



3. SEMESTER PROJEKT  
RAPPORT

---

UDVIKLING AF ET BLODTRYKMÅLESYSTEM  
Projektrapport

---

Navn	AU ID	Studienummer
Caroline Kaagaard Dahl Laursen	AU572444	201611025
Nicolai Bæch	AU580049	201704646
Mathias Egsgaard	AU590400	201705031
Thea Plenus Kjeldahl Kristensen	AU577124	201707180
Sarah Krohn Fenger	AU577425	201707931
Mikkel Rugholm Boisen	AU578833	201708119
Kajene Elankanathan	AU594051	201710472

VEJLEDER: SAMUEL ALBERG THRYSØE

DATO: 19/12-2018

ANTAL SIDER: XX

# Resumé

# Abstract

# Indholdsfortegnelse

	Side
<b>Kapitel 1 Ordliste</b>	<b>6</b>
<b>Kapitel 2 Indledning</b>	<b>7</b>
<b>Kapitel 3 Projektformulering</b>	<b>8</b>
3.1 Problemformulering . . . . .	8
<b>Kapitel 4 Teori</b>	<b>9</b>
4.1 Fysiologi . . . . .	9
4.2 Blodtryksmåling . . . . .	9
<b>Kapitel 5 Systembeskrivelse</b>	<b>10</b>
5.1 Systemoversigt . . . . .	10
5.2 Systembeskrivelse . . . . .	10
5.2.1 Alarmer . . . . .	11
<b>Kapitel 6 Kravspecifikation</b>	<b>12</b>
6.1 Krav . . . . .	12
6.2 Use Case diagram . . . . .	13
6.3 Aktørbeskrivelser . . . . .	13
6.4 Use Case beskrivelser . . . . .	14
6.4.1 Use Case 1: Mål og vis blodtryk og puls . . . . .	14
6.4.2 Use Case 2: Justér grænseværdier . . . . .	14
6.4.3 Use Case 3: Alarmering . . . . .	14
6.4.4 Use Case 4: Gem data . . . . .	14
6.4.5 Use Case 5: Kalibrering . . . . .	14
6.5 Afgrænsning . . . . .	15
6.5.1 Ikke-funktionelle krav . . . . .	15
<b>Kapitel 7 Arkitektur og design</b>	<b>16</b>
7.1 Hardwarearkitektur . . . . .	16
7.1.1 BDD . . . . .	16
7.1.2 IBD . . . . .	16
7.2 Hardwaredesign . . . . .	16
7.2.1 Forstærker . . . . .	16
7.2.2 Anti-aliaserings filter . . . . .	16
7.2.3 Printplade . . . . .	16
7.3 Softwarearkitektur . . . . .	16
7.3.1 Domænemodel . . . . .	16

7.4	Softwaredesign . . . . .	16
<b>Kapitel 8</b>	<b>Implementering og test</b>	<b>18</b>
8.1	Hardwareimplementering . . . . .	18
8.2	Test af hardware . . . . .	18
8.2.1	Forstærker . . . . .	18
8.2.2	Anti-aliaseringsfilter . . . . .	18
8.2.3	Test af printplade . . . . .	18
8.3	Softwareimplementering . . . . .	18
8.4	Test af software . . . . .	18
8.4.1	Modultest . . . . .	18
8.4.2	Integrationstest . . . . .	18
<b>Kapitel 9</b>	<b>Resultater</b>	<b>19</b>
9.1	Hardwareresultater . . . . .	19
9.2	Diskussion af resultater . . . . .	19
9.3	Softwareresultater . . . . .	19
9.4	Diskussion af resultater . . . . .	19
<b>Kapitel 10</b>	<b>Udviklingsproces</b>	<b>20</b>
10.1	Metode . . . . .	20
10.1.1	Samarbejdsaftale . . . . .	20
10.1.2	Udviklingsforløb . . . . .	20
10.1.3	Arbejdsfordeling . . . . .	21
10.2	Proces . . . . .	21
10.3	Programmer . . . . .	21
<b>Kapitel 11</b>	<b>Erfaringer</b>	<b>22</b>
<b>Kapitel 12</b>	<b>Fremtidigt arbejde</b>	<b>23</b>
<b>Kapitel 13</b>	<b>Konklusion</b>	<b>24</b>
<b>Kapitel 14</b>	<b>Bilag</b>	<b>25</b>
<b>Kapitel 15</b>	<b>Referencer</b>	<b>26</b>



