



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI
3. SEMESTERPROJEKT

Rapport

Gruppe 2

Anne Bundgaard Hoelgaard (201404492)
Mette Hammer Nielsen-Kudsk (201408391)
Ditte Heebøll Callesen (201408392)
Martin Banasik (201408398)
Albert Jakob Fredshavn (201408425)
Johan Mathias Munk (201408450)

Vejleder

Studentervejleder
Peter Johansen
Aarhus Universitet

Resumé

Dette projekt beskæftiger sig med blodtryksmåling, hvor der ud fra et blodtrykssignal kan bestemmes en systolisk- og diastolisk værdi, samt en pulsværdi. Formålet er at bygge en hardwaredel, der kan trykændre et analogt signal. Via et filter skal signalet forstærkes og sendes videre igennem en Data Acquisition (DAQ), der kan lave det analoge signal om til et digitalt signal. Herefter skal signalet videre ind i softwaren. Her er formålet at få programmeret en algoritme, der kan detektere en puls samt en systolisk- og diastolisk værdi. Disse systole- og diastoleværdier kan findes ud fra det digitale signal. Signalet bliver observeret fra starttidspunktet, hvor værdien for trykket begynder at vokse, indtil grafen når maksimum. Når værdien igen begynder, kan programmet detektere den højeste værdi, som er den systoliske værdi. Efter maksimum begynder grafen for signalet at falde igen, da trykket bliver mindre, indtil det når det laveste punkt, hvor værdien igen begyndte at stige en smule. Dette lavpunkt er værdien for diastolen. Grafen er periodisk og ud fra den systoliske- og diastoliske værdi kan pulsen findes.

Herudover skal programmet kunne kalibreres og nulpunktsjusteres når forskeren synes nødvendigt. Projektet er lavet til forskningsbrug og derfor blev det vurderet, at det er nødvendigt for en forsker, at kunne gemme sine målinger. Programmeringen skal derfor også gøre det muligt for forskeren, at kunne gemme sine målinger i en database.

Resultatet for dette projekt viser at det er muligt vha. hardware- og softwaredelen, at måle et blodtryk. Gennem videreudvikling kan det blive muligt at detektere hypertension (Forhøjet blodtryk) og hypotension (Lavt blodtryk) i samme program. Dette ville være nyttigt, hvis projektet i fremtiden skulle bruges til patienter. I projektet er der dog udelukkende blevet arbejdet med måling af blodtryk.

Abstract

This project has primarily been focused on blood pressure measurement, where systolic- and diastolic values, as well as the pulse, could be determined. The purpose of this project is to construct hardware and software which can manipulate the pressure of an analog signal as well as depict this signal using developed software. By using an analog filter, the analog signal is amplified and sent to a Data Acquisition (DAQ), where the signal will be transformed from an analog signal to a digital signal. This digital signal will afterwards enter a computer, running the software program. Here the purpose of the software is to program an algorithm, which can detect a pulse as well as the systolic- and diastolic values. These two values are found by looking at the visualized signal. From the beginning of the graph, and until the signal ascends to maximum and falls again. This maximum value is the systolic value. Afterwards the signal will begin to descend until a minimum is reached, where this minimum is the diastolic value. The graph will then slightly climb again and begin a new period. From these systolic- and the diastolic values, the pulse can be found. In addition to the above requirements, the program is able to be calibrated and zero adjusted, whenever the researcher finds it necessary.

This project is made for research use and it was therefore decided that a necessity for a researcher, would be to save his measurements in a database.

The result of this project shows that it is possible to measure blood pressure through a combination of hardware and software. Through future development it could be possible for the program to detect hypertension (high blood pressure) and hypotension (low blood pressure). This would be useful if the project, in the future, would be used on patients. In this project however, the main goal has exclusively been measurements of blood pressure.

Underskrifter

Gruppemedlemmer

Anne Bundgaard Hoelgaard (201404492) Dato

Mette Hammer Nielsen-Kudsk (201408391) Dato

Ditte Heebøll Callesen (201408392) Dato

Martin Banasik (201408398) Dato

Albert Jakob Fredshavn (201408425) Dato

Johan Mathias Munk (201408450) Dato

Vejleder

Peter Johansen Dato

Godkendelsesformular

Godkendelsesformular

Forfattere:

Anne Bundgaard Hoelgaard

Mette Hammer Nielsen-Kudsk

Ditte Heebøll Callesen

Martin Banasik

Albert Jakob Fredshavn

Johan Mathias Munk

Godkendes af Peter Johansen

Antal sider 28

Kunde Aarhus Universitet

Ved underskrivelse af dette dokument accepteres det af begge parter som værende kravene til udviklingen af det ønskede system.

Dato: 16/12-2015

Kundens underskrift

Leverandørens underskrift

Ordliste

Ord	Forklaring
(F)URPS+	Et akronym, der repræsenterer en model til klassificering af softwareens kvalitet
GUI	Graphical User Interface (Grafisk brugergrænseflade)
VPN	Virtual Private Network
DAQ	Data acquisition
SysML	Systems Modeling Language – sprog til visuel fremstilling af systemer
UML	Unified modelling language – sprog til oversigtsfremstilling af klasser i programmering

Indholdsfortegnelse

Resumé	i
Abstract	ii
Underskrifter	iii
Godkendelsesformular	iv
Ordliste	v
Kapitel 1 Indledning	1
Kapitel 2 Projektformulering og afgrænsning	3
Kapitel 3 Blodtryk	5
3.1 Hvad betyder blodtryk?	5
3.2 Hypertension	6
3.3 Hypotension	7
Kapitel 4 Systembeskrivelse	8
Kapitel 5 Krav	10
Kapitel 6 Projektbeskrivelse	13
6.1 Projektgennemførelse	13
6.2 Projektstyring	15
6.3 Metoder	15
6.4 Systemarkitektur	16
6.4.1 Hardware	17
6.4.2 Software	17
6.5 Problemidentifikation (design)	17
6.5.1 Hardware	17
6.5.2 Software	17
6.6 Implementering	17
6.6.1 Hardware	17
6.6.2 Software	21
6.7 GUI-beskrivelse	21
6.7.1 Algoritmer (grænseværdier)	21
6.7.2 Filteret/Ufiltreret	21
6.7.3 Lagring af data i Database	21
6.8 Test	21
6.8.1 Hardware	21

6.8.2 Software	23
6.9 Resultater og diskussion	23
6.10 Udviklingsværktøjer	24
6.11 Opnåede resultater	25
6.12 Perspektivering - Fremtidigt arbejde	25
Kapitel 7 Konklusion	26
Kapitel 8 Bilag	27
Litteratur	28
Figurer	28

Indledning 1

I dette projekt arbejdes der med blodtryksmålere. Vi har valgt at udarbejde en blodtryksmåler til forskningsbrug. Blodtryksmåleren skal kunne modtage en spænding fra en transducer, nulpunktsjustere og kalibere efter en forskers ønske. Signalet skal vises i en graf, på et display, hvor værdier for puls, systoliske- og diastoliske tryk vises. Her fra forskeren starter og gemmer sine målinger.

I kravspecifikationen findes de krav, der er blevet stillet for projektet. Herunder er også de krav, som blev stillet mellem os og vores vejleder.

Under systemarkitekturen findes informationer om, hvordan software- og hardwaredelen er opbygget. I afsnittet integrationstest kan der læses om, hvordan projektet er blevet testet.

Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse
0.1	04-11-2015	MHNK	Oprettelse af LaTex dokumenter
0.2	11-11-2015	ABH	Udviklingsværktøjer og krav
0.3	11-11-2015	DHC	Metoder
0.4	18-11-2015	ABH	Systembeskrivelse
0.5	19-11-2015	DHC	Perspektivering - Fremtidigt arbejde
0.6	24-11-2015	MHNK	Blodtryk
0.7	24-11-2015	AJF	Projektgennemførelse og -styring
0.8	24-11-2015	MM	Projektformulering og afgrænsning
1.0	01-12-2015	MB, MHNK	Referencelister i LaTex
1.1	02-12-2015	MB, MHNK	Figurlister i LaTex
1.2	02-12-2015	MHNK	Resume
1.3	07-12-2015	MHNK	Abstract
1.4	09-12-2015	MHNK	Konklusion
1.5	11-12-2015	ABH	Tilrette krav og systembeskrivelse ift. endelig løsning

Projektformulering og afgrænsning 2

I daglig klinisk praksis er der ofte behov for kontinuert at monitorere patienters blodtryk, i særdeleshed på intensive afdelinger samt operationsstuer, hvor blodtrykket er en vigtig parameter i monitorering af patienters kardiovaskulære status.

Denne kontinuere monitorering er også nødvendig i forskningsverdenen. Det er i forskerens interesse, at kunne måle blodtrykket når der laves hæmodynamiske undersøgelser. Her skal det være muligt for forskeren at kunne aflæse det diastoliske- og systoliske tryk, pulsen samt få vist en pæn kurve over blodtrykket. Det er målet, at opbygge en prototype, der kan registrere de spændinger, i milivolt (mV), der kommer fra tryktransduceren og analogt forstærke samt filtrere signalet. Dette signal skal derefter konverteres til det digitale domæne.

Herfra skal der programmeres en brugergrænseflade, der fremfører disse målinger, samt gør det muligt, for forskeren, at gemme målingen i en database til senere brug. Resultatet bliver derfor et elektronisk kredsløb med forbindelse til et software program. For at de gemte data kan sammenlignes, kræver det at de alle er blevet gemt med samme forudsætning, dvs. at målingerne er blevet kalibreret og nulpunktsjusteret. Dette bliver håndteret i softwaredelen, hvor beregninger implementeres. Når forskeren kigger på blodtryksgrafen, vil han normalt se på et filtreret signal. I tilfælde af at det er i hans interesse at se på et ufilteret signal, vil dette være muligt, ved et tryk på en knap.

MoSCoW¹

Must

- Et elektronisk kredsløb, som forstærker signalet fra tryktranduceren og filtrerer det med ét indbygget analogt filter
- Et program til at vise blodtrykket som funktion af tiden. Programmet skal opfylde en række obligatoriske krav. Det skal kunne:
 - Programmeres i C#
 - Kunne kalibrere blodtrykssignalet og foretage en nulpunktsjustering
 - Vise blodtrykssignalet kontinuert
 - Kunne gemme de målte data i en database

¹Metode, der benyttes til at beskrive hvad programmet skal kunne og det der ønskes, men ikke vil ske
- Must have, Should have, Could have, og Would like but won't get

- Kunne filtrere blodtrykket i selve programmet via et digitalt filter, som skal kunne slås til og fra (monitor mode = filtreret og afrundet; signaldiagnose mode = råt signal med alle udsving)
- Afbildning af systolisk/diastolisk blodtryk med tal

Should

Could

- Hardwaredelen skal bestå af ét VEVO Board med indbyggede komponenter

Would

- Alarmering, hvis blodtrykket afviger indbyggede grænseværdier
- Forskeren skal kunne hente de gemte data ned igen

Ansvarsområder

Idet gruppens størrelse ikke lægger op til samlet, at arbejde på alle dele samtidig, er projektets ansvarsområder blevet fordelt som følgende:

Navn	Ansvarsområder
Ditte Heebøll Callesen	Hardwaredesign, dokumentation
Albert Jakob Fredshavn	Hardwaredesign, dokumentation
Martin Banasik	Hardwaredesign, dokumentation
Johan Mathias Munk	Softwaredesign, algoritmeopbygning, dokumentation
Mette Hammer Nielsen-Kudsk	Softwaredesign, algoritmeopbygning, dokumentation
Anne Hoelgaard	Softwaredesign, algoritmeopbygning, dokumentation

Blodtryk 3

3.1 Hvad betyder blodtryk?

Alle har på et tidspunkt i deres liv fået målt blodtryk, men hvad betyder det egentlig? Blodtryksmåling er en meget enkelt undersøgelse, der giver vigtige informationer om blodkarrenes og hjertets tilstand. Blodtryk kan måles både invasivt og ikke-invasivt. Dette projekt omhandler invasivt blodtryksmåling, som er en måling af trykket direkte i en blodåre. Blodtrykket måles i enheden millimeter kviksølv (mmHg). Et normalt blodtryk ligger på omkring 120/80 mmHg [1].

Hjertet pumper iltet blod ud i hele kroppen via arteriesystemet og sørger dermed for tilførslen af ilt og næringssubstanse til alle muskler og organer. De røde blodlegemer (erythrocytterne) er en vigtig bestanddel af blodet. Det er hæmoglobinet i de røde blodlegemer, som binder ilten og sørger for at ilten transporteres frem til vævene i kroppen. Når ilten er afgivet fra blodet til musklerne og andre væv, transporteres blodet tilbage til højre side af hjertet via venesystemet. Det af-iltede blod pumpes af højre hjertepumpekammer ud i lungerne, hvor blodet iltes på ny og derfra strømmer til venstre hjertehalvdel for igen at blive pumpet ud i kroppen af venstre hjertepumpekammer.

Det høje tryk er det systoliske tryk, som kan måles når venstre ventrikkel trækker sig sammen. Det diastoliske blodtryk er blodtrykket i hjertets afslapningsfase (diastolen). Når hjertet trækker sig sammen (systolen) skaber det en trykbølge som forplanter sig ud igennem arteriesystemet. Trykbølgjen kan erkendes som pulsen, der let kan mærkes f.eks. ved palpation af a. radialis ved håndleddet. Trykket i venesystemet er meget lavere end i arteriesystemet, da blodet passivt blot skal strømme tilbage til højre forkammer, hvor trykket er lavt.

Venstre ventrikkel pumper iltet blod, under højt tryk, ud i aorta og arterierne. Disse blodårer er derfor tykvæggede og elastiske i modsætning til veneerne, der er ganske tyndvæggede, fordi de kun udsættes for et lavt tryk.. Hjertet overfører, gennem systolen, energi til arterievæggen, som bruges i den resterende del af hjertets cyklus, til at presse blod gennem karsystemet.

For at forstå det følgende afsnit introduceres tre vigtige begreber nedenfor:

- Væskevolumen, der løber igennem et rør pr. tidsenhed, kaldes for væskestrømmen.
- Distancen, som en væske flytter sig pr. tidsenhed er strømningshastigheden.
- Blodvolumen, der løber gennem et væv pr. tids- og vægtenhed er gennemblødning.

Der er en trykforskel imellem begyndelsen og slutningen af et rør. Væskestrømmen i røret afhænger af trykforskellen hen over røret og modstanden i røret. Dette kan udtrykkes i

følgende ligning som minder om Ohms lov for elektriske kredsløb:

$$Væskestrøm(Q) = \frac{Trykforskell(\Delta P)}{Modstand(R)} \quad (3.1)$$

Drivkraften for væskestrømningen (Q) er trykforskellen (ΔP) gennem røret. Hjertets kontraktioner gør, at strømmen i røret går fra et højere, til et lavere tryk. Modstanden (R) i en arterie er bestemt af bl.a. gnidningsmodstanden mellem arterievæggen og blodet, blodets viskositet og diameteren af arterien. Når blodet løber igennem arterierne, falder trykket efterhånden i blodet. Ved stigende modstand mod væskestrømmen forøges trykfaldet. Når modstanden i rørvæggen stiger, formindskes væskestrømningen, hvis trykforskellen samtidig ikke er steget.

Et rørs modstand bestemmes ud fra tre parametre:

- Længden af røret.
- Den indre diameter på røret.
- Viskositeten af væsken.

Jo kraftigere hjertet pumper, desto større bliver trykforskellen og dermed blodstrømningen. Blodkarrets diameter er det, der har størst betydning for modstanden mod blodstrømmen. Hvis blodet presses igennem et snævert kar, er der en større del af blodet, der er tæt på karvæggen og bliver derved bremset af friktionskraft. Modsat, hvis diameteren på karret havde været større, ville en mindre del af blodet være i kontakt med væggen og derved ville det ikke blive bremset lige så meget. Modstanden er derfor mindre og blodstrømningen større, i et stort kar.

Blodets viskositet stiger, jo flere røde blodlegemer, der findes i blodet. Jo flere røde blodlegemer, desto højere viskositet. Hvis blodet har en høj viskositet og derved er tyktflydende, så skal der et større tryk til at holde en hvis væskestrøm [1].

3.2 Hypertension

Hypertension er en meget almindelig lidelse, ca. 30% af den danske befolkning har forhøjet arterielt blodtryk [2]. Derfor er det vigtigt ofte at få målt sit blodtryk, da forhøjet blodtryk ikke kan mærkes og er den vigtigste årsag til hjerte-kar-sygdomme. Der er tale om hypertension når blodtrykket er 140/90 mmHg eller højere. Ved hypertension bliver arbejdsbelastningen af hjertet forøget, da der skal pumpes blod ud af hjertet mod en større modstand i arteriesystemet.

Forhøjet blodtryk gør at arbejdsbelastningen bliver større. Derved sker der lige så stille en fortykkelse af muskulaturen i venstre ventrikkel, da hjertet skal bruge flere krafter på at pumpe blodet ud i aorta. Det øgede tryk påvirker blodkarrenes belastning, og kan medføre at mindre blodkar kan briste under det store tryk. Hvis der er tale om blodkar i hjernen kan dette føre til en hjerneblødning. Hypertension er den hyppigste årsag til hjerneblødninger. Hypertension kan føre til en række andre komplikationer i form af åreforkalkning, hjerteinsufficiens [3], akut myokardieinfarkt, hjertekrampe, nyreskader og apopleksi. Forhøjet blodtryk behandles med lægemidler, i form af blodtryksnedsættende

medicin (antihypertensiva). Samtidig er non-farmakologiske metoder, som rygestop, motion, reduktion af saltindtagelse, vægtab og reduktion af alkoholforbrug vigtige i behandlingen af hypertension.

3.3 Hypotension

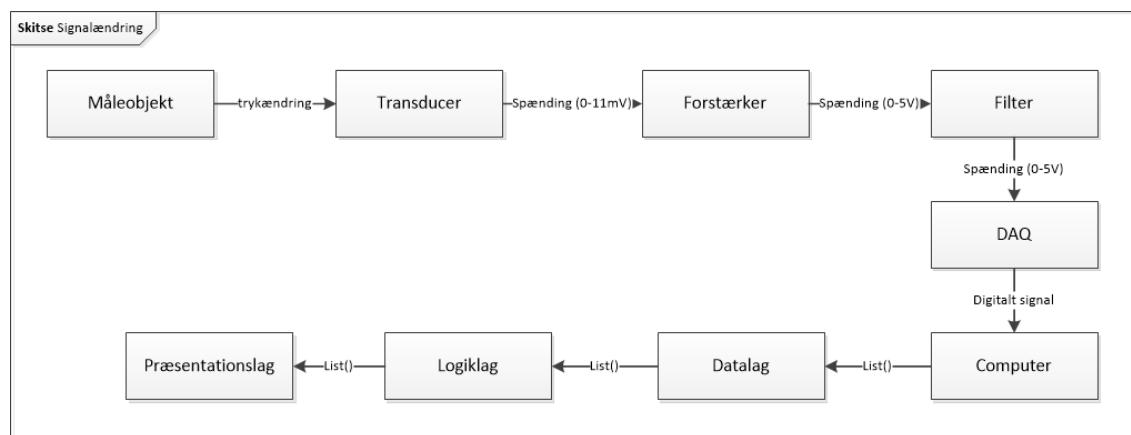
Hypotension er modsat hypertension et lavt blodtryk, og ikke nær så almindeligt. Der er tale om hypotension når blodtrykket er 90/60 mmHg. Hypotension ses især ved en række alvorlige akutte tilstænde som akut myokardieinfarkt, lungeemboli, sepsis eller alvorlige blødninger. Patienten kan i disse situationer være i en livstruende shock tilstand. Lavt blodtryk kan også i sjældne tilfælde forekomme kronisk, især ved sjældne stofskiftesygdomme med nedsat produktion af binyrebarkhormoner [4].

