

««< HEAD ===== »»> 9656cb1b8b2eb4bcac12ea1597f7c5a821fee169

««<

HEAD

=====

»»> 9656cb1b8b2eb4bcac12ea1597f7c5a821fee169



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 Elakanathan	AU594051	201710472

VEJLEDER: SAMUEL ALBERG THRYSSØE

DATO: 19/12-2018

ANTAL SIDER: XX

Resumé

Abstract

Indholdsfortegnelse

	Side
Kapitel 1 Ordliste	6
Kapitel 2 Indledning	7
Kapitel 3 Projektformulering	8
3.1 Problemformulering	8
Kapitel 4 Teori	9
4.1 Fysiologi	9
4.2 Blodtryksmåling	9
Kapitel 5 Systembeskrivelse	10
5.1 Systemoversigt	10
5.2 Systembeskrivelse	10
5.2.1 Alarmer	11
Kapitel 6 Krav	12
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
«««< HEAD	
6.4.5 Use Case 5: Kalibrering	14
6.5 Afgrænsning	15
6.5.1 Ikke-funktionelle krav	15
Kapitel 7 Arkitektur og design	16
7.1 Hardwarearkitektur	16
7.1.1 BDD	16
7.1.2 IBD	17
7.2 Hardwaredesign	18
7.2.1 Forstærker	18
7.2.2 Subtractor	18
7.2.3 Anti-aliaserings filter	19
7.2.4 Printplade	19

7.3	Softwarearkitektur	20
7.3.1	Domænemodel	20
7.4	Softwareedesign	20
Kapitel 8	Implementering og test	21
8.1	Hardwareimplementering	21
8.2	Test af hardware	22
8.2.1	Forstærker	22
8.2.2	Anti-aliaseringsfilter	22
8.2.3	Test af printplade	22
8.3	Softwareimplementering	22
8.4	Test af software	22
8.4.1	Modultest	22
8.4.2	Integrationstest	22
Kapitel 9	Resultater	23
9.1	Hardwareresultater	23
9.2	Diskussion af resultater	23
9.3	Softwareresultater	23
9.4	Diskussion af resultater	23
Kapitel 10	Udviklingsproces	24
10.1	Metode	24
10.1.1	Samarbejds aftale	24
10.1.2	Udviklingsforløb	24
10.1.3	Arbejdsfordeling	25
10.2	Proces	25
10.3	Programmer	25
Kapitel 11	Erfaringer	26
Kapitel 12	Fremitidigt arbejde	27
Kapitel 13	Konklusion	28
Kapitel 14	Bilag	29
Kapitel 15	Referencer	30
<hr/>		
6.4.5	Use Case 5: Kalibrering	15
6.5	Afgrænsning	16
6.5.1	Ikke-funktionelle krav	16
Kapitel 7	Arkitektur og design	18
7.1	Softwarearkitektur	18
7.1.1	BDD	18
7.1.2	IBD	18
7.2	Hardwaredesign	18
7.2.1	Forstærker	18

7.2.2	Anti-aliaserings filter	18
7.2.3	Printplade	18
7.3	Softwarearkitektur	18
7.3.1	Domænemodel	18
7.4	Softwaredesign	18
Kapitel 8	Implementering og test	20
8.1	Hardwareimplementering	20
8.2	Test af hardware	20
8.2.1	Forstærker	20
8.2.2	Anti-aliaseringsfilter	20
8.2.3	Test af printplade	20
8.3	Softwareimplementering	20
8.4	Test af software	20
8.4.1	Modultest	20
8.4.2	Integrationstest	20
Kapitel 9	Resultater	21
9.1	Hardwareresultater	21
9.2	Diskussion af resultater	21
9.3	Softwareresultater	21
9.4	Diskussion af resultater	21
Kapitel 10	Udviklingsproces	22
10.1	Metode	22
10.1.1	Samarbejds aftale	22
10.1.2	Udviklingsforløb	22
10.1.3	Arbejdsfordeling	23
10.2	Proces	23
10.3	Programmer	23
Kapitel 11	Erfaringer	24
Kapitel 12	Fremitidigt arbejde	25
Kapitel 13	Konklusion	27
Kapitel 14	Bilag	28
Kapitel 15	Referencer	29
»»»> 9656cb1b8b2eb4bcac12ea1597f7c5a821fee169		

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 invasivt, 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. Vi har valgt at arbejde med projektet i forhold til den valgte brugssituation på en operationsstue. 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. Monitorering af blodtrykket invasivt bruges på operationsstuer og intensive afdelinger verden over. Derfor er der udarbejdet en standard for, hvordan alarmer af forskellige prioritet skal alarmere både visuelt og auditivt. Denne standard har vi valgt at følge ift. alarmering visuelt og auditivt samt hvilke parametre, der skal gemmes ved en alarm.

