

Bloodpressure 3000

3. semesterprojekt

Version 2.5

Ingeniørhøjskolen Aarhus



Vejleder: Peter Johansen

Afleveret: 17/12-2014

Studienummer	Navn	Studieretning
201371026	Camilla Lund Pedersen	ST
201371015	Charlotte Søgaard Kristensen	ST
201370750	Kasper Kronborg Larsen	ST
201370826	Kathrine Duus Kinnerup	ST
201205775	Mads Morgen Kusk	ST
201371061	Thuvaharan Karunakaran	ST

Abstract

The purpose of this project has been to develop an invasive blood pressure monitor with the ability to amplify, convert and display a blood pressure signal. The finale product must have the capability to obtain and visualize these values continuously on a graph.

Obtaining the values happens by connecting the system to an artificial heart pump mechanism, which gives a physiological correct heart signal. The pressure from the pump mechanism is converted into small voltages, which must be amplified by an amplifier. The amplifier along with a low-pass filter is part of the assignment and must be implemented in the project. The implemented low-pass filter filtrates the signal past the 50 Hz cut-off frequency. This is the hardware part of the project.

The software part consist of obtaining the values continuously. An A/D converter digitalizes the signal and is connected to the computer via an USB-portal. By using the architecture, 3-tier-layer method the application has been split in three layers. The Data Access Layer makes the connection to the A/D converter and obtain the values through there. In the Business Layer the signal is converted into pressure, which the Presentation Layer displays to the user.

Working on the project the group gained experience with different learning tools e.g. containing the use of SysML, programming, designing and implementing hardware. Managing the project has been done by using parts of the project process SCRUM.

The project achieved most of its goals, but some parts of the software were lacking. The alarm and digital filter did not work due to overloading the system, but in a future project this could be fixed given more time. The acceptance test revealed that most of the project has been a success.

Ansvarsfordeling

Gruppen har fra start været opdelt i en software og en hardware gruppe, bestående af tre gruppemedlemmer i hver af de to grupper. Software gruppen består af Camilla Lund Pedersen, Charlotte Søgaard Kristensen og Thuvaharan Karunakaran, mens hardware gruppen består af Kasper Kronborg Larsen, Katrine Duus Kinnerup og Mads Morgen Kusk. Alle overordnede beslutninger er vedtaget i fællesskab, og et samlet slutresultat er opnået.

Resume

Formålet med dette projekt har været at udvikle et invasiv blodtryksmålesystem, der kan forstærke, konvertere og vise et blodtrykssignal. Det endelige produkt skal kunne vise blodtryksværdier og -graf kontinuert.

Måling af værdier sker ved at forbinde systemet til en pumpe, der kan simulere et fysiologisk korrekt hjertesignal. Trykket fra pumpen bliver konverteret til små spændinger, som skal forstærkes af en forstærker. Forstærkeren er sammen et lavpasfilter en del af projektets krav. Det implementerede lavpasfilter filtrerer signalet ved en 50 Hz knæfrekvens. Dette udgør hardwaredelen af projektet.

Software delen består af at behandle data kontinuert. En A/D converter digitaliserer signalet og forbindes til en computer via en USB-port. Ved brug af 3-lagsmodellen deles programmet i tre lag. Data Acces Layer laver forbindelsen til A/D converteren og opsamler data herfra. I Business Layer bliver data omregnet til tryk, og disse data viser Presentation Layer til operatøren.

I arbejdet med projektet har gruppen arbejdet med forskellige metoder, som fx SysML, programmering, design og implementering af hardware. Til projektstyring er der brugt dele af metoden SCRUM.

De fleste mål med projektet er opnået, men nogle funktioner i softwaren er ikke opnået. Alarm og det digitale filter virker ikke grundet overbelastning af programmet. Dette kunne i et fremtidigt arbejde rettes op på. Accepttesten viser at størstedelen af projektet har været en succes.

Indholdsfortegnelse

ABSTRACT.....	2
ANSVARSFORDELING.....	3
RESUME.....	3
INDHOLDSFORTEGNELSE	4
VERSIONSHISTORIK	6
ORDLISTE.....	7
1.0 PROJEKTBEKRIVELSE	9
1.1 PROJEKTAFGRENSNING	9
2.0 INDLEDNING	10
3.0 SYSTEMBEKRIVELSE	11
4.0 TEORIAFSNIT.....	12
4.1 HJERTETS ANATOMI OG FYSIOLOGI	12
4.2 BLODETS VEJ GENNEM HJERTET	13
4.3 BLODTRYK.....	13
4.3.1 Diastolen	13
4.3.2 Systolen	14
4.3.3 Normalværdier og forhøjet blodtryk	15
4.4 HARDWARETEORI.....	16
4.4.1 Det medicinske instrumenteringssystem.....	16
4.4.2 Tryktransducer	16
4.4.3 A/D converter	18
4.5 SOFTWARE TEORI	18
4.5.1 3-lagsmodellen	18
4.5.2 SOLID-principperne.....	19
4.5.3 Et godt Presentation Layer.....	19
4.5.4 Producer consumer princippet	19
4.5.5 Observer-mønster.....	20
4.5.6 Blocking collection.....	20
5.0 PROJEKTGENNEMFØRSEL	21
5.1 KOMMUNIKATIONEN	21
5.1.1 Sociale medier	21
5.1.2 Samarbejdsaftale	21
5.1.3 Tidsplan	21
5.1.4 Logbog.....	21
5.1.5 Dropbox.....	21
5.1.6 Rollefordelinger	21
5.2 UDVIKLINGSVÆRKTØJER.....	22
5.2.1 Programmer	22
5.2.2 Måleudstyr	22
5.3 METODER.....	22
5.3.1 SysML (System Modelling Language).....	22
5.3.2 SCRUM.....	23
5.3.3 V-modellen	24
5.3.4 Vandfaldsmodellen.....	25
5.3.5 FURPS	26

5.3.6 MoSCoW.....	26
5.3.7 Reviews.....	27
5.3.8 Enhedstest	27
5.3.9 Integrationstest	27
6.0 KRAV.....	28
7.0 UDVIKLINGSDOKUMENTATION.....	30
7.1 HARDWARE.....	30
7.1.1 Design.....	30
7.1.1.1 Allokering.....	30
7.1.1.2 Forstærker	30
7.1.1.3 Analogt lavpasfilter	31
7.1.2 Implementering	32
7.1.2.1 Kredsløbsopstilling.....	33
7.1.2.2 Kalibrering.....	33
7.2 SOFTWARE.....	34
7.2.1 Design.....	34
7.2.1.1 Klasser i softwaren.....	34
7.2.1.2 Digitale filter	35
7.2.2 Implementering	37
7.2.2.1 Indhentning af data	39
7.2.2.2 Presentation Layer	39
7.2.2.3 Business Layer.....	40
7.2.2.4 Data Access Layer	41
7.2.2.5 Metoder og klasser	41
7.2.2.6 Digitalt filter	43
7.3 TESTS.....	44
7.3.1 Hardware.....	44
7.3.2 Software	44
8.0 RESULTATER OG DISKUSSION	45
8.1 RESULTATER.....	45
8.2 DISKUSSION	46
8.2.1 Diskussion af accepttest	46
8.2.2 Diskussion af software	47
8.2.2.1 Diskussion af digitalt filter	48
8.2.3 Diskussion af hardware	49
8.2.4 Diskussion af arbejdsproces og metode	50
8.2.4.1 SCRUM	50
8.2.4.2 Logbog	50
8.2.4.3 Ansvarsfordeling	50
9.0 PERSPEKTIVERING.....	51
10.0 KONKLUSION	52
11.0 BILAG	52
12.0 REFERENCELISTE	53

Versionshistorik

Version	Dato	Beskrivelse af ændringer	Initialer
1.0	1/12-2014	Dokument oprettet. Tilføjelse af allerede skrevne afsnit.	MMK, TK
2.0	2/12-2014	Opbygning ændret.	MMK, TK
2.1	2/12-2014	Projektbeskrivelse, systembeskrivelse, materialer og metoder + udviklingsdokumentation (HW) tilføjet	MMK, TK
2.2	11/12-2014	Rettelser	MMK, TK
2.3	12/12-2014	Rettelser	MMK
2.4	15/12-2014	Indsat rettelse i afsnittet udviklingsdokumentation Diverse rettelser af formuleringer og afsnit	CLP, CSK
2.5	16/12-2014	Endelige rettelser i hele dokumentet	Alle

Ordliste

Herunder ses en liste over de forkortelser og fagtermer, der er brugt i rapporten og deres betydning samt en ordforklaring til de anvendte begreber:

BDD	Blok definitions diagram – bruges til systembeskrivelse i forbindelse med SysML.
Boundary klasse	En klasse, der benyttes i en applikationsmodel, og som er den klasstype, der har forbindelse til brugeren.
Cirkulær buffer	Hukommelsescyklus i programmering, der tømmes og fyldes konstant.
CMRR	Common Mode Rejection Ratio – hvor godt et filter er til at filtrere common mode støj fra.
Common mode støj	Støj der påvirker på to tætsiddende og lige lange ledninger, og som begge refererer til stel.
Controller klasse	En klasse, der benyttes i en applikationsmodel, og som er den klasstype, der har med logiske operationer at gøre.
Cut-off frekvens	Knækfrekvens. Den frekvens der bestemmer hvornår signalet skal dæmpes.
Disposed	Når en tråd eller task er afbrudt.
Domæne klasse	En klasse, der benyttes i en applikationsmodel, og som er den klasstype, der har med systemets information at gøre.
Gain	Forstærkning.
Gridlines	Linjer, der inddeler en graf i intervaller.
IBD	Internt Blok Diagram – bruges til systembeskrivelse i forbindelse med SysML.
IHA	Ingeniørhøjskolen Aarhus.
Instrumentationsforstærker	En type operationsforstærker egnet til brug af fysiologiske målinger.
Invasiv	Indføring af instrumenter gennem naturlige kanaler i legemet.
LSB	Least significant bit.
Peak	Kraftigt udslag i amplitude på karakteristikker.
Sensitivitet	Sensitivitet beskriver en transducers statiske overføringsfunktion. Det vil sige, sensitivitet er hvor meget output ændrer sig, når input ændres.

Støj	Uønskede frekvenser.
Task	En form for tråd.
Tråd	Et objekt i programmering, der gør parallel afvikling af opgaver mulig.
Transducer	Komponent til omdannelse af én energiform til en anden – i dette tilfælde fra tryk til spænding.

1.0 Projektbeskrivelse

Dette projekt har til opgave at udvikle et system til invasiv måling af blodtryk. Projektet består af en hardware- og en softwaredel. Der skal i hardwaredelen laves et elektronisk kredsløb til forstærkning og filtrering af et simuleret blodtrykssignal. Softwaredelen skal gøre det muligt at afbillede blodtrykket på en grafisk brugerflade og gemme dette signal i en CSV-fil. Der skal ydermere designes et digitalt filter, som skal implementeres i blodtryksapplikationen.

Projektet vil være dokumenteret vha. følgende dokumenter:

- Rapport
- Kravspecifikation
- Accepttest
- Softwaredesign
- Hardwaredesign
- Softwareimplementering
- Hardwareimplementering

1.1 Projektafgrænsning

Der skal undervejs i projektet være tests af implementerede delelementer, og arbejdsprocessen vil blive bygget op efter V-modellen.

IHA stiller teknik til rådighed til gennemførsel af projektet. Hver gruppe har fået udleveret en Digilent Analog Discovery der vha. programmet WaveForms kan simulere et analogt signal. En A/D converter i form af National Instruments DAQ 6009 vil blive brugt til opsamling og konvertering af data analog til digital form.

Sidst bliver en pumpeopstilling i CAVE-lab stillet til rådighed. Pumpeopstilling kan levere et signal lignende et korrekt fysiologisk blodtrykssignal. Dette bruges til udføre accepttesten for systemet.

Der er fra IHA's side defineret nogle krav til projektets indhold:

- Et elektronisk kredsløb, som forstærker signalet fra det væskefyldte kateter og filtrerer det med et indbygget lavpasfilter
- Et program til at vise blodtrykket som funktion af tiden. Programmet skal have en række obligatoriske krav. Det skal:
 - Programmeres i C#
 - Vise blodtrykssignalet kontinuert
 - Kunne lagre de målte data i comma-separated values (CSV) filer
 - Kunne filtrere blodtrykket i selve programmet (tændt/slukket)

Der stræbes efter at implementere et fuldt funktionsdygtigt blodtryksmålingssystem. Ydermere har gruppen valgt at implementere en alarmfunktion, som alarmerer, hvis blodtrykket overskrider indtastede grænseværdier og afbillede blodtryksværdier samt middelblodtryk.

2.0 Indledning

Blodtryk er et udtryk for hjertets pumpefunktion og kan bruges til at danne et billede af en patients helbred. Det kan måles på mange forskellige måder, hvor den direkte blodtryksmåling anvendes, når der er brug for nøjagtige målinger. En sådan direkte måling udføres invasivt på patienten, hvorimod en indirekte måling foretages med en trykmanchet non-invasivt.

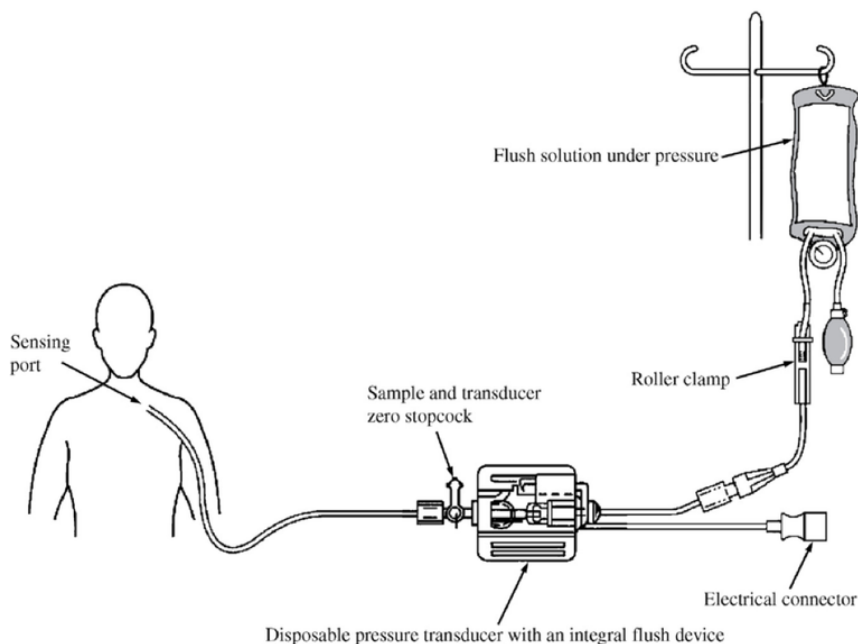
I det danske sundhedsvæsen anvendes invasive blodtryksmålinger, der kontinuert viser et nøjagtigt blodtryk, som danner et billede af patientens tilstand over en længere tidsperiode. Ofte bruges den direkte målemetode på intensive afdelinger og under operationer, hvor det er vigtigt at kende til pludselige svingninger i patientens blodtryk.