# 1

## Ordliste

Ord	Forklaring
AV-klap	Atrioventrikulærklap
GUI	Graphical User Interface
Ext	Extension
VS	Microsoft Visual Studio
AD-converter	Analog-Digital Converter
CPR	Centrale personregister
MoSCoW	Must have, should have, could have, won't have
FURPS+	Functionality, usability, reliability, performance, supportability
BDD	Block Definition Diagram
IBD	Internal Block Diagram

**Tabel 1.1.** Ordliste

# 2

## Indledning

Vores fornemste opgave som sundhedsteknologer er at lave udstyr, som kan hjælpe fagpersoner i sundhedssektoren. Især på sygehuse er der brug for mange sundhedsteknologier, som vi skal være med til at udvikle og gøre bedre. I daglig klinisk praksis er der ofte behov for kontinuert at monitorere patienters blodtryk, i særdeleshed på intensive afdelinger samt operationsstuer, hvor blodtrykket er en vigtig parameter til monitorering af deres helbredstilstand. Formålet med dette projekt har været at udvikle et system, der kan opfylde netop dette behov. Da systemet skal kunne bruges på operationsstuer, stiller dette selvfølgelig mange krav, da det er livsnødvendigt, at systemet er funktionelt og intuitivt at bruge. Systemet består både af hardware og software, som i sammenspil kan måle, behandle og visualisere blodtryk samt puls på en computerskærm.

Projektet har givet os mulighed for at anvende vores viden fra fysiologi-kurser, elektronik-kurser, signalbehandlingskursus og ikke mindst programmeringskurser. Teoretisk viden fra disse kurser har dannet grundlag for, at vi har været i stand til at designe og udvikle systemet.

Når man skal måle blodtryk, skal man først og fremmest bruge en tryktransducer, som måler signalet. Dette analoge signal skal behandles ved hjælp af hardware inden det bliver digitaliseret. Når signalet er digitaliseret, kan det bruges i en software applikation, som sørger for at lave beregner på signalet, visualisere det, alarmere hvis der er pludselige ændringer i og kontinuerligt give et overblik over patientens status. Dette leder frem til den konkrete problemformulering, som vi har arbejdet ud fra

# 3 Projektformulering

## 3.1 Problemformulering

Projektet har til formål at udvikle et invasivt blodtryksmålesystem, der kan tilsluttes patienters arterier via et væskefyldt kateter. Dette målesystem indeholder to elementer.

- Et elektronisk kredsløb, som forstærker signalet fra tryktransduceren og filtrerer det med et analogt filter for at undgå aliasering
- Et program til at vise blodtrykket som funktion af tiden. Programmet skal opfylde en række obligatoriske krav. Det skal:
  - Programmeres i C#
  - Kunne kalibrere blodtrykssignalet og foretage en nulpunktsjustering
  - Vise blodtrykssignalet kontinuert
  - Kunne lagre de målte data
  - Kunne filtrere blodtrykket i selve programmet via et digitalt filter, dette skal kunne slås til og fra (monitor = filtreret og afrundet signal/diagnose mode = rå signal med alle udsving)
  - Kunne afbilde systolisk/diastolisk blodtryk/middel blodtryk med tal
  - Kunne alarmere hvis blodtrykket overstiger indbyggede grænseværdier
  - Kunne måle og afbilde puls



# 4 Teori

## 4.1 Fysiologi

Hjertet består af to halvdele og er adskilt af en kraftig skillevæg. Ud over de to hjertehalvdele består hjertet også af andre elementer, hér vigtigt at nævne hjerteklapperne, som sørger for, at blodet kun kan løbe én vej. Hjertet har fire hjerteklapper: To AV-klapper, aortaklappen og pulmonalklappen. Hjerteklappernes åbning og lukning bestemmes af trykket på henholdsvis den ene og den anden side. Højre side af hjertet pumper blod ud i lungekredsløbet, også kaldt det lille kredsløb, mens venstre side af hjertet pumper blod ud i legemskredsløbet, også kaldt det store kredsløb. Hjertets pumpefunktion er en vigtig funktion, der har til hovedopgave at transportere blod rundt i kroppen.