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 mäter 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 beregninger 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 = råt 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, her 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 pumpefunktionen 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 højst - 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 ventrikkel overstiger trykket i aorta åbnes aortaklappen og blodet vil strømme ud.

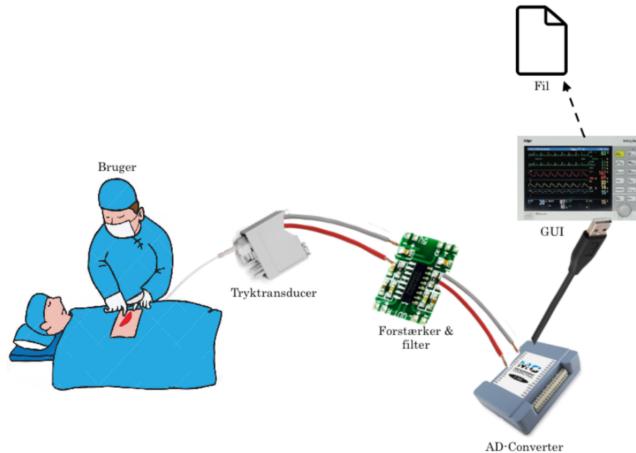
Blodtryk angives normalvis i mmHg. Et blodtrykket 100 mmHg betyder det, at blodtrykket er 100 mmHg højere end det atmosfæriske tryk. Ved blodtryksmåling foretaget i hjertehøjde siges det, at et almindeligt blodtryk hos en ung voksen der slapper af ligger på ca. 120/70 mmHg. 120 angiver det systoliske 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 kanyle 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 systemoversigten for vores 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 højtalere
- 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 Krav

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 højtaleren, 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. 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, digitalt filter status, kalibreringsdato samt dens værdi og alarmer iløbet af operationsstuen samt alarmbetingelserne iflg. uddraget af standard 60601-1-8 i en fil. 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 fire 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.
- 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.

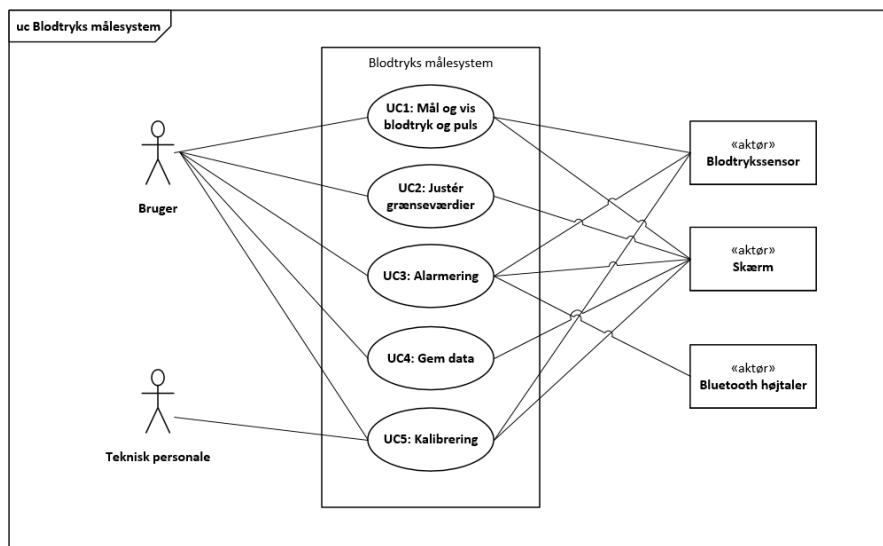
Uden det ikke-funktionelle krav vedrørende brugergrænsefladen ville udviklerne kunne vise de opsamlede data som de ville, og ikke som kunden ønsker det. I dokumentet kravspecifikation i bilaget ses flere ikke-funktionelle krav opstillet med fokus på brugergrænsefladen, så den er

beskrevet så specifikt som muligt.

De 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		<p>Brugeren betjener blodtryks målesystemet.</p> <p>Brugeren nulpunktsjusterer, igangsætter og stopper måling. Brugeren er en sundhedsfaglig person, der arbejder på operationsstuen.</p>	<p>1. Da der kun er én blodtryksmåler i systemet, kan der kun være en bruger.</p>
Tekniske personale	Primær		<p>Tekniske personale er teknisk uddannet og skal foretage kalibreringen af systemet.</p>	<p>1. Da der kun er én blodtryksmåler i systemet, kan der kun være et teknisk personale</p>

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 højtalер	Sekundær	Bluetooth højtaleren afspiller lyden for de alarmer der står i systembeskrivelsen.	1. Da der kun er én blodtryksmåler i systemet, kan der kun være en bluetooth højtalér
Skærm	Sekundær	Skærmens bruges til at vise brugergrænsefladen meget større end på computeren. Skærmens gör det nemt og overskueligt for hele personalet at følge med i blodtrykmålingen og de målte værdier af systolisk-, diastolisk-, median blodtryk og puls.	Flere. Da skærmens er uafhængig af antal blodtryksmåler i systemet, vil der kunne være flere skærme

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 i bilaget kravspecifikation 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 blodtryksmåling, digitalt filter status og alarmer i en fil. Personale ID, patientens CPR-nummer samt dato og tid indtastes. Kendes patientens CPR-nummer ikke indtastes "0000000-0000".