Da denne målemetode fortæller meget om patientens helbred og er ofte klinisk anvendt, kunne det være interessant at arbejde med dette. De blodtryksmålinger, som kommer fra patienten, kræver et system, der kan aflæse patientens data og afbillede dem, så en sundhedsfaglig kan benytte sig af dem.

Derfor vil der i dette projekt udvikles en invasiv blodtryksmåler, som kan vise blodtryksmålingerne grafisk på en brugergrænseflade.

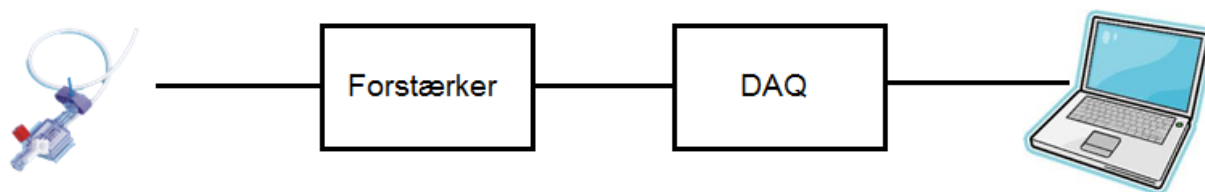
3.0 Systembeskrivelse

Systemet skal invasivt måle et arterielt blodtryk gennem et kateter, vha. en transducer. Blodtrykket føres igennem det væskefyldte kateter, og transduceren registrerer trykændringer, som transformeres til elektriske spændinger. Opstilling med væskefyldt kateter og patient ses på figur 1.



Figur 1 Direkte blodtryksmåling med væskefyldt kateter (21)

Disse spændinger/signaler sendes videre til en instrumentationsforstærker, som forstærker signalet op til en målbar størrelse, og som en computer senere kan registrere. Inden signalet kan modtages af computeren/softwareprogrammet, skal det konverteres fra et analogt til et digitalt signal. Derfor sendes signalet gennem en A/D converter. Til sidst ender signalet i computeren, hvor der kan analyseres på det forstærkede digitale signal og vises en blodtrykskurve på brugergrænsefladen. Dette flow ses på figur 2.



Figur 2 Beskrivelse af systemet

Når systemet testes, kan det ikke gøres helt korrekt med en invasiv blodtryksmåling på en patient. I stedet bruges en testopstilling i CAVE-lab, som kan simulere flowet og trykket i venstre ventrikel. Derudover kalibreres systemet med en statisk væskesøjle.

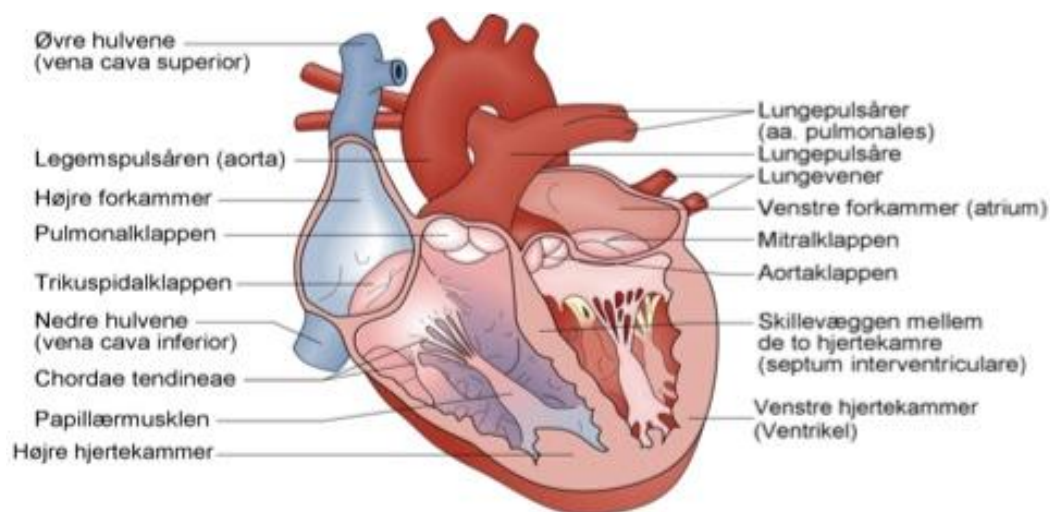
4.0 Teoriafsnit

4.1 Hjertets anatomi og fysiologi

Alle kroppens organismer og celler afhænger af, at der er en konstant tilførsel af næringsstoffer og ilt samtidig med, at affaldsstofferne fragtes væk. Hjertet er det centrale muskulære organ, der sikrer, at blodet bliver pumpet rundt i kredsløbet.

Hjertet består af fire kamre - to forkamre, som er højre og venstre atrium og to hjertekamre, der kaldes ventrikler. Det, der adskiller hjertekamrene fra forkamrene, er en plade af bindevæv, annulus fibrosus, der går skråt gennem hjertet. Annulus fibrosus består af fire bindevævssringe, der er forbundet med hinanden, hvor to af bindevævssringene udgør åbningerne mellem atrierne og ventriklerne. De to andre bindevævssringe danner åbningerne mellem højre ventrikel og lungepulsåren, pulmonalarterien, samt venstre ventrikel og hovedpulsåren, aorta. Ud fra disse bindevævssringe går der tynde klapper ud. Imellem atrierne og ventriklerne sidder atrioventrikulærklapperne, mens semilunarklapperne er placeret imellem ventriklerne og udløbsarterierne. Klapperne er udstyret med flige, hvis bevægelser er styret af de lokale tryk- og flow-forhold.

AV-klapperne består af bicuspidalklappen/mitralklappen, som er den venstre AV-klap, og tricuspidalklappen, som er den højre AV-klap. Netop disse hjerteklapper er hjertet udstyret med for, at blodet flyder den rigtige vej, og dermed fungerer klapperne som en slags ventil. Tricuspidal-, pulmonal- og aortaklappen sørger altså for at regulere og styre blodet i den rigtige retning, (16)



Figur 3 Hjertets opbygning (3)

4.2 Blodets vej gennem hjertet

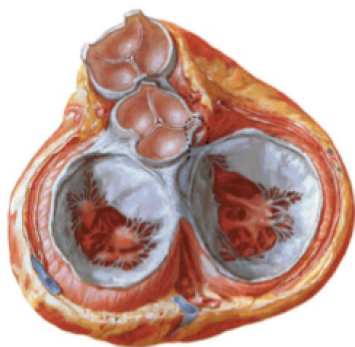
Hjertet har til formål at pumpe blodet rundt i kroppen, som sker ved at skabe trykforskelle i hjertet, så blodet kan komme ud i kredsløbet. Hjertets pumpecyklus deles op i to hovedfaser – den systoliske og den diastoliske fase. Når hjertet kontraherer, kaldes det systole, og når det relaxerer kaldes det diastole.

4.3 Blodtryk

4.3.1 Diastolen

Diastolen begynder, når ventriklernes påførte tryk bliver lavere end trykket i atrierne. Når dette sker, vil AV-klapperne åbnes, hvorved blod strømmer ned i ventriklerne. Højre ventrikel modtager iltfattigt blod fra kroppen igennem de store hulvener og pumper det ud igennem pulmonalarterien og videre til lungerne hvor der sker en gasudveksling af O_2 - og CO_2 . Venstre ventrikel modtager det iltede blod fra lungerne igennem pulmonalvenen, hvor det igennem aorta pumpes ud i kroppen. I den første tredjedel af diastolen udvider hjertet sig hurtigt, hvor opfyldning af ventriklen sker hurtigt grundet den samlede mængde af blod i atrierne under ventriklens kontraktion. I den anden tredjedel af diastolen flyder der kun en lille mængde blod direkte fra venesiden gennem atrierne ned i ventriklerne. I sidste tredjedel sker der en kontraktion af atrierne, hvor de sidste 20-30 % af ventriklernes fyldning sker.

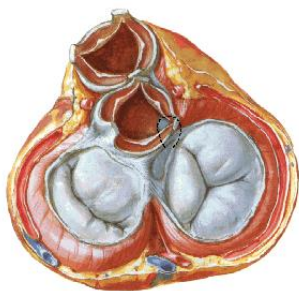
Diastolen kaldes også fyldningsfasen/afslapningsfasen. (16)



Figur 4 Hjerteklapperne under diastolen (8)

4.3.2 Systolen

Systolen er kendt som kontraktionsfasen af ventriklerne. Når ventriklerne kontraheres, vil trykket i ventriklerne blive højere end trykket i atrierne, hvilket resulterer i, at AV-klapperne lukkes, hvor blodet vil være forhindret i at strømme til atrierne.

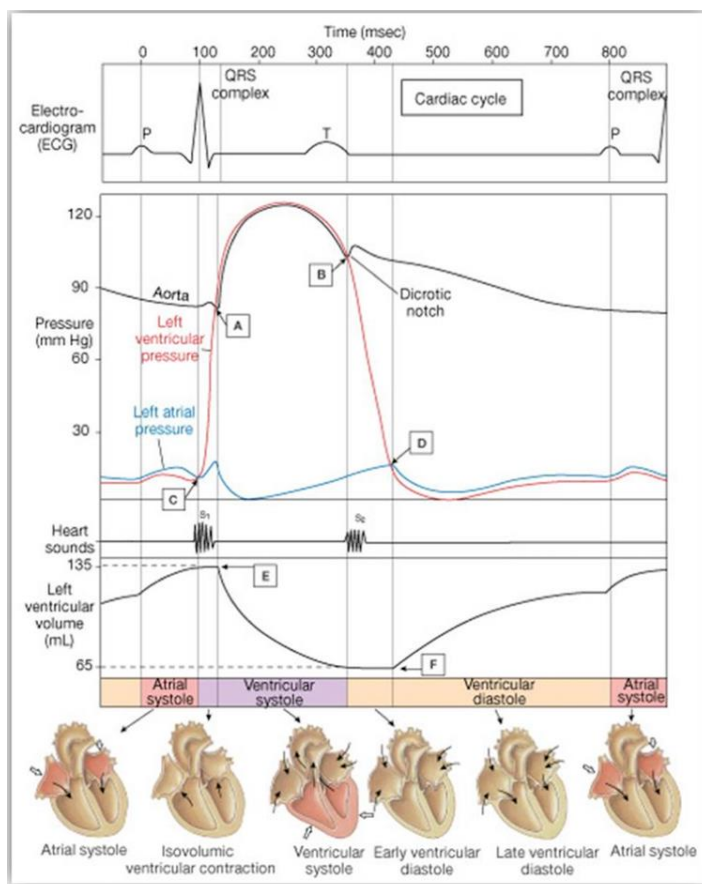


Figur 5 Hjerterklapperne under systolen (8)

På figur 6 ses tre trykkurver samt en tidsakse, hvor mmHg ses op af x-aksen. Den sorte graf er trykket i aorta, den røde graf er trykket i venstre ventrikel, og den blå graf er trykket i venstre atrium. Det er kun venstre side af hjertet, der er illustreret. Trykket i venstre ventrikel stiger i forbindelse med sammentrækningen af ventriklens muskel. Når trykket i venstre ventrikel overstiger trykket i aorta, åbnes aorta klapperne. Blodet kommer ud i aorta, og trykket stiger. Trykket i aorta stiger indtil det bliver højere end trykket i den venstre ventrikel, hvorefter det maksimale systoliske tryk opnås.

I slutningen af systolen afslappes muskelfibre i ventriklen, hvorved trykket i ventriklen falder og bliver lavere end trykket i aorta. Det betyder, at aortaklappen lukkes. Det er også det man kan se på grafen, hvor der er en lille bølge top, som kaldes aortic notch.

I punkt D krydser de to trykkurver hinanden, og her bliver mitralklappen lukket. Grunden til at mitralklapperne lukkes er, at ventriklerne afslappes, og trykket inde i dem bliver lavere end i atrierne. Når dette sker, vil AV-klapperne åbnes, hvorefter hjertet vil gå ind i en ny hjertecyklus. En hjertecyklus består altså af en fyldningsfase/afslapningsfase, som er diastolen og en udtømnings-/kontraktionsfase, som er systolen. (8) (9)



Figur 6 Overblik over forskellige signaler undervejs i hjertets cyklus (9)

4.3.3 Normalværdier og forhøjet blodtryk

Blodtrykket afhænger af, hvor stor en kraft hjertet trækker sig sammen med, når blodet skal pumpes rundt, samt hvor stor modstanden er i de perifere blodkar. Hvis det systoliske tryk er under 140 mmHg, og det diastoliske tryk er under 90 mmHg, i hvile, har man et normalt blodtryk. Der er forskellige faktorer, der spiller ind og har indflydelse på blodtrykket, og derfor er det en tommelfingerregel. Der er fx andre retningslinjer for diabetespatienter.

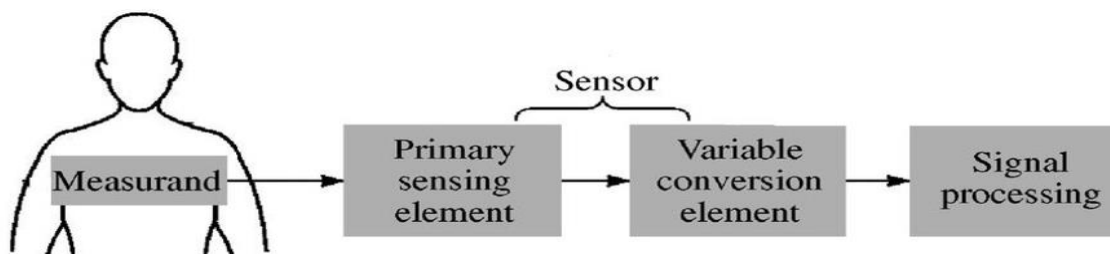
Hvis blodtrykket f.eks. er 160mmHg/100mmHg vil man have et forhøjet blodtryk, som kaldes hypertension. Ved hypertension hos en patient er der på sigt anledning til hjerte/karsygdomme, som eksempelvis blodprop i hjertet, åreforkalkning og hjertesvigt.

Hypertension ses hos 10-20% af befolkningen. Omkring 10% af befolkningen fra 35-85+ år har permanent hypertension, hvilket kræver blodtryksænkende behandling. (17)

Ifølge Dansk Almen Medicinsk Database får 33,2%, af de 37,651 patienter med dokumenteret hypertension blodtrykskontrol. (18) Altså kun en tredjedel af patienter med behov for behandling, får denne foretaget. Ved indlæggelser er det derfor relevant at monitorere disse patienter.