For at hjertet kan transportere blodet rundt til hele kroppen kræver det et tryk, der dannes ved, at hjertet trækker sig sammen med jævne mellemrum. Når hjertet slapper af er blodtrykket lavest - denne fase i hjertets cyklus kaldes diastolen, deraf kommer det diastoliske blodtryk. Når hjertet istedet kontraherer sig er blodtrykket højest - denne fase i hjertets cyklus kaldes systolen, deraf kommer det systoliske blodtryk.

Diastolen starter, når trykket i ventriklerne er lavere end i arterierne. Dette medfører, at AV-klappen åbnes, så hjertet kan fyldes med blod. Under diastolen er aortaklappen lukket.

Systolen starter, når trykket i ventriklerne igen overstiger trykket i arterierne. AV-klapperne lukkes for at forhindre tilbagestrømning af blodet, der derved vil løbe i den forkerte retning. Når trykket i venstre ventrikel overstiger trykket i aorta åbnes aortaklappen og blodet vil strømme ud.

Blodtryk angives normalvis i mmHg. Er blodtrykket 100 mmHg betyder det, at blodtrykket er 100 mmHg højere end det atmosfæriske tryk. Ved blodtryksmåling foretaget i hjertehøjde siges det, at et almindeligt blodtryk hos en ung voksen der slapper af ligger på ca. 120/70 mmHg. 120 angiver det systoliske blodtryk mens 70 angiver det diastoliske blodtryk..

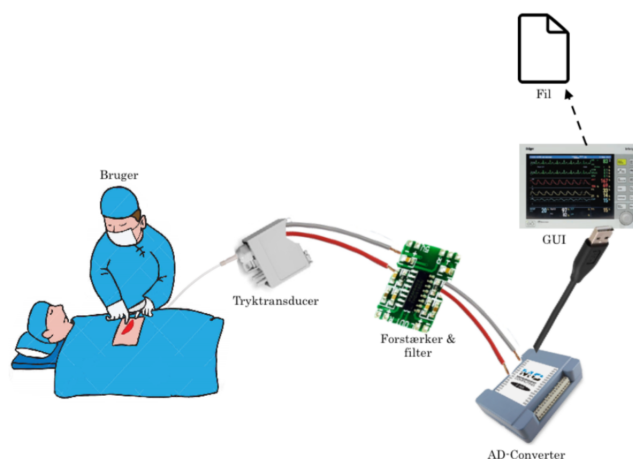
## 4.2 Blodtryksmåling

Blodtryk kan måles invasivt eller non-invasivt. Blodtrykket måles enklest ved den non-invasive metode, hvor man eksempelvis kan bruge en oppustelig blodtryksmanchet tilsluttet et manometer. Proceduren er hurtig og smertefri. Den invasive metode kræver indlæggelse af en kanyler i et af kroppens arterier. Metoden kan derfor være mere besværlig end ved den non-invasive blodtryksmåling. Den invasive metode kan med fordel bruges hos meget syge patienter eller på operationsstuer, hvor patienten enten bløder meget eller er i risiko for at komme til at bløde meget.

# 5

## Systembeskrivelse

### 5.1 Systemoversigt



**Figur 5.1.** Systemoversigt

### 5.2 Systembeskrivelse

På figur 5.1 ses en systemoversigt over et blodtrykmålesystem, der har til formål at kunne måle og vise en patients blodtryk og puls på en brugergrænseflade.

Systemet består af:

- Tryktransducer
- Forstærker
- Subtraktor
- Anti-aliaseringsfilter
- AD-Converter (NI DAQ 6009 USB)
- Computer
- Bluetooth højttaler
- Skærm
- Brugergrænseflade

Brugergrænsefladen er bygget op som et Graphical User Interface (GUI) hvorpå det målte blodtryk visualiseres kontinuert i form af en graf. Det systoliske-, diastoliske-, median blodtryk og puls vises i form af tal. Brugergrænsefladen består af knapper, der har forskellige funktioner. For nærmere beskrivelse af brugergrænsefladens funktioner henvises der til dokumentet kravsspecifikation.