6.4.5 Use Case 5: Kalibrering

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

6.5 Afgrænsning

I kravspecifikationen er der beskrevet de ikke-funktionelle krav vha. MoSCoW- modellen. Vores fokus har været invasiv blodtryksmålingen, og derfor har vi valgt ikke at prioritere at implementere fx EKG-måling. Vores won't have krav lyder derfor således, "Systemet kan ikke vise andet end blodtryksmålingen samt angive puls, diastolisk/systolisk blodtryk (F)".

De softwaremæssige afgrænsninger i systemet er blandt andet, at der ikke kan hentes, samt åbnes en tidligere måling fra en fil. Herudover har vi valgt at gemme målingens værdier i en fil, frem for i en database. Ift brugssituationen, ville det give god mening at benytte både en fil og en database, da dataen kan tilgås og ses fra flere enheder. Eftersom det at gemme i en fil er et must have krav, er det blevet prioriteret højere at opfylde dette end could have kravet om at gemme i en database. Yderligere er et could have krav at systemet kunne have et lydsignal ved afsluttet blodtryksmåling, dette blev dog ikke prioriteret, da det kun var et could have krav. Derimod er det prioriteret, at systemet skal kunne beregne og vise middelblodtrykket på GUI'en. Dette er et must have krav, da man gennem en operation skal have en indikation på, hvordan blodtrykket for patienten lægger.

Det er nødvendigt at man manuelt skal foretage en nulpunktsjustering inden hver måling igangsættes, for at sikre sig at man ikke mäter 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. Derfor er dette et must krav. Endnu et must krav er, at der årligt skal 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 mäter 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 $5,0 \mu\text{V}/(\text{V}^*\text{mmHg})$, da forstærkningsfaktoren er beregnet ud fra denne, og dermed er modstandsvalget afhængig af en tryktransducer med en sensitivitet på $5,0 \mu\text{V}/(\text{V}^*\text{mmHg})$.

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

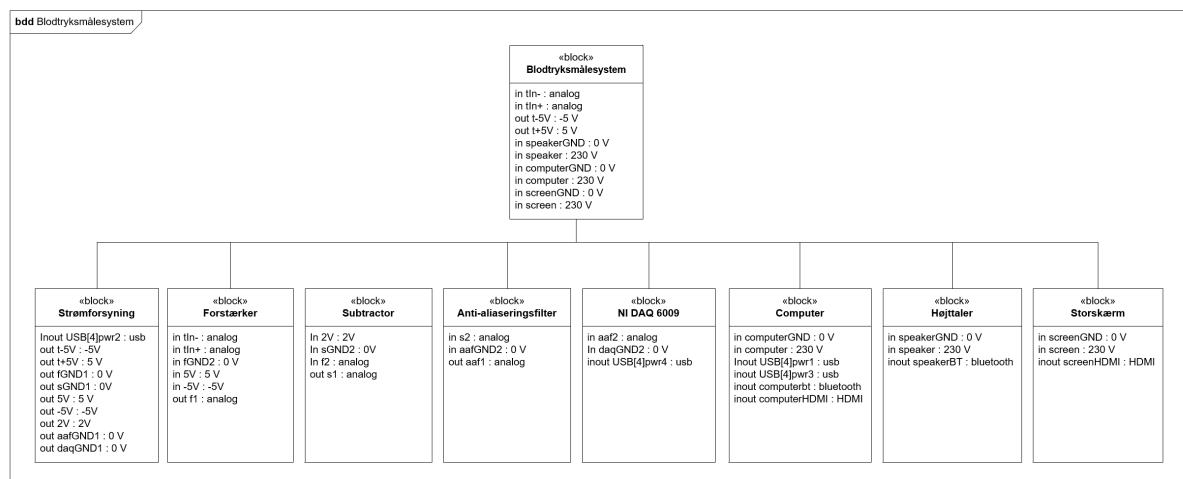
Arkitektur og design

7.1 Hardwarearkitektur

I den følgende sektion om hardwarearkitektur beskrives opbygningen af systemet ved brug af sysML. Hardwarearkitekturen indholder diagrammerne BDD og IBD samt en kort forklaring af disse. For ydeligere information om interaktionen i systemet henvises til dokumentet arkitektur og design for flere detaljer. Dette dokument indeholder bloktabel og signaltabel, der fortæller om alle blokke, porte og signaler, der ses i diagrammerne. Yderligere information omkring valg af elementer i systemet findes i dokumentet analyse, som findes i bilaget.

