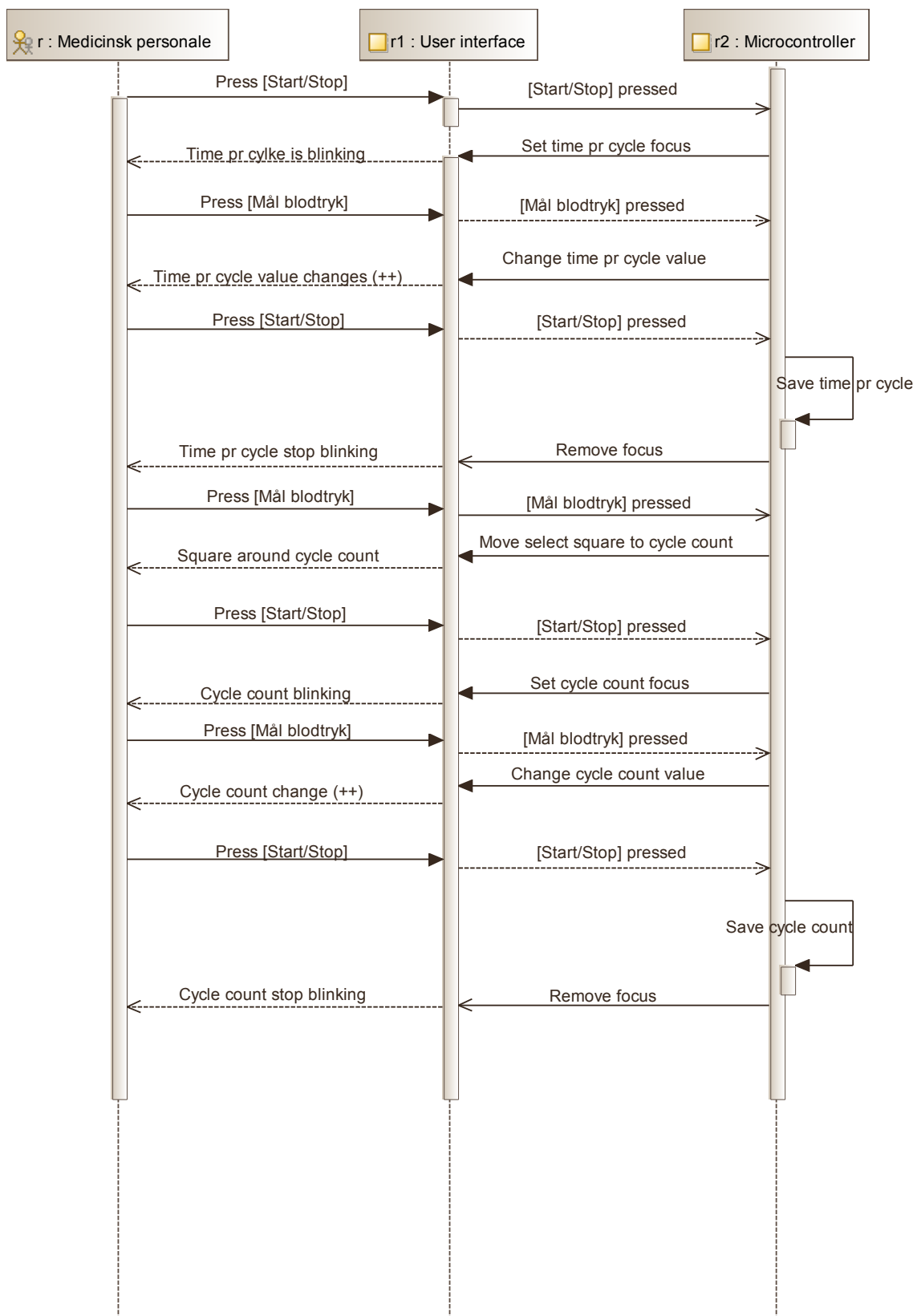
**Figur 3.12.** Sekvens diagram over forløb i use case 6