4.4 Hardwareteori

4.4.1 Det medicinske instrumenteringssystem



Figur 7 Det medicinske instrumenteringssystem med de funktionelle komponenter (21)

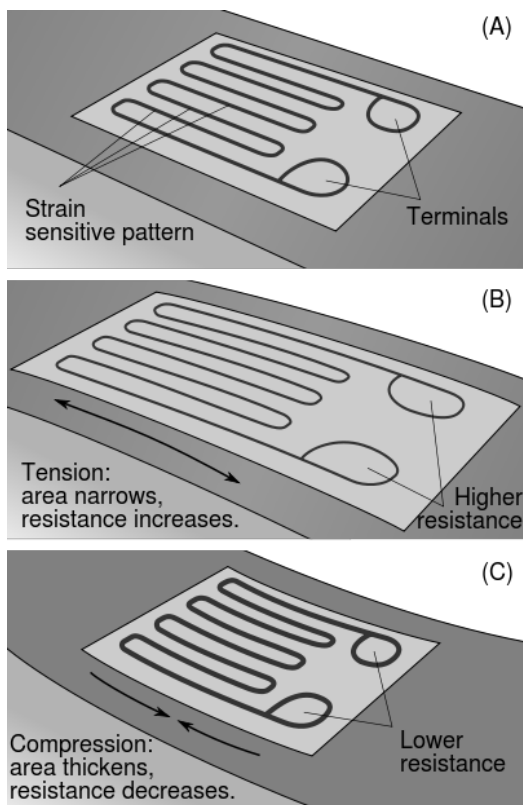
Et medicinsk instrumenteringssystem består af funktionelle komponenter, som ses på figur 7. På figuren går flowet fra venstre mod højre. Først ses målestørrelsen, som er det fysiske målebånd. I dette tilfælde er det blodtrykket, der måles. Transduceren/sensoren omformer signalet fra en fysisk måleenhed til et elektronisk signal. Der er to sensorelementer i en transducer – et primært og et variabelt element. Det primære sensorelement omformer forskydning til en modstandsændring, fx vha. en strain gage, mens det variable element omformer modstandsændring til spænding, fx vha. en wheatstone bro. Det ene sensorelement beskriver transducerens egenskaber ved et konstant input. Det andet sensorelement beskriver transducerens egenskaber ved et tidsvarierende input, hvor sensorer typisk er 1. og 2. orden systemer. Tryktransduceren og strain gages er uddybet i afsnit 4.4.2. Efter sensorens omformning fra tryk til spænding, skal spændingssignal behandles. Dette sker vha. forstærkning og filtrering. Til sidst skal signalet kobles til softwareprogrammet og vises fx som en graf.

4.4.2 Tryktransducer

En tryktransducer konverterer tryk til spænding. Konverteringen fra tryk til spænding sker ved fysisk deformation, fx vha. strain gages. En strain gage består af meget tynde ledere, og hvis disse ledere strækkes og ændres i opbygning, sker der ændring i lederens modstand. Denne modstand, før deformation, kan skrives på følgende måde:

$$R = \frac{\rho * L}{A}$$

I ligningen er modstanden i elektriske leder givet ved ρ , L er længden af den elektriske leder og A er tværsnitsarealet af den elektriske leder. Når disse elektriske ledere deformeres i én retning, sker deformationen også i de andre retninger – altså, en ændring i længden giver en ændring i tværsnitsarealet, som giver ændring i modstanden. Jo større tværsnitsareal, jo mindre modstand får strain gagen, da disse er omvendt proportionale. Dette ses også på figur 8.



Figur 8 Princippet i strain gages (10)

Når der sker en deformation, kan den relative ændring i modstand skrives på følgende måde:

$$\frac{\Delta R}{R} = \frac{\Delta L}{L} - \frac{\Delta A}{A} + \frac{\Delta \rho}{\rho}$$

Strain gages er karakteriseret ved en gage faktor, G . Denne faktor er et udtryk for hvor let strain gagen er at deformere, og dermed hvor let det er at skabe modstandsændring. Heraf kan det siges; jo større en gage faktor jo mindre deformation er nødvendig for at skabe ændring i modstand.

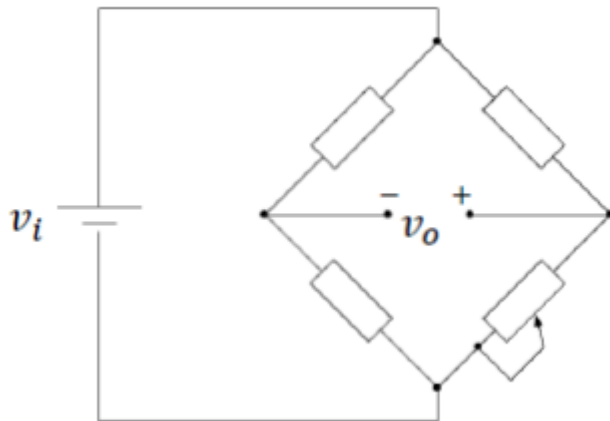
Denne ændring i modstand er proportional med spændingen.

Disse strain gages er tilkoblet en wheatstone bro. En wheatstone bro er et kredsløb der består af to balancerede spændingsdelere. Denne opbygning bevirker at man kan måle den differentielle spændingsforskel mellem disse to spændingsdelere, fremfor at måle absolutte spændingsniveauer. Dette er velegnet til at måle meget små ændringer i modstandene.

Det er muligt at konfigurere sin bro, så man bestemmer hvor mange af de 4 modstande der skal påvirkes af strain gagen. En 1/1-bro giver påvirkning af samtlige modstande i broen, mens en 1/2-bro påvirker halvdelen af broens modstande.

Hvis alle modstandene i broen er ens, vil det give et differentielt output lig 0. Når en eller flere modstandene ændrer sig, vil der ske en ændring i outputtet og dette vil blive forskelligt fra 0. Denne

ændring i modstandene forårsages, når der sker en deformation i strain gagen. På denne måde kommer et elektrisk signal ud af en transducer.



Figur 9 Wheatstone broen (10)

4.4.3 A/D converter

En A/D converter konverterer analoge signaler til digitale signaler, binære tal, hvilket bevirker at computeren kan aflæse og behandle disse signaler. Gruppen har fået tildelt en National Instruments DAQ 6009 til denne opgave, og den tilsluttes til en computer vha. USB 2.0.

4.5 Software teori

4.5.1 3-lagsmodellen

Ud fra filosofien om det gode design, er der taget udgangspunkt i 3-lagsmodellen for at designe softwaren. Denne model deles op i de tre lag, Presentation Layer (PL), Business Layer (BL) og Data Access Layer (DAL). Den fungerer ved, at man laver kald ned gennem lagene. Fra PL kan man kalde metoder i BL, og fra BL kan man kalde metoder i DAL. Der må ikke kaldes metoder den modsatte vej – der må kun sendes returmeddelelser. PL er selve brugergrænsefladen, det øverste lag i applikationen. Lagets ansvar er at oversætte data og gøre disse synlige for operatøren. BL's ansvar er at behandle data og stå for samtlige omregninger i hele systemet. DAL er det nederste lag, og det er her, at alt angående indhentning og lagring af data foregår.

3-lagsmodellen bruges til større programmer, fordi det skaber en naturlig struktur. De forskellige lag bør være forholdsvis uafhængige af hinanden, så man kan udtage fx PL og bruge i en anden applikation. (6)

4.5.2 SOLID-principperne

SOLID-principperne er en måde, hvorpå man kan opbygge sin applikation på den mest hensigtsmæssige måde.

S i SOLID står for Single Responsibility. Dette sikrer, at én klasse har ét ansvar og skaber en bro mellem klasser. O står for Open/closed. Her skal software være lukket for modifikation men åben for udvidelse. L står for Liskov substitution, som sikrer, at objekter kan erstattes med instanser af deres undertyper uden at påvirke programmet negativt. I står for Interface segregation. Her laves der adskillige specifikke interfaces til hver klient/aktør. D står for Dependency inversion. Dette går mod det konventionelle afhængighedsprincip – nemlig at high-level moduler ikke skal være afhængig af lavere moduler i applikationen. De skal begge være afhængige af abstraktioner. (11)

4.5.3 Et godt Presentation Layer

En velfungerende brugerflade er essentiel for, at brugeren af systemet har en god og konstruktiv oplevelse med programmet. For at sikre dette designes der efter 16 principper, som har designet af en god brugergrænseflade i betragtning.

Disse principper indeholder bl.a. at brugerfladen skal være let at overskues, så brugeren nemt kan se, hvad der skal trykkes på for at opnå et ønsket resultat. Dette gøres ved at designe et system sådan, at knapperne og deres størrelse symboliserer deres betydning. Samtidig skal typografien være skrevet sådan, at brugeren kan læse funktionen på f.eks. knapper. Man skal tage højde for brugerens forståelse af programmet og ikke programdesignerens logik. En yderligere vigtighed er, at det skal være tydeligt, om programmet arbejder med noget, eller om det blot venter på at operatøren trykker på en knap. I den sammenhæng er det også vigtigt at programmet viser en reaktion, når operatøren starter et event, så det kan observeres, at der sker en handling. Operatøren bør altid være i kontrol af programmet, så han kan afbryde en handling, hvis det ønskes. Dette er vigtigt for, at man ikke skal vente på, at programmet fuldfører en handling, der ved en fejl blev igangsat. (5)

4.5.4 Producer consumer princippet

For at indhente data til en applikation kan man bruge producer consumer princippet. En producer bruges, når et indhold af data skal overleveres fra en klasse til en anden klasse. Consumeren benytter sig af disse data, og den klasse vil gerne vide, når der sker en ændring af data. Observer-mønstret kan bruges til at opfylde dette. (9)

4.5.5 Observer-mønster

Mønsteret er en metode, hvor man adskiller kode, så man får en lav kobling i et system, og det dermed er nemt at vedligeholde klasserne. Der vil produceren informere consumeren, når data ændres. Consumeren står dermed og forespørger producer-klassen om der er tilgængelige data periodisk.

Mønsteret har fire typer af klasser, man skal have implementeret. En generel Observer-klasse, som ofte laves som et interface, en generel Subject-klasse, en konkret Subject-klasse og en konkret Observer-klasse. Den konkrete Observer-klasse arver fra den generelle Observer-klasse og implementere metoden Update. Den konkrete Subject-klasse arver fra den generelle Subject-klasse og kalder metoden Notify.

Den konkrete Subject-klasse agerer som produceren. Den konkrete Observer-klasse agerer som consumeren og modtager opdateringer om tilgængelig data, når metoden Notify kaldes i den konkrete Subject-klasse. (9)

4.5.6 Blocking collection

For at føre data fra DAL og op til BL er der gjort brug af Blocking collection, der benyttes ved multiple tråde. Det er benytter producer consumer princippet med en bedre trådsikkerhed, da den blokerer for dataflow. Man er sikker på, hvilket data man får. Consumeren har bestemt, hvor mange antal samples, som skal videregives af gangen. Produceren vil derfor indhente data, indtil det antal er udfyldt, hvorefter consumeren får data, og produceren indhenter yderlige data. Data står dermed i kø til at blive hentet op. (13)

5.0 Projektgennemførelse

5.1 Kommunikationen

5.1.1 Sociale medier

Facebook er blevet brugt til at udveksle generelle informationer omkring mødetidspunkter og diverse andre ting. For at gøre det overskueligt og nemt at finde ud af, hvornår gruppen skulle mødes, er Campusnet blevet brugt til at skemalægge de forskellige gruppemøder.

5.1.2 Samarbejdsaftale

I projektets start udformede gruppen også en samarbejdsaftale, som skal klargøre forventningerne hvert gruppemedlem havde til projektet og arbejdsformen.

5.1.3 Tidsplan

Tidsplanen er lavet for at holde styr på reviewmøder og deadlines. I projektet var der på forhånd indarbejdet nogle faste deadlines for reviews og tests for at undgå at være presset til sidst i projektet. Første deadline var kravsspecifikationen og accepttest. Anden deadline var design af hardware og software. Tredje deadline har været at udføre accepttesten for vejlederen, mens fjerde og sidste deadline har været selve projektafleveringen. Tidsplanen er lavet på baggrund af disse deadlines.

5.1.4 Logbog

Efter hvert internt gruppemøde er der blevet skrevet logbog, hvor dagens arbejde og vigtige beslutninger er skrevet ned. Dette har givet gruppen et overblik over opgaver, der var afsluttede, igangværende og opgaver der skulle laves til næste gang.

5.1.5 Dropbox

Dropbox har været brugt til at dele dokumenter med hinanden. Dette har givet alle mulighed for at rette og oprette nye dokumenter som har været tilgængelige for alle. For hver ændring er der gemt en ny version af dokumentet, hvilket er angivet i filnavn og i selve dokumentet. Dette har Dropbox ikke gjort automatisk og er derfor gjort manuelt.

5.1.6 Rollefordelinger

I dette projektforsløb har der ikke været faste administrative roller som ordstyrer og referent, men alle gruppens medlemmer har taget del i de forskellige roller i løbet af projektet. Undervejs i projektforsløbet har gruppen haft kontakt med vejlederen, hvor gruppen fik afklaring på eventuelle spørgsmål. Efter hvert vejledermøde er der blevet skrevet mødereferat skiftevis af gruppens medlemmer.

Under udviklingen af systemets overordnede funktionalitet hvor kravene til systemet blev opstillet, har alle gruppens medlemmer taget del i dette. Dette har betydet, at alle har haft indflydelse på kravene til systemet. Herefter blev softwaredelen og hardware delt op, så tre af gruppens medlemmer hovedsageligt arbejdede med softwaren, mens de resterende arbejdede med hardwaredelen.

5.2 Udviklingsværktøjer

5.2.1 Programmer

Følgende udviklingsværktøjer er anvendt i forbindelse med udviklingen af blodtryksmålesystemet.

- Microsoft Visio 2013 er anvendt for at kunne lave forskellige SysML- og UML diagrammer.
- Microsoft Visual Studio 2013 er anvendt for at udvikle selve programmet og dens brugergrænseflade.
- Microsoft Word er blevet anvendt til selve udarbejdelsen af projektrapporten, projektdokumentationen samt andre relevante tekstdokumenter.
- Multisim er anvendt i forbindelse med simulering og test af det teoretisk opstillede kredsløb.
- Matlab er anvendt i forbindelse med design og implementering af det digitale filter.

5.2.2 Måleudstyr

Følgende udstyr er brugt i forbindelse projektet.

- Funktionsgeneratoren er anvendt til at levere prøvesignaler ved forskellige frekvenser i udviklingen af det elektroniske kredsløb.
- Spændingsforsyning er anvendt som spændingskilde.
- Måleprope er anvendt til at måle med.
- Oscilloskop er anvendt som et måleinstrument, der er brugt til at visualisere elektriske signaler i form af kurver, som vises på en skærm.

5.3 Metoder

5.3.1 SysML (System Modelling Language)

Beskrivelsen af systemet er blevet udarbejdet ved brug af SysML, et modelleringssprog, der bruges til at lave diagrammer/modeller af systemets hard- og software. Modellerne beskriver de forskellige aspekter af systemet både overordnet og i en større detaljegrad.

I projektet er der anvendt både struktur- og adfærdsdiagrammer til at beskrive systemet. Af adfærdsdiagrammer er der benyttet følgende modeller:

- ACT, Aktivitetsdiagram, er en grafisk afbildning af flowet af aktiviteter og aktioner. Den viser iterationer og valg, som finder sted i systemet. Aktivitetsdiagrammet er brugt til at beskrive kodeelementer i en større detaljegrad.
- SD, Sekvensdiagram, bruges til at vise, hvordan programmets objekter interagerer med hinanden. Sekvensdiagrammet er blevet lavet for hver Use Case over de forskellige scenarier samt et sekvensdiagram over det samlede program.
- STM, state machine diagram, bruges til at beskrive tilstande i programmet. Tilstanden beskriver programmets opførsel på et givent tidspunkt, indtil et event får den til at skifte tilstand. Modellen er i projektet benyttet til at illustrere alarmeringen og filtreringen.
- UC, Use Case Diagram, fungerer som en simpel visuel repræsentation af aktørernes interaktion med systemet.