7.1.1 BDD

Et BDD viser opbygningen af et system med brug af blokke. Nedenstående diagram på figur 7.1 viser opbygningen af vores blodtrykssystem, der består af en række forskellige elementer.



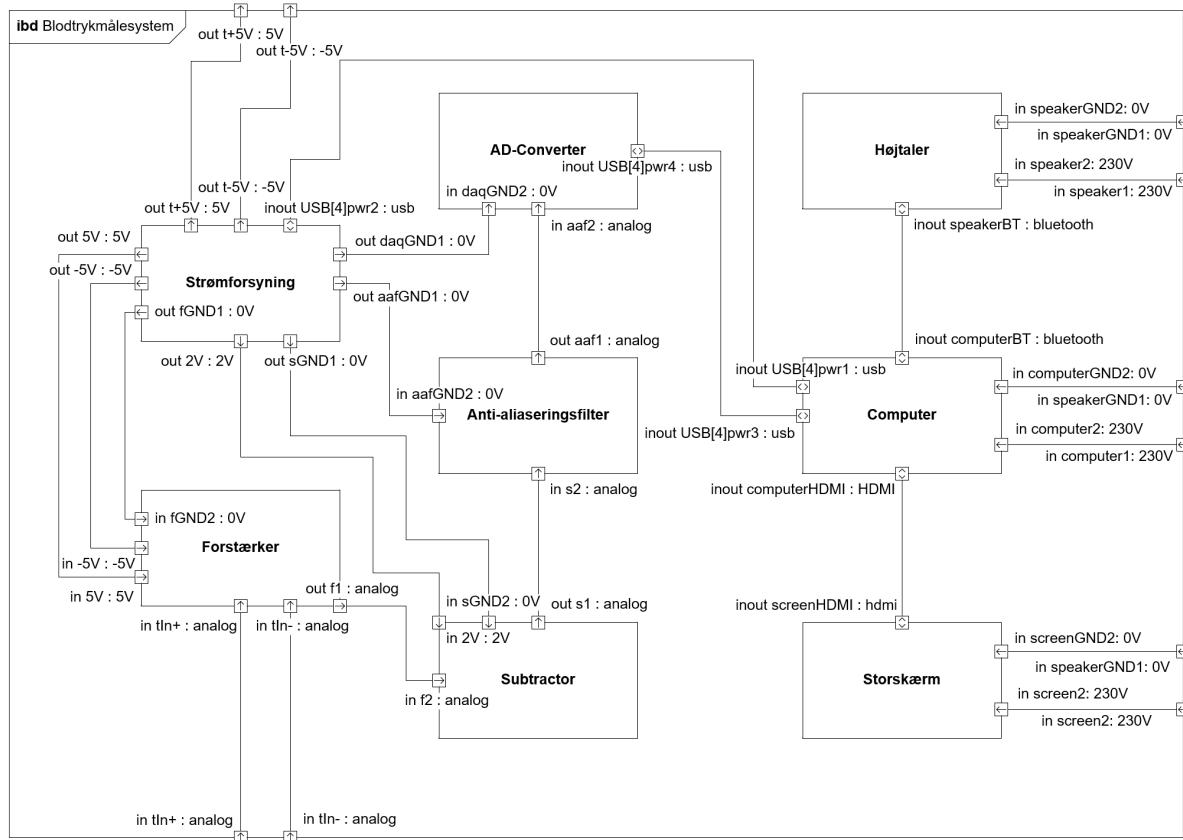
Figur 7.1. BDD Blodtryksmålesystem

På BDD'et ses de forskellige elementer, der er en del af blodtrykmålesystemet. Den øverste blok "Blodtryksmålesystem" viser systemets interaktioner med udefrakommende objekter. I denne blok findes tilslutningen til tryktransduceren, som ikke er en del af selve systemet. I blokken findes også tilslutningen af strøm til henholdsvis computer, højttaler og storskærm.

På diagrammet ses også otte blokke, der repræsenterer de forskellige dele af systemet. Systemet består altså af en strømforsyning, en forstærker, en subtractør, et filter, en AD-Converter, en computer, en højttaler og en storskærm. Strømforsyningen giver strøm til tryktransducer, forstærker og subtractør og udgøres af Analog Discovery. Forstærker, subtractør og filter er designet til systemet i forbindelse med projektet. Mere info om disse elementer i afsnit om design af hardware.

7.1.2 IBD

Et IBD viser en mere detaljeret opbygning af et system fordi alle forbindelser mellem blokkene, der findes i BDD'et er tegnet. På nedenstående figur 7.2 ses derved opbygningen af blodtryksmålesystemet inklusiv alle forbindelser mellem blokkene.