Systembeskrivelse 4

Med udgangspunkt i projektformuleringen kommer dette projekts endelige system til at bestå af et software og hardware system, der kan tilsluttes et måleobjekt, hvorpå et blodtryk kan måles. Systemet skal kunne implementeres i forskningsmiljøer, hvor en eller flere forskere ønsker at analysere indhentede blodtrykssignaler. Visionen er, at systemet skal være let tilgængeligt og effektivt, hvilket vil komme til udtryk ved, at systemet fungerer stabilt.

I dette projekt realiseres en prototype af systemet. Det vil sige at flere dele af systemet udvikles ud fra forsimplede metoder ift. hvordan det vil være optimalt at implementere dem i virkeligheden. Her tænkes på hardware-, såvel som software-elementer. Hardwaren i prototypen realiseres på et VEVO Board, så det er muligt at tage den med sig, samt er mere holdbar over tid. Softwaren i prototypen består af flere moduler. Disse er opbygget efter principperne i en trelagsmodel, hvilket vil sige, at koden indeholder et database-lag, logik-lag samt præsentations-lag. Dette er valgt for, at skabe et overblik over hvilke dele af software-koden der har ansvaret for de enkelte funktionaliteter i systemet.

Hardwaren består af en forstærker og et filter. Forstærkerens opgave består i at forstærke det analoge signal fra max 11 millivolt til 5 volt. Filtret sørger for at filtrere unødig støj fra det analog signal. Signalændringen fra måleobjektet til visning af signal på en graf er skitseret herunder. Det skal pointeres at dette kun er en skitse for at skabe overblik, derfor er flere processer i softwaren udeladt af diagrammet. Database-laget består af en



Figur 4.1: Skitse af signalændring

lukket database samt indhentningen af blodtrykssignalet fra måleobjektet til transduceren, gennem hardware, inden det rammer software-delen. I den lukkede database gemmes det indhentede blodtrykssignal i en tabel. Signalet gemmes med et tidsstempel, samt

under et autogenereret Id, sammensat med det forsøgsnavn som forskeren indtaster på brugergrænsefladen ved begyndelsen af en måling.

Logik-laget er handlingslaget, og alt kommunikation til de resterende lag går gennem dette lag. Laget indeholder flere klasser, der indeholder metoder til indhentning af systoliske-, diastoliske og puls-værdier, ud fra det indhentede blodtrykssignal. Derudover indeholder laget også klasser, der har ansvaret for at foretage en filtrering af signalet når dette er valgt.

Præsentationslaget er forskerens vej ind i systemet, dette lag har til ansvar, at udskrive valgte data på brugergrænsefladen.

Systemet skal udadtil have en brugergrænseflade i form af en touch skærm eller almindelig computerskærm med tilhørende tastatur. Det er denne skærm som den primære aktører til systemet, altså forskeren, interagerer med. Det tilstræbes, at opbygge brugergrænsefladen simpelt og efter forskerens logik, så opbygningen giver mening for systemets bruger. Efter indhentning af blodtrykssignal er systemet i stand til grafisk at vise signalet kontinuerligt, samt udskrive blodtrykssignalets systoliske-, diastoliske- og puls-værdier.

Krav 5

Til projektet er der opstillet to krav, som er formuleret i projektoplægget. Det er et krav at disse funktioner implementeres i produktet. Kravene lyder på at der skal udvikles et system, som kan tilsluttes et væskefyldt kateter, samt vise en blodtrykskurve på en computerskærm. Mere detaljeret vil det sige, at systemet skal indeholde to elementer. Først et elektronisk kredsløb, som forstærker signalet fra tryktransduceren og filtrerer det med et indbygget analogt filter. Derefter et program til at vise blodtrykket, som funktion af tiden.

Yderligere skal dette program opfylde kravene:

- Programmeres i C#
- Kunne kalibrere blodtrykssignalet og foretage en nulpunktsjustering
- Vise blodtrykket kontinuert
- Kunne lagre de målte data i en tekstfil eller en database
- Kunne filtrere blodtrykket via et digitalt filter, denne funktionskunne slås til og fra

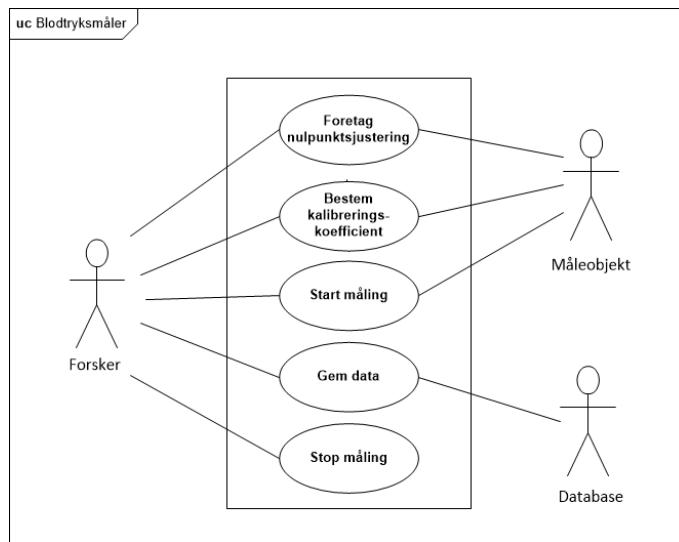
På baggrund af disse krav er der opstillet fem Use Cases, der tager højde for disse krav, samt beskriver aktørens interaktion med systemet. Disse Use Cases benyttes som kravspecifikation, der har til formål at specificere, hvilke krav der stilles til projektet. Udeover ovennævnte krav vil der også blive arbejdet hen imod at systemet skal afbillede puls, det systoliske blodtryk og det diastoliske blodtryk med tal.

Kravene opstilles ud fra kundens ønsker samt leverandørens mulighed for realisering. Systemet består af en computer med programkode, en NI-DAQmx, en transducer og hardware VERO Board med et filter og en forstærker. Den fulde beskrivelse af hver enkelt Use Cases (fully dressed Use Cases) findes i dokumentationen.

Aktørbeskrivelse

Use Case diagrammet viser de tre aktører: Forsker, Database og Transducer. Herunder er der en detaljeret beskrivelse af hver aktør.

Forsker er en primær aktør. Det er denne aktør, som foretager blodtryksmålingen, kalibrer og fortager nulpunktsjustering. Målingerne for blodtrykssignalet vises på displayet, som forskeren har tilgang til.



Figur 5.1: Use Case diagram

Transducer er en sekundær aktør. Transduceren har til formål at modtage blodtrykssignalet fra måleobjektet, som kan bestå af In Vitro, patient eller anden, som kan skabe et blodtrykssignal.

Database er en sekundær aktør. Denne aktør er en database, hvori blodtrykssignalets rådata gemmes. Ligeledes gemmes det indtastede forsøgsnavn og det autogenerede Id.

Use Case beskrivelse

Use Case diagrammet viser ligeledes de fem Use Cases, der er for systemet: Foretag nulpunktsjustering, Bestem kalibreringskoefficient, Start Måling, Gem data, Stop måling. Disse Use Cases beskriver interaktionen mellem aktørerne og systemet. Herunder er der en kort beskrivelse formålet med hver Use Case.

UC1: Foretag nulpunktsjustering Når systemet startes op vil det første der møder forskeren være en GUI, hvorfra nulpunktsjustering kan foretages. Det forudsættes at forskeren har åbnet for transduceren, så den modtager atmosfærisk tryk inden at forskeren trykker på foretag. Systemet indhenter så en nulpunktsjusteringsværdi, denne værdi gemmes i softwaren og alt indhentet signal herefter vil dermed være indstillet til en offset på nul. Herefter går systemet videre til næste Use Case

UC2: Bestem kalibreringskoefficient Kalibreringen foregår uafhængig af om systemet kører. Forsker tilslutter hardware til væskesøjle ved 50 mmHg. Så aflæses output spænding fra hardwaren. Kalibreringskoefficienten kan så bestemmes ved en simpel beregning ud fra tryk og output spænding. Denne værdi indtastet så i fil, hvorfra softwaren kan tilgå koefficienten og derefter benytte den som omsætningskoefficient mellem volt og mmHg. Dermed er en kalibrering udført.

UC3: Start måling Det kræves at måleobjekt, hvorpå blodtrykssignal ønskes fra, er tilsluttet. Derfor tilslutter forskeren transduceren til måleobjektet. Derpå kan målingen startes ved, at forskeren trykker på knappen START MÅLING. Herefter indhenter systemet blodtrykssignalet, som bliver udskrevet på display. Værdier for puls, systolisk- og

diastolisk blodtryk udskrives på display.

UC4: Gem data Det er muligt for forskeren at gemme det indhentede blodtrykssignals rådata, indenfor en periode, valgt af forskeren. Dette gøres ved, at forskeren trykker på knappen START GEM. Systemet vil herefter begynde at sende rådata ind i databasen, hvor dataene gemmes. Dette vil systemet blive ved med at udføre indtil forskeren trykker på knappen STOP GEM. Ved tryk på STOP GEM vises filnavn på display.

UC5: Stop måling Det er muligt for forskeren at stoppe visning af blodtrykssignalet. Dette gøres ved, at forskeren trykker på knappen STOP MÅLING. Systemet vil herefter lukke forbindelsen til indhentning af data og grafen i chart vil fastholdes.

Ikke-funktionelle krav beskrivelse

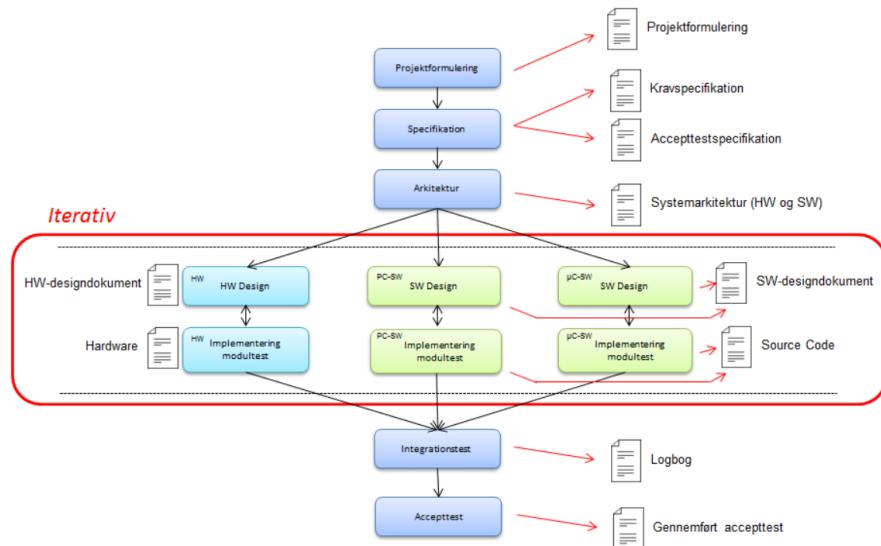
Ikke-funktionelle krav er struktureret efter (F)URPS+, hvor krav til systemets funktionalitet, brugervenlighed, pålidelighed, præsentation samt vedligehold er beskrevet. Disse krav er primært software-krav. Der opstilles bl.a. et krav om en maksimal tid, der må gå fra, at der er trykket på en knap, til at systemet reagerer. Når der er trykket på en knap, skal systemet foretage den ønskede proces, hvilket eksempelvis er, at ved tryk på GEM-knappen, at sende det ufiltrerede blodtrykssignal til databasen, hvor signaldata bliver gemt. Ligeledes er opbygning af display også en del af de ikke-funktionelle krav.

Projektbeskrivelse

6

6.1 Projektgennemførelse

Dette projekt er gennemført vha. forskellige udviklingsprocessor, hvilket er med til at sikre kvalitet, og at deadlines overholdes. En af disse modeller er ASE-modellen [5]. Denne model er en udviklingsmodel, der er udarbejdet af Aarhus Ingeniørhøjskole. Modellen er en gentagelig udviklingsproces drevet ud fra projektets Use Cases. Modellen er benyttet på den måde, at gruppemedlemmerne fastlægger en projektformulering, kravspecifikation og systemarkitektur, for derefter at designe og implementere de enkelte hardware- og softwaredelene. Gennem en integrationstest ses det om hardware- og softwaredelene fungere som de skal. Dette ender med en gennemført accepttest, således at det testes om systemet lever op til kravene.

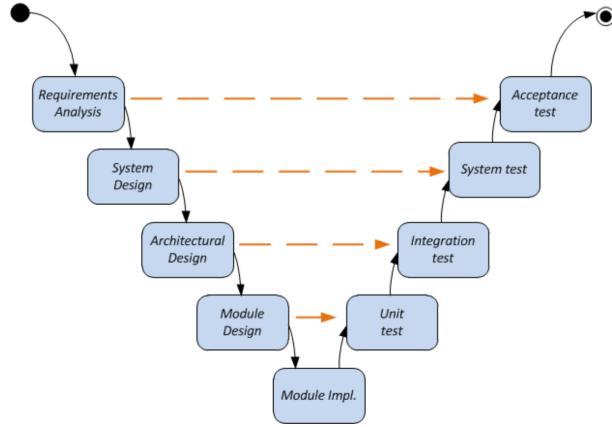


Figur 6.1: ASE-modellen

For at forstå ASE-modellen [5] er det vigtigt at gennemgå Use Cases; et værktøj, som skal beskrive interaktioner mellem diverse aktører og selve systemet. Sammen med de ikke-funktionelle krav opnås et overblik over hvilke funktionalitetskrav, der stilles til systemet. På baggrund af kravspecifikationen kan accepttesten efterfølgende udarbejdes. I dette projekt er hardware- og software design implementering på lige fod, da projektet består af begge ting ligeligt.

V-modellen [5] er en faseopdelt udviklingsmodel, der også er værd at nævne i dette pro-

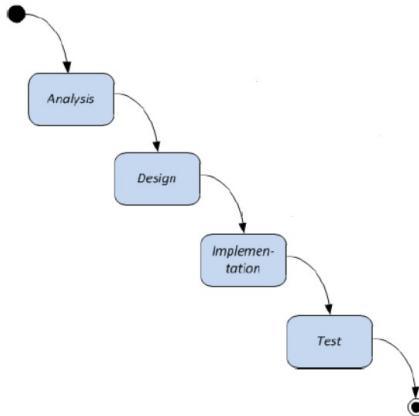
jekt. Den beskriver udviklingsfaserne og testfaserne sideløbende i forhold til projektet, og den er derfor benyttet til dette projekt sideløbende med ASE-modellen. Modellen fungerer således, at specifikationen af test foregår parallelt med udviklingen af selve systemet.



Figur 6.2: V-model

Den er blevet benyttet til hardware- og softwareudviklingen. Hardware og software skulle begge teste deres funktioner inden nye faser blev igangsat, for at verificere om disse funktioner virkede korrekt gennem forløbet. Fordelen ved at teste på forskellige niveauer er, at det skal sikre de udviklede delsystemer, således at de virker som planlagt. Det er vigtigt, at hver fase er udført, før den næste fase påbegyndes.

Vandfaldsmodellen [5] er også benyttet under dette projekt. Softwareudviklingen bærer præg af vandfaldsmodellen, da udviklingen er opdelt i faser, hvor hver fase er blevet gennemført, før den næste er påbegyndt. Dette er i relation til V-modellen, som blev beskrevet før og den er konstant strømmende nedad.



Figur 6.3: Vandfalds modellen

6.2 Projektstyring

Projektet er udarbejdet over et helt semester, hvor undervisningen og forelæsningerne delvist har udgjort grundlaget for teorien benyttet i dette projekt. Der blev i starten udarbejdet en tidsplan, som dog var mulig at ændre undervejs, men med faste deadlines, der skulle overholdes. Den endelige tidsplan kan ses i bilag.

Projektgruppen har bestået af 6 gruppemedlemmer, som er blevet delt i 2 fokusområder, hardware og software. Fordelingen blev udarbejdet efter den enkeltes ønske. Gruppen har derfor været afhængig af at der var god kommunikation mellem de to undergrupper, derfor har hver arbejdsgang været i samarbejde. Da gruppen har været opdelt, har der været projektmøde hver uge, hvor gruppen har opdateret hinanden og vejleder.

Under projektet har alle medlemmer været med til, at sikre en administrativ kæde af deadlines til individuelle opgaver, samt dagsordener til hvert møde. Disse deadlines har sikret at opgaverne er blevet opfyldt op til møderne, for at hindre langsomme arbejdsprocesser. Da der har været overlap mellem de forskellige opgaver er arbejdet foregået flydende for at sikre, at der blev testet.

6.3 Metoder

Til at kunne overskue arkitektur og designet af projektet, er flere forskellige arbejdsmetoder benyttet for at skabe det bedst mulige resultat. For at finde, hvad blodtryksmåleren skal gøre, er der blevet udarbejdet Use Cases. Disse beskriver systemet funktionalitet. Use Cases viser, hvad brugeren skal opleve fra systemet, men ikke, hvordan det sker. I Use Case diagrammet bliver det også vist, hvilke aktører der findes og hvordan de interagerer med systemet.

I projektet bruges accepttest til at teste blodtryksmåleren. Dette gøres ud fra kravspecifikationen, hvor det er angivet, hvilke krav der er stillet til systemet.

Accepttesten er en test, hvor der beskrives, hvad der skal ske og, hvad brugeren skal gøre. Testen er for at undersøge om produktet opfylder de krav, der er blevet sat for det. Accepttesten giver et godt overblik for udvikleren og for kunden, der nemt og hurtigt kan se om produktet virker som det skal.

Til beskrivelse af design af software og hardware er diagrammer og skemaer blevet udarbejdet i SysML og UML. SysML er et grafisk modelleringsprog, som kan bruges til at overskueliggøre systemer.

Til software er der blandt andet lavet en applikationsmodel i SysML, som består af et domæne-, klasse- og sekvensdiagram.

Domænemodellen viser sammenhængen mellem blokkene i systemet. Blokkene findes i Use Casene og derved bliver disse to ting koblet sammen.

Klassediagrammet viser, hvilke metoder blokkene har og hvordan de kommunikerer med hinanden. Her findes domæne-, kontrol- og grænsefladeklasser. Kontrolklasserne beskriver, hvordan data behandles mellem domæne- og grænsefladeklasser. Domæneklasser indeholder funktionalitet fra den pågældende softwareblok. Grænsefladeklasserne viser, hvordan, systemet interagerer med omverdenen. Diagrammet gør det nemmere at fremme en lav kobling og høj samhørighed i softwaren.