Der er anvendt følgende modeller til at beskrive systemets struktur:

- BDD, Block Definition Diagram, definerer hvordan systemet er bygget op af blokke. Modellen beskriver ligeledes, hvordan hierarkiet og sammenhængen mellem de forskellige blokke, da en blok i systemet kan udgøres af flere andre blokke. Denne diagramtype er anvendt til at beskrive, hvordan blodtryksmåleren er opbygget.
- IBD, Internal Block Diagram, beskriver sammenkoblingerne og grænsefladerne mellem en blok og den information, som flyder mellem dem. Denne type diagram er lavet ud fra BDD'et, da det altid relaterer sig til en blok i BDD'et. Derved benyttes det til at beskrive de interne forbindelser mellem blokkens bestanddele. IBD'et er i projektet benyttet til at beskrive, hvordan information flyder mellem de forskellige hardwarelementer.

Andre former for SysML har været domænemodellen og allokering, hvor domænemodellen er en slags skitse for koncepterne i det overordnede system. Allokeringen beskriver, hvordan man kan gå fra en teoretisk blok, der har nogle krav, til en konkret blok, som opfylder dette. (2)

5.3.2 SCRUM

I dette projekt er der arbejdet med nogle elementer af arbejdsmetoden SCRUM, som er en iterativ og fleksibel ramme for udarbejdelse af et projekt. I projektets indledende fase blev der brugt en variation af

SCRUM, hvor der er brugt en variation af elementet SCRUM board. Et SCRUM board skaber overblikket over status på diverse dele i projektet. I projektet er der blevet benyttet en digital tidsplan og to-do lister, som tilsammen har udgjort SCRUM boardet.

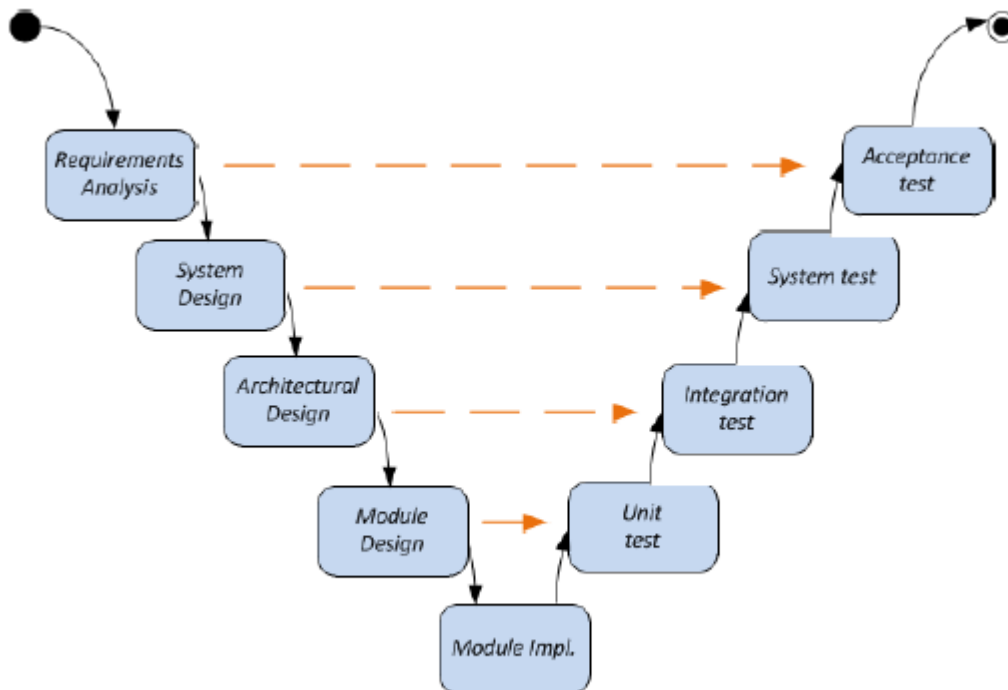
En vigtig del af SCRUM er, at udviklingen sker i en serie af sprints. Undervejs i projektforløbet har der været indlagt flere deadlines, som skulle være overholdt. Tiden frem til deadline har fungeret som overordnede sprints. Derudover er møder blevet aftalt en hel uge afgang, hvor der også er diskuteret, hvad der skal være færdigt i det igangværende sprint. Dermed er der også anvendt en form for "sprint backlog" idet, at der hver uge er blevet valgt elementer, som skulle være færdiggjort.

Metoden SCRUM er valgt, da denne giver et godt overblik over hvilke dele, der mangler at blive lavet. Den er god til at lave aftale efter, da man også for fyldt op på de arbejdsfordelinger, man har haft undervejs i forløbet. Det er også en fordel, at man kan nøjes med at bruge delelementer og ikke følge metoden slavisk.
(2)

5.3.3 V-modellen

I dette projekt er der gjort brug af V-modellen. Der er blevet arbejdet ud fra tre udviklingsfaser, som til sidst har mundet ud i implementeringen. Parallelt med de tre udviklingsfaser er der lavet tilsvarende tests. Ved at teste på forskellige niveauer, er der blevet sikret, at de udviklede delsystemer virker efter hensigten. Den første udviklingsfase har været at udarbejde selve kravspecifikationen, hvor kravene til systemet blev specificeret. Hertil er der blevet lavet en accepttest, som skal sørge for at opfylde de overordnede krav til systemets funktionalitet. Herefter er hardware- og softwaredesignet udviklet og implementeret, og til test af dette er der brugt unittest og integrationstest.

Der er arbejdet iterativt med V-modellen. Der har løbende været ændringer i formuleringerne af kravene, hvilket kendetegner et iterativt forløb. (2)

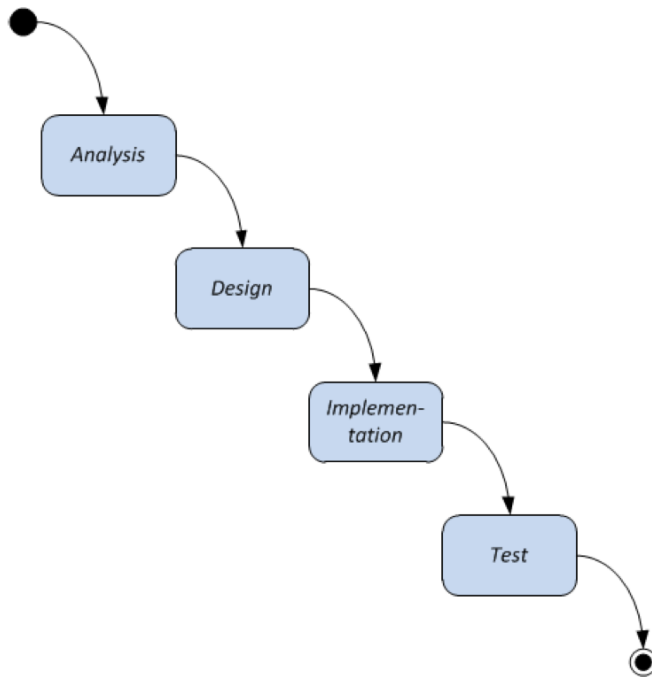


Figur 10 V-modellen (2)

5.3.4 Vandfaldsmodellen

Udgangspunktet for projektet har været v-modellen, som hardwaredelen er udarbejdet efter. Som udgangspunkt er softwaredelen også udarbejdet efter v-modellen, dog med elementer af vandfaldsmodellen. Dette diskuteres nærmere i afsnittet 8.0 Resultater og diskussion.

Vandfaldsmodellen er karakteriseret ved, at faseovergangene sker sekventielt. Når først en fase er afsluttet, er der mulighed for at gennemføre en ny fase. Vandfaldsmodellen består af en analysefase, en designfase og en implementeringsfase, som til sidst munder ud i en test af systemet. Hvert led i systemet afsluttes for sig, og man kan ikke gå tilbage i faserne. Der testes kun til sidst, og hvis der opstår fejl, er det svært at finde frem til årsagen. (2)



Figur 11 Vandfaldsmodellen (2)

5.3.5 FURPS

FURPS definerer de ikke-funktionelle krav, som systemet har. FURPS+ står for:

F: Funktionelle krav, som er angivet i Use Cases.

U: Usability

R: Reliability

P: Performance

S: Supportability

+: ekstra krav til systemet, som ikke hører ind under ovenstående.

Punkt F i FURPS+ er i dette projekt ændret til fysiske krav. Dette punkt beskriver krav til hardwaren. (2)

5.3.6 MoSCoW

MoSCoW-metoden betegner de ikke-funktionelle krav, som systemet skal opfylde (must), bør realisere (should), kunne realisere, men ikke har indvirkning på de andre krav (could), og de krav som omhandler fremtidige opdateringer og udvidelser (would). Kravene i kravspecifikation er prioriteret efter denne fremgangsmåde. (2)

5.3.7 Reviews

Review er blevet brugt i projektet i forbindelse med de to første deadlines i projektet. Der er altså foretaget review af kravsspecifikation og accepttest og af hardware- og softwaredesign, og ved hvert review har der været tilknyttet en reviewgruppe.

Fremgangsmåden har været som følger:

- Reviewgrupperne har udvekslet dokumenter og efterfølgende rettet og skrevet kommentarer til hinandens dokumenter.
- Der er afholdt et reviewmøde, hvor de to grupper har præsenteret rettelser for hinanden.
- Kommentarerne fra reviewgruppen er diskuteret i gruppen og brugt til rettelser af dokumenterne.

Denne metode har gjort, at gruppen har fået et andet syn på projektet, og fundet frem til fejl, som ellers ikke ville blive opdaget.

5.3.8 Enhedstest

Enhedstest bruges til at teste enkelte dele af et system, inden man tester det samlet. I projektet er der lavet enhedstest af både software og hardware. Enhedstest af hardware har bestået i at teste komponenter, lavpasfilter og forstærker for sig (jf. Hardwareimplementering, s. 11). Enhedstest af software har bestået i at teste de forskellige metoder ved at lave nye projekter og bruge de klasser, der er oprettet i det samlede program. Dette er dog først gjort til slut i forløbet og ikke undervejs, når en klasse/metode er færdiggjort (jf. Softwareimplementering, s. 32). (2)

5.3.9 Integrationstest

Integrationstest bruges til at teste det samlede system, efter der udført godkendte enhedstest. Der er lavet integrationstest for hardware for sig og software for sig, og accepttesten har fungeret som integrationstest af det samlede system bestående af både hardware og software. (2)

6.0 Krav

Der er udviklet både software- og hardwarekrav til blodtrykssystemet. For at opnå projektets formål og de obligatoriske opstillede krav fra IHA, er der for det samlede system opstillet tre former for krav: fysiske krav til hardwaren, ikke-funktionelle og funktionelle krav.

Hardwaren består af en forstærker, som skal være i stand til at omforme og forstærke et blodtryksignal til en given spænding. Derfor er der opstillet en række krav (jf. Kravspecifikation, s. 7), som også beskriver udseendet på forstærkeren.

De ikke-funktionelle krav er opstillet for at sikre systemets pålidelighed, mulighed for support og præstationsevne (jf. Kravspecifikation, s. 7).

Det er de funktionelle krav, som beskriver funktionaliteten af systemet, og de beskrives i Use Cases. Systemets sammenspil med aktørerne operatør, patient og datalagring er beskrevet i fully-dressed Use Cases.

UC1 Nulpunktsjustér stiller kravet om en nulpunktsjustering programmet ift. det atmosfærisk tryk (jf. Kravspecifikation, s. 11).

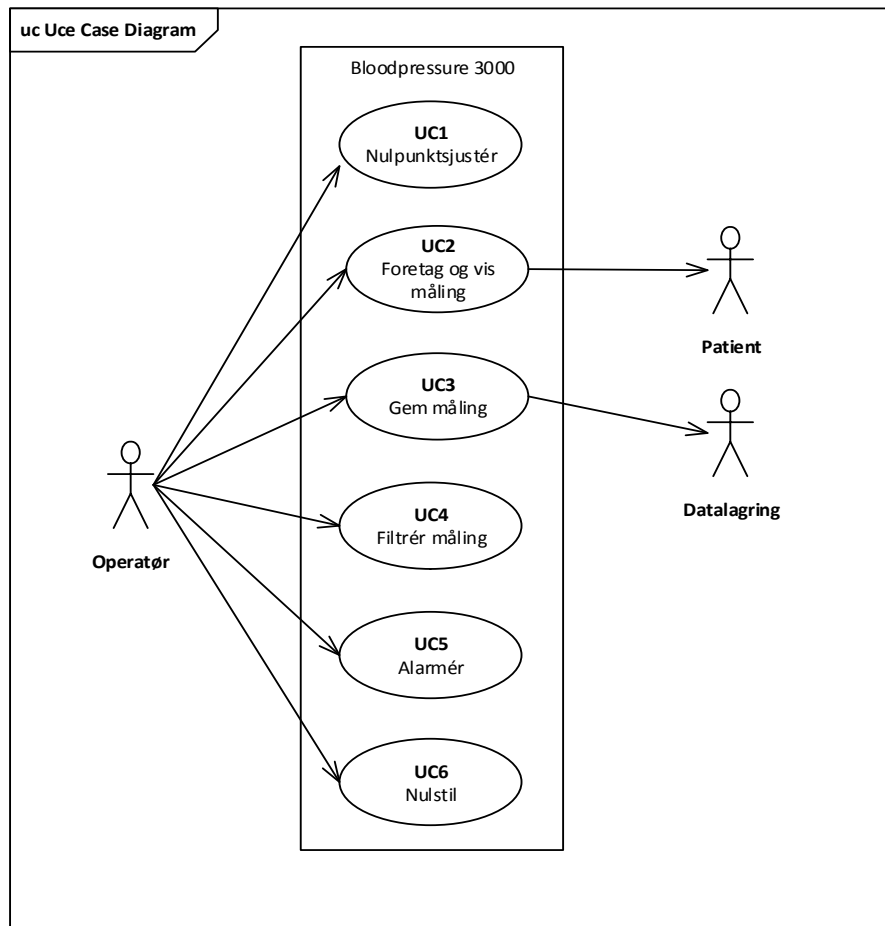
UC2 Foretag og vis måling beskriver kravene til opsamlingen af data, samt hvordan disse skal vises på brugerfladen (jf. Kravspecifikation, s. 12).

UC3 Gem måling indeholder krav om, hvordan målingerne skal gemmes (jf. Kravspecifikation, s. 13).

UC4 Filtrér måling beskriver kravene til det obligatoriske krav om et digital filter (jf. Kravspecifikation, s. 14).

UC5 Alarmér omhandler kravene til en alarmfunktion (jf. Kravspecifikation, s. 15).

UC6 Nulstil er kravene til nulstilling af brugerflade (jf. Kravspecifikation, s. 16).



Figur 12 Use Case diagram

I figur 12 ses en oversigt over systemets Use Cases og aktørernes forbindelser til disse. I projektet er der seks Use Cases, hvoraf alle har en interaktion med operatøren, men patient har kun indflydelse på UC2 Foretagelse af Måling, og datalagring har en interaktion med UC3 Gem Måling.