Figur 7.2. IBD Blodtryksmålesystem

Den yderste ramme omkring diagrammet illustrerer den øverste blok "Blodtryksmålesystem" på figur 7.1 på forrige side. Her ses det tydeligt hvordan nogle af blokkene interagerer med objekter uden for selve systemet fordi forbindelserne er tegnet ud til rammen. Hver port har et unikt portnavn. IBD'et er med til at forbinde portnavnene fra BDD'et sammen, så overblikket over systemet som en helhed giver mening.

7.2 Hardwaredesign

I det følgende afsnit vil de vigtigste overvejelser for designet af systemets hardware fremtræde. En mere detaljeret begrundelse for valg findes i dokumentet analyse i bilaget. Dokumentet analyse indeholder også alle de beregninger der er foretaget i forbindelse med valg af forstærkning, filtertype og subtractor.

7.2.1 Forstærker

Grundet at det signal, der udsendes gennem tryktransduceren er for småt til, at det kan analyseres og anvendes i vores system, skal signalet derfor forstærkes. Til projektet blev det givet, at signalet skulle forstærkes med en operationsforstærker af typen INA114. Denne type operationsforstærker har et variabelt gain, der afhænger af hvilken modstand, der forbindes til operationsforstærkeren.

For at vide hvor meget vi ønsker at forstærke signalet fra tryktransduceren er det vigtigt at kende til størrelsen af signalet. Mængden af forstærkning bestemmes derfor ud fra tryktransducerens følsomhed. Denne følsomhed findes i databladet for tryktransduceren. Vi udregner hvor meget vi ønsker at forstærke signalet op til:

$$V_{out} = 5 \cdot \frac{\mu V}{V \cdot mmHg} \cdot 10V \cdot 250mmHg = 12500\mu V = 12,5mV$$

Fordi vi kender følsomheden på tryktransduceren og ved, at vi ønsker at kunne forstærke blodtrykssignaler op til 250 mmHg og har en forsyningsspænding på -5 til 5V via Analog Discovery får vi ovenstående resultat ved design af forstærkeren.

Forstærkerens gain kan udregnes med følgende formel:

$$G = 1 + \frac{50k\Omega}{R_G}$$

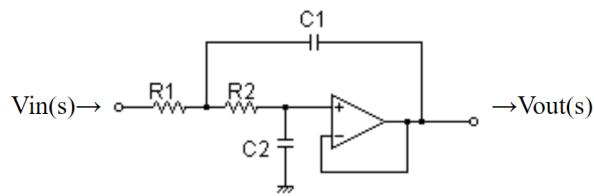
Vi ønsker at forstærke disse 12,5 mV op til 4V. Resultatet af udregningerne blev derfor, at vi forstærker signalet fra transduceren 320 gange og bruger en modstand på 156 Ω . Alle beregninger og beskrivelser findes i dokumentet analyse i bilaget.

7.2.2 Subtractor

For at benytte alle 14-bit på AD-Converten, NI-DAQ 6009, har vi i forbindelse med designet af forstærkeren besluttet at forstærke fra 0-4V. Alle tal vi får ind fra tryktransduceren vil derfor ligge i dette område. For at få vores indgangssignal til at passe med indgangene på AD-Converten og derved udnytte alle bit er det nødvendigt at nedjustere signalet. Vi bruger derfor en subtractor af typen OP27G, som er en standard operationsforstærker, der findes i laboratoriet. Subtractoren justerer vores signal så det går fra -2V til 2V i stedet for 0-4V. Alle udregninger vedrørende subtractoren findes ligeledes i dokumentet analyse i bilaget.