Sekvensdiagrammet fortæller, hvad der sker i selve koden. Igen går det ud fra Use Casene,

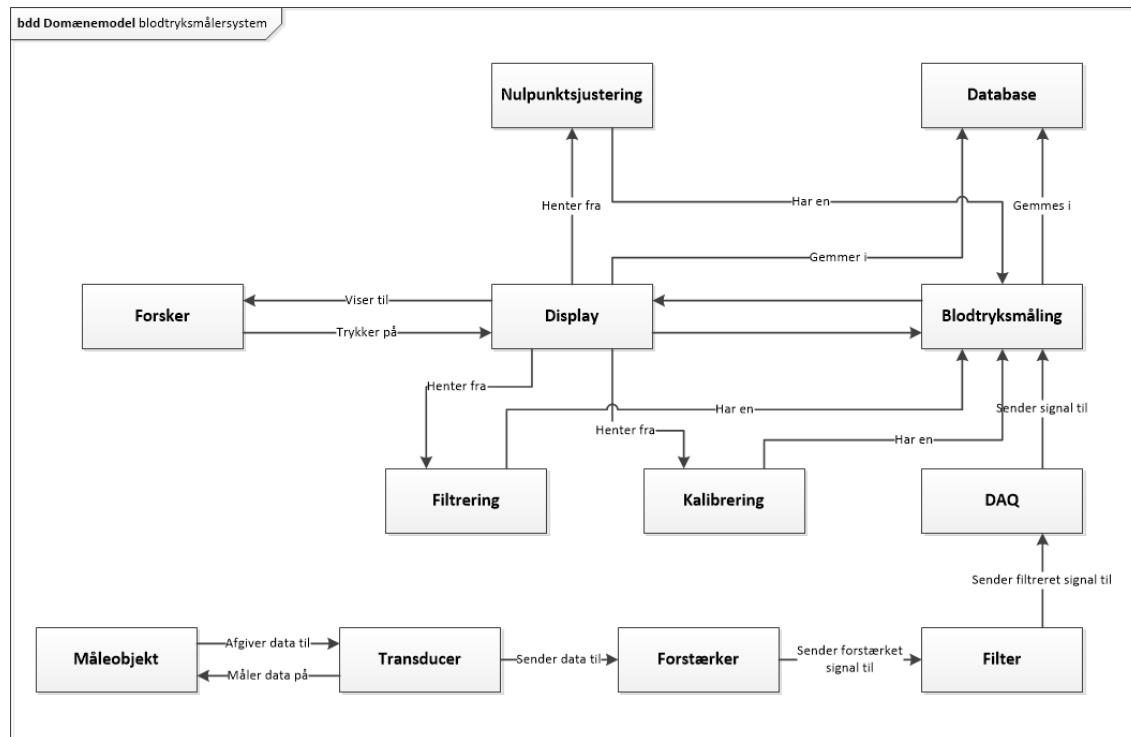
hvor vægten nu er på softwaredelen. Derved beskrives det, hvordan metoder bliver kaldt og hvordan de forskellige klasser interagerer. Hver Use Case skal her gennemgås i softwaren, så der skabes et overblik over vejen gennem koden.

For at skabe et overblik og indsigt i koden, er der i UML udarbejdet et aktivitetsdiagram og et klassediagram. Aktivitetsdiagrammet går i dybden med en specifik metode. Det er kun blevet gjort for relevante metoder. Her tydeliggøres det, hvordan hver metode fungere og, hvad den indeholder. Klassediagrammet fortæller hvilke metoder, en klasse indeholder og hvordan klasserne hænger sammen.

Til hardwaren er der blevet brugt Block Definition Diagram(BDD), som viser hvilke blokke et system indeholder og hvilke porte de har. BDD er lavet til at give et overblik over systemet. Ud fra BDD'et er et Internal Block Diagram(IBD) blevet lavet. Her vises, hvilke signaler, som findes i systemet og hvordan de sendes rundt. Her vises portene igen og der skal være overensstemmelse mellem BDD og IBD.

Til udarbejdelsen af kredsløb blev Analog Discovery brugt til at simulerer signalet, som i sidste ende skal komme fra transduceren. Først blev kredsløbet opbygget på et fumlebræt, hvor det blev testet for at afprøve om det lever op til kravene. Når det opfylder kravene, flyttes det over på et VEVO Board. VEVO Boardet bliver igen testet før aflevering.

6.4 Systemarkitektur



Figur 6.4: Domæne Model

6.4.1 Hardware

I hardwaredelen skal der ligge en forstærkning og et lavpasfilter. Det differentieret signal fra transduceren skal forstærkes og filtrers før det kan sendes ind i DAQ'en.

6.4.2 Software

6.5 Problemidentifikation (design)

6.5.1 Hardware

Forstærkning

Transduceren mäter en trykændring, som den omsætter til en spænding. Dette er udtrykt ved et differentieret signal, som sendes ind i forstærker-blokken.

Signalet fra transduceren er en lav spænding, som skal forstærkes op, for at passe med DAQ'ens input. Denne forstærkning udregnes ud fra det maksimale output fra transduceren og det maksimale input til DAQ'en. Se beregningerne under implementering.

Under simulering bruges Analog Discovery som en funktionsgenerator, der simulerer det differentieret signal.

Lavpas

I projektet skal der laves et 2. ordens lavpasfilter. Filteret skal laves for at sikre, at der ikke opstår aliasering.

Aliasering [6] er, hvor signalet bliver gentaget. Når man har signalet i det digitale domæne, bliver spektret for signalet en periodisk funktion.

Det skal sikres, at der ikke kommer overlap mellem signalet og et alias. Da det ellers kunne give anledning til misforståelser. Derfor laves et lavpasfilter, som sikrer at der ikke ligger noget signal ved den halve samplingsfrekvens.

Lavpasfilteret skal være et Sallen-Key Butterworth-filter med en knækfrekvens på 50 Hz og en samplingsfrekvens på 1kHz. Ud fra oplysninger givet til projektet, vides det at filteret skal dæmpe signalet med 20 dB, under antagelse af at den forekommende støj er mindre end signalet, også når det forekommer over knækfrekvensen.

Ved en typisk blodtryksmåling forekommer der ikke signal over 50 Hz, samtidigt er signalet her aftaget med ca. 70 dB. For at få signalet, ved den halve samplingsfrekvens til at være $1/2 \cdot LSB$ (Least Significant Bit), skal det ydeligere dæmpes 20 dB. Derfor oplyses filteret til at være 50 Hz, da dette giver en minimum dæmpning på 20 dB pr. dekade.

6.5.2 Software

6.6 Implementering

6.6.1 Hardware

Forstærkning

For at få den rette forstærkning er det blevet valgt, at benytte instrumentationsforstærkeren INA-114. Her kan transduceren sættes på med det differentierede signal. INA-114 er valgt da følgende gælder[7] for instrumentationsforstærkere:

- Differentielt input - single ended output
- Gain justering med ændring af kun én modstand
- Meget høj indgangsimpedans
- Stor Common Mode Rejection Ratio(CMRR)

For at udregne den korrekte forstærkning, bruges følsomheden fra transduceren og eksistationsspændingen. Først udregnes det maksimale output fra transduceren:

$$9V \cdot 250mmHg \cdot 5\mu \cdot 10^{-5}uV/V/mmHg = 11.25mV \quad (6.1)$$

Da det er besluttet at det maksimale input til DAQ'en [6] er 5V, kan forstærkningen (Gain) nu udregnes:

$$\begin{aligned} 5V &= 11.25mV \cdot G \\ G &= 444.44 \end{aligned} \quad (6.2)$$

For at få den rette forstærkning udregnes den eksterne modstand (R_g) til INA-114 [8]. INA-114's forstærkning afhænger af størrelsen på R_g , hvis modstanden er stor, er forstærkningen lille og omvendt. R_g udregnes ved formlen:

$$\begin{aligned} G &= 1 + \frac{50k\Omega}{R_g} \\ 444.44 &= 1 + \frac{50k\Omega}{R_g} \Rightarrow R_g = 112.75\Omega \end{aligned} \quad (6.3)$$

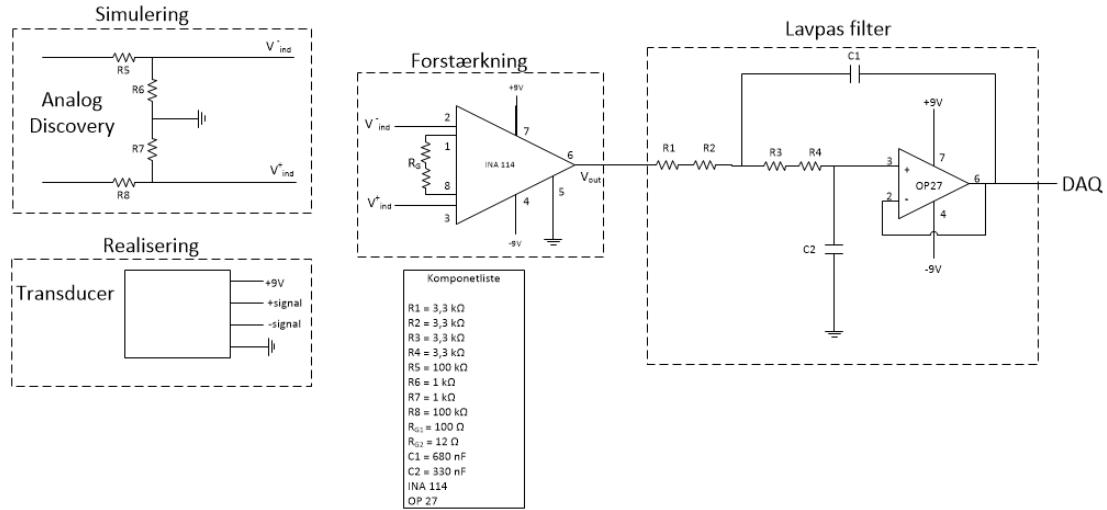
Derved fås en værdi for den eksterne modstand til INA-114, som skaber den ønskede forstærkning.

Den ønskede forstærkning kan bruges, da det passer overens med båndbredden. Dette kan undersøges da produktet af forstærkning og båndbredde er konstant og båndbredden skal ligge over knækfrekvensen for filteret. Se beregning i Dokumentation ligning 3.4.

For at imødekommme usikkerheden ved Analog Discovery, der bruger lave spændinger, laves et kredsløb efter spændingsdelerprincippet. Signalerne fra Analog Discovery skal sendes igennem dette kredsløb, hvor de efter spændingsdelerprincippet gøres mindre.

Derved kan Analog Discovery sende signaler med en højere spænding ind i kredsløbet og usikkerheden mindskes. Hvis INA-114 skal have 11.25mV skal Analog Discovery sende 1.1352V ind.

På figur 6.5 ses et diagram af det endelige kredsløb med komponentværdier. Her ses, hvordan det ser ud ved realiseringen med transduceren og ved simuleringen ved Analog Discovery.



Figur 6.5: Diagram over HW

Lavpas

For at opnå den ønskede effekt i lavpasfilteret, blev det oplyst at $f_c = 50$ Hz, $f_s = 1\text{kHz}$, $R_1 = R_2$ og $C_2 = 680\text{nF}$. Ud fra disse værdier, udregnes de resterende komponentværdier for filteret.

Overføringsfunktionen for et 2. ordens filter er:

$$H(z) = \frac{\omega_n^2}{(s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2)} \quad (6.4)$$

For at finde overføringsfunktionen for det gældende system, vides det at følgende ligninger gælder [9]:

$$\begin{aligned} \omega_n &= 2 \cdot \pi \cdot 50 = \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \\ 2 \cdot \zeta \cdot \omega_n &= \frac{1}{C_2} \cdot \left(\frac{R_1 + R_2}{R_1 \cdot R_2} \right) \end{aligned} \quad (6.5)$$

Dette indsættes i den generelle overføringsfunktion og det simplificeres, blandt andet ved at det vides at $R_1 = R_2$. Se Beregning af overføringsfunktion i Bilag for nærmere udregninger:

$$H(z) = \frac{\frac{1}{C_1 \cdot C_2 \cdot R^2}}{s^2 + s \cdot \frac{2}{R \cdot C_2} + \frac{1}{C_1 \cdot C_2 \cdot R^2}} \quad (6.6)$$

Når der arbejdes med et 2. ordens Butterworth filter, vides det at udsvinget ζ skal have værdien 0.7 [10]. Under beregningerne, var der usikkerhed omkring, hvad værdien af ζ skulle være. Da kredsløbet skulle realiseres og dokumenteres blev det derfor overvejet at ændre samtlige komponentværdier så de passede med en $\zeta = 1$. Inden dette blev gjort blev det dokumenteret at det gælder at ζ skal have en værdi på 0.7.

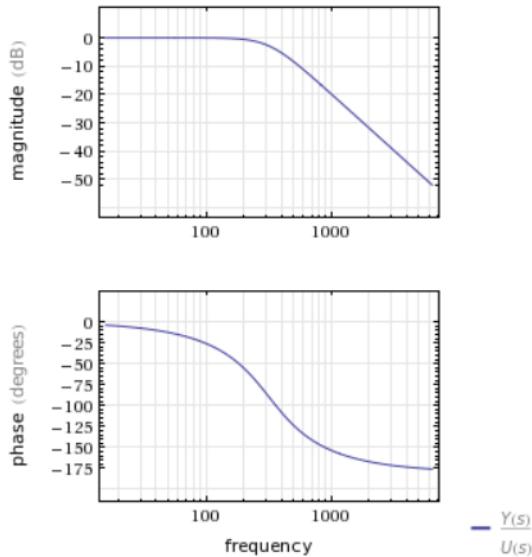
Den sidste overføringsfunktion sammenlignes med den generelle for 2. ordens systemer. Det gælder at $C2 = 680 \cdot 10^{-9}nF$. Det er så muligt at isolerer forskellige led. Først isoleres der for modstanden(midterste led i nævneren) og den udregnes til $R = 6687\Omega$.

Nu kan det tredje led i nævneren isoleres og kondensatoren $C1$ udregnes til $C1 = 333 \cdot 10^{-9}nF$. Nærmere beregninger kan ses under Systemarkitektur i Dokumentationen.

Alle komponentværdierne til lavpasfilteret er fundet og det kan nu realiseres.

Under udviklingen af lavpasfilteret er komponentstørrelserne, blevet ændret for at kunne realisere det. De brugte komponentværdier er: $R = 6.6k\Omega$, $C1 = 330 \cdot 10^{-9}nF$ og $C2 = 680 \cdot 10^{-9}nF$. For at være sikker på at filteret har de ønskede karakteristika, laves et bodeplot for den endelig overføringsfunktion:

$$H(z) = \frac{62500000000}{610929 \cdot \left(s^2 + \frac{250000}{561} \cdot s + \frac{62500000000}{610929} \right)} \quad (6.7)$$



Figur 6.6: Bodeplot

Ud fra den nye overføringsfunktion udregnes en nu ζ for at kontrollere at værdien ikke har ændret sig. Denne udregnes til $\zeta = 0.709$. Derfor kan det konkluderes at filteret stadig har den ønskede funktionalitet.

6.6.2 Software

6.7 GUI-beskrivelse

6.7.1 Algoritmer (grænseværdier)

6.7.2 Filteret/Ufiltreret

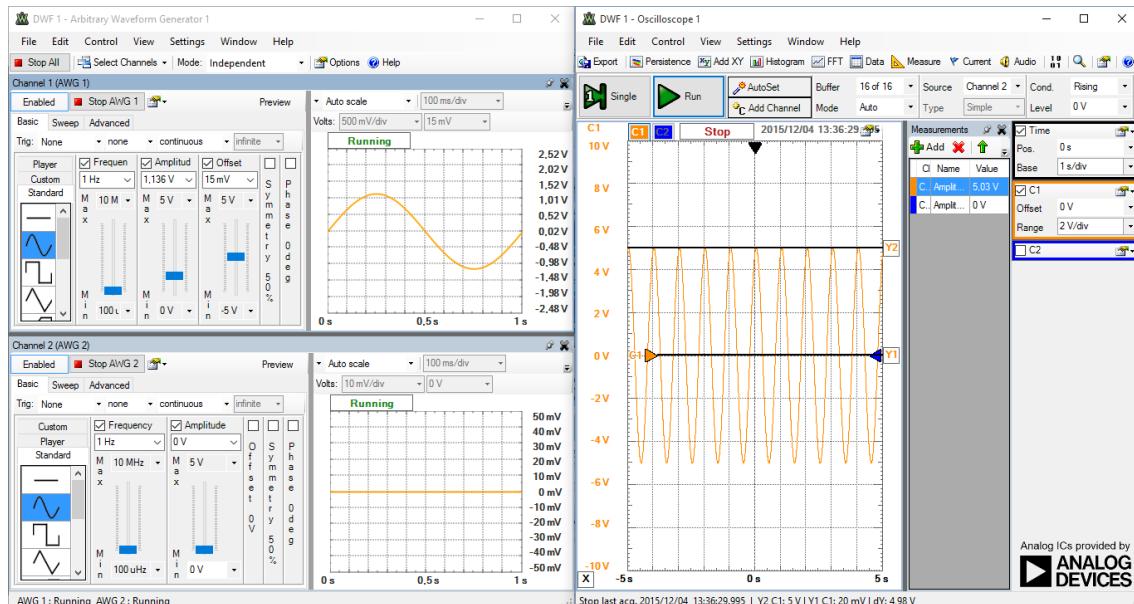
6.7.3 Lagring af data i Database

6.8 Test

6.8.1 Hardware

Forstærkning

For at teste forstærkningen, sendes et differentieret signal ind vha. Analog Discovery. Her observeres der hvor meget signalet bliver forstærket. På figur 6.7 ses det signal, som sendes ind i forstærkningsblokken og det, der måles på udgangen af blokken.



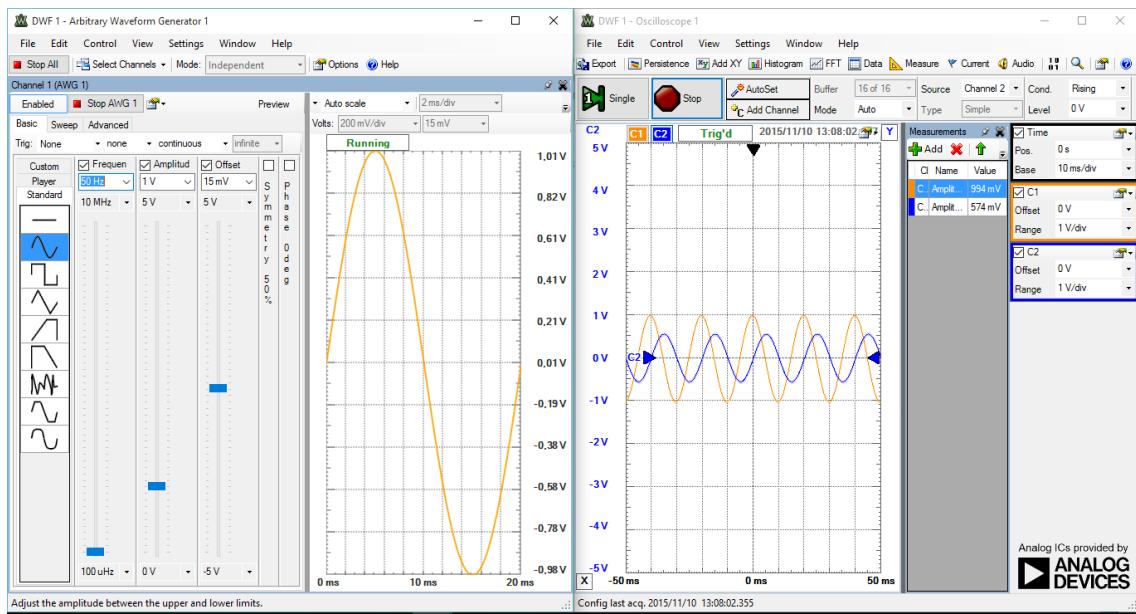
Figur 6.7: Forstærkningsblok

Der sendes et differentieret signal ind i INA-114. På udgangen ses det at signalet er blevet forstærket op til 5V DC. Herved er maksimum input til forstærkningsblokken blevet forstærket så det passer med maksimum input til DAQ'en. Signalet bliver ikke ændret på andre måde i denne blok.

Lavpas

For at teste lavpasfilteret foretages målinger med en sinus, hvor frekvensen varierer for hver måling. Derved aflæses fasen mellem indgang- og udgangssignal, samt amplituden for hver måling. Der laves flere målinger, både før, ved og under knækfrekvensen. Ved knækfrekvensen skal fasedrejningen være 90°. Dette kan aflæses på figur 6.8.

Se Modultest i Dokumentationen for mere dokumentation.

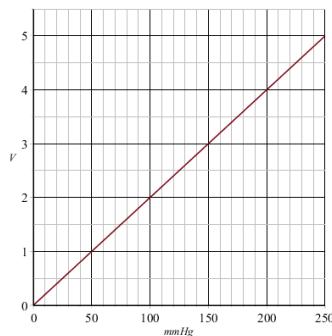


Figur 6.8: Måling for 50 Hz

Kalibrering med vandsøjle

Efter forstærkning og lavpasfilteret er blevet testet hver for sig, udføres en kalibrering af systemet vha. en vandsøjle. Her bruges en udleveret vandsøjle med tre målepunkter, hvor det er angivet hvor højt trykket (målt i milimeter kviksølv, mmHg) er ved hvert af disse punkter. Derved kan det testes om hardwaren måler den rigtige spænding i forhold til mmHg.