7.0 Udviklingsdokumentation

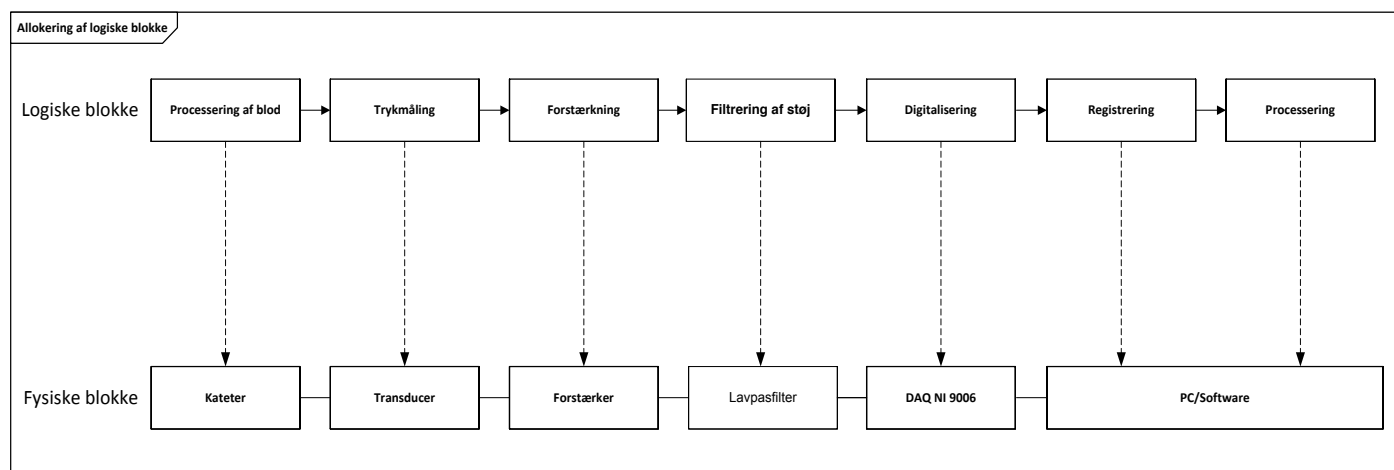
7.1 Hardware

I det følgende afsnit beskrives de væsentligste punkter angående hardwaredesign og –implementering for projektet. For uddybende beskrivelse af design og implementering henvises til dokumenterne ”Hardwaredesign” og ”Hardwareimplementering”.

7.1.1 Design

7.1.1.1 Allokering

I nedenstående diagram gives et overblik over systemets logiske og fysiske blokke, og deres forbindelser.



Figur 2 Allokering af systemet

Ud fra figuren ses det, hvordan blodtrykket ændres gennem systemet. Transduceren omsætter tryk til spænding, hvorefter denne bliver forstærket i forstærkeren. Herefter løber signalet gennem et lavpasfilter, hvor støj bliver filtreret fra. Det filtrerede signal bliver nu digitaliseret i DAQ NI 6009, hvorefter det bliver registreret og behandlet af softwaren.

7.1.1.2 Forstærker

Der er valgt at benytte en instrumentationsforstærker til forstærke det svage signal fra tryktransduceren, så dette kan aflæses af A/D converteren, DAQ NI6009, der blev udleveret fra projektets begyndelse.

Instrumentationsforstærkeren er karakteriseret ved en høj indgangsimpedans, differentielt input og single-ended output, samt gain justering ved ændring af kun én modstand. De tre krav opfyldes af typen INA 114 (jf. Hardwareimplementering, s. 6). Indgangsimpedansen for INA 114 er $10^{10}\Omega$, og forstærkerens CMRR er

115 dB. (21) Samme datablad beskriver formler til beregningerne af både gain og gain-modstanden, R_G . Der beregnes, at den nødvendige gain for forstærkeren er 740, og R_G bliver heraf ca. 68Ω (jf.

Hardwareimplementering, s. 7).

7.1.1.3 Analogt lavpasfilter

Designet af et passende lavpasfilter til blodtrykssystemet tager udgangspunkt i frekvensen for arterielle blodtryksmålinger. En blodtryksmåling indeholder brugbare informationer i intervallet fra 0-50 Hz. (22)

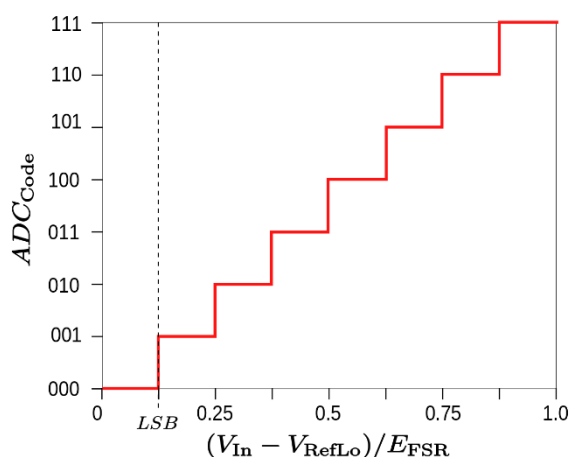
Ved brug af fourieranalyse er der opnået et udtryk for blodtrykssignaler ved at summere et passende antal simple sinusoider og dermed få afbilledet det komplekse blodtrykssignal (jf. Hardwareimplementering, s. 7-9).

Signalet indeholder dog også støj med en frekvens på 50 Hz brum. Denne støj ses som et kraftigt peak ved 50 Hz på frekvenskarakteristikken. Dette skulle gerne filtreres væk med et passende 1. ordens lavpasfilter. Der vælges en cut-off frekvens på 50 Hz, da blodtrykssignalets frekvens er på 0-50 Hz. På denne måde bibeholdes de informationer, der kommer fra signalet (jf. Hardwaredesign, s. 15).

Der vælges en samplingsfrekvens på 1 kHz, som er 20 gange signalfrekvensen for at få så præcis en repræsentation som muligt. Den høje samplingsfrekvens giver det mest nøjagtige resultat og gør, at aliasering undgås (jf. Hardwaredesign, s. 18).

Et 1. ordens lavpasfilter er tilstrækkeligt for at dæmpe støjen fra, da der skal dæmpes med -20 dB/dekade. Argumenterne for at vælge denne type filter omhandler, A/D converteren ift. spændingsområde og bits og blodtryksmålingens frekvenskarakteristik.

Helt grundlæggende får A/D converteren analog spænding, som omsættes til digitale tal. For at finde ud af, hvor lave spændingsforskelle A/D converteren minimum kan arbejde med, dvs. hvad LSB svarer til i spænding, benyttes A/D converterens operative spændingsområde på 10V og de 14 bits, A/D converteren har. For at være sikker på, at der ikke kommer afrunding, udregnes den halve LSB. Når man er under den halve LSB er alle værdierne 0, og rent teoretisk set registrerer converteren ingen ting. Den halve LSB udregnes til 0,3 mV (jf. Hardwareimplementering, s. 18). Dette betyder, at når A/D converteren har 14 bits, vil denne bestå af 2^{14} spændingsintervaller.

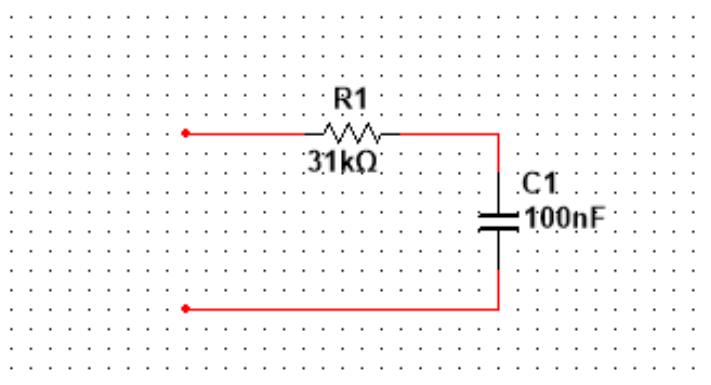


Figur 13 A/D converter kurve over bits (1)

Ved at se på blodtryksspektret ses det, at effekten i signalet ved 50 Hz er 10.000.000 mindre end ved grundtonen ved 1 Hz. Ved udregninger af signalamplituden og kendskab til Nyquist-frekvensen udregnes, at forstærkningen bliver -20db. Dette resulterer i, at der skal gøres brug af et filter med en dæmpning på -20dB/dekade, altså et 1. ordens lavpasfilter (jf. Hardwareimplementering, s. 19-20).

7.1.2 Implementering

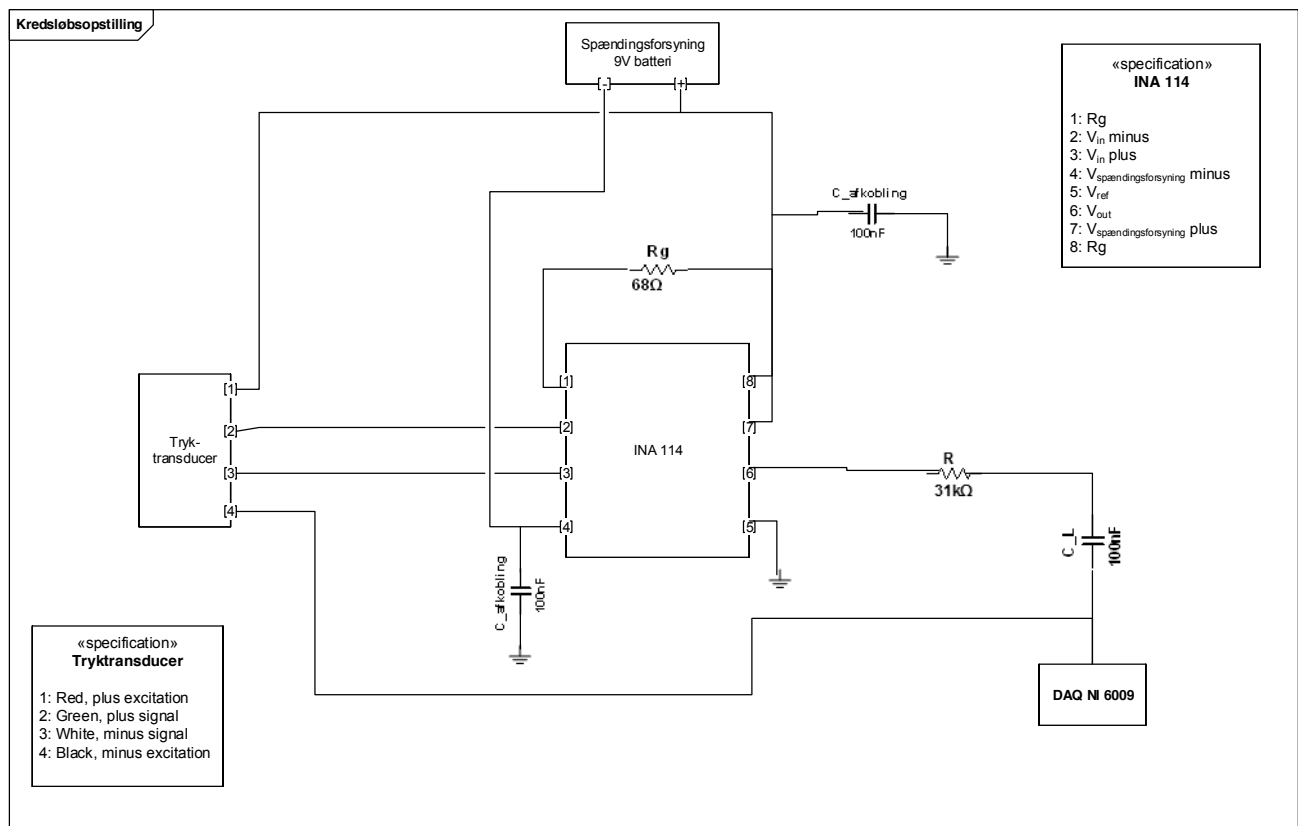
Med en cut-off frekvens på 50 Hz udregnes komponenterne for lavpasfilteret. Filtret kommer til at se ud som figur 14. Der er i implementeringen valgt en modstand på 33 kΩ, grundet de komponenter, der var til rådighed (jf. Hardwareimplementering, s. 21).



Figur 14 Lavpasfilter med komponentværdier

7.1.2.1 Kredsløbsopstilling

Ud fra alle udregningerne kan det elektroniske kredsløb opbygges som på figur 15.



Figur 15 Kredsløbsopstilling

7.1.2.2 Kalibrering

Omregningsfaktoren er fundet ved at tage udgangspunkt i tryktransducerens sensitivitet, som er fundet i databladet. Denne omregningsfaktor benyttes til at øge nøjagtigheden for blodtryksmålingerne. Ved beregninger findes omregningsfaktoren fra tryk og spænding til at være $BP[mmHg] = \frac{V[Volt]}{33,3 \frac{mV}{mmHg}}$. Det betyder, at denne værdi benyttes i softwaren, når omregnes fra spænding til blodtryk (jf. Hardwareimplementering, s. 21).

7.2 Software

I det følgende afsnit beskrives de væsentligste punkter angående softwaredesign og –implementering for projektet. For uddybende beskrivelse af design og implementering henvises der til dokumentationsdokumentet for software.

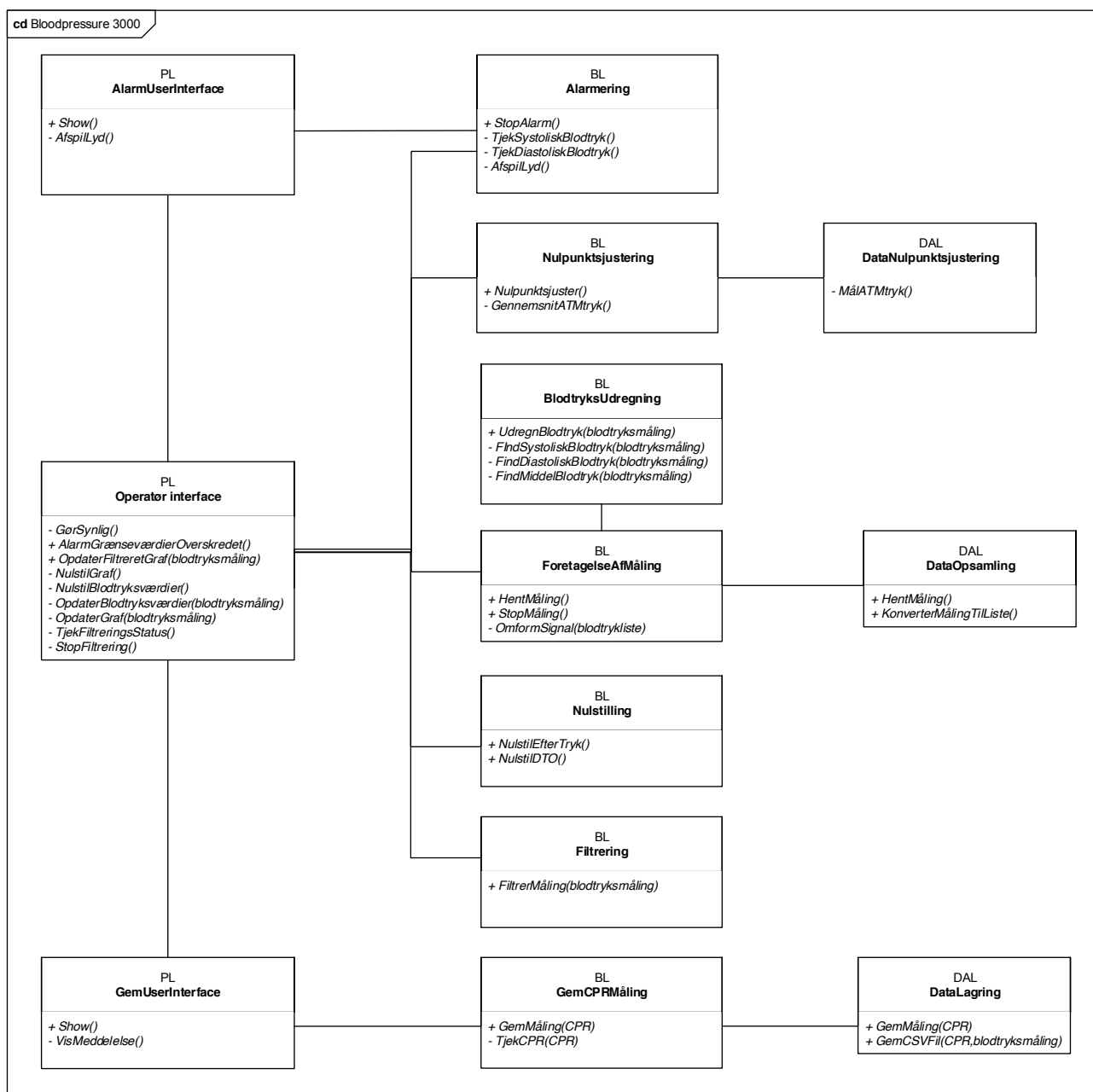
7.2.1 Design

Designet er lavet på baggrund af principperne om lav kobling og høj samhørighed. Disse er essentielle for at opnå et godt design, da det gør systemet nemmere at vedligeholde, give det en ny funktionalitet og finde fejl. Dette opnås ved at bruge SOLID-principperne (jf. afsnit 4.5.2 SOLID-principperne).