### 5.2.1 Alarmer

Systemet skal i henhold til standard 60601-1-8 vedrørende alarmering kunne alarmere ved eventuelle fejl eller ændringer. I forhold til brugssituationen af blodtrykmålesystemet vurderes det at systemet skal kunne alarmere ved følgende:

1. Fejl i nulpunktsjustering
2. Tid til kalibrering
3. Fald eller stigning i blodtryk uden for de justerbare grænseværdier
4. Fald eller stigning i puls uden for de justerbare grænseværdier
5. Fejl i gem data

Alarmerne prioriteres efter hvorvidt de vurderes til at være i kategorien *high priority*, *medium priority* eller *low priority*. Alarmerne alarmerer enten visuelt, auditivt eller begge dele alt efter, hvor højt de er prioriteret. For krav til alarmerne henvises til dokumentet kravspecifikation afsnit 1.12 og dokumentet standard 60601-1-8.

# 6

## Kravspecifikation

### 6.1 Krav

For ethvert system er der opstillet en række funktionelle og ikke-funktionelle krav for at sikre at kunden og udvikleren er enige om, hvilke funktionaliteter systemet skal have samt systemets kapacitet og betingelser.

For dette system er der tilknyttet fem aktører; to primære og tre sekundære. Den ene primære aktør er en bruger, som betjener blodtryksmåleren. Den anden primære aktør er det tekniske personale, som skal foretage kalibreringen af systemet. De sekundære aktører er blodtrykssensoren, som er et interface for patienten som er koblet til systemet, bluetooth højttaleren, der bruges til at afspille alarmlydene, samt storskærmen, der skal bruges til at vise brugergrænsefladen meget større, så den er let at se for alle på operationsstuen.

Aktørerne interagerer med systemet, hvilket opstiller 5 use cases (funktionelle krav). UC1 er hovedscenariet idet, der måles og vises blodtryk og puls, og angiver. UC2 omhandler om at justér grænseværdier for puls samt systolisk- og diastolisk blodtryk. UC3 omhandler alarmeringen ift. ændringer i blodtryk samt puls eller ved fejl. UC4 omhandler funktionen at gemme patientens CPR, de opsamlede blodtryksdata samt puls gemmes i en privat database. UC5 omhandler kalibreringen af systemet. For yderligere detaljer se kravspecifikationen.

Når der opstilles en række use cases, opstilles der samtidig en række ikke-funktionelle krav, der skal overholdes for at use casene kan udføres optimalt. Den samlede oversigt over de ikke-funktionelle krav ses i kravspecifikationen. Her er der valgt at beskrive tre funktionelle krav der har fået prioriteringen must have ift. MoSCOW-modellen.

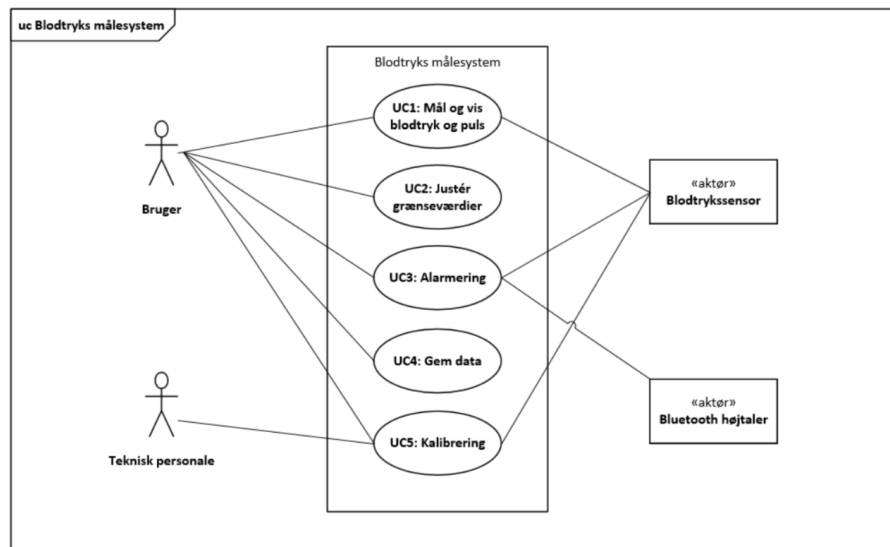
- Systemet skal have en brugergrænseflade
- Systemets GUI skal alarmere i henhold til punkt 1.12.4, hvis pulsen falder til under 60 slag i minuttet og overstiger 120 slag i minuttet og at Systemet skal alarmere i henhold til punkt 12.1.4, hvis blodtrykket falder til under 60 diastolisk eller stiger til over 160 diastolisk.
- Diagrammet samt de viste værdier for puls skal kunne læses på op til 1,5 meters afstand af person uden synshandicap, eller som benytter korrekt korrigerende midler, og må ikke indeholde farver som påvirker folk med farveblindhed.