Ud fra den maksimale spænding (målt i Volt, V) og mmHg kan det udregnes, hvad hardware skal vise ved 100 mmHg, se figur 6.9.



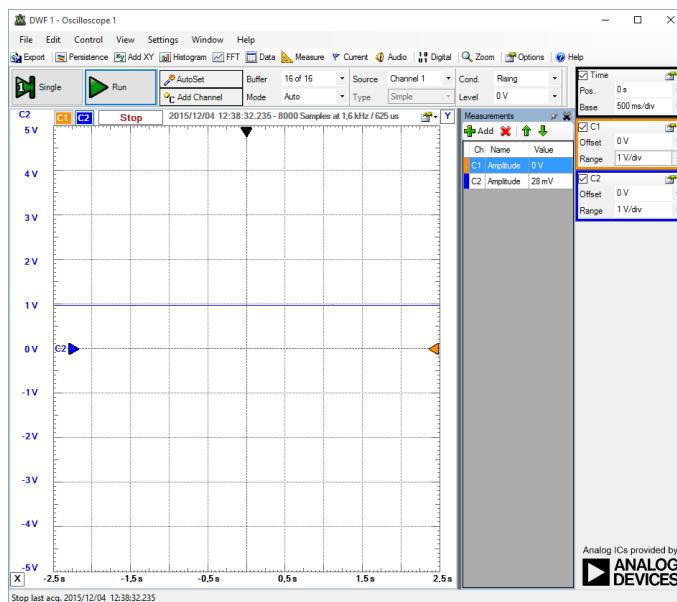
Figur 6.9: Graf til kalibrering, fra udregninger

Testen udføres ved at fylde vand i søjlen til et bestemt punkt. Transduceren skal være tilkoblet netop dette målepunkt, mens de andre er lukket til. Transduceren er sat til hardwaren, der hvor Analog Discovery tidligere har været sat til. Transduceren er tilkoblet 0-9V ved batterierne. På samme måde, som ved simuleringen, aflæses målingen på computeren ved hjælp af programmet WaveForms. Da det vides hvilken trykændring der måles på, ved vi fra grafen til kalibreringen hvilken spænding den skal vises. Dette fortages for de tre målepunkter på vandsøjlen, hvor hver måling sammenlignes med den udregnede graf. For hver måling, skal transduceren flyttes til et af de andre målepunkter.



Figur 6.10: Opstilling

Ud fra grafen i figur 6.9 vides, hvad svaret til hver måling skal være. På figur 6.11 ses målingen, da transduceren var tilkoblet målepunktet for 50 mmHg. Ud fra figur 6.9 vides det at målingen skal vise 1V DC.



Figur 6.11: Måling ved 50 mmHg

Ud fra figur 6.9 kendes udgangsspændingen også for de to andre målepunkter. Se under Modeltest i Dokumentationen for billeder af målingerne ved 10 mmHg og 100 mmHg. Målingen for 50 mmHg er her valgt ud, da denne måling ligger til baggrund for kalibreringen i softwaren.

6.8.2 Software

6.9 Resultater og diskussion

Resultaterne for projektet er at ...

Under projektet er vores tidsplan skredet adskillige gange. Blandt andet pga. mange uforudsete problemstillinger og udefrakommende faktorer.

Ved softwaren er det specielt teorien bag tråde og derfor forståelsen derom, som har manglet og derved skabt flere problemer. Teorien for tråde skulle være kommet i undervisningen i kurset IT3, dette har dog ikke været fyldestgørende i forhold til brugen i projektet.

Ved hardware har der været en manglende forståelse for de signaler som sendes ind og måles ved udgangen. Det har givet en stor usikkerhed for, hvornår de forskellige blokke har virket korrekt. Derfor er der blevet spildt tid på at tro at systemet ikke virkede, mens det i virkeligheden gjorde som det skulle. De udefrakommende faktorer som har spillet ind var en eksamen, som har lagt midt i det hele. Den har naturligvis være kendt hele forløbet, men det satte en stopper for projektet på ca. 1 uge.

6.10 Udviklingsværktøjer

Gennem projektarbejdet har vi anvendt en række forskellige værktøjer til udvikling af blodtryksmåler-systemet. Disse er yderligere uddybet herunder.

Visual Studio 2013

Softwaredelen af projektets programmering er skrevet i sproget C-sharp. Her er Visual Studio 2013 anvendt som kompiler, da programmet gør det nemt at omskrive tekst til kode. Visual Studio 2013 indeholder også funktionen Windows Form Application, der visuelt kan fremstille de ønskede resultater i form af knapper, grafer og labels mv. i en samlet brugergrænseflade, som aktøren interagerer med.

Microsoft Visio 2016

Microsoft Visio er et tegne værktøj, der i dette projekt er anvendt til at designe både SysML og UML diagrammer, som benyttes ved organisering af hardware og software design. Microsoft Visio er det oplagte valg, da diagrammer lavet i programmet får et enkelt og overskueligt udseende, og dermed fremstår det tydeligt for læseren hvad diagrammet vil vise.

Analog Discovery og Waveform fra Digilent

Analog Discovery og Waveform er i projektet benyttet som omformer og signal generator under testfasen. Her fungerer Analog Discovery som en Waveform generator, så et analog signal kan sendes videre ind i lavpasfiltret, forstærkeren og derefter ind i DAQ'en. I den endelig implementering erstattes Analog Discovery og Waveform med transduceren.

NI-DAQmx

NI-DAQmx er et værktøj udarbejdet af National Instruments, som anvendes til at omforme det indkomne analoge signal fra transduceren (Analog Discovery) til et digital signal. Værdier fra NI-DAQmx er af en type som kan anvendes i selve softwarekoden.

LaTeX

LaTeX er anvendt i projektet til design og opsætning af projektrapport og projektdokumentation. LaTeX er god til tekstformatering, hvor opsætning og strukturer defineres samlet for hele rapport, samt god til versionsstyring. Til at skrive selve koden benyttes programmet TeX-maker som kombiler.

6.11 Opnåede resultater

6.12 Perspektivering - Fremtidigt arbejde

I fremtiden vil blodtryksmåleren kunne udvides gennem flere muligheder. Da blodtryksmåleren er lavet til forskningsbrug, er der ingen idé i at udvide mod patienter. En forlængelse af systemet kunne derimod være en metode, som skal kunne vise gemte målinger.

Et log-in vindue er en anden ting, som kunne forbedre systemet, for på den måde at skabe større sikkerhed for forskeren og dataen. Et log-in vindue vil gøre at, en forsker kan være sikker på at hans målinger og forskning ikke kan tilgås af andre. Det kræver en større udvidelse, hvor der skal laves et log-in vindue og en database, hvor password og brugernavn gemmes. Der skal også laves en metode, som kan tjekke om det indtastede password og brugernavn passer overens med det i databasen.

Generelt skal de standarter, som findes for blodtryksmålere undersøges grundigere. Specielt brugergrænsefladen, men også resten af systemet som enheder og visning af graf, skal rettes til efter de passende standarter.

Hvis systemet ydeligere skulle tilpasses forskning, kunne det gøres gennem en bedre navngivning af data i tabellen eller et bedre overblik over, hvordan data bliver gemt f.eks. gennem en liste for de gemte målinger. På den måde vil det blive nemmere for forskeren at finde frem til gamle målinger.

I forhold til hardware er målet, at det hele skal samles i en kasse. Så det på den måde ikke er muligt at ændre eller stille ved det. Derved skal filteret og forstærkningen laves på en printplade. Samtidigt skal det ved kassen være en plads til batterierne, hvor det er muligt at kunne skifte dem, når nødvendigt. Derved fås en kasse, som nemt kan flyttes rundt på og som ikke er i fare for at gå i stykker.

Konklusion 7

Formålet med projektet var, at udvikle en blodtryksmåler, som består af en hardware- og en softwaredel. Hardwaredelen består af en forstærker blok og et lavpasfilter. Hardwaren modtager et differentieret signal fra transduceren som forstærkes og filtreres inden det sendes gennem en DAQ og videre ind i softwaren. Softwaren vil gennem algoritmer, kunne detektere systoliske- og diastoliske værdier. Ud fra de to værdier ville programmet kunne udregne pulsen.

Da projektet blev lavet til forskningsbrug skulle forskeren kunne gemme sine målinger i en database. Herudover var formålet med projektet, at blodtryksmåleren skulle kunne blive kalibreret og nulpunktsjusteret, når det var ønsket.

I dette projekt er det lykkedes, at udvikle en repræsentativ prototype, der opfylder de overordnede formål. Systemet viser, på under tre sekunder, den systoliske- og diastoliske værdi. Det er derfor hurtigt og nemt, for en forsker, at få målbare data.

Selvom dette kun er en prototype, så er hardwaredelen blevet udviklet til at være brugervenlig. Kredsløbet er blevet loddet på et VEVO Board og lagt i en kasse. Batterierne holdes udenfor kassen, da det på denne måde vil være ubesværet for forskeren, at skifte batterierne. På kassen vil en lille diode lyse, når batterierne er sat til.

På trods af et funktionsdygtigt slutprodukt, må det konkluderes, at arbejdsprocessen blev stressede. Både hardware- og softwaredelen tog længere tid end forventet. Hensigten med den første tidsplan var god, men pga. DSB miniprojekter og en KSS eksamen, blev tidsplanen skubbet og især softwaredelen blev presset hen mod slutningen. Derfor er det ikke lykkedes at opfylde alle punkter i kravspecifikationen (disse punkter er beskrevet i problemrapporten).

Alt i alt er der blevet udviklet en fornuftig prototype, der med en smule videreudvikling, uden besvær, ville kunne bruges af en forsker.

Bilag 8

Bilagene kan findes på den tilførende CD. Herunder findes en liste over bilagene.

1. Samarbejdsaftale
2. Beregninger af overføringsfunktion
3. OP27 Datasheet
4. INA114 Datasheet
5. Transducer Datasheet
6. NI-6009 DAQ Datasheet

Litteratur

- [1] Egil Haug og Jan G. Bjålie Olav Sand, Øystein V. Sjaastad. *Menneskets anatomi og fysiologi*. Gads forlag, 2008.
- [2] Egil Haug og Jan G. Bjålie Olav Sand, Øystein V. Sjaastad. *Menneskets anatomi og fysiologi*. Gads forlag, 2008.
- [3] Hanne Ramløv Ivarsen og Ahmed Aziz. *Sygdomslære for sundhedsprofessionelle*. Gads forlag, 2013.
- [4] <http://prodoktor.dk/lavt-blodtryk/> : d. 26.11.2015 kl. 09:33.
- [5] Kim Bjerge. *Development Processes*. IHA, 2015.
- [6] Richard G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall, 2011.
- [7] Peter Johansen. *Instrumentationsforstærkeren*. IHA, 2014.
- [8] *INA114 Datasheet*.
- [9] https://en.wikipedia.org/wiki/Sallen%E2%80%93Key_topology.
- [10] Gregory J. Toussaint Rolande E. Thomas, Albert J. Rose. *The Analysis and Design of Linear Circuits*. Wiley, 2012.

Figurer

4.1	Skitse af signalændring	8
5.1	Use Case diagram	11
6.1	ASE-modellen	13
6.2	V-model	14
6.3	Vandfalds modellen	14
6.4	Domæne Model	16
6.5	Diagram over HW	19
6.6	Bodeplot	20
6.7	Forstærkningsblok	21
6.8	Måling for 50 Hz	22
6.9	Graf til kalibrering, fra udregninger	22
6.10	Opstilling	23
6.11	Måling ved 50 mmHg	23



AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI
3. SEMESTERPROJEKT

Dokumentation

Gruppe 2

Anne Bundgaard Hoelgaard (201404492)
Mette Hammer Nielsen-Kudsk (201408391)
Ditte Heebøll Callesen (201408392)
Martin Banasik (201408398)
Albert Jakob Fredshavn (201408425)
Johan Mathias Munk (201408450)

Vejleder

Studentervejleder
Peter Johansen
Aarhus Universitet

13. december 2015

Gruppemedlemmer

Anne Bundgaard Hoelgaard (201404492)	Dato
Mette Hammer Nielsen-Kudsk (201408391)	Dato
Ditte Heebøll Callesen (201408392)	Dato
Martin Banasik (201408398)	Dato
Albert Jakob Fredshavn (201408425)	Dato
Johan Mathias Munk (201408450)	Dato

Vejleder

Peter Johansen	Dato
----------------	------

Ordliste

Ord	Forklaring
(F)URPS+	Et akronym, der repræsenterer en model til klassificering af softwarens kvalitet
GUI	Graphical User Interface (Grafisk brugergrænseflade)
VPN	Virtual Private Network
DAQ	Data acquisition, NI-6009 DAQ

Indholdsfortegnelse

Ordliste	ii
Kapitel 1 Indledning	1
Kapitel 2 Kravspecifikation	2
2.1 Versionshistorik	2
2.2 Godkendelsesformular	3
2.3 Indledning	4
2.4 Systembeskrivelse	4
2.5 Funktionelle krav	4
2.5.1 Aktør-kontekstdiagram	4
2.5.2 Aktørbeskrivelse	5
2.5.3 Use case-diagram	5
2.5.4 Use Cases	6
2.6 Ikke-funktionelle krav	9
2.6.1 (F)URPS+	9
Kapitel 3 Systemarkitektur	12
3.1 Hardware	13
3.1.1 Design	13
3.1.2 Implementering	14
3.1.3 Modultest	18
3.2 Software	23
3.2.1 Design	23
3.2.2 Implementering	29
3.2.3 Modultest	36
Kapitel 4 Accepttest	37
4.1 Accepttest af Use Cases	37
4.2 Indledning	37
4.2.1 Use Case 1	38
4.2.2 Use Case 2	38
4.2.3 Use Case 3	39
4.2.4 Use Case 4	40
4.2.5 Use Case 5	42
4.3 Accepttest af ikke-funktionelle krav	43
4.4 Godkendelsesformular	50
4.5 Problemrapport	51
Litteratur	52

Figurer

52

Indledning 1

I dag bruges blodtryksmålere mange steder, både på hospitalet og i hjemmet. Blodtryksmålere kan måle en persons blodtryk, hvor den viser puls, samt diastoliske- og systoliske tryk i numerisk form og afbilledet i en graf.

Vi har valgt at arbejde ud fra, at blodtryksmåleren skal bruges til forskning. Derfor skal systemet gemme samtlige målinger, så en forsker senere kan tilgå dem. Samtidig skal puls og tryk vises på en graf, som skal være nem at aflæse. Bruger'en vil kunne benytte målere gennem et interface, hvor han kan starte og gemme målinger. Det er også her grafen vises. Der var fra start givet en række krav til systemet, samtidig har gruppen valgt at tilføje nogle flere for at få de ting løst, gruppen synes var vigtigt. Disse kan findes i Kravspecifikationen og under Krav.

Nærmere informationer om opbygning af hardware og software kan findes under Systemarkitektur, som er delt ind efter Hardware og Software. Her under findes også Modultest. Under Modultest kan det læses, hvordan vi har testet systemet samlet ogenkelt hvis for hardware og software. Under Accepttest ses det, om systemet opfylder kravene der blev sat.

Ansvarsområde

Initialer:

Albert Jakob Fredshavn - AJF

Martin Banasik - MBA

Mette Hammer Nielsen-Kudsk - MHNK

Ditte Heebøll Callesen - DHC

Johan Mathias Munk - JMM

Anne Bundgaard Hoelgaard - ABH

Afsnit	Ansvarlig
Indledning	
Kravspecifikation	
Hardware arkitektur	
Software arkitektur	
Software implementering	
Accepttest	
Fejlrapport	

Kravspecifikation 2

2.1 Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse
0.1	21-09-2015	MHNK og MBA	Oprettelse og udfyldning af kravspecifikation
0.2	24-09-2015	DHC og ABH	Omskrivning af UC1 - UC5
0.3	28-09-2015	ABH	Ikke-funktionelle krav
0.4	08-10-2015	Alle	Tilrette efter review med Grp. 1
0.5	15-10-2015	MBA	Indskrevet i LaTex
0.6	11-11-2015	ABH	Ændre Use Case 1 og 2 efter review med Grp. 4
0.7	20-10-2015	MHNK	Tilretning
0.8	26-11-2015	MHNK	Retning af hele kravspec.
0.9	09-12-2015	DHC	Rettelser ift. software
1.0	09-12-2015	MHNK	Rettelse af afsnit i rapport og dokumentation
1.1	10-12-2015	DHC, ABH	Rettelser i forhold til slutprodukt

2.2 Godkendelsesformular

Forfattere Anne Hoelgaard, Ditte Heebøll Callesen, Martin Banasik, Albert Fredshavn, Mathias Munk og Mette Hammer Nielsen-Kudsk

Godkendes af Peter Johansen

Antal sider 53

Kunde IHA

Ved underskrivelse af dette dokument accepteres det af begge parter, som værende kravene til udviklingen af det ønskede system.

Sted

Dato

Kundens underskrift

Leverandørens underskrift

2.3 Indledning

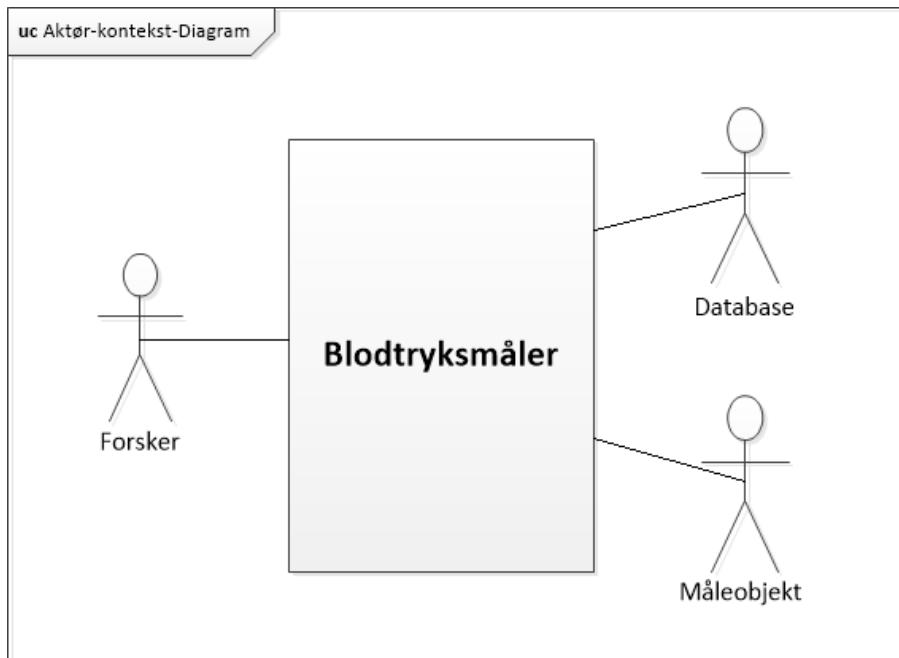
På baggrund af krav fra kunden samt, hvad leverandøren finder muligt, er denne kravspecifikation blevet udarbejdet. Kravspecifikation har til formål at specificere kravene til produktet. Dette projekt tager udgangspunkt i en blodtryksmåler, hvortil der er en række aktører, som interagerer med systemet, der er beskrevet yderligere nedenfor.

2.4 Systembeskrivelse

Blodtryksmålersystemet ønskes udviklet således at systolisk og diastolisk blodtryk samt puls kan bestemmes ud fra en invasiv ateriel blodtryksmåling. Der udvikles instrumentering til den udleverede transducer som hardware og et software-program til kontinuerligt visning af målt blodtryk, samt til udskrivelse af løbende systoliske, diastoliske og puls værdier. Disse to dele udgør til sammen systemet.