7.2.1.1 Klasser i softwaren

Ud fra Use Cases og domænemodellen er der lavet en applikationsmodel. Denne indeholder et klassediagram, sekvensdiagrammer og tilstandsdiagrammer. Dette giver et overblik over, hvilke metoder systemet er tiltænkt skal indeholde, og hvordan disse skal virke. Først findes de konceptuelle klasser ud fra Use Casene, hvor der derefter laves et klassediagram med associationer indeholdende domæne-, controller- og boundaryklasser. Ved derefter at lave sekvensdiagrammer udfyldes disse med relevante metoder for systemet. (jf. Softwaredesign, s. 4).

Det endelige klassediagram med metoder for designet ses på figur 16.



Figur 16 Klassediagram, design

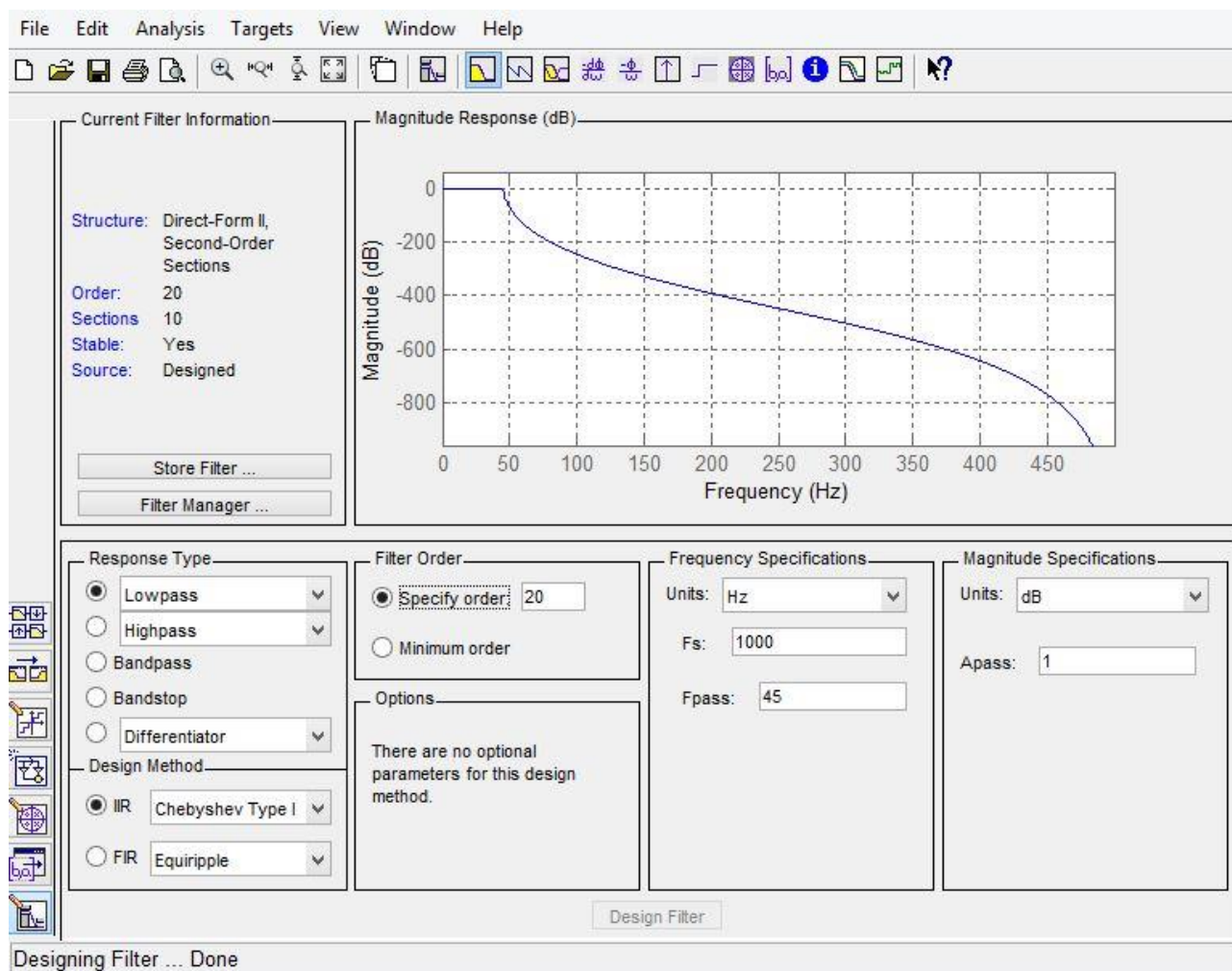
7.2.1.2 Digitale filter

Blodtrykssignalet er overlejret med 50 Hz støj. Da det analoge filter ikke fjerner denne støj med et 1.ordens lavpasfilter, så skal det digitale kunne filtrere denne støj fra.

Yderligere information om signalets frekvenser og karakteristika findes i udviklingsdokumentationen for hardware og i afsnittet i rapporten.

For at kunne designe et passende digitalt filter til filtreringen af de 50 Hz brum støj, benyttes Matlabs filter design og analyse værktøj. Det vides, at blodtrykssignalet indeholder nyttig information i frekvensområde 0-50 Hz, og derfor vælges et lavpasfilter med en cut-off frekvens på 45 Hz.

Fremgangsmåden omkring design af digitale filtre, med henblik på orden, stabilitet og dæmpning, er at prøve sig frem, og kigge på bodeplots for de forskellige filter designs.



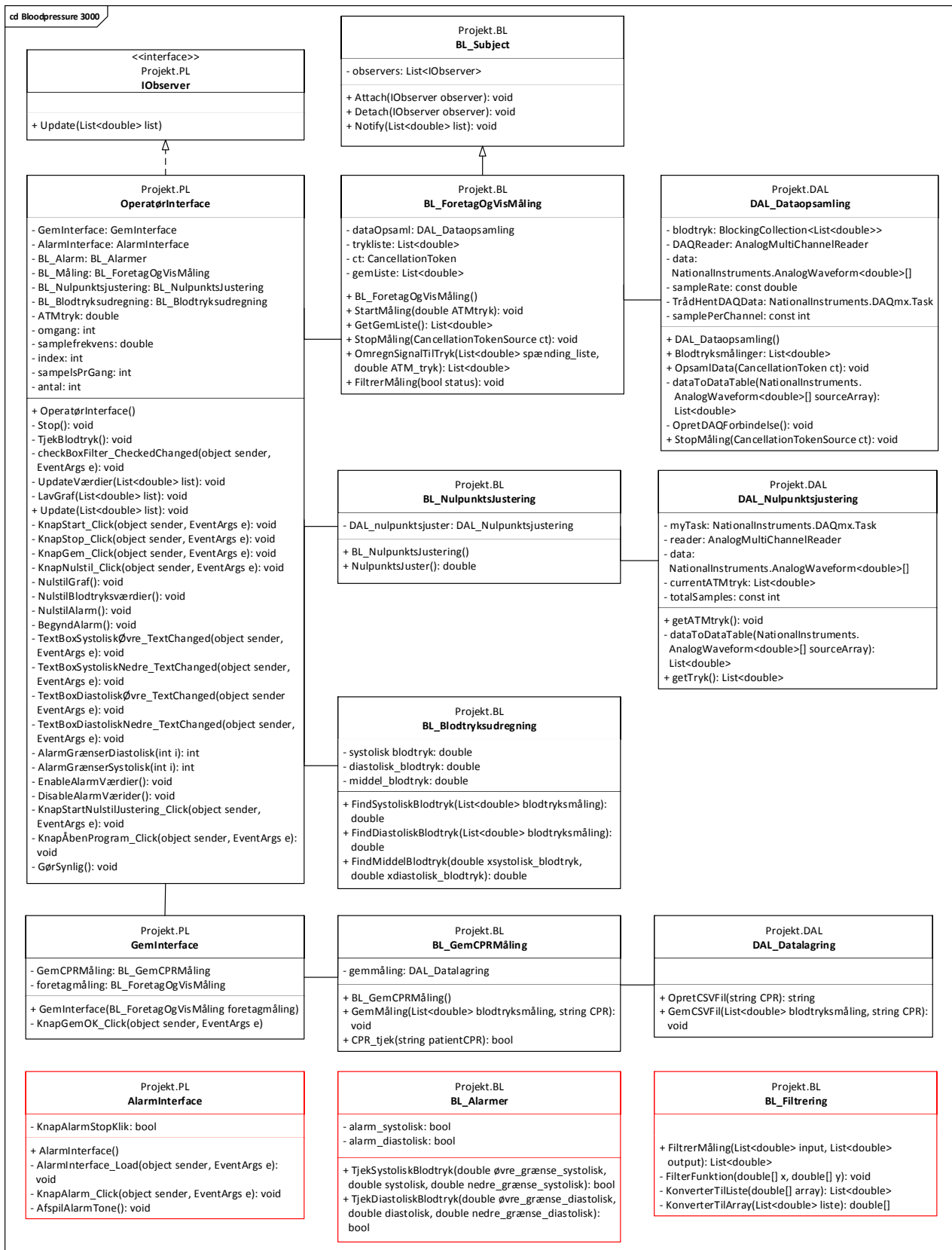
Figur 17 Filter design tool i Matlab

Ud fra fremgangsmåden vælges et 20.ordens Chebyshevs type 1 lavpasfilter, som ved 50 Hz dæmper med ca. 73 dB. Dette fjerner helt sikkert de 50 Hz støj. Filteret skal have en høj orden for at dæmpe tilstrækkeligt ved 50 Hz. Jo højere orden, jo større dæmpning ses ved 50 Hz. Det vurderes, at et 20.ordens filter er tilstrækkeligt.

7.2.2 Implementering

Ved implementering er programmet opdelt i de 3 namespaces Projekt.PL, Projekt.BL og Projekt.DAL, som følger 3-lagsmodellen. Opbygningen af blodtrykssystemet kan ses i klassediagrammet på figur 18. Hvis man sammenligner det med klassediagrammet fra design på figur 16, ses det tydeligt, at der er foretaget mange ændringer, siden designfasen er afsluttet. I gennem softwareimplementering er softwaredesignet altså ikke fulgt slavisk. Dette diskuteres i afsnittet 8.2.2 Diskussion af software.

Interaktionen mellem klasserne kan ses i de forskellige sekvensdiagrammer (jf. Softwareimplementering, s. 22).



Figur 18 Endeligt klassediagram for det implementerede program

7.2.2.1 Indhentning af data

Et vigtigt element i implementeringen er indhentningen af data. For stadigvæk at overholde 3-lagsmodellen er det vigtigt, at der kun sker kald ned gennem systemet og ikke op til systemet. Det har derfor været nødvendigt at implementere efter producer consumer princippet (jf. afsnittet 4.5.4 Producer consumer princippet).

Ift. observer-mønstret er BL_ForetagelseAfMåling implementeret som det konkrete subjekt, som nedarver fra klassen Subjekt. Her benyttes metoden Notify til at fortælle PL, at målinger er klar til at blive brugt. OperatørInterface arver fra IObserver og implementer metoden Update.

Blocking collection er implementeret først i DAL_dataopsamling, hvor den starter en ny Task, som indhenter data fra A/D converteren. Der fyldes efterhånden en liste på 800. I DAL_dataopsamling er der implementeret metoden Blodtryksmålinger, hvori Take-metoden opsamler data, når samtlige 800 målinger er fyldt op i listen. Da det er en cirkulær buffer har det været nødvendigt at implementere CancellationsTokens for at kunne stoppe opsamlingen af målingerne. Dette er med som parameter hele vejen ned fra brugerfladens "Stop" knap.

7.2.2.2 Presentation Layer

Der findes tre GUI's med tilhørende klasser i PL. Disse er OperatørInterface, GemInterface og AlarmInterface. OperatørInterface startes med nulpunktsjustering med to knapper, hvor knappen til at åbne selve programmet først er tilgængelig efter endt nulpunktsjustering. Når der trykkes på åben program vise selve OperatørInterface. Her findes tekstbokse til værdierne for både det systoliske, diastoliske og middelblodtrykket. Operatøren kan vælge intervallet for alarmgrænserne, dog med enkelte begrænsninger på minimum 20 mmHg. Grafen indeholder gridlines til at overskue værdierne for blodtryksmålingen. De fire knapper, "Start", "Stop", "Gem" og "Nulstil" er til for at operatøren skal være i kontrol over programmet.

GemInterfacet åbnes, hvis der er trykket på knappen "Gem", og her vises som default, at målingen ikke er gemt endnu. Dette skal tydeliggøre over for operatøren, at der stadig mangler en aktiv handling fra dennes side. Der skal nu indtastes et gyldigt CPR, og hvis ugyldigt vises beskeden "Ugyldig CPR – indtast venligst et gyldigt CPR". Hvis CPR-nummeret er korrekt, vil brugerfladen lukke og målingen vil være gemt.

Mens der sker en optagelse af måling, vil AlarmInterface vises, hvis grænseværdierne overskrides. Samtidig vil en lyd, "tadar" afspilles, og den vil først stoppe efter, operatøren har trykket på knappen "Stop Alarm". AlarmInterfacet tydeliggør, at grænseværdierne er overskredet og advarer operatøren. Dette er gjort for at overholde UC5 Alarmer (jf. Kravsspecifikation, s. 15).