Uden det ikke-funktionelle krav vedrørende brugergrænsefladen ville udviklerne kunne vise de opsamlede data som de ville, og ikke som kunden ønsker det. I dokumentet kravsspecifikation i bilaget ses flere ikke-funktionelle krav opstillet med fokus på brugergrænsefladen, så den er beskrevet så specifikt som muligt.

Det ikke-funktionelle krav, der henvender sig til alarmeringen er meget vigtig, da brugeren skal reagere hurtigt hvis patientens blodtryk eller puls falder eller stiger, da det kan være livstruende.

I henhold til brugssituationen i en operationsstue kan det ske, at brugeren måler patientens blodtryk og samtidig skal ordne noget ved underekstremiteterne, og dermed skal kunne læse værdien for puls og blodtryk på 1,5 meters afstand. Ligeledes skal der tages højde for at brugeren kan være farveblind, og dermed skal der benyttes andre farver end rød og grøn.

## 6.2 Use Case diagram



**Figur 6.1.** Use Case Diagram

## 6.3 Aktørbeskrivelser

Aktør	Type	Beskrivelse	Samtidige forekomster
Bruger	Primær	Brugeren betjener blodtryks målesystemet. Brugeren nulpunkts-justerer, igangsætter og stopper måling. Brugeren er en sundhedsfaglig person, der arbejder på operationsstuen.	1. Da der kun er én blodtryksmåler i systemet, kan der kun være en bruger.
Teknisk personale	Primær	Teknisk personale er teknisk uddannet og skal foretage kalibreringen af systemet.	1. Da der kun er én blodtryksmåler i systemet, kan der kun være et teknisk personale
Blodtryksensor	Sekundær	Blodtryksensoren er et interface for patienten som er koblet til systemet	1. Da der kun er én blodtryksmåler i systemet, kan der kun være en blodtryksensor

Bluetooth højttaler	Sekundær	Bluetooth højttaleren afspiller lyden for de alarmer der står i systembeskrivelsen.	1. Da der kun er én blodtryksmåler i systemet, kan der kun være en bluetooth højttaler
Skærm	Sekundær	Skærmen bruges til at vise brugergrænsefladen meget større end på computeren. Skærmen gør det nemt og overskueligt for hele personalet at følge med i blodtrykmålingen og de målte værdier af systolisk-, diastolisk-, median blodtryk og puls.	1. Da der kun er én blodtryksmåler i systemet, kan der kun være en skærm

**Tabel 6.1.** Aktørbeskrivelser

## 6.4 Use Case beskrivelser

Dette afsnit giver en kort beskrivelse af systemets Use Cases. For en yderligere og mere detaljeret beskrivelse af Use Cases henvises til dokumentet kravspecifikation afsnit xx.

### 6.4.1 Use Case 1: Mål og vis blodtryk og puls

Før en måling kan igangsættes skal systemet nulpunktjusteres. Patientens målte blodtryk visualiseres kontinuert i form af en graf på brugergrænsefladen. Systolisk-, diastolisk-, median blodtryk og puls vises i form af tal ligeledes på brugergrænsefladen.

### 6.4.2 Use Case 2: Justér grænseværdier

Før og under en måling er det muligt at justere grænseværdier for blodtryk og puls. Default værdier for grænseværdierne findes i systembeskrivelsen afsnit xx.

### 6.4.3 Use Case 3: Alarmering

Før og under en måling skal systemet kunne alarmere ved eventuelle fejl eller ændringer. Systemet alarmerer i henhold til standarden og prioriteres efter *high priority*, *medium priority* eller *low priority*.

### 6.4.4 Use Case 4: Gem data

Efter endt måling er det muligt at gemme patientens blodtryk og puls i en fil. Personale ID, patientens CPR-nummer samt dato og tid indtastes. Kendes patientens CPR-nummer ikke indtastes "000000-0000".

### 6.4.5 Use Case 5: Kalibrering

Kalibrering af systemet skal foretages af teknisk personale. Kalibreringen foregår én gang årligt.