2.5 Funktionelle krav

2.5.1 Aktør-kontekstdiagram



Figur 2.1: Aktør-kontekstdiagram

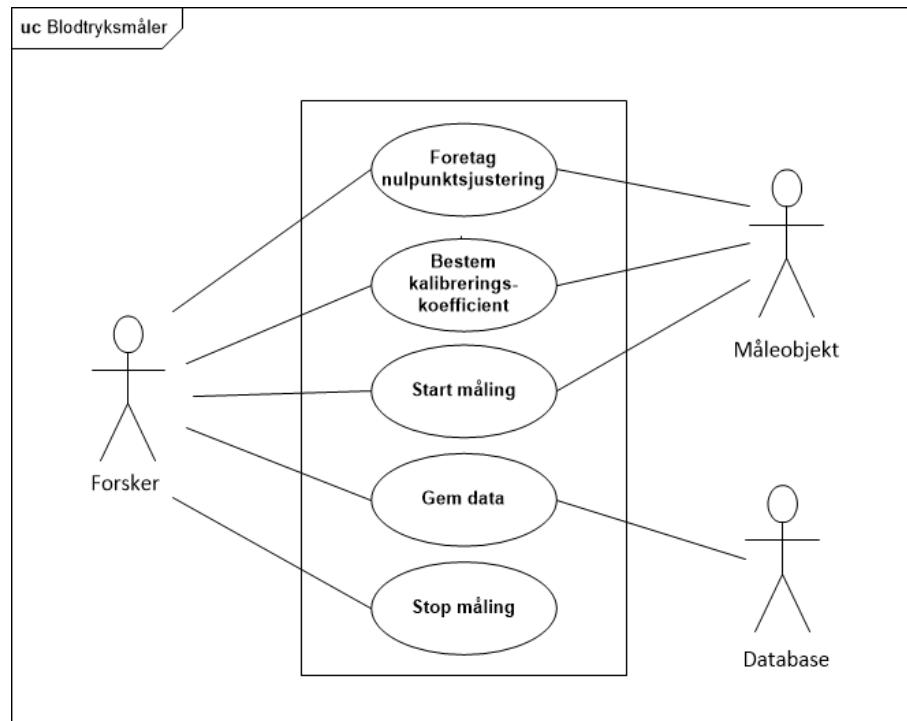
På figur 2.1 ses aktørerne til at være: Forsker, Måleobjekt og Database. Herunder er der en detaljeret beskrivelse af hver aktør.

2.5.2 Aktørbeskrivelse

Aktørnavn	Type	Beskrivelse
Forsker	Primær	Forskeren er aktøren, der starter måling, giver besked om at data ønskes gemt, navngiver målingen samt afslutter måling af blodtryk
Database	Sekundær	Databasen er aktøren, hvori måledata bliver gemt
Måleobjekt	Sekundær	Måleobjekt er aktøren, hvorfra blodtrykssignalet indhentes. Måleobjektet er tilkoblet transduceren I den endelige version er måleobjektet In Vitro maskinen, som findes i Cave Lab Under løbende test i udviklingsprocessen benyttes Analog Discovery og Waveform

Tabel 2.3: Aktørbeskrivelse

2.5.3 Use case-diagram



Figur 2.2: Use Case-diagram

Diagrammet ovenfor viser systemets fem Use Cases: Foretag nulpunktsjustering, Bestem kalibreringskoefficient, Start måling, Gem data og Stop måling. Herunder følger en nærmere beskrivelse af de enkelte Use Cases, gennem et fully-dressed Use Case skema.

Systemet består af en computer, hvor softwaren er placeret, en NI-DAQ, en Analog Discovery samt en transducer med tilhørende implementering. Systemet gør det muligt at foretage en blodtryksmåling på et måleobjekt, som er tilsluttet og sender disse signaldata ind i systemet via transduceren og det tilhørende hardware, hvor signalet vises. Det øn-

skede interval af blodtrykssignalet gemmes så i databasen.

I softwaren benyttes algoritmer til at analysere signalet, ud fra opsatte grænseværdier, så systolisk, diastolisk og puls værdier hentes frem. Disse algoritmer undersøger signalet for, hvor signalets bølgetoppe og -bunde er placeret. Da top er signalets systoliske værdi og bund er signalets diastoliske. Puls bestemmes ved at tælle antallet af blodtryksperioder pr. minut.

Brugergrænseflade er det, som forsker initierer med, altså hvorfra systemet aktiveres. Brugergrænsefladen forkortes til GUI.

2.5.4 Use Cases

Use Case 1

Scenarie	Hovedscenarie
Navn	Foretag nulpunktsjustering
Mål	At få foretaget en nulpunktsjustering
Initiering	Startes af Forsker
Aktører	Forsker (primær), Måleobjekt (sekundær)
Referencer	
Samtidige forekomster	Én nulpunktsjustering pr. kørsel
Forudsætninger	Alle systemer er ledige og operationelle
Resultat	Nulpunktsjustering er blevet fortaget efter ønske
Hovedscenarie	<ol style="list-style-type: none"> 1. Pop-up vindue for nulpunktsjustering er åbent 2. Forsker trykker på Foretag-knap: 3. Systemet fortager nulpunktsjustering og vinduet lukker ned.
Undtagelser	-

Tabel 2.4: Fully dressed Use Case 1

Use Case 2

Scenarie	Hovedscenarie
Navn	Bestem kalibreringskoefficient
Mål	At få bestemt kalibreringskoefficienten
Initiering	Startes af Forsker
Aktører	Forsker (primær)

Referencer	Ingen
Samtidige forekomster	Én kalibrering pr. måling
Forudsætninger	Alle systemer er ledige og operationelle. Væskesøjle og computer med en WaveForm er tilgængeligt
Resultat	Kalibreringskoefficient er blevet indtastet i XML-fil
Hovedscenarie	<ol style="list-style-type: none"> 1. Forsker tilslutter WaveForm og væskesøjle ved 50 mmHg til systemets hardware. 2. Output spænding fra hardware aflæses i WaveForm 3. Beregning foretages 4. Forsker indtaster beregnet kalibreringskoefficient i konfigurations XML-fil 5. Kalibreringskoefficienten kan tilgås af systemet
Undtagelser	-

Tabel 2.5: Fully dressed Use Case 2

Use Case 3

Scenarie	Hovedscenarie
Navn	Start Måling
Mål	At få foretaget en blodtryksmåling
Initiering	Startes af Forsker
Aktører	Forsker (primær), Måleobjekt (sekundær)
Referencer	Use Case 1
Samtidige forekomster	Ét signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, samt alle systemer kører og er klar til at foretage en måling
Resultat	Systolisk-, diastolisk blodtryk, puls og blodtryksgraf bliver vist på GUI
Hovedscenarie	<ol style="list-style-type: none"> 1. Forsker indtaster Forsøgsnavn 2. Filteret signal er valgt per default af systemet 3. Forsker trykker på Start-knap på GUI 4. Signal for blodtryk vises på GUI

5. Systolisk og diastolisk blodtryk samt puls bliver vist i bokse på GUI

[*Udvidelse 1:*] Forsker vælger filtreret/ufiltreret signal

Undtagelser og Udvidelser [*Udvidelse 1:*] Forsker vælger filtreret/ufiltreret signal

- a. Forsker vælger ufiltreret signal
- b. Det viste signal er nu ufiltreret
- c. Forsker vælger filtreret signal
- d. Det viste signal er nu filtreret

Tabel 2.6: Fully dressed Use Case 3

Use Case 4

Scenarie	Hovedscenarie
Navn	Gem data
Mål	At gemme data i databasen
Initiering	Startes af Forsker
Aktører	Forsker (primær), Database(sekundær)
Referencer	Use Case 1 og Use Case 3
Samtidige forekomster	Et signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, Use Case 3 kører. VPN er tilsluttet
Resultat	Signalets rådata er blevet gemt i en Database under Forsøgsnavn og et autogenereret Id

- Hovedscenarie
1. Forsker trykker på Start Gem-knap
 2. Systemet gemmer det fremadrettede signals rådata i Databasen
 3. Forsker trykker på Stop Gem-knap for at stoppe med at gemme

[*Undtagelse 1:*] Forsker trykker på Stop Måling-knap

4. Det vises at data er gemt ved at filnavnet(Forsøgsnavn og Id) for målingen vises på GUI.

Undtagelser	[<i>Undtagelse 1:</i>] Forsker trykker på Stop Måling-knap
	a. Systemet gemmer ikke målingen og blodtryksgrafen fastholdes.

*Tabel 2.7: Fully dressed Use Case 4***Use Case 5**

Scenarie	Hovedscenarie
Navn	Stop måling
Mål	At stoppe målingen af blodtryk
Initiering	Startes af Forsker
Aktører	Forsker (primær)
Referencer	Use Case 1 og 3
Samtidige forekomster	Ét signal pr. måling
Forudsætninger	Use Case 1 er kørt succesfuldt, Use Case 3 kører
Resultat	Måling af blevet stoppet

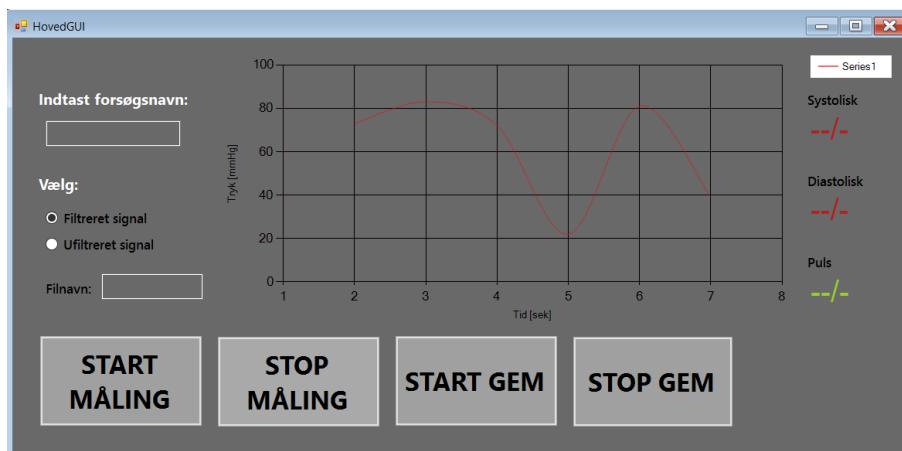
- | | |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hovedscenarie | <ol style="list-style-type: none"> 1. Forsker trykker på Stop Måling-knap 2. Målingen stopper og blodtryksgrafen fastholdes. |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|

Undtagelser	-
-------------	---

*Tabel 2.8: Fully dressed Use Case 5***2.6 Ikke-funktionelle krav****2.6.1 (F)URPS+****Functionality**

1. Blodtryksmåleren skal indeholde en Start Måling-knap til at igangsætte målingerne.
2. Blodtryksmåleren skal indeholde en Stop Måling-knap, hvorfra måling kan stoppes.
3. Blodtryksmåleren skal indeholde en Start Gem-knap til påbegyndelses af at gemme måling i Database.

4. Blodtryksmåleren skal indeholde en Stop Gem-knap til afslutning af at gemme måling i Database.
5. Blodtryksmåleren skal indeholde en tekstboks til forsøgsnavn, hvori forsker indtaster det pågældende forsøgsnavn.
6. Blodtryksmåleren skal indeholde radiobutton til filtreret signal, denne skal være default valget.
7. Blodtryksmåleren skal indeholde radiobutton til ufiltreret signal.
8. Blodtryksmåleren skal indeholde tekstbokse til puls, systolisk og diastolisk blodtryk som vises med op til tre cifre.
9. Blodtryksmåleren skal indeholde en tekstboks som viser filnavn(forsøgsnavn og id) på malingen, efter maling er gemt.
10. GUI'en skal se ud som vist på figur 2.3:



Figur 2.3: Skitse af GUI

Usability

1. Forskeren skal kunne starte en default-måling maksimalt 30 sekunder efter systemet er startet.

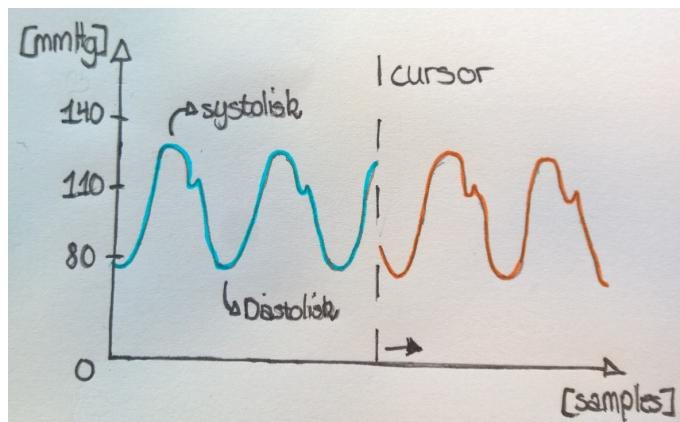
Reliability

1. Det skal maksimalt tage 5 timer at gendanne systemet (MTTR - Mean Time To Restore)
2. Systemet skal have en oppeid uden nedbrud på minimum 1 måned (720 timer) (MTBF - Mean Time Between Failure).
3. Systemet skal have en oppeid/køretid på:

$$\text{Availability} = \frac{MTBF}{MTBF + MTTR} \cdot 100 = \frac{720}{720 + 5} \cdot 100 = 99,31\% \quad (2.1)$$

Performance

1. Blodtryksmåleren skal, indenfor 3 sekunder, kunne vise systolisk og diastolisk blodtryk via graf. Dette accepteres med en tolerance på +/- 15 %.
2. Blodtryksmåleren skal, indenfor 5 sekunder fra der er trykket på Stop Gem-knap, have gemt målingerne i Databasen. Dette accepteres med en tolerance på +/- 15 %.
3. Grafen vises i ét vindue, hvor y-aksen måles i mmHg (millimeter kviksølv) og x-aksen i tid i sekunder.
4. Hvert 3. sekund skal værdier for systolisk og diastolisk blodtryk samt puls opdateres. Dette accepteres med en tolerance på +/- 15 %.
5. Graf for blodtryk skal køre kontinuerligt i GUI efter følgende princip (figur 2.4), hvor det blå signal erstatter det orange signal ved, at den seneste måling altid sættes ved cursorens placering.



Figur 2.4: Graf for blodtryk

6. Når der trykkes på Stop Gem-knap gemmes signals rådata under det indtastede forsøgsnavn og et autogenereret id. "forsøgsnavn_id".
7. Systemet skal kunne måle blodtrykværdier fra 0 til 250 mmHg.

Supportability

1. Forskeren skal kunne udskifte batterierne til hardwaren på 2 minutter.
2. Softwaren skal opbygges med lav kobling.

Systemarkitektur 3

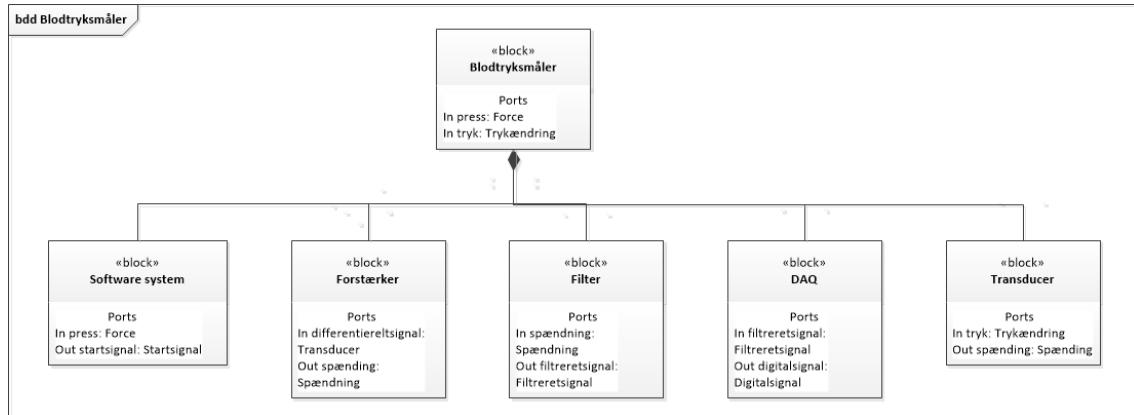
Version	Dato	Ansvarlig	Beskrivelse
0.1	03-11-2015	MBA	Oprettelse
0.2	10-11-2015	DHC, MBA	HW Start af skrivning, indsætning af billeder
0.3	10-11-2015	ABH	SW Start på design, indsætning af diagrammer
0.4	11-11-2015	DHC	HW Design Forstrækning
0.5	13-11-2015	ABH	SW Design klasse- og metodeidentifikation
0.6	18-11-2015	ABH	HW Rettelse af diagrammer
0.7	18-11-2015	DHC, AJF	HW Implementering Forstrækning, Modultest Lavpas
0.8	18-11-2015	MHNK, JMM	SW Design, Rettelse af domænemodel
0.9	18-11-2015	ABH	SW Design, Mere metodeidentifikation
1.0	20-11-2015	MHNK	SW Indskrivning af alle sekvensdiagrammer
1.1	26-11-2015	DHC	HW Modultest, Kalibrering ved vandsøjle
1.2	26-11-2015	DHC, AJF	HW Design Lavpas
1.3	02-12-2015	DHC	HW Referencer
1.4	02-12-2015	MHNK	HW Rettelser i tekst
1.5	02-12-2015	DHC, MBA	HW Modultest
1.6	04-12-2015	ABH	SW Implementering, Generelt, Analyse og Digitalt filter
1.7	06-12-2015	ABH	SW Implementering, Kalibrering og nulpunktsjustering
1.8	09-12-2015	DHC	Rettelser i tekst
1.9	09-12-2015	ABH, JMM	SW Implementering Observer-Strategy, Analyse og Digital Filter

I det følgende beskrives arkitekturen for systemet. Systemarkitekturen er vores udviklingsramme for den videreudvikling af design og implementering af blodtrykssystemet. Designet af systemet er grebet an således at, der først kigges på det overordnede system, hvorefter systemet arbejdes ned i mindre brudstykker. Dette gøres ved at benytte diagrammer med tilhørende beskrivelser.

3.1 Hardware

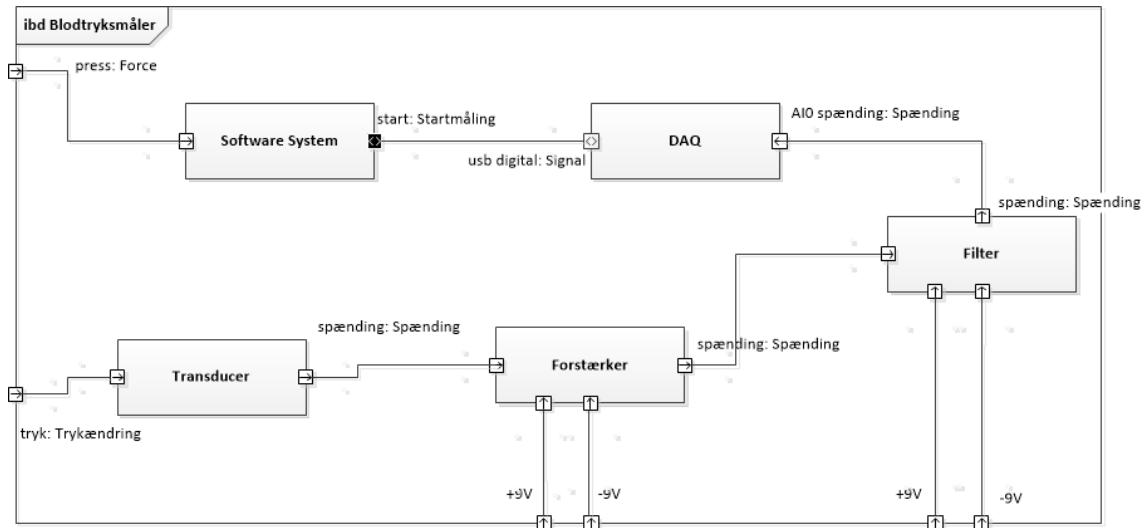
3.1.1 Design

Systemets hardware kan illustreres i et BBD. Det ses på figur 3.1 at systemet består af fem hardware blokke: software system, forstærker, filter, DAQ og transducer. Disse fem blokke udgør til sammen selve blodtryksmåleren.