7.2.3 Anti-aliaserings filter

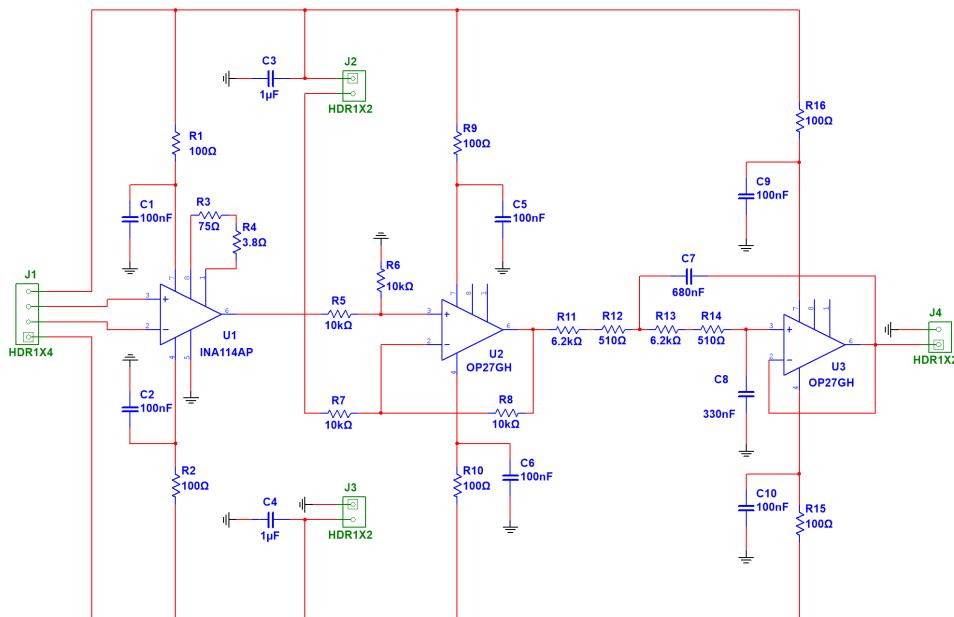
For at fjerne støj er det nødvendigt at filtrere signalet. Til systemet har vi derfor valgt at benytte et aktivt 2. ordens lavpasfilter af typen Sallen Key. Alt i alt ønskes signalet dæmpet med 90 dB , men fordi signalet allerede dæmpes 70 dB er det kun nødvendigt at dæmpe med 20 dB yderligere. Dette kan klares med et 1. ordens filter men for at være på den sikre side, har vi valgt at benytte et 2. ordens filter, der dæmper med 40 dB pr. dekade. Alt dokumentation omkring valg af filtertype, beregninger og begrundelse for skift at filter undervejs i projektet er beskrevet nærmere i dokumentet analyse i bilaget. Nedenfor ses opbygningen af et Sallen Key filter. Vi bruger igen en operationsforstærker af typen OP27G til at bygge filteret.



Figur 7.3. Sallen Key: Aktivt 2. ordens lavpasfilter

7.2.4 Printplade

Før printpladen kan designes er det nødvendigt at opbygge systemet i Multisim. Multisim danner en simuleret version af det kredsløb, der undervejs er opbygget og testet på fumlebræt. Nedenfor vises det endelige kredsløb i Multisim, som også er brugt til at designe printpladen.



Figur 7.4. Kredsløbet i Multisim

En detaljeret beskrivelse af opbygningen af diagrammet, herunder designovervejelser, findes i dokumentet ?? i bilaget.

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 principippet. 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 frameworkt. Dette viste sig også at være en god løsning på kommunikation mellem alarmeringscontrolleren og hovedmenuen.

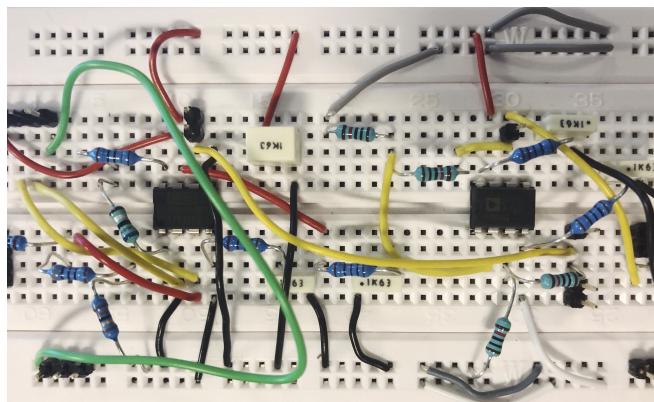
Vi har desuden valgt, at vi vil have vist en puls på det første sekund af målingen, da vi mener, at i forhold til brugssituation er det bedst at have en puls med det samme, og så bliver den så mere præcis med tiden end at man skal vente 10 sekunder, hvor man ikke har en puls.

8

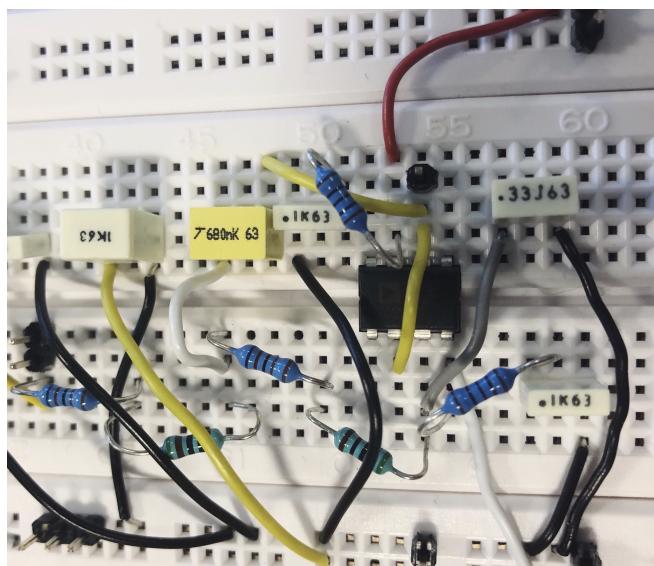
Implementering og test

8.1 Hardwareimplementering

«««< HEAD Alle dele af projektets hardware er blevet analyseret og designet og derefter implementeret. Vi startede med at udregne komponenter til forstærkeren, som vi på forhånd vidste hvordan skulle opbygges. Efter beregningerne kunne forstærkeren opbygges på fumlebræt. Efter at hver del er blevet implementeret på fumlebrættet er det blevet modultestet. På denne måde er vi sikre på, at hvert delelement af projektets hardware virker som forventet. På fumlebrættet er forstærkeren, subtractoren og Sallen-Key filteret derved implementeret og modultestet enkeltvis for at se om de gav de forventede værdier. Efterfølgende blev det hele sat sammen til et kredsløb og systemet kunne derefter integrationstestes.