7.2.2.3 Business Layer

BL står for alle former for udregninger og omformninger af data. Her er implementeret seks klasser, som hver især har et ansvarsområde. Disse klasser er BL_Subjekt, BL_ForetagOgVisMåling, BL_NulpunktsJustering, BL_Blodtryksudregning, BL_GemCPRMåling og BL_Alarmer. BL_Nulpunktsjusteringen er den første klasse i BL, som bliver benyttet. Her gemmes værdien for det omkringliggende blodtryk.

Subjekt er en del af observer-mønsteret, der sørger for at få data fra BL til PL. Den får målingen fra BL_ForetagOgVisMåling, som har de fleste ansvarsområder. Denne klasse startes oppe fra OperatørInterfacet ved tryk på "Start", hvorefter den fungerer som en stub mellem PL og DAL. Dog bliver dataene undervejs først omregning fra spænding til tryk ved brug af omregningsfaktoren (jf. Hardwareimplementering, s. 21).

BL_Blodtryksudregning sørger for at udregne minimum og maksimum samt middelværdi for værdier i listen på de 800 målinger, hvor middelværdien angives til at være $\frac{1}{3} \cdot systolen + \frac{2}{3} \cdot diastolen$ (jf. Softwareimplementering, s. 34). Opdateringen af tekstboksene for blodtrykket i PL bliver derfor opdateret hvert 0,8 sekund, men ved et stabilt blodtryk vil ændringen ikke være særlig stor og opdateringen vil synes langsommere.

I BL-klassen tilknyttet GemInterface, BL_GemCPRMåling, er dens vigtigste funktion at validere CPR-målingen. Denne metode uddybes nærmere i afsnittet 7.2.2.5 Metoder. Ved et korrekt CPR vil dette sendes med parameter med til DAL, hvor det skal gemmes.

Klassen BL_Alarmer tjekker værdierne fra tekstboksene i OperatørInterface. Dog er denne metode ikke implementeret, og dette er begrundet nærmere under afsnittet 8.2.2 Diskussion af software.

7.2.2.4 Data Access Layer

Den første gang DAL benyttes er i klassen DAL_NulpunktsJustering, hvor der er en reader, som gør det muligt at læse data fra A/D converteren. I en liste opsamles 100 målinger, som derefter vil kunne benyttes i BL_NulpunktJustering.

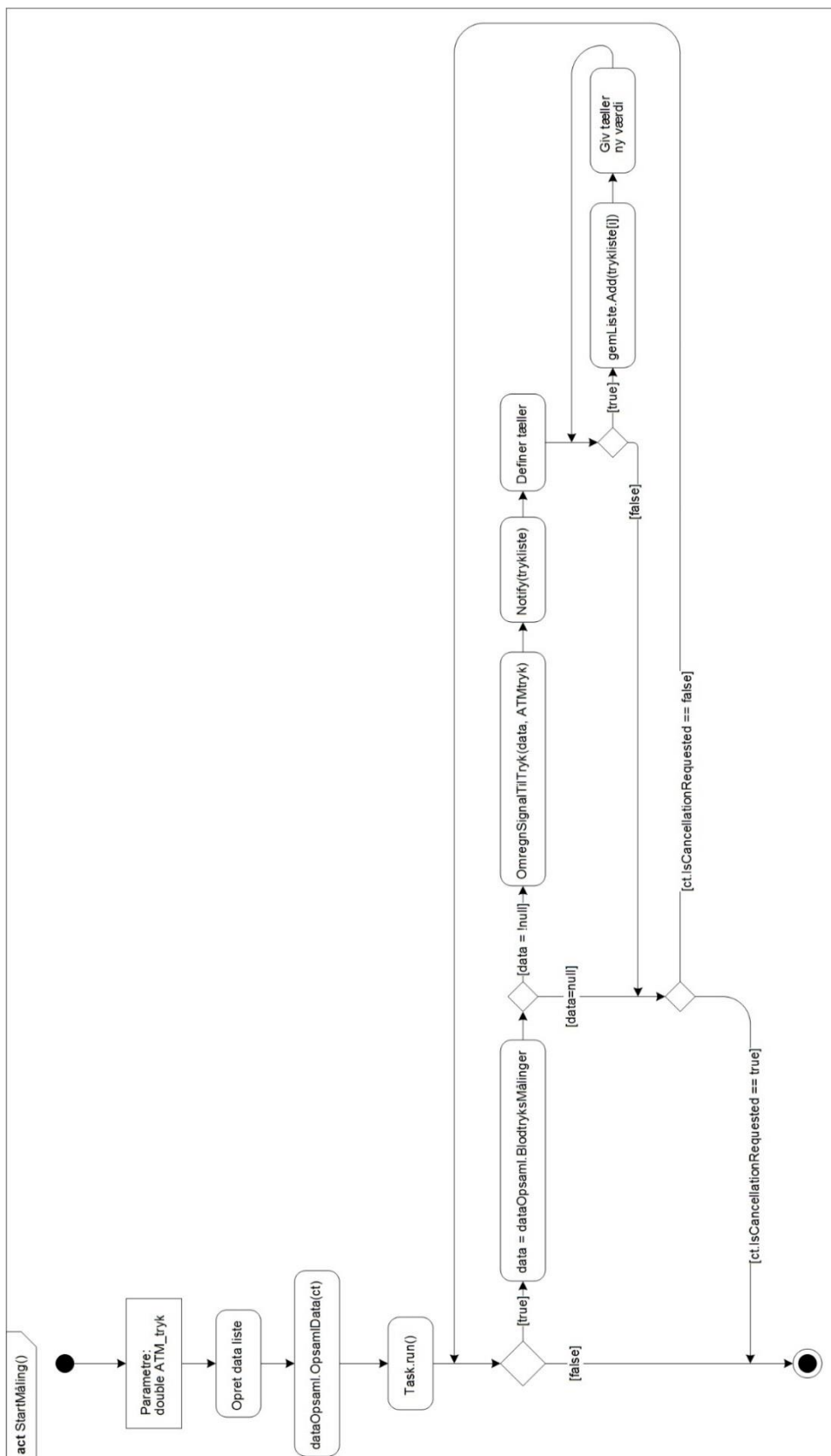
DAL_Dataopsamling har integreret en blocking collection, der fungerer som en cirkulær buffer. Her oprettes forbindelse til DAQ'en på nogenlunde samme måde som i DAL_NulpunktsJustering, men forskellen er, at den her ikke bliver disposed efter et bestemt antal samples. Den cirkulære buffer vil fylde en datakø op, hvorefter den opsamles fra BL_ForetagOgVisMåling. Derefter vil datakøen endnu engang fyldes op, og sådan vil forløbet køre indtil systemet stoppes af CancellationTokenSourcen.

Efter endt måling kan der vælges, at målingerne skal gemmes. Listen ved gemListe sendes med fra BL_ForetagOgVisMåling og benyttes i DAL_Datalagring. Her oprettes en CSVfil med navnet CPR-nummeret_måling. Hvis der allerede findes en sådan fil, vil den gamle blive overskrevet.

7.2.2.5 Metoder og klasser

Nogle af metoderne og klasser er på forhånd indbygget i Visuel Studio, men programmet indeholder også metoder, som der er lavet algoritmer for. Til de mest relevante af disse metoder og algoritmer er der lavet aktivitetsdiagrammer på for bedre at kunne gennemskue dem. Samlingen af alle metoder og klasser i programmet kan ses i dokumentet Softwareimplementering, hvor udvalgte metoder er beskrevet af aktivitetsdiagrammer (jf. Softwareimplementering, s. 26).

En vigtig metode i programmet er metoden StartMåling i klassen BL_ForetagOgVisMåling. Denne illustreres i aktivitetsdiagrammet på figur 19.



Figur 19 Aktivitetsdiagram for StartMåling()

7.2.2.6 Digitalt filter

Til at implementere det digitale filter, findes der en funktion man kan integrere i MatLab. Vha. funktionen "sos2csharp" er det muligt at kalde denne på sit filter. Funktionen "sos2csharp" generer herefter en C#-klasse til filtret man har brugt funktionen på. (19)

Det er herefter muligt at tilføje den generede klasse til sit projekt i Visual Studio. I denne klasse er der en filtreringsmetode implementeret som herefter kan kaldes på ønskede objekter.

7.3 Tests

7.3.1 Hardware

Der er anvendt enheds- og integrationstest på alle dele af hardwaren. Der er udført enhedstest på forstærkeren og lavpasfilteret enkeltvis på fumlebræt. Disse blev udført for at sikre, at forstærkeren og lavpasfilteret fungerede optimalt, før de blev tilsluttet hinanden. Efterfølgende blev der lavet integrationstest af det samlede system på fumlebræt, før det blev loddet sammen på printpladen. Ligeledes blev dette gjort for at undgå fejl i samspillet mellem forstærkeren og lavpasfilteret. Til sidst blev der lavet integrationstest af det færdige produkt på printpladen for at teste det helt endelige.

7.3.2 Software

Der er blevet enhedstestet på kodens forskellige klasser og deres respektive metoder. Der er løbende blevet lavet integrationstest samt enhedstest forskudt af hinanden ved fejlfinding af eventuelle problemer. Dette har som nævnt ikke været i den optimale rækkefølge (jf. afsnittet 5.3.8 Enhedstest).

Der er lavet enhedstest over følgende klasser og metoder:

- BL_Blodtryksudregning, metoden FindMiddelBlodtryk
- BL_Alarmer, metoden TjekSystoliskBlodtryk
- BL_Nulpunktsjustering, metoden Nulpunktsjuster
- BL_GemCPRMåling, metoden CPR_tjek
- DAL_Datalagring, metoden GemCSVFil
- DAL_Nulpunktsjustering, metoden getATMtryk
- BL_ForetagOgVisMåling, metoden OmregnSignalTilTryk

8.0 Resultater og diskussion

8.1 Resultater

Der blev udført en accepttest for at afprøve og vurdere funktionaliteten af systemets forskellige aspekter. Testene er udført for både fysiske krav, samt systemets funktionelle og ikke-funktionelle krav, som er opstillet.

De fysiske krav, som blev testet er:

- F1: Forstærkeren skal kunne forstærke signalet fra transduceren, så det analoge udgangssignal, forstærkeren giver, ligger nær den optimale indgangsspænding i A/D converteren, som er 10V.
- F2: Forstærkeren skal have et 4 pin-indgangsstik fra transduceren.
- F3: Forstærkeren skal have 2 pin-udgangsstik til A/D converteren.
- F4: Det skal være muligt at tilkoble en spændingsforsyning til forstærkeren.
- F5: Forstærkeren må max være 30 cm x 30 cm x 5 cm i dimensionerne.

Her er alle de observerede resultater for de fysiske krav godkendte.

De ikke funktionelle krav, som blev testet er:

- U3: Der skal vises en blodtryksmålingskurve på GUI'en, hvor enheden for blodtryk, mmHg, vises op af y-aksen fra 0 mmHg til 300 mmHg.
- R4: Programmet må have en fejlmargen på max 2%.
- P1: Startoptid for programmet skal være max 10 sekunder.
- P2: Svartid skal være max 1 sekund ved tryk på en given knap.
- P3: Data skal samples med en samplingsfrekvens på 1000 Hz.
- P4: Programmet skal kunne måle op til 300 mmHg.

Her er to af de observerede resultater for de ikke-funktionelle krav ikke godkendt. Dette gælder for kravene P2 [Stop-knappen] Svartid skal være max 1 sekund ved tryk på en given knap og P3 Programmet skal kunne måle op til 300 mmHg.

Testen af de funktionelle krav er en test af alle Use Cases og deres extensions. Disse tests er:

- UC1: Nulpunktsjuster
- UC2: Foretag og vis måling
- UC3: Gem måling

- UC3: Gem måling – Extension 2a: ugyldig CPR
- UC4: Filtrer måling scenarie A
- UC4: Filtrer måling scenarie B
- UC5: Alarmer
- UC6: Nulstil

Her er 3 af de observerede resultater for de funktionelle krav ikke godkendt. Dette gælder for kravene UC4: Filtrer måling scenarie A og B, samt UC5: Alarmer.

8.2 Diskussion

8.2.1 Diskussion af accepttest

Af de tests, som blev opstillet for systemet, er fem af kravene ikke godkendt. Alle fysiske krav blev godkendt, men både de funktionelle og ikke-funktionelle krav havde fejl og mangler. Under udførelsen af testen for P2 Svartid, skal svartiden være max 1 sekund ved tryk på en given knap var der en fejl. Resultatet af Stop-knappen afhang af, hvor lang tid programmet havde indhentet målinger. Dvs. at testen kunne accepteres de gange, hvor systemet ikke havde været belastet for længe og gjort programmet langsommere. Årsagen til dette uddybes nærmere under afsnittet 8.2.2 Diskussion af software. Programmet skal kunne måle op til 300 mmHg, hvilket omregnet til spænding, svarer til 9,03 V. Dette var ikke muligt at godkende, da forstærkeren kun forsynes med 9V. Teoretisk kan signalet forstærkes til 9V, men i praksis går noget spænding tabt. Reelt set forstærkes signalet op til 7V svarende til 245 mmHg. For at løse dette problem, skulle forstærkeren forsynes med højere spænding. Dette uddybes nærmere under afsnittet 8.2.3 Diskussion af hardware.

Der er fejl i kravene for UC4 Filtrer måling scenarie A og B, samt UC5 Alarmér. Test af Use Casene og deres resultater afhænger af software designet og implementeringen af metoder. Dette uddybes nærmere under afsnittet 8.2.2 Diskussion af software.

8.2.2 Diskussion af software

Udviklingen af softwaren har som udgangspunkt fuldt V-modellen (jf. afsnittet 5.3.3 V-modellen). Dette har dog ikke at være muligt at gøre fyldestgørende, og især den korrekte rækkefølge af test er ikke fulgt.

Fremgangsmåden har været at prøve frem med, hvad der virker og løbende få hjælp af forskellige faglærere. Dette er sket, fordi gruppens programmeringserfaring ikke har givet mulighed for andet. Der er blevet testet lidt løbende, men det har ikke været systematisk, og enhedstest er først lavet efter de testede klasser er implementeret i det samlede program. På den måde afviger arbejdsmetoden fra v-modellen, og i og med, at der først er testet systematisk til sidst minder dette mere om vandfaldsmodellen.

Der er samtidig arbejdet iterativt, fordi gruppen har afvejet fra designet under implementeringen og på den måde gentaget processen. Det er dog ikke gjort optimalt, da man ikke har gentaget alle faser under en iteration, men mere kun har rettet i programmeringsdelen.

Det lykkedes at få alle værdier vist kontinuerligt på en graf, og blodtryksværdierne opdateres som ønsket. Programmet overbelaster dog CPU'en, og det er specielt opdateringen af grafen i GUI-tråden som optager en stor del af computerens ydeevne. Opbygningen af softwaren er ikke udført hensigtsmæssigt, og dette skyldes manglen på tråde. Tråde er nødvendige for at applikationen kan udføre flere opgaver på én gang, men det blev vedtaget i samspil med faglærere at anvende tasks til udførsel af disse opgaver.

Sådan som det er etableret nu, har programmet en task som har et alt for stort ansvar, hvilket overbelaster CPU'en. Den høje samplingsfrekvens og den overbelastede CPU, gør at programmet ikke kan behandle de mange data der kommer ind. Dette er blandt årsag til at Stop-knappen ikke reagerer hurtigt nok, når programmet har kørt i en vis tid.