Figur 3.1: Block Definition Diagram for hardware

Ovenstående BDD-diagram fører videre til udarbejdelsen af IBD for hardware komponenterne. I IBD diagrammet vises koblingen mellem de forskellige blokke gennem port forbindelser. Det ses at signalet starter ved transduceren, hvorefter det bliver behandlet gennem forstærker, filter og DAQ. Til sidst sendes det ind i software systemet, som bliver påvirket af tryk på knapper på GUI.



Figur 3.2: Internal Block Diagram for hardware

Forstærkning

Transduceren måler en trykændring som den omsætter til en spænding. Dette er udtrykt ved et differentieret signal, som sendes ind i forstærker-blokken. Da signalet fra transduceren er en lav spænding, skal det forstærkes op, for at passe med DAQ'ens

input. Denne forstærkning udregnes ud fra det maksimale output fra transduceren og det maksimale input til DAQ'en. Se beregningerne under Implementering.

Under simulering bruges Analog Discovery som en funktionsgenerator, der simulere det differentieret signal. Analog Discovery har en usikkerhed, når der arbejdes med små spændinger. Dette kan modarbejdes vha. spændingsdeler princippet. Dette gør at Analog Discovery kan sende en højere spænding ind i systemet, så usikkerheden mindskes. Dette bruges kun under simulering og teste af hardwaren.

Lavpas

I projektet skal der laves et 2. ordens lavpasfilter. Filteret skal laves for at sikre, at der ikke opstår aliasering.

Aliasering [1] er, hvor signalet bliver gentaget. Når man har signalet i det digitale domæne, bliver spektret for signalet en periodisk funktion. Det vil sige, at den gentager sig selv, efter et bestemt stykke tid.

Det skal sikres, at der ikke kommer overlap mellem signalet og et alias. Da det ellers kunne give anledning til misforståelser. Derfor laves et lavpasfilter, som sikre at der ikke ligger noget signal ved den halve samplingsfrekvens. Signalet her kan med fordel gøres så lille at DAQ'en ikke kan læse det, dvs. signalet skal være mindre end $1/2 \cdot LSB$ (Least Significant Bit).

Lavpasfilteret skal være et Sallen-Key Butterworth-filter med en knækfrekvens på 50 Hz og en samplingsfrekvens på 1kHz. Ud fra oplysninger givet til projektet, vides det at filteret skal dæmpe signalet med 20 dB, under antagelse af at, den forekommende støj er mindre end signalet, også når støjen forekommer over knækfrekvensen.

Ved en typisk blodtryksmåling forekommer der ikke signal over 50 Hz, samtidigt er signalet her aftaget med ca. 70 dB. For at få signalet, ved den halve samplingsfrekvens til at være $1/2 \cdot LSB$, skal det ydeligere dæmpes 20 dB. Derfor oplyses filterets til at være 50 Hz, da dette giver en minimums dæmpning på 20 dB pr. dekade.

3.1.2 Implementering

Forstærkning

For at få den rette forstærkning er det blevet valgt, at benytte instrumentationsforstærkeren INA-114. Her kan transduceren sættes på med det differentierede signal. INA114 er valgt da følgende gælder[2] for instrumentationsforstærkere:

- Differentielt input - single ended output
- Gain justering med ændring af kun én modstand
- Meget høj indgangsimpedans
- Stor Common Mode Rejection Ratio(CMRR)

Under opbygning og modultestning vil det differentierede signal blive simuleret af Analog Discovery.

For at udregne den korrekte forstærkning, bruges følsomheden fra transduceren og eksistationsspændingen. Først udregnes det maksimale output fra transduceren:

$$9V \cdot 250mmHg \cdot 5\mu \cdot 10^{-5}uV/V/mmHg = 11.25mV \quad (3.1)$$

Da det er besluttet at det maksimale input til DAQ'en [1] er 5V, kan forstærkningen (Gain) nu udregnes:

$$\begin{aligned} 5V &= 11.25mV \cdot G \\ G &= 444.44 \end{aligned} \quad (3.2)$$

[3] For at få den rette forstærkning udregnes den eksterne modstand (R_g) til INA114. INA114's forstærkning afhænger af størrelsen på R_g , hvis modstanden er stor, er forstærkningen lille og omvendt. R_g udregnes ved formlen:

$$\begin{aligned} G &= 1 + \frac{50k\Omega}{R_g} \\ 444.44 &= 1 + \frac{50k\Omega}{R_g} \Rightarrow R_g = 112.75\Omega \end{aligned} \quad (3.3)$$

Derved fås en værdi for den eksterne modstand til INA114, som skaber den ønskede forstærkning.

Det skal nu sikres at dette kan lade sig gøre. Derfor sikres det, at den ønskede forstærkning kan ske ved båndbredden. Dette kan undersøges da produktet af forstærkning og båndbredde er en konstant. Konstanten aflæses i databladet for INA114[3].

$$\begin{aligned} 1000000Hz &= G \cdot BW \\ BW &= 2250Hz \end{aligned} \quad (3.4)$$

Da båndbredden ligger over knækfrekvensen for lavpas filtret, er dette godkendt. Hvis båndbredde havde ligget under knækfrekvensen vil operationsforstærkeren ikke have kunnet arbejde med de ønskede frekvenserne. Derfor er det vigtigt at båndbredden er bred nok til at kunne indeholde frekvenser fra begge side af knækfrekvensen.

For at imødekommme usikkerheden ved Analog Discovery med lave spændinger, laves et kredsløb efter spændingsdelerprincippet. Signalerne fra Analog Discovery skal sendes igen nem dette kredsløb, hvor de efter spændingsdelerprincippet gøres mindre. I kredsløbet benyttes to modstande, hvis værdier er $R_1 = 100k\Omega$ og $R_2 = 1k\Omega$. Da vi kender signalet som skal ind i INA114 og modstandene i kredsløbet, kan størrelsen af den spænding, som skal sendes fra Analog Discovery, findes:

$$\begin{aligned} U_{INA} &= U_{analog} \cdot \frac{R_2}{R_1 + R_2} \\ 11.25mV &= U_{analog} \cdot \frac{1k\Omega}{100k\Omega + 1k\Omega} \Rightarrow U_{analog} = 1.1362V \end{aligned} \quad (3.5)$$

Derved kan Analog Discovery sende signaler med en højere spænding ud og usikkerheden mindskes. Der er taget højde for at, hvis modstandene i kredsløbet bliver for store, vil det skabe en termisk usikkerhed. Derfor er modstandene valgt som de er. Dette er kun under simulering, når transduceren benyttes, bruges spændingsdeleren ikke.

Lavpas

For at opnå den ønskede effekt i lavpasfilteret, blev det oplyst at $f_c = 50$ Hz, $f_s = 1\text{kHz}$, $R_1 = R_2$ og $C_2 = 680\text{nF}$. Ud fra disse værdier, udregnes de resterende komponentværdier for filteret.

Overføringsfunktionen for et 2. ordens filter er:

$$H(z) = \frac{\omega_n^2}{(s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2)} \quad (3.6)$$

For at finde overføringsfunktionen for det gældende system, vides det at følge ligninger gælder [4]:

$$\begin{aligned} \omega_n &= 2 \cdot \pi \cdot 50 = \frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \\ 2 \cdot \zeta \cdot \omega_n &= \frac{1}{C_2} \cdot \left(\frac{R_1 + R_2}{R_1 \cdot R_2} \right) \end{aligned} \quad (3.7)$$

Derved fås en overføringsfunktion som hedder:

$$H(z) = \frac{\left(\frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \right)^2}{s^2 + \left(\frac{1}{C_2} \cdot \left(\frac{R_1 + R_2}{R_1 \cdot R_2} \right) \cdot s \right) + \left(\frac{1}{\sqrt{R_1 \cdot R_2 \cdot C_1 \cdot C_2}} \right)^2} \quad (3.8)$$

Da det bliver oplyst at $R_1 = R_2$, kan funktionen reduceres. Den kan samtidig simplificeres. I sidste ende fås overføringsfunktionen, se Beregninger til overføringsfunktion under Bilag for nærmere udregninger:

$$H(z) = \frac{\frac{1}{C_1 \cdot C_2 \cdot R^2}}{s^2 + s \cdot \frac{2}{R \cdot C_2} + \frac{1}{C_1 \cdot C_2 \cdot R^2}} \quad (3.9)$$

Da der arbejdes med et 2. ordens Butterworth filter, vides det at udsvinget ζ skal være 0.7 [5]. Den sidste overføringsfunktion sammenlignes med den generelle for 2. ordens systemer. Det gælder at $C_2 = 680 \cdot 10^{-9}\text{nF}$. Det er muligt at isolerer forskellige led. Først isoleres for modstanden:

$$\begin{aligned} \frac{2}{R \cdot C_2} &= 2 \cdot \zeta \cdot \omega_n \\ \frac{2}{R \cdot 680 \cdot 10^{-9}} &= 2 \cdot 0.7 \cdot (2 \cdot \pi \cdot 50) \\ &\Downarrow \\ R &= 6687\Omega \end{aligned} \quad (3.10)$$

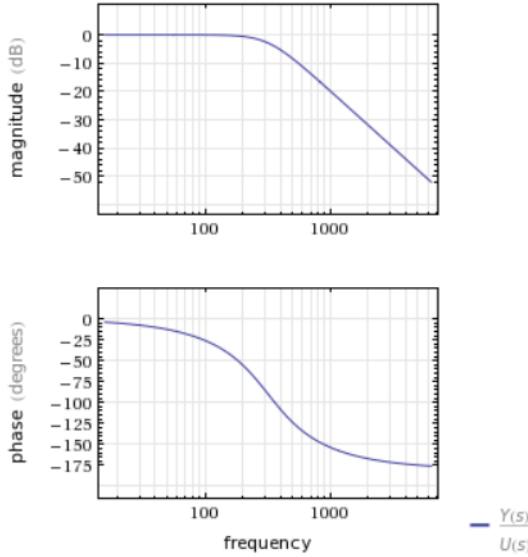
Derved er modstandene udregnet til $R = 6687\Omega$. Nu kan der isoleres for kondensator C_1 :

$$\begin{aligned} \frac{1}{C_1 \cdot C_2 \cdot R^2} &= \omega^2 \\ \frac{1}{C_1 \cdot 680 \cdot 10^{-9} \cdot 6687^2} &= (2 \cdot \pi \cdot 50)^2 \\ &\Downarrow \\ C_1 &= 333 \cdot 10^{-9}\text{nF} \end{aligned} \quad (3.11)$$

Dette betyder, at $C1 = 333 \cdot 10^{-9}nF$ og $C2 = 680 \cdot 10^{-9}nF$. Derved er alle komponentværdierne til lavpasfilteret fundet og det kan nu realiseres.

Under udviklingen af lavpasfilteret er komponent størrelserne, blevet ændret for at kunne realisere det. De brugte komponent størrelser er: $R = 6.6k\Omega$, $C1 = 330 \cdot 10^{-9}nF$ og $C2 = 680 \cdot 10^{-9}nF$. For at være sikker på at filteret har de ønskede karakteristika, laves et bodeplot for den endelig overføringsfunktion:

$$H(z) = \frac{62500000000}{610929 \cdot \left(s^2 + \frac{250000}{561} \cdot s + \frac{62500000000}{610929} \right)} \quad (3.12)$$

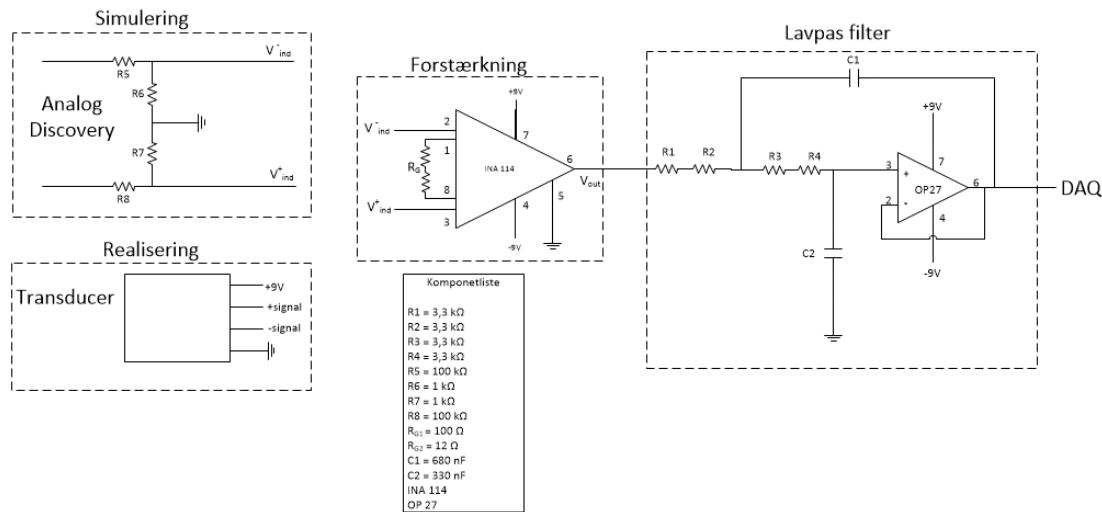


Figur 3.3: Bodeplot

Udregning af den præcise oversving ζ ud fra de benyttet komponentværdier:

$$\begin{aligned} \frac{2}{R \cdot C1} &= 2 \cdot \zeta \cdot \omega_n \\ \frac{2}{6600 \cdot 680 \cdot 10^{-9}} &= 2 \cdot \zeta \cdot (2 \cdot \pi \cdot 50) \\ &\downarrow \\ \zeta &= 0.709 \end{aligned} \quad (3.13)$$

Dvs. de små ændringer i komponent værdierne ikke har haft betydende indflydelse på værdien for ζ .



Figur 3.4: Diagram over HW

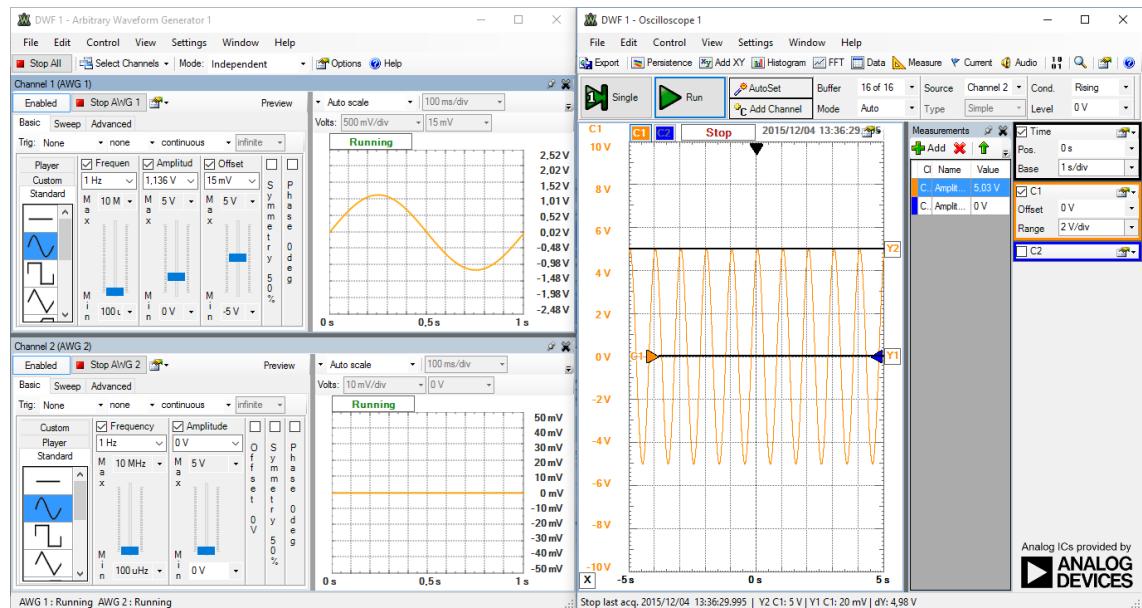
På figur 3.4 ses et diagram over, hvordan kredsløbet er bygget op. Her ses kredsløbet for realiseringen med transduceren og for simuleringen med Analog Discovery.

3.1.3 Modultest

Forstærkning

For at teste forstærkningen sendes et differentieret signal ind vha. Analog Discovery. Signalet måles ved udgangen og der ses på, hvor meget signalet er blevet forstærket.

På figur 3.5 ses det signal, som sendes ind i forstærknings blokken og det, der måles på udgangen af blokken.



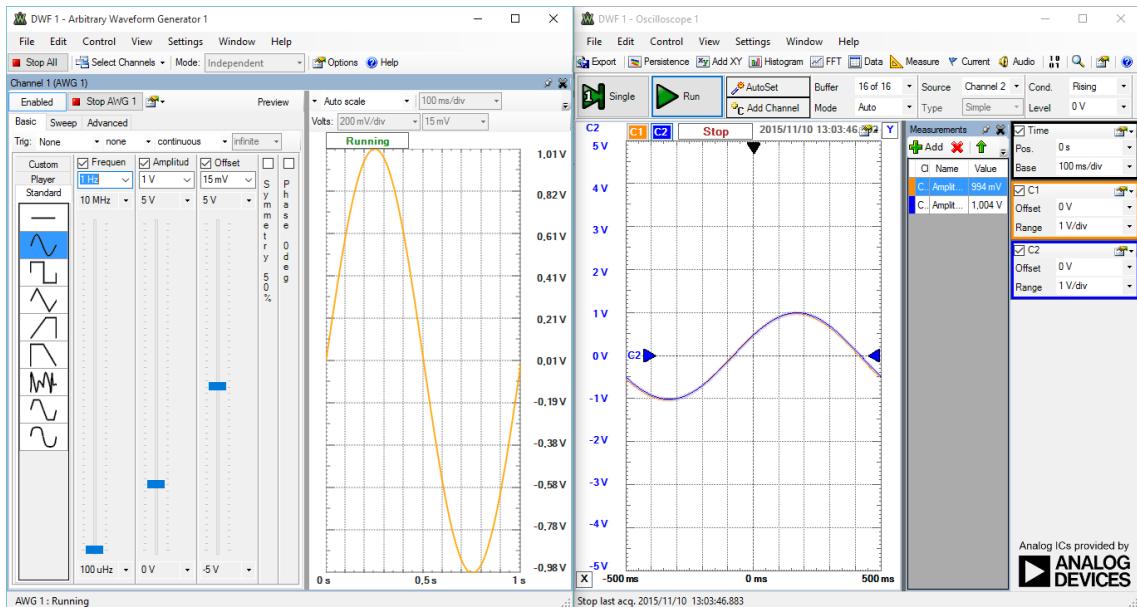
Figur 3.5: Forstærknings blok

På udgangen, ses det at signalet er blevet forstærket op til 5 V DC. Herved er maks. output

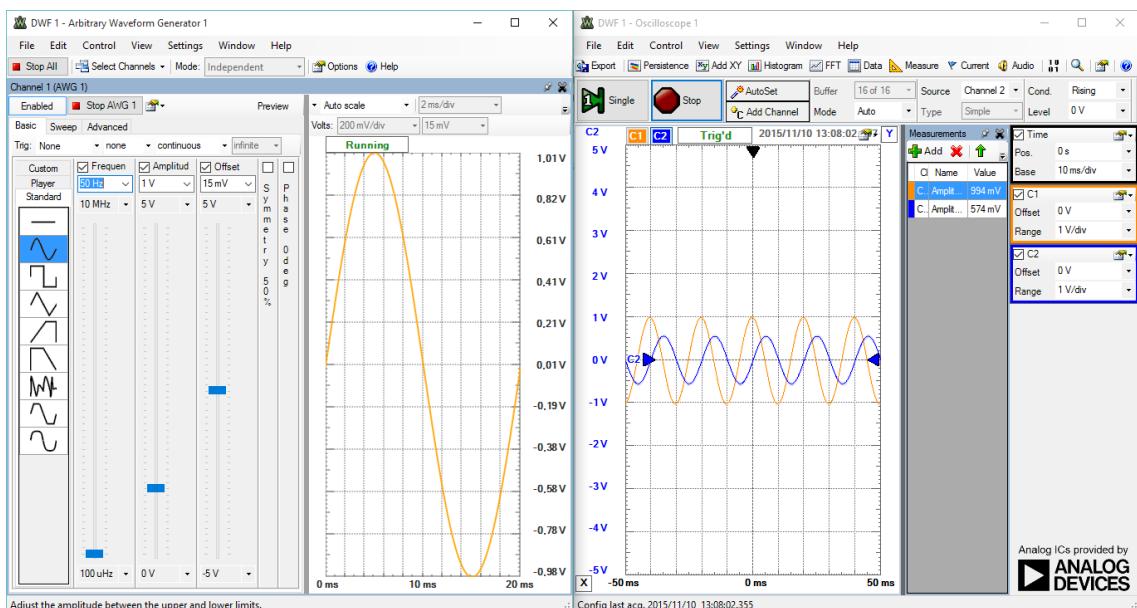
fra transduceren blevet forstærket så det passer med maks. input til DAQ'en. Signalet bliver ikke ændret på andre måde i forstærker blokken.