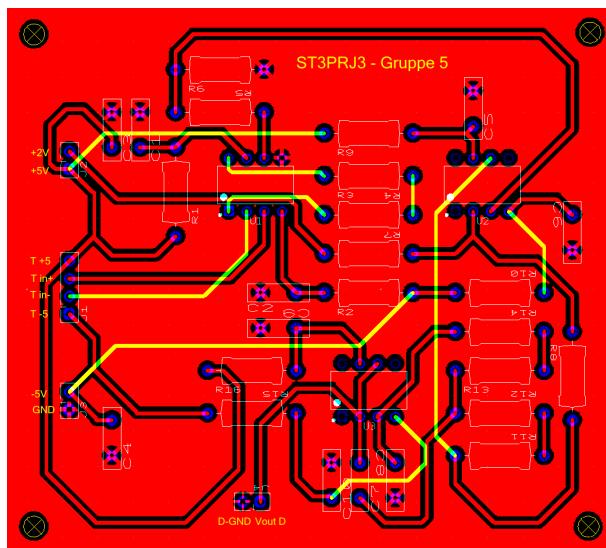
Figur 8.1. Opbygning af forstærker(til venstre) og subtractor (til højre) på fumlebræt



Figur 8.2. Opbygning af filter på fumlebræt

Efter integrationstesten blev kredsløbet tegnet i Multisim. Det tegnede kredsløb i Multisim blev til slut overført til ultiboard, hvor de tilhørende komponenter blev placeret på printpladen. Printet blev testet og sendt til EuroCircuits. De beregnede komponenter i analyse-delen er til slut loddet på den færdige printplade og systemet blev testet en sidste gang.

For kredsløbet i Multisim henvises til figur 7.4 på side 19. Nedenfor vises designet af printpladen i Ultiboard da det var klar til print samt printpladen, der er loddet med alle komponenter.



Figur 8.3.

===== 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 +- 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. »»> 9656cb1b8b2eb4bcac12ea1597f7c5a821fee169

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

I implementeringsfasen startede gruppen ud med at lave et "skelet" for programmeringen ud fra vores klassediagram, så det var opdelt og klar til at blive kodet efter 3 lags-modellen. Da vi blev klar til at gå i gang med at kode begav vi os til at uddelegerere forskellige programmeringsopgaver. Hvert medlem startede med at påtage sig nogle små opgaver, som var nogle basale ting. Efter udførslen af små-opgaverne blev der uddelegeret use cases, som man fik ansvaret for, at var klar og funktionel til deadline. Herefter blev de forskellige use cases sat sammen, og testet på tværs af lagene ud fra programmets funktionalitet, der er defineret i kravspecifikationen.

Hele vores system er skrev i programmeringssproget C# og er skrevet vha. programmeringsprogrammet Visual Studios 2017. Vi har gennem 1., 2. og 3. semester skrevet i sproget C# i undervisningen, og derfor har det været oplagt at skrive i dette sprog. Alle i gruppen har downloaded en udvidelse til Visual Studios kaldet "GitHub", hvilket har gjort det muligt at alle gruppens medlemmer har kunne sidde og arbejde i det samme dokument. Dette har gjort programmeringsprocessen en hel del mere overskuelig idet vi får alle dele til at spille sammen med det samme, og ikke sætter det tilsammen til sidst, hvor der kan forkomme et store problemer.