## 6.5 Afgrænsning

I kravsspecifikationen er der beskrevet de ikke-funktionelle krav vha. MoSCoW- modellen. Vi har valgt at holde fokus på blodtryksmålingen, og har derfor valgt ikke at implementere mere i vores system. Vores won't have krav lyder derfor således, "Systemet kan ikke vise andet end blodtryksmålingen samt angive puls, diastolisk/systolisk blodtryk (F)". HENVISNING

En af de software mæssige afgrænsninger i systemet, er at der ikke kan hentes samt åbnes en tidligere måling fra en fil, og kan derfor kun visualisere igangværende målinger. MERE OM SOFTWARE

Det er nødvendigt at man manuelt skal foretage en nulpunktsjustering inden hver måling igangsættes, for at sikre sig at man ikke måler det atmosfæriske tryk med. Sker dette ville vores system ikke kunne registrere værdierne, da vi har bygget systemet efter at kunne håndtere et max tryk på 250 mmHg. Derudover skal der årligt laves en kalibrering af systemet, for at sikre en vedvarende præcision af systemet. Dette foretages af en teknisk person, som sørger for at et kendt tryk svarer til en kendt spænding, således systemet måler korrekt.

Der er valgt en række værdier ifm. Udviklingen af hardwaren, hvilket gør at den er begrænset i forhold til hvilke størrelser af værdier den kan håndtere. Forstærkeren kræver nogle specifikke modstande, for at operationsforstærkeren skal forstærke op til de 4 V, som vi har valgt vores system skal kunne. Derudover skal systemet benytte en tryktransducer med en sensitivitet på  $5,0 \mu\text{V}/(\text{V} \cdot \text{mmHg})$ , da denne værdi indgår i en beregning af  $V_{\text{out}}$  som forstærkeren skal kunne levere.

### 6.5.1 Ikke-funktionelle krav

MoSCoW bruges til at definere forskellige krav til systemet. I dette afsnit vil nogle af de vigtigste krav være opstillet. For alle krav henvises til dokumentet kravspecifikation afsnit xx.

#### Must have

1. Systemet skal kunne kalibreres (F, U)
2. Systemet skal kunne nulpunktsjusteres (F, U)
3. Systemet skal kunne måle og vise et blodtryk på 250 mmHg ved normal atmosfærisk tryk (P)
4. Systemet skal kunne måle og vise en puls på mellem 30 og 220 slag pr. minut
5. Systemets GUI skal kunne vise en graf inden for 10 sekunder (R)
6. Systemets GUI skal kunne stoppe målingen inden for 5 sekunder (R)
7. Systemets GUI skal kunne vise systolisk-, diastolisk-, og median blodtryk samt puls i form af heltal (U)
8. Systemet skal kunne alarmere i henhold til systembeskrivelsen afsnit xx i dokumentet kravspecifikation (P)

# 7

# Arkitektur og design

## 7.1 Hardwarearkitektur

### 7.1.1 BDD

### 7.1.2 IBD

## 7.2 Hardwaredesign

### 7.2.1 Forstærker

### 7.2.2 Anti-aliaserings filter

### 7.2.3 Printplade

## 7.3 Softwarearkitektur

### 7.3.1 Domænemodel

## 7.4 Softwaredesign

Dette afsnit beskriver, hvordan softwarearkitekturen er realiseret, samt hvordan de forskellige lag og klasser kommunikerer med hinanden. Overordnet set er det tilstræbt at designe softwaren ud fra paradigmet om lav kobling. Dette er først og fremmest gjort ved, som beskrevet i arkitekturafsnittet, at dele de forskellige use cases op i egne klasser med interfaces imellem. Derudover er der brugt dependency injection, som ligeledes fremmer en lav kobling. Et knudepunkt i softwaren er klassen HovedmenuGUI, som har mange ansvar. Dette ansvar er forsøgt mindsket ved at lave en GuiFactory klasse, som sørger for at oprette instanser af de tre andre GUI'er. Dette sikrer også, at det ville være nemt at tilføje en anden GUI, hvis softwaren skal udvides med en ny funktionalitet.

Kommunikationen mellem datalaget og logiklaget er designet ud fra producer-consumer princippet. Dette var en oplagt løsning, da vores måleenhed hele tiden producerer data, som skal bruges og behandles af controllerklasserne i logiklaget. En anden vigtig feature ved producer-consumer er, at en datakø er indbygget, som sørger for, at der ikke går måledata tabt mellem datalaget og logiklaget, som kører i hver sin tråd. Køen har vi realiseret ved at bruge AutoResetEvents, som forhindrer, at producertråden overskriver måledata, inden de er blevet consumed i logiklaget. På den måde undgås fejl i kommunikationen mellem datalaget og logiklaget.