Lavpas

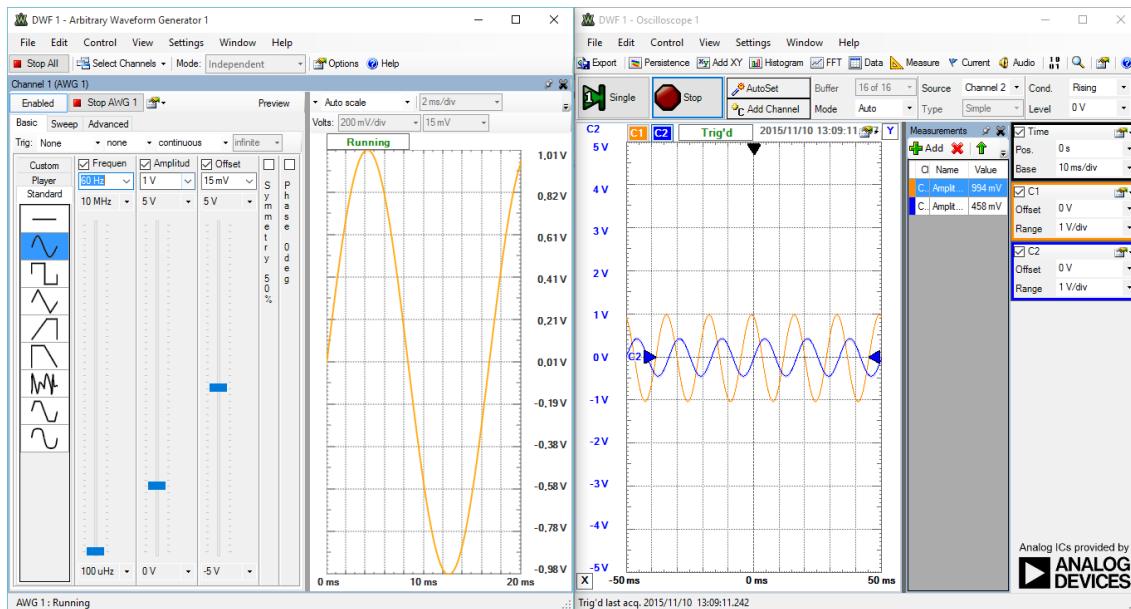
For at teste lavpasfilteret foretages målinger med en sinus, hvor frekvensen varierer for hver måling. Fasen aflæses mellem indgang- og udgangssignal. Amplituden aflæses ligeledes for hver måling. Ved knækfrekvensen skal fasedrejningen være 90° . Dette kan aflæses på figur 3.7. Efter knækfrekvensen skal amplituden gå mod nul. Ved målingen for 60 Hz figur 3.8, kan det ses hvordan amplituden er faldet drastisk efter knækfrekvensen.



Figur 3.6: Måling for 10 Hz



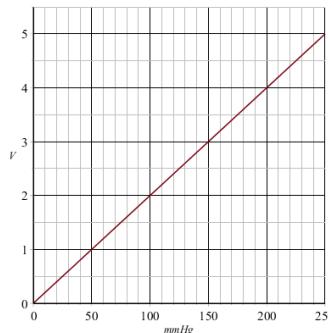
Figur 3.7: Måling for 50 Hz



Figur 3.8: Måling for 60 Hz

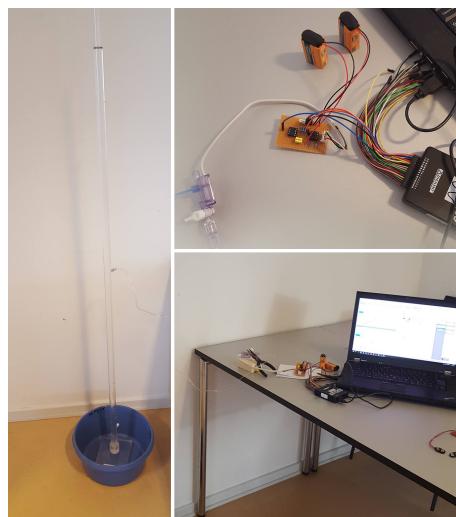
Kalibrering med vandsøjle

Efter forstærkning og lavpasfilteret er blevet testet hver for sig, udføres en kalibrering af systemet vha. en udleveret vandsøjle med tre målepunkter, hvor det er angivet, hvor højt trykket(mmHg) er ved hvert af disse punkter. Derved kan det testes om hardware-delen mäter den rigtige spænding i forhold til millimeter kviksølv(mmHg). Ud fra det maksimale antal volt (V) spænding og millimeter kviksølv(mmHg) kan det udregnes, hvad hardware skal vise ved 100 mmHg.



Figur 3.9: Graf til kalibrering, fra udregninger

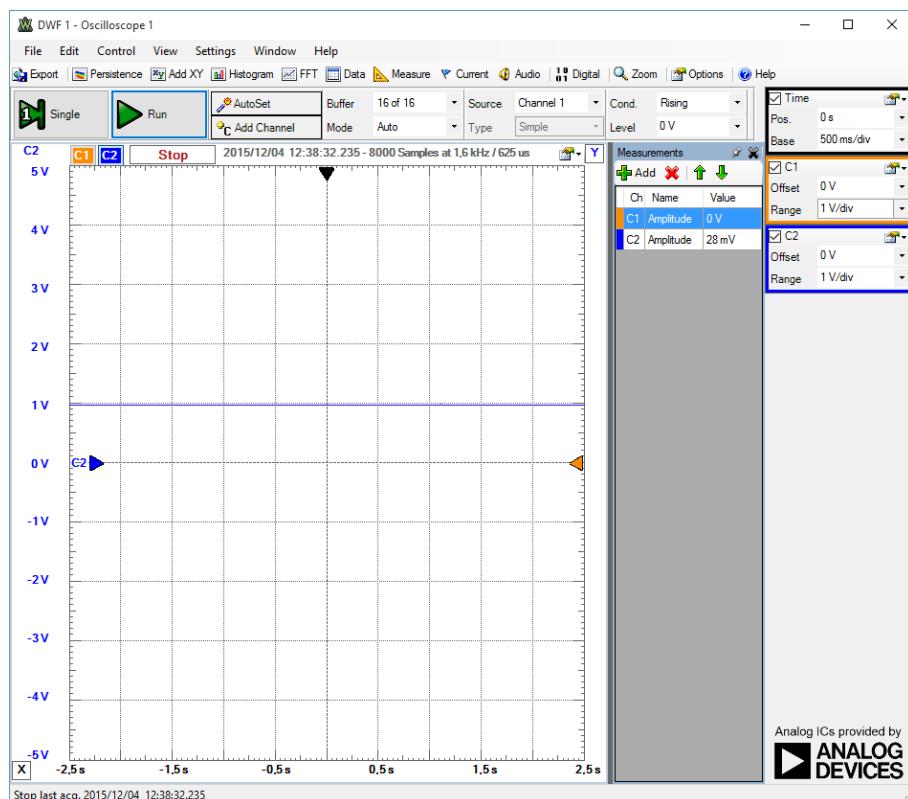
Testen udføres ved, at fylde vand i søjen til et bestemt punkt. Transduceren skal være tilkoblet et af de tre målepunkter, mens de andre er lukket til. Transduceren er sat til forstærkningen, der hvor Analog Discovery tidligere har været sat til. Transduceren er tilkoblet 0-9V, ved batterierne. På samme måde som ved simuleringen aflæses målingen på computeren ved hjælp af programmet WaveForms. Da det vides, hvilken trykændring der måles på, ved vi fra grafen til kalibreringen, hvilken spænding den skal vise. Dette fortages for de tre målepunkter på vandsøjlen, hvor hver måling sammenlignes med den udregnede graf. For hver måling, skal transduceren flyttes til et af de andre målepunkter.



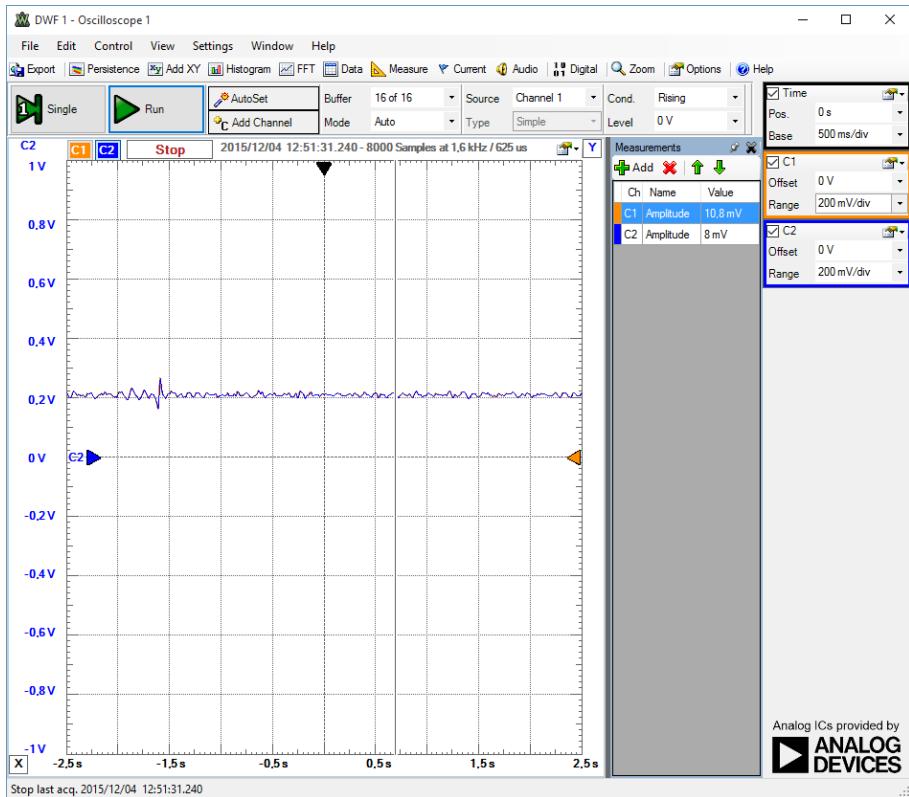
Figur 3.10: Opstilling

Opstillingen er gjort klar og der hentes ekstra vand under testen. Vandet skal bruges til at fyldе vandsjølen på til de forskellige målinger.

Ud fra grafen i figur 3.9 vides, hvad svaret på hver måling skal være. På figur 3.11 ses målingen, da transduceren var tilkoblet målepunktet for 50 mmHg. Ud fra figur 3.9 vides det at målingen skal vise 1V DC.

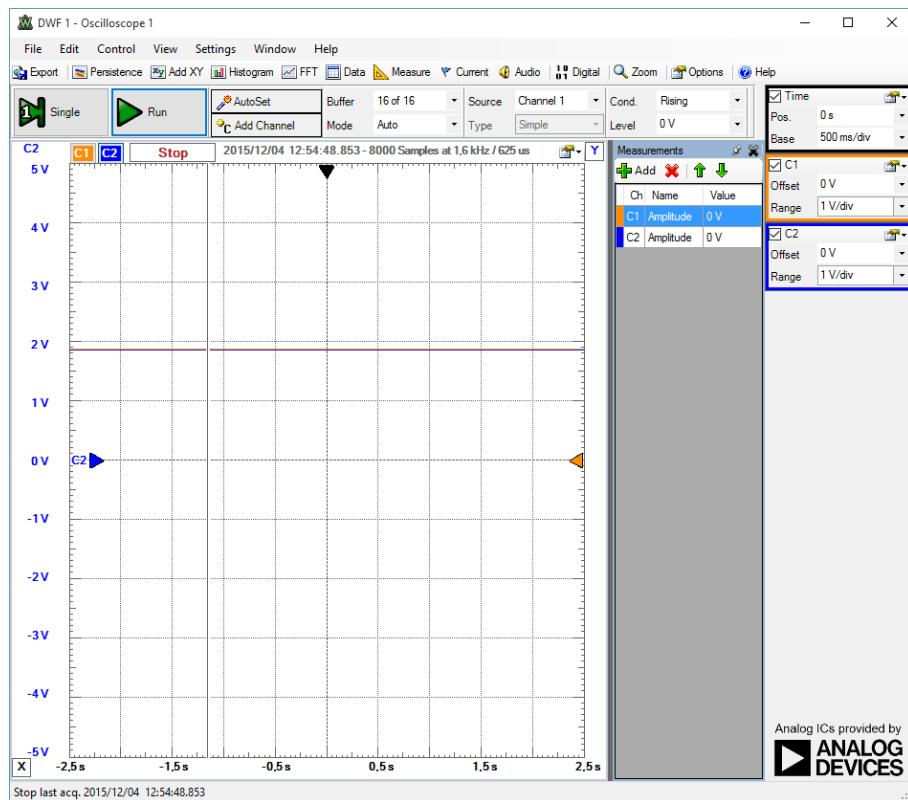


Figur 3.11: Måling ved 50 mmHg



Figur 3.12: Måling ved 10mmHg

På målingen for 10 mmHg ses en del rystelser(udsving på signalet). Som det ses på figur 3.12 ligger signalet ikke præcist på 0.2V, dette kan skyldes at under testen, skal transduceren være i højde med målepunktet. Pga. korte ledninger, blev det under testen derfor nødvendigt at løfte og holde transducer, VEVO Board og Analog Discovery i højde med målepunktet.



Figur 3.13: Måling ved 100mmHg

Ved målingen for 100mmHg skulle der måles en spænding på 2V. Som det ses på figur 3.13 ligger den ikke præcis på 2V. Som under målingen for 10 mmHg skal transduceren være i samme højde som målepunktet. Her er målepunktet lavt, men det skaber stadig en del usikkerhed.

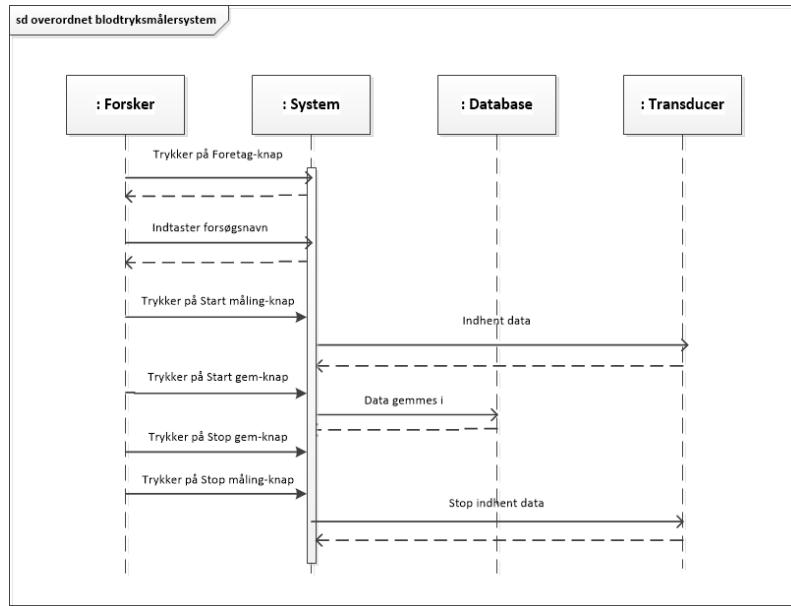
3.2 Software

3.2.1 Design

I dette beskrives systemets softwaredesign på baggrund af systembeskrivelsen og kravspecifikationen. De overvejelser som er gjort i forbindelse med design af software vil blive præsenteret i dette afsnit.

Overordnet sekvensdiagram

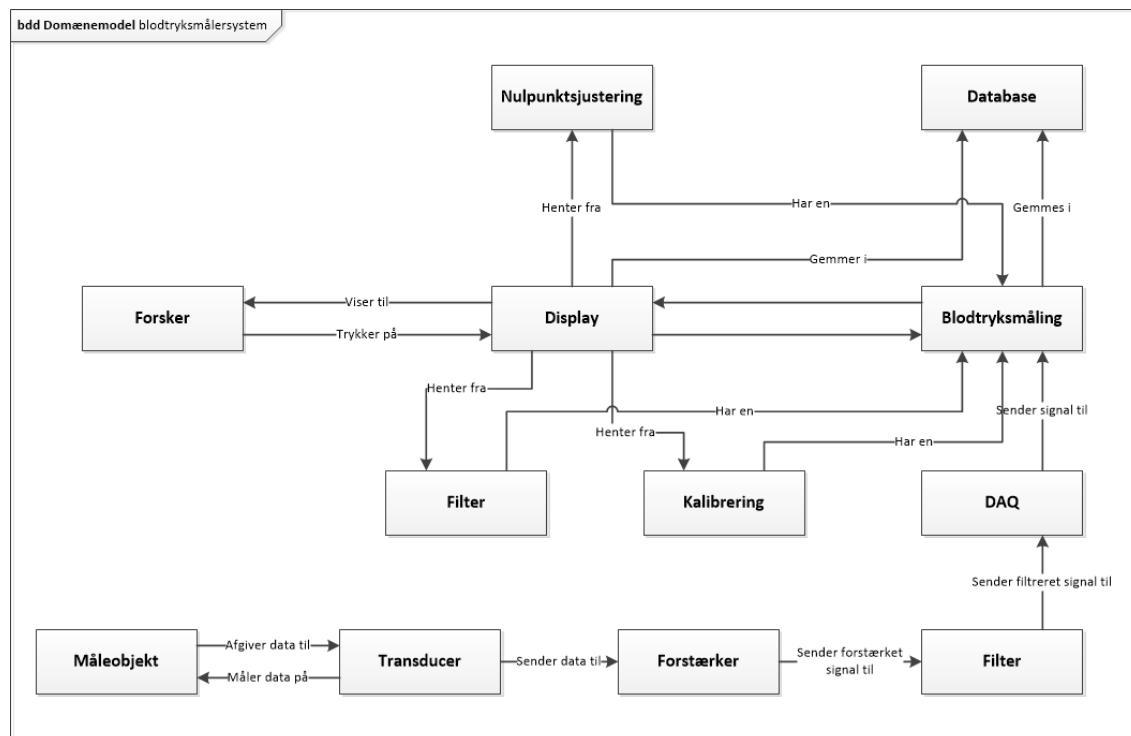
Overordnet set ønskes det at udvikle et system, der kan interagerer med en forsker. Diagrammet herunder viser at forskerens opgave består i at starte, tage stilling til nulpunktsjustering og kalibrering samt gemme de ønskede data. Diagrammet er en simpel illustration som viser systemets adfærd gennem alle fem Use Cases. Formålet med dette diagram er udelukkende at skabe et overblik over det samlede system.