8.4 Test af software

8.4.1 Modultest

8.4.2 Integrationstest

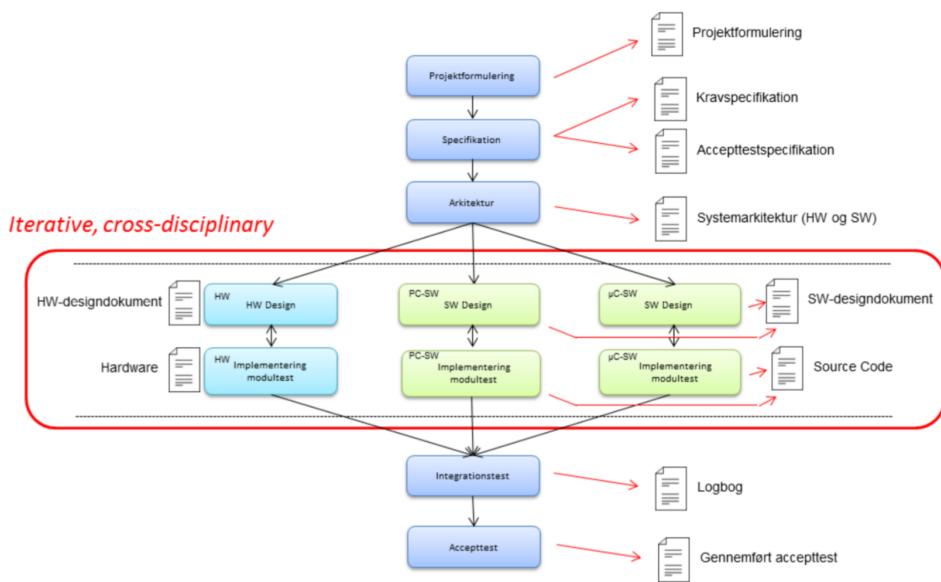
9 Resultater

- 9.1** Hardareresultater
- 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 Samarbejds aftale

Vi har anvendt en samarbejds aftale, som er udarbejdet i fællesskab i gruppen[1]. Udbyttet ved brug af samarbejds aftalen, er et bedre gruppearbejde. Når de regler og krav der er stillet til gruppedellemmerne samt projektet, bliver formuleret og skrevet ned, vil man undgå diskussioner forvirring omkring disse, da man altid kan hive samarbejds aftalen 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]. Udo over 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

I fremtidige versioner af blodtryksmåleren er der dele af systemet, som ville være oplagt at lave forbedringer af både i henhold til funktionalitet og brugervenlighed.

Gemme-funktion

I vores program foregår processen med at gemme oplysninger fra operation til slut, hvilket er meget risikabelt idet at vi ikke imødekommer naturlige uforudsigelige fejl, såsom strømafrydelse af systemet. Denne uheldige situation kan blive en mindre kritisk situation, hvis der i løbet af operation bliver gemt data hver 5. minut fremfor kun at gemme til sidst. Udo over at det kunne være optimalt at gemme flere gange i løbet af en operation burde den fremtidige udgave også gøre det muligt at tilføje patienten til en database, hvor data'en kan blive gemt fremfor at gemme i en tekst fil. På den måde er alle målinger, der er blevet foretaget under operationen hele tiden koblet til patientens oplysninger. Ydermere ville det være ideelt, hvis der i programmet var mulighed for at indhente tidligere målinger en patient, når patienten er blevet tilføjet i systemet. Det vil gøre at man kan gå ind og analysere tidligere målinger. En sammenkobling med den elektroniske patientjournal ville gøre at alle målinger automatisk blev gemt i den pågældende patients journal og kunne være med til at optimere programmet.

Teknisk personale

Det tekniske personale har til opgave at udføre kalibreringen korrekt. Når der er tid til en kalibrering skal en fra det tekniske personale udføre kalibreringen uden at logge ind. For at optimerer denne proces ville det være godt, hvis der blev knyttet et ID til det tekniske personale. Dette vil gøre at det tekniske personale logger ind med et ID hver gang der skal foretages en kalibreringen, og dermed kant tracke, hvem der har udført processen.

Brugergrænseflade

Ifølge standarderne 60601-1-8, så er der nogle krav til symbolerne der bruges på brugergrænsefladen. På vores brugergrænseflade benytter vi primært tekst i stedet for symboler, og det kunne derfor være en god idé i fremtiden at gøre brug af symboler i stedet for tekst.

Alt efter indretning af operationsrummet ville det være optimalt, hvis man skifte farven for baggrunden på brugergrænsefladen. Når systemet bliver anvendt om natten kunne det derfor være behageligt, for både patient og personalet at baggrunden er sort, da der vil være mindre lys fra skærmen. Der kunne med fordel vælges en lysere baggrund om dagen, så det er nemmere at se skærmen.

Hardware

Rent hardwaremæssigt kunne der laves nogle forbedringer i fremtiden. Først og fremmest kunne det være oplagt, hvis strømforsyningen kom fra computeren i stedet for at have Analog Discovery som strømforsyning. Dette vil gøre at systemet vil bestå af mindre dele, da Analog Discovery kan blive unødvendig idet computeren nu kan virke, som vores strømforsyning. For at det kan

lade sig gøre kræver det, at printpladen laves større, så der er plads til en spændingsdeler på printet. Spændingsdelen vil dele de 5 volt, som computeren afgiver så subtraktoren stadig får 2 volt. Vi har i vores printplade fået lavet 1 hul i hvert hjørne, så den er klar til at blive skruet fast inde i en boks, som vil have funktionen at beskytte vores printplade. Det kunne derfor være en smart ting at få udarbejdet i fremtiden.

13 Konklusion

14 Bilag

15 Referencer