Mellem logiklaget og præsentationslaget var det i første omgang planen at bruge observer pattern. Det viste sig dog, at dette ikke var særlig hensigtsmæssigt, da der er forskel på, hvor tit blodtryksgrafen og beregningerne på hovedmenuen skal opdateres. Dette gjorde det vanskeligt at bruge observer standardmønstret. Der var behov for en mere generisk løsning.



Kommunikationen mellem logiklaget og præsentationslaget blev i stedet løst ved at bruge events, som på mange måder minder om observer, men er en mere fleksibel metode, der er indbygget i .NET frameworket. Dette viste sig også at være en god løsning på kommunikation mellem alarmeringscontrolleren og hovedmenuen.

# 8

## Implementering og test

### 8.1 Hardwareimplementering

Hardware er implementeret således at den indeholder en PC, NI DAQ 6009( AD-converter), Sallen-Key Lavpas-filter, transducer og substraktor. På fumlebrættet er selve forstærkeren, filteret og substraktoren opbygget. Disse tre dele er hver især blevet testet igennem, for at se om de gav de forventede værdier. Det hele blev sat sammen til et kredsløb og der blev testet for hele systemet. Dette blev gjort for at sikre, at systemet i helhel virkede som der blev forventet. Efter testningen af hele systemet, blev kredsløbet lavet inde på multisim, hvor multisimkredsløbet derefter blev overført til ultiboard, hvor de tilhørende komponenter blev placeret og sat fast på printpladen. Printet blev derefter sendt til afsted og vi fik vores printplade. Herefter loddede vi på egen hånd de udregnede komponenter.

#### BILLEDE AF PRINTET OG ULTIBOARD

Systemet hænger sammen på den måde, at strømforsyningen forsyner med strøm til substraktoren, transduceren, og forstærkeren. Analog discovery forsyner spænding til forstærker, transducer og filter. Signalet, som i form af et blodtrykssignal, kommer fra transduceren til Analog Discovery. Forstærkeren forstærker signalet fra transduceren, der sender signalet gennem filteret, som bliver filtreret. Signalet går igennem substraktoren, som neddæmper signalet til at gå fra 0-4 V til  $\pm 2$  V . Dette signal bliver sendt videre til DAQ'en, som omdanner det analoge signal til et digitalt signal, der sender signalet videre til PC'en.