## 3.3.8 Afbryd - UC7



**Figur 3.13.** Sekvens diagram over forløb i use case 7

## 3.3.9 Setup - UC8

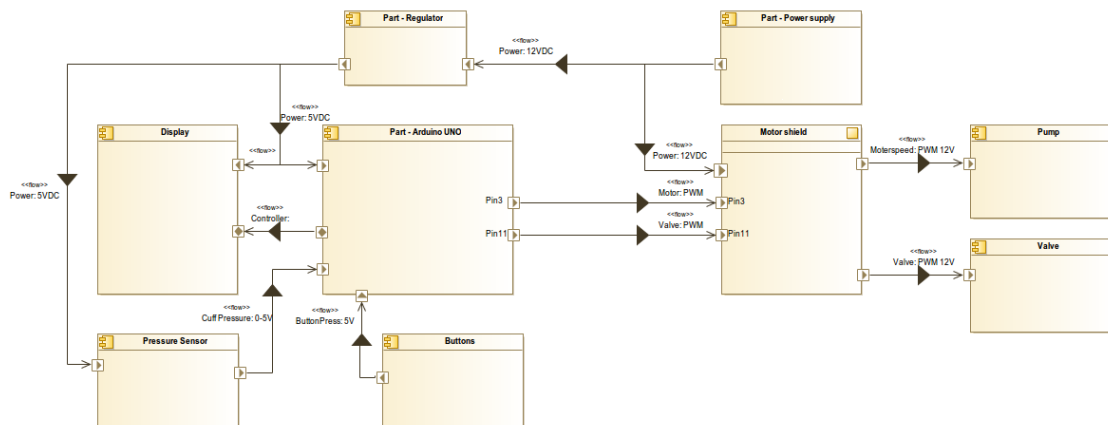


Figur 3.14. Sekvens diagram over forløb i use case 8

## 3.4 Implementation

### 3.4.1 Hardware

Diagrammet viser interaktioner mellem systemets hardware dele



**Figur 3.15.** Internal block diagram over flow for hardwaren i *Konditioneringsapparatet*

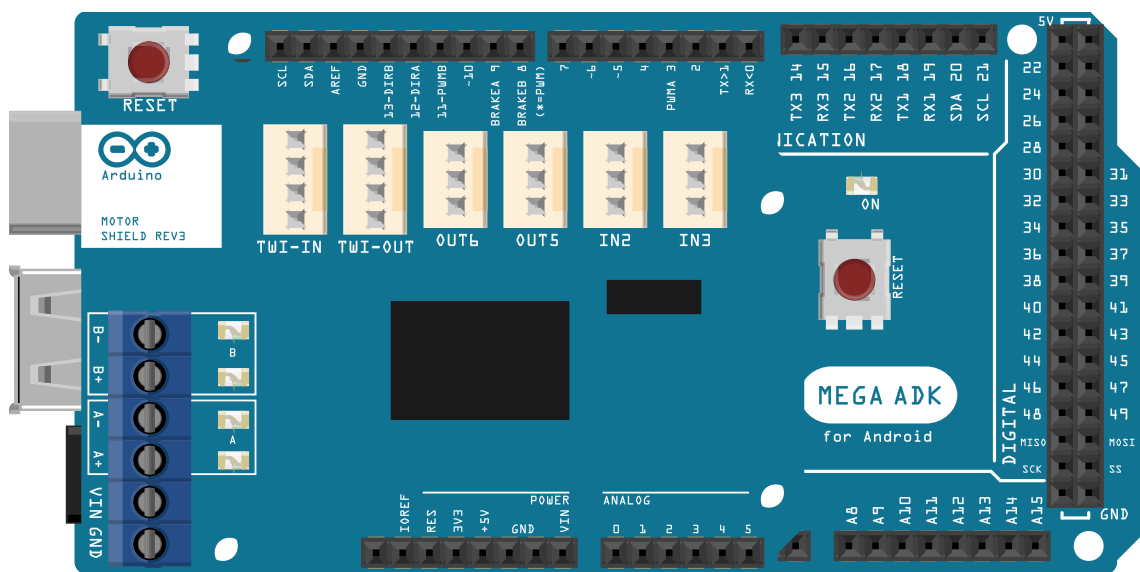
(Skal rettes til PDF)

#### 3.4.1.1 Beskrivelse af hardware

INDSÆT reference til kapitel 2

#### 3.4.1.2 Arduino Mega og Motor Shield

Til udvikling af prototypen bruges en arduino Mega og et 12V motor shield. Se oversigt tegnenfor



**Figur 3.16.** Arduino Mega med 12V motor shield monteret

### 3.4.2 Software

*Konditioneringsapparatet* består kun af ét software system og det kræver ikke flere software systemer for at apparatet fungere. Dette giver derfor et simpelt struktur for softwaren

#### 3.4.2.1 Class diagram

Diagrammer præsenterer software klasser med funktioner og de er opdelt i hver sit namespace. Denne struktur er valgt for at opnå høj samhørighed og lav kobling.



*Figur 3.17.* Class diagram for softwaren i *Konditioneringsapparatet*

#### 3.4.2.2 Sprog

- C++ - brugt til at skrive source code til arduino
- SysML - brugt til udvikling af system diagrammer

#### 3.4.2.3 3-lags modellen

Softwaren er struktureret efter 3-lags princippet. Dette design er valgt for at give klare grænseflader og ansvarsfordeling. Det tre lags er presentationslaget, logik laget og datalaget. Lagdelingen giver stor fleksibilitet fordi det er nemmere at vedligeholde og genbruge kode fra et bestemt lag uden at den influere med andre lag. De tre lag er repræsenteret som hvert sit namespace i software arkitekturen



#### 3.4.2.4 Udviklingsværktøjer

##### Eclipse

IDE til udvikling af C og C++. Dette miljø bruges til at skrive koden til arduinoen og dermed styringen af prototypen. Der er valgt versionen: “Juno Service Release”. Denne version understøtter integration af Arduino’ eget IDE, samtidig med at man kan gøre brug af Eclipses programmerings funktioner.

##### Arduino IDE

Arduinos eget udviklingsmiljø bruges som et plugin via Eclipse. Den brugte version er 1.5.5. Grundet manglende funktion er det blevet fravalgt at bruge Arduino IDE alene.

##### Git, GitHub Desktop og SmartGit

Til versionsstyring af til projekt dokumentation og source code. Bachelor gruppen har købt et privat repository grundede den igangværende patentsag. Som bruger interface for git er blevet brugt henholdsvis GitHub Desktop og SmartGit

##### Eksterne biblioteker

Arduino biblioteker

- EEPROM - muliggøre operationer på arduinoen indbyggede hukommelse
- SD - muliggøre operationer med SD kort
- TFT - funktioner til at bruge tft displays til arduino

## 3.5 Deployment

Da produktet Konditioneringsapparat er en prototype beskæftiger projektet sig ikke med de redskaber der bruges i deployment view. Dette view beskriver blandt andet hvordan software mappes på hardware, kaldet et deployment diagram. Konditioneringsapparatet består kun af én software eksekverende enhed og derfor er dette overflødigt at beskrive.