Gruppen har forsøgt at implementere en blocking collection mellem PL og BL. Denne blocking collection var tiltænkt som værende en cirkulær buffer, der skulle tømmes og fyldes med data kontinuerligt. Blocking collections har ikke været introduceret i undervisningen, og der har derfor ikke været kendskab til disse, inden udvikling af programmet blev påbegyndt. Det har derfor været svært at implementere korrekt, men tanken var, at dette skulle lette arbejdsbyrden mht. dataopsamling og visning af data på GUI.

Det blev vedtaget at observer-mønstret skulle anvendes i softwaren og brugen af dette bliver kompliceret, når man vil udføre flere forskellige opgaver på samme tidspunkt, hvilket var tilfældet i projektet. Den bedste løsning var at få implementeret flere blocking collections, da dette giver en trådsikkerhed, som observer-mønstret ikke tilbyder, og den belaster ikke systemet på samme måde.

De funktioner, som ikke fungerede i koden, var filtreringen og alarmeringen. Problemer med disse dele af applikationen kunne være blevet løst med mere tid til at implementere flere tråde. Kontrolleringen af alarmeringen sker først, når blodtryksværdierne har været opdateret på brugerfladen. Dette bør ske før, så tråden i GUI'en ikke bliver for stor og belaster systemet for meget. Altså er det selve arkitekturen som bør remodelleres, da programmet først bør gå gennem alarmudregningerne inden det vises på brugerfladen. Dette vil også bevirke, at 3-lagsmodellens retningslinjer overholdes korrekt.

Med hensyn til navngivning af klasser og metoder, er koden opbygget på en rodet måde. Det varierer i koden, hvornår der bruges store og små bogstaver, og gruppen har ikke været konsekvente med hvornår der er brug af " _ ". Dette er kommet af, at flere forskellige gruppemedlemmer har programmeret samtidig uden at have nogen stringent retningslinje at følge.

8.2.2.1 Diskussion af digitalt filter

Til implementering af det digitale filter, blev der arbejdet med et 20. ordens chebyshev type 1 filter med en cut-off frekvens på 45 Hz. Det kunne ses, at dette filter ville filtrere alle frekvenser på 50 Hz helt væk, hvilket var formålet. Ved implementering af det digitale filter til koden i Visual Studio, resulterede det høje antal af ordener, at et indgangssignalet blev dæmpet fuldstændig, så udgangssignalet ikke var brugeligt. Et digitalt filter af en lavere orden ville her være bedre egnet, da et lavere antal af ordener stadig ville dæmpe signalet uden at dæmpe det helt.

Da det digitale filter ikke blev implementeret, havde systemet kun et 1. ordens lavpasfilter med en cut-off frekvens på 50 Hz. Det vil sige, at der ikke bliver filtreret støj ved 50 Hz fra, som var det egentlige formål med filtrene. Det lykkedes dog alligevel at få vist en kontinuerlig og nøjagtig kurve på grafen uden for meget synlig støj.

8.2.3 Diskussion af hardware

For at kunne forstærke signalet tilstrækkeligt, så programmet kan opfylde kravet om at vise 300 mmHg på GUI, skulle forsyningsspændingen på forstærkeren være højere. Gruppen valgte at tilføje samme spænding på transducerens excitationsspænding og som forsyningsspænding på forstærkeren. For at afhjælpe problematikken omkring visningen af 300 mmHg, kunne gruppen have valgt forskellig spændingsforsyning på transduceren og forstærkeren. Hvis gruppen fra projektets start havde vidst, at det forstærkede blodtrykssignal skulle være omkring 9-10V for at vise 300 mmHg, kunne en løsning have været at vælge en anden spændingsforsyning på forstærkeren. Det er vigtigt at fastholde, at transduceren skal have 9V i excitationsspænding, for at have den sensitivitet, som forstærkningen på INA 114 er regnet ud fra. Det kunne være fordelagtigt at have valgt en variabel spændingsforsyning på forstærkeren.

Kalibreringen blev udregnet ved brug af væskesøjlen. Her forekom nogle fejlkilder, der bevirkede, at kalibreringen kunne blive upræcis. Gruppen erfarede, at der forekom forskellige resultater, for hver gang kalibreringen med væskesøjlen blev udført. Det lykkedes dog at få nogle fornuftige resultater, der blev brugt til udregning af kalibreringen.

Teoretisk set skulle det målte tryk fra væskesøjlen ind i transduceren, blive vist som en jævn kurve, da trykket var konstant. Dette var ikke tilfældet, da der forekom små udsving på kurven. Støj kan være skyld i dette, og da der intet digitalt filter var til at filtrere denne støj fra, blev målingerne upræcise.

Ud over dette holdes tryktransduceren i hånden, hvilket er med til at give små rystelser, som kan skabe endnu en usikkerhed. Derfor er det vigtigt, at tryktransduceren holdes i samme position under alle 3 målinger og i samme niveau som væskesøjles målepunkt.

Væskesøjlen var også en anelse utæt og har sandsynligvis været med til at give et udslag i trykket, så det ikke forblev konstant.

Gruppen valgte at lave et 1. ordens lavpasfilter med en cut-off frekvens på 50 Hz. Dette 1. ordens lavpasfilter viste sig, under enhedstest, at dæmpe frekvenser før cut-off frekvensen på 50 Hz, hvilket ikke var hensigtsmæssigt, da formålet netop var at fjerne støj, der forekom ved 50 Hz og derefter. Et filter med højere orden ville give mere dæmpning efter cut-off frekvensen og mindre dæmpning før.

Derudover var det ønskeligt at filtrere 50 Hz støj fra, hvilket ikke var muligt med det analoge lavpasfilter, da der er valgt en cut-off frekvens på 50 Hz. For at filtrere støjen skulle det digitale filter være brugt.

8.2.4 Diskussion af arbejdsproces og metode

8.2.4.1 SCRUM

Gruppen har anvendt SCRUM til projektarbejdet. I løbet af arbejdsprocessen, har det kun været udvalgte aspekter fra SCRUM metoden, der har været taget i brug. Generelt har det fungeret godt og givet et godt overblik over status i projektarbejdet. Gruppen har dog ikke altid anvendt daglige og ugentlige SCRUM møder, hvilket har resulteret i, at gruppen har været meget opdelt i to grupper; hardware og software. Disse to arbejdsgrupper har til tider ikke haft nok information omkring hinandens arbejde, når valg skulle tages undervejs. Det var først til sidst i forløbet, grupperne fulgte op på hinandens arbejde og opnåede forståelse for det. Ved at afholde daglige SCRUM møder, ville denne opdeling have været undgået, og et mere samlet projektarbejde ville være opnået.

8.2.4.2 Logbog

Der er skrevet logbog for hele forløbet, men denne er undertiden blevet opdateret nogle dage, efter eventuelle beslutninger er blevet vedtaget og møder blevet afholdt. Det har været svært for de to individuelle grupper at føre logbog for hinanden, pga. den manglende indsigt. Førelsen af logbog i dette projekt har til tider været mangelfuld, og er et punkt der kunne forbedres. Det kunne evt. afhjælpes ved at vælge en logbogsansvarlig.

8.2.4.3 Ansvarsfordeling

Fra starten af forløbet var der en klar opdeling i hardware- og softwaregruppe. Gruppen opdeltes med tre personer i hver gruppe, hvilket viste sig ikke at være optimalt. Det forventedes, at hardware-delen ville blive den mest problematiske, hvor det var software-delen, der skabte flest problemer.

Det viste sig, at software-delen var den mest tidskrævende og klart sværeste del, hvilket set i bakspejlet ville kræve størstedelen af gruppens medlemmer.

Den egentlige arbejdsfordeling skulle derfor have været to mand på hardware og fire mand på software.

9.0 Perspektivering

Alarm og filtrering har fra starten været funktioner, der er ønsket implementeret, men dette er som nævnt, ikke lykkedes. Dette er selvfølgelig noget man bør arbejde videre med i et eventuelt senere arbejde. Samtidig er der andre funktioner, der kunne være interessante at implementere i systemet. Dette kunne fx være måling af puls og en angivelse af dette med bip-lyd, hvilket er en meget almindelig funktion i monitoreringssystemer, der bruges klinisk. Det giver en tydelig indikation af patientens hjertefunktion. En anden ting, der er relevant klinisk, er at kunne gemme data i en database. Dette kunne derfor også være en funktion, der kunne arbejdes på i en videre udvikling af systemet. I det danske sundhedsvæsen bruges EPJ (Elektronisk Patient Journal) til lagring af forskellige kliniske data som fx blodtryksmålinger, og derfor en muligheden for gemme i en database meget relevant.

Et blodtryksmålingssystem som udviklet i dette projekt er ofte ved klinisk brug integreret i andre monitoreringssystemer. Når en patient har behov en invasiv arteriel måling af blodtryk, er der også behov for at måle andre størrelser som fx EKG. Derfor er det praktisk at flere grafer og værdier kan vises på samme brugerflade, og klinisk benyttes ofte monitoreringssystemer, der kan dette. Systemet, der er udviklet i dette projekt, har derfor behov for at blive integreret i et større system, så det er mere klinisk anvendeligt. Dette kræver ændringer i brugerfladen og funktionerne, som nu er målrettet et program, der udelukkende har med måling af blodtryk at gøre.

Det er også nødvendigt at se på problemet omkring CPU belastning. I klinikken monitoreres patienter 24 timer døgnet, så projektprogrammet vil belaste CPU for meget til, at dette vil kunne lade sig gøre. Det er også et problem for at kunne integrere det i et større system og køre med så få fejl, som det er nødvendigt for at kunne bruge systemet klinisk.

10.0 Konklusion

På baggrund af problemstillingen, som var at udvikle et system, der kan tilsluttes et væskefyldt kateter og vise en blodtrykskurve, kan det konkluderes, at opgaven er løst på tilfredsstillende vis. Essensen i systemet var at kunne måle tryk via en tryktransducer, forstærke og filtrere signalet, samt indhente dette i et softwareprogram. Dette blev løst fyldestgørende, og dermed er de obligatoriske krav opnået. Det lykkedes at forstærke signalet, for derefter at filtrere støj fra med et analog lavpasfilter. Samtidig blev det muliggjort at lagre de målte data i en CSV-fil, som var et stille krav. Dog blev det sidste obligatoriske krav ikke imødekommet, da det digitale filter ikke kunne implementeres i det endelige produkt. Som ekstra funktionalitet til produktet valgte gruppen at tilføje visning af blodtryksværdier og et alarmeringssystem til programmet. Her kan der konkluderes, at alarmeringssystemet ikke blev implementeret pga. dårlig softwarearkitektur. Til gengæld blev blodtryksværdierne på brugerfladen vist med stor nøjagtighed. I arbejdsprocessen skulle arbejdsfordelingen have været anderledes, og det blev erfaret, at softwareholdet krævede flere hænder end hardwaregruppen, da softwaredelen var den mest tidskrævende del. Overordnet set kan det konkluderes, at projektet er løst i en tilfredsstillende grad.

11.0 Bilag

Bilagene er vedlagt på CD.

1. Projektrapport
2. Kravspecifikation
3. Accepttest
4. Hardwaredesign
5. Hardwareimplementering
6. Softwaredesign
7. Softwareimplementering
8. Referenceliste
9. Kode
10. Versionshistorik over kode
11. Logbog
12. Mødereferater
13. Samarbejdsaftale
14. Tidsplan
15. Kode til testprojekter brugt til enhedstest
16. Enhedstest og integrationstest screenshots
17. Enhedstest og integrationstest billeder af forsøgsopstillinger

12.0 Referenceliste

(1) A/D-converter - Picture. Available at:

http://upload.wikimedia.org/wikipedia/en/thumb/5/52/ADC_voltage_resolution.svg/1273px-ADC_voltage_resolution.svg.png. Accessed 16/12, 2014.

(2) Bjerger K. Introduction to System Engineering Compendium (T-006). 2014(IHA).

(3) Deichgræber C. Hjertets opbygning billede. 2013; Available at:

<http://www.auhb.dk/resources/Opbygning.jpg>. Accessed 16/12, 2014.

(4) Edwards Lifesciences LLC. Retrieved from Accuracy For You and Your Patients. 2006; Available at:

http://www.hi-techsurgical.com/nephrology/edward_lifescience/blood_management_system.pdf.

Accessed 16/12, 2014.

(5) Fog A. Man Machine Interface. 2000; Available at: www.eit.ihk-edu.dk/subjects/mmi. Accessed 16/12, 2014.

(6) Häuser L. ST2ITS2 Lektion 4 - 3-lagsmodellen. 2014.

(7) Häuser L. ST3ITS3 Lektion 3-4. 2014.

(8) Johansen P. Hæmodynamik og hjerte-karsystemet - PowerPoint. 2014.

(9) Johansen P. Hæmodynamik og hjerte-karsystemet. 2009.

(10) Kidmose P. Lektion 2 Transducer Principper. 2014.

(11) Koirala S. SOLID architecture principles using simple C# examples. 2014; Available at:

<http://www.codeproject.com/Articles/703634/SOLID-architecture-principles-using-simple-Csharp>.

Accessed 16/12, 2014.

(12) Livingstone C. Frequency Content of the Arterial Pressure Waveform. 2000; Available at:

http://web.squ.edu.om/med-Lib/MED_CD/E_CDs/anesthesia/site/content/v03/030261r00.HTM. Accessed 16/12, 2014.

(13) Microsoft. **BlockingCollection Overview**. 2014; Available at: <http://msdn.microsoft.com/en-us/library/dd997371%28v=vs.110%29.aspx>.

Accessed 16/12, 2014.

(14) National Instruments. Analog Sampling Basics. 2014; Available at: <http://www.ni.com/white-paper/3016/en/>.

Accessed 16/12, 2014.

(15) National Instruments. User Guide and Specifications NI USB-6008/6009. Available at:

<http://www.ni.com/pdf/manuals/371303m.pdf#G36557>. Accessed 16/12, 2014.

(16) Nielsen BK, Haug E, Sand O, Sjaastad ØV, Bjålie JG. Menneskets anatomi og fysiologi. 2nd ed.: Gads Forlag; 2008.

- (17) Norvatis Danmark. Forhøjet blodtryk, hypertension. Available at:
http://www.novartis.dk/sygdomsomraader/forhojet_blodtryk.shtml. Accessed 16/12, 2014.
- (18) Practicus. Hypertension i almen praksis: Hvor mange bliver behandlet og hvor godt bliver de behandlet? 2013; Available at: <http://www.practicus.dk/flx/artikler/?m=showArticle&aid=223>. Accessed 16/12, 2014.
- (19) Squire J. Convert Matlab SOS filters to C#. 2010; Available at:
<http://www.mathworks.com/matlabcentral/fileexchange/23530-convert-matlab-sos-filters-to-c->. Accessed 16/12, 2014.
- (20) Texas Instruments. Precision Instrumentation Amplifier. 1998; Available at:
<http://www.ti.com/lit/ds/symlink/ina114.pdf>. Accessed 16/12, 2014.
- (21) Webster JG. Medical Instrumentation Application and Design. 4th ed.: John Wile & Sons Inc.; 2009.