#### BILLEDE AF FUMLEBRÆT

### 8.2 Test af hardware

#### 8.2.1 Forstærker

##### 8.2.1.1 Test med Analog Discovery

##### 8.2.1.2 Test med væskesøjle

#### 8.2.2 Anti-aliaseringsfilter

##### 8.2.2.1 Test med Analog Discovery

##### 8.2.2.2 Test med væskesøjle

#### 8.2.3 Test af printplade

### 8.3 Softwareimplementering

#### 8.4 Test af software

##### 8.4.1 Modultest

##### 8.4.2 Integrationstest

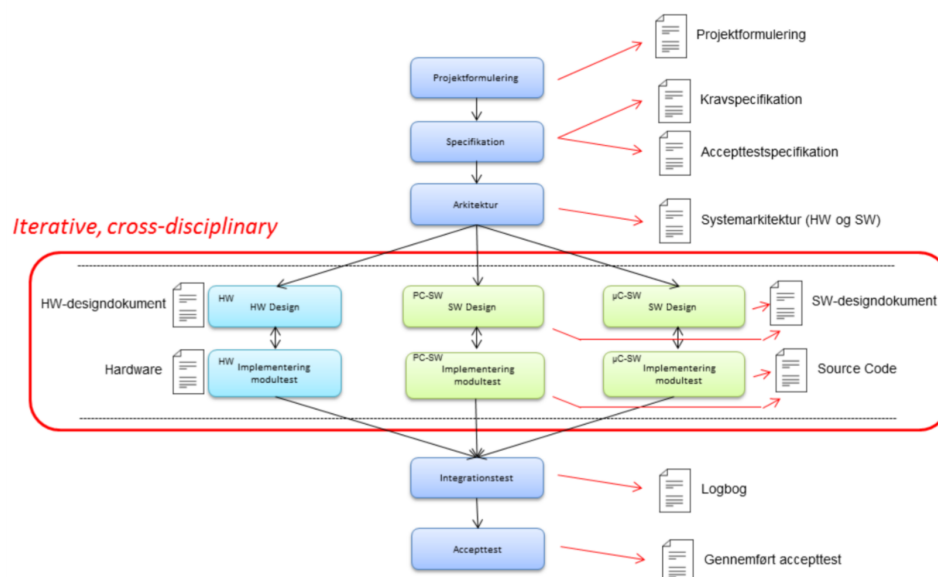
# 9 Resultater

- 9.1 Hardwareresultater
- 9.2 Diskussion af resultater
- 9.3 Softwareresultater
- 9.4 Diskussion af resultater

# 10 Udviklingsproces

## 10.1 Metode

I dette semesterprojekt har vi arbejdet ud fra ASE-udviklingsmodellen, som er vist i figur 1. Når man arbejder ud fra denne model betyder det, at man undervejs i projektet gennemfører arbejdet i små bidder, også kaldet iterationer, og afleverer disse. I vores tilfælde har dette foregået i reviewgrupper, hvor vi har rettet hinandens dele, og herefter gennemgået rettelserne i fællesskab. Der har været review af tre omgange; review af kravspecifikation og accepttest, review af hardware arkitektur, og slutteligt review af software arkitektur. Denne feedback har hjulpet gruppen videre ved eventuelle problemer eller tvivl.



### 10.1.1 Samarbejdsaftale

Vi har anvendt en samarbejdsaftale, som er udarbejdet i fællesskab i gruppen[1]. Udbyttet ved brug af samarbejdsaftalen, er et bedre gruppearbejde. Når de regler og krav der er stillet til gruppemedlemmerne samt projektet, bliver formuleret og skrevet ned, vil man undgå diskussioner forvirring omkring disse, da man altid kan hvide samarbejdsaftalen frem.

### 10.1.2 Udviklingsforløb

Som arbejdsmetoden i dette projekt, har vi anvendt Scrum. Vores projektleder Sarah har fungeret som Scrum Master, som har taget sig af det administrative og haft lederrollen. Derudover har vi benyttet os af ugentlige sprints, fra onsdag til onsdag. Til gruppemødet, som sædvanligvis ligger onsdag morgen, bliver der lavet review af det seneste sprint. Til review af sprint bliver gennemgået hvad gruppen har nået, hvilke problemer vi er stødt på, og hvordan disse er blevet

løst. Herudover aftales der hvad næste uges sprint skal bestå af, og hvilke deadlines der skal være. Både review af sprint og planlægning af det nye sprint dokumenteres i et tekstdokument [2]. Udover gruppemøde ca. hver onsdag, har vi haft to ugentlige standup-møder, på omkring fem minutter. Disse møder finder sted mandag og fredag eftermiddag. At arbejde med Scrum, som vi har gjort det, har været positivt. De ugentlige sprints har været med til at bryde projektet ned i endnu mindre dele og gøre det overskueligt. Samtidig har standup møderne giver overblik og indsigt i hvad resten af gruppen har arbejdet med og er kommet i mål med.

### 10.1.3 Arbejdsfordeling

Gruppen tog en beslutning om at fordele projektarbejdet i to teams; et softwareteam og et hardwareteam, da projektet er så stort at det ikke er muligt at være inde over det hele. Softwareteamet er bestående af Mathias, Nicolai og Thea, mens Hardwareteamet er bestående af Caroline, Kajene og Mikkel. Sarahs rolle er flyver, som skiftevis er med softwareteamet og hardwareteamet, alt afhængigt af hvor der er brug for ekstra hjælp. Denne opdeling har fungeret udmærket, da den har givet mulighed for at specialisere sig, og komme dybere ned i stoffet. Dog har det også givet anledning til frustration i forhold til ikke at være helt inde over hvad det modsatte team arbejder med. De ugentlige standup-møder har bidraget til at begge teams får en forståelse for hvad der bliver arbejdet med i både hardware og software. Projektarbejdet er også blevet uddelegeret, så hvert gruppemedlem har fået primære og sekundære områder, hvilket ses på nedenstående skema:

## 10.2 Proces

## 10.3 Programmer

# 11 Erfaringer

# 12 Fremtidigt arbejde

# 13 Konklusion



# 14 Bilag

# 15 Referencer