Figur 3.14: Overordnet sekvensdiagram for systemet

Problemidentifikation

Første step i software designet er at klarlægge hvilke klasser systemet skal bestå af. Til dette er en domænemodel derfor udarbejdet med udgangspunkt i de fem Use Cases. I de fem Use Cases er de konceptuelle klasser blevet identificeret, og derefter indført som klasser i nedestående domænemodel. Modellen har til formål at vise hvilke dele systemet skal holde styr på.

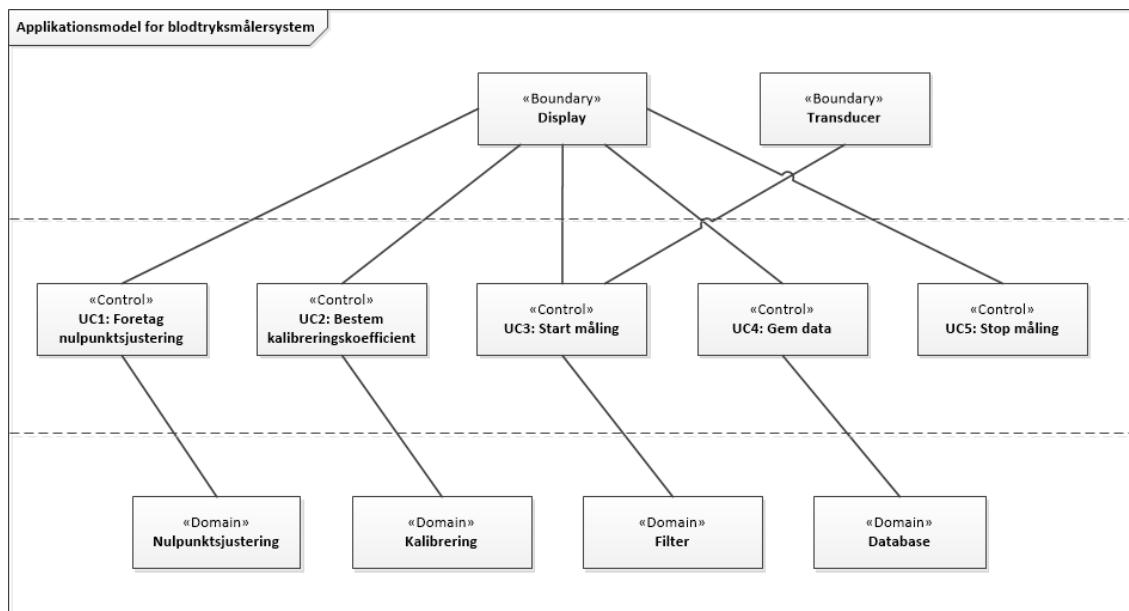


Figur 3.15: Domænemodel

Diagrammet viser tydeligt forskerens interaktion med display, samt hvilke handlinger denne interaktion starter i system. Hardware-komponenterne er medtaget for at vise signalets vej fra måleobjekt til system.

Klasseidentifikation

Ud fra domænemodellen kan et klassediagram udarbejdes, således tager dette diagram også udgangspunkt i de fem Use Cases. Hensigten med et klassediagram er at klarlægge hver klasses individuelle formål.



Figur 3.16: Applikationsmodel for software

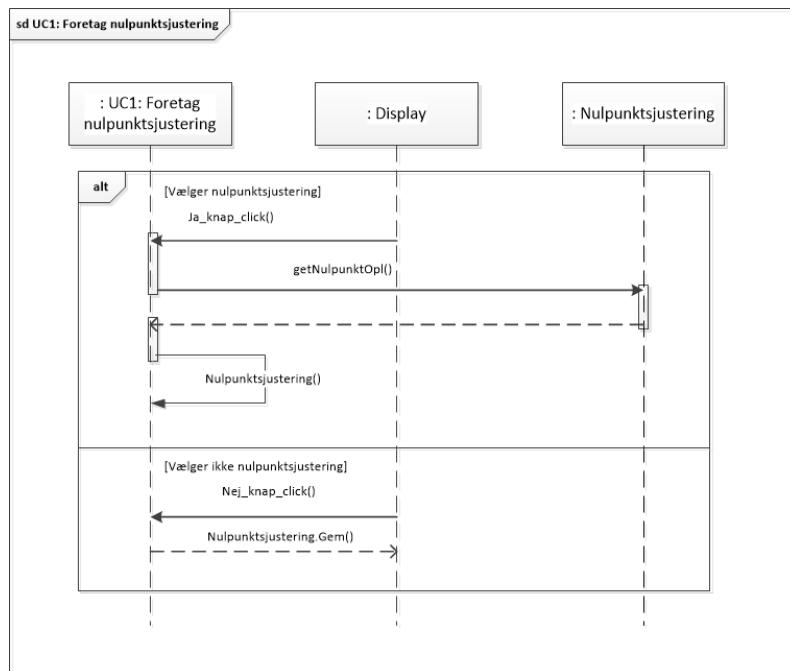
Dermed ses det at denne model er delt op i tre niveauer:

1. Grænsefladeklasse
 - a) Transducer - Indhentet data fra måleobjekt
 - b) Display - Brugergrænseflade til forsker
2. Kontrolklasse
 - a) UC1: Foretag nulpunktsjustering
 - b) UC2: Bestem kalibreringskoefficient
 - c) UC3: Start måling
 - d) UC4: Gem data
 - e) UC5: Stop måling
3. Domænekklasse
 - a) Database

Metodeidentifikation

Klasserne i ovenstående klassediagram er med til at definere, hvilke blokke de følgende sekvensdiagrammer må indeholde. Det er yderst vigtigt at der er en sammenhæng mellem klasserne i klassediagrammet og blokkene i sekvensdiagrammet. Vi har valgt at udarbejde et sekvensdiagram for hver enkelt Use Case, hvori systemets interne kommunikation beskrives, når normalforløb og udvidelser gennemløbes. I alle diagrammerne beskrives forløbet via de metodekald, der er nødvendige for at få de ønskede handlinger mellem blokkene udført.

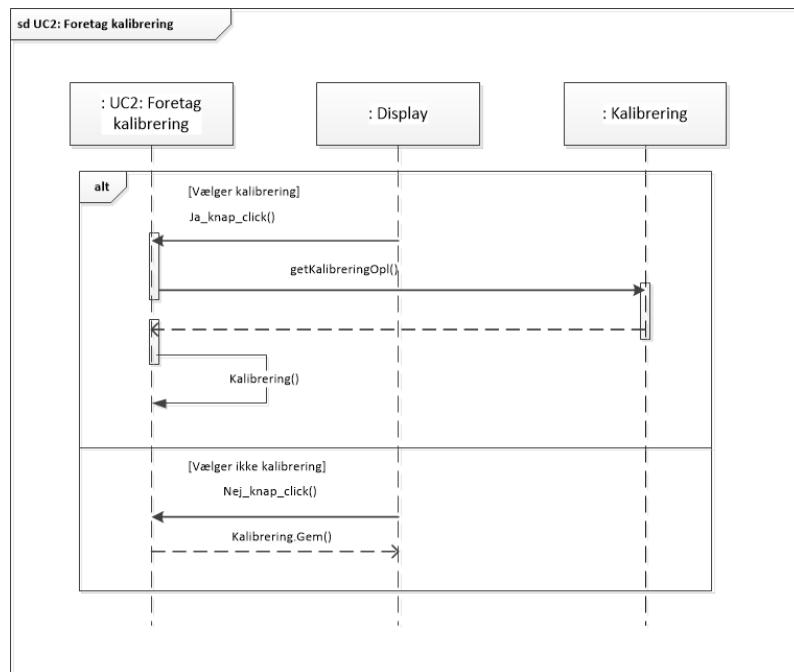
Use Case 1



Figur 3.17: Sekvensdiagram for Use Case 1

Det ses af ovenstående sekvensdiagram at forsøker interagerer med display. Her er der to mulige udfald "Vælger nulpunktsjustering" og "Vælger ikke nulpunktsjustering", disse implementeres som to muligheder forsøker kan vælge imellem.

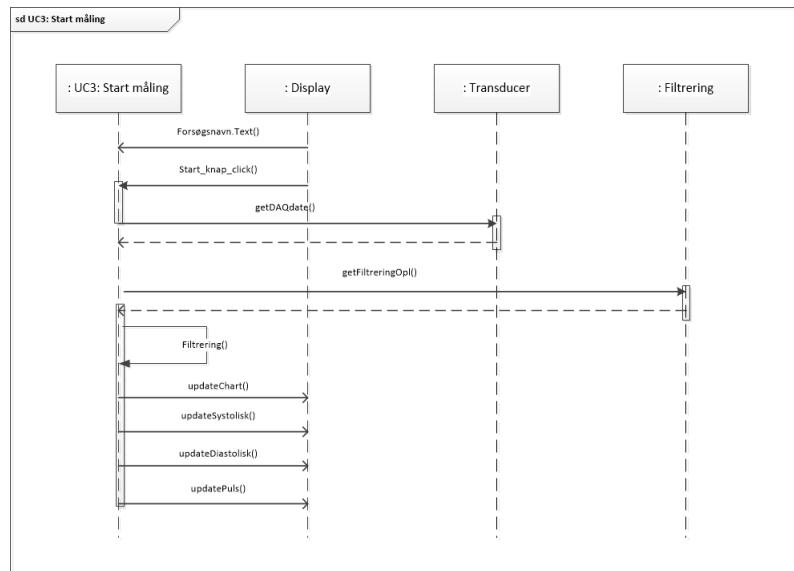
Use Case 2



Figur 3.18: Sekvensdiagram for Use Case 2

Diagrammet ovenfor viser at forsker interagerer med display, hvor der ved tryk enten vælges ja eller nej til kalibrering. Afhængig af valg foretager systemet de nødvendige kald.

Use Case 3

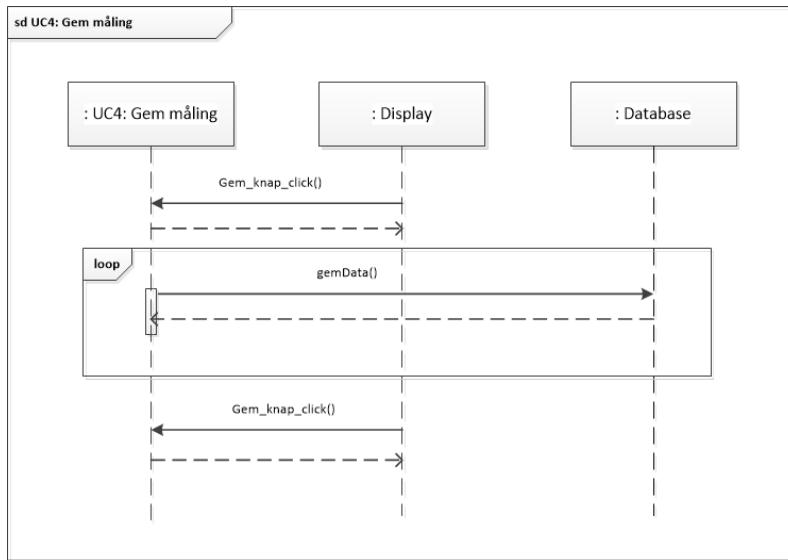


Figur 3.19: Sekvensdiagram for Use Case 3

Ved Use Case 3 - Start måling ses det at display, transducer og filtreringsklassen vil komme i spil. Her modtages besked ved indtastning af forsøgsnavn og tryk på start-knap på display om, at signaldata fra transduceren skal hentes ind i systemet. Herefter foretages filtrering af signalet, samt visning af signal i graf, systoliske-, diastoliske og puls-værdier på display. Use Casen indeholder en undtagelse hvor filtrering af signal ikke ønskes foretaget, denne

er ikke medtaget i sekvensdiagrammet.

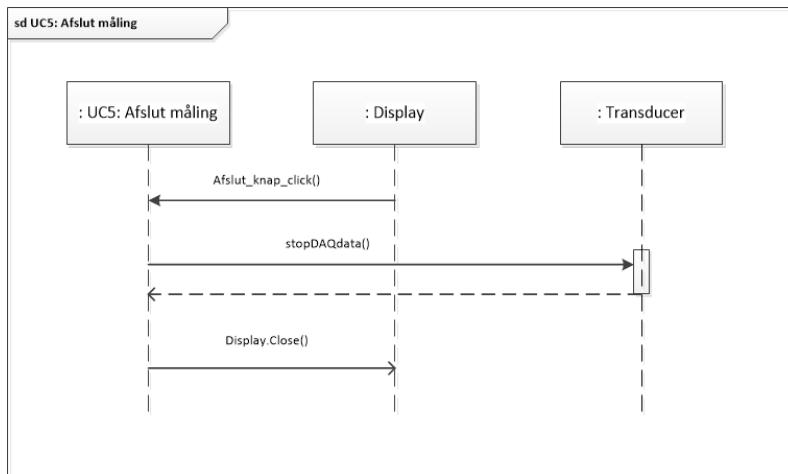
Use Case 4



Figur 3.20: Sekvensdiagram for Use Case 4

Ovenstående diagram viser at for at få gemt data fra signalet, kræver det at der trykkes på Gem-knap på display, hvor efter systemet konstant vil sende data ned i databasen indtil der igen trykkes på Gem-knappen, for at stoppe gemning af data.

Use Case 5



Figur 3.21: Sekvensdiagram for Use Case 5

Ved afslutning af en måling ses det at forsker trykker på Afslut-knap på display, hvorefter indhentingen af data fra DAQ stoppes og programmet lukker ned.

3.2.2 Implementering

Indledende implementeringsovervejelser

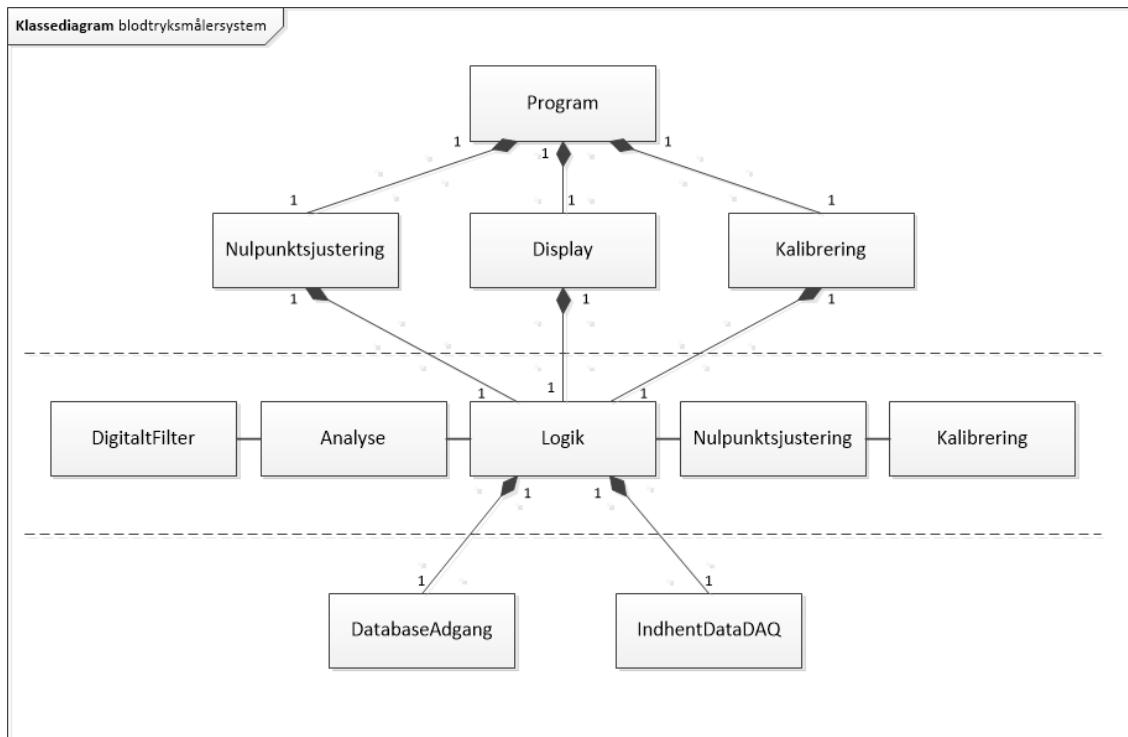
På baggrund af designfasen for softwaren kan implementeringen af softwaren påbegyndes. Softwaredesignet viser at systemet skal implementeres med en GUI applikation, som aktøren kan interagere med systemet gennem. Derudover er det kendt at softwaren skal indeholde en række klasser, hvor i funktionalitetér som kalibrering, nulpunktsjustering, digitalt filter og indhentning af systolisk-, diastoliske- og puls-værdier skal placeres. I det følgende beskrives de overvejelser vi har gjort i forhold til implementering af disse funktionaliteter og hele softwaresystemet generelt.

Implementeringen af softwaren sker i Visual Studio 2013 i sproget C#. Dette er valgt da programmet er godt til arbejde med GUI applikationer, samt til håndtering af tråde og tråd kommunikation. Tråde benyttes i softwaren, da systemet der skal implementeres er et eventdrevet system, hvilket vil sige at systemet skal kunne håndtere mange handlinger på en gang. Handlingerne igangsættes af events der kommer af aktørens interaktion med systemet. Tråd kommunikationen fungerer således at en tråd kan sende et signal ud som andre tråde kan reagere på.

Klasse implementering

På baggrund af designmodellerne er det besluttet at opbygge systemkoden efter principperne i en trelagsmodel. Trelagsmodellen indeholder et præsentations-lag, et logik-lag og et data-lag. Præsentations-laget består af de klasser som systemets aktører har tilgang til. Logik-laget er det analyserende lag. Det er således i dette lag at signalet behandles. Logik-laget har tilgang til de andre lag som det eneste. Det betyder at præsentations-laget og data-laget ikke kan kommunikere sammen, derved skal denne kommunikation foregå gennem logik-laget. Data-laget er tilgangen til den implementerede database og til indhentning af blodtrykssignalet fra hardware.

Fordelen ved trelagsmodel opbygningen er at det skaber et godt overblik i koden, og skaber en kode med lav kobling, da hver enkelt klasse har hvert sit specifikke ansvar. Hvilket gør at koden er let at vedligeholde og ændre hvis funktionaliteter ønskes opbygget anderledes. Et overordnet klassediagram over systemet er udarbejdet på baggrund af præcisering af applikationsmodellen, se figur 3.22.



Figur 3.22: Klassediagram

Brugergrænseflade

Displayet (GUI) er aktørens, i dette tilfælde forskerens, indgang til systemet. Derfor er det vigtigt at den er opbygget efter hvad der følger forskerens logik. Til at klarlægge dette er principperne om en god brugergrænseflade taget i mente. Brugen af disse kommer til udtryk ved, at det tydeligt fremgår af hver knap eller label hvad dens formål er, samt at størrelsen af det enkelte komponent er tilstrækkelig stor til at det ikke er til at overse. Komponenterne på display er logisk placeret, det vil sige at de dele som forsker først skal forholde sig til og eventuelt udfylde er placeret i venstre side af display. Dette vil give mening såfremt systemet benyttes af personer fra den vestlige verden, hvor læseretningen er fra venstre mod højre.

Det er et krav at forsker indtaster et forsøgsnavn inden at en måling startes, derfor er komponenterne implementeres således at knappen "Start måling" først bliver aktiveret når der er indtastet noget i tekstdoboksen hvori forsøgsnavn skal indtastes. Systoliske-, diastoliske og puls-værdi er placeret efter hvilken rækkefølge det typisk ses på standard blodtryksapparater.

Indsæt figur - Evt. alle tre form i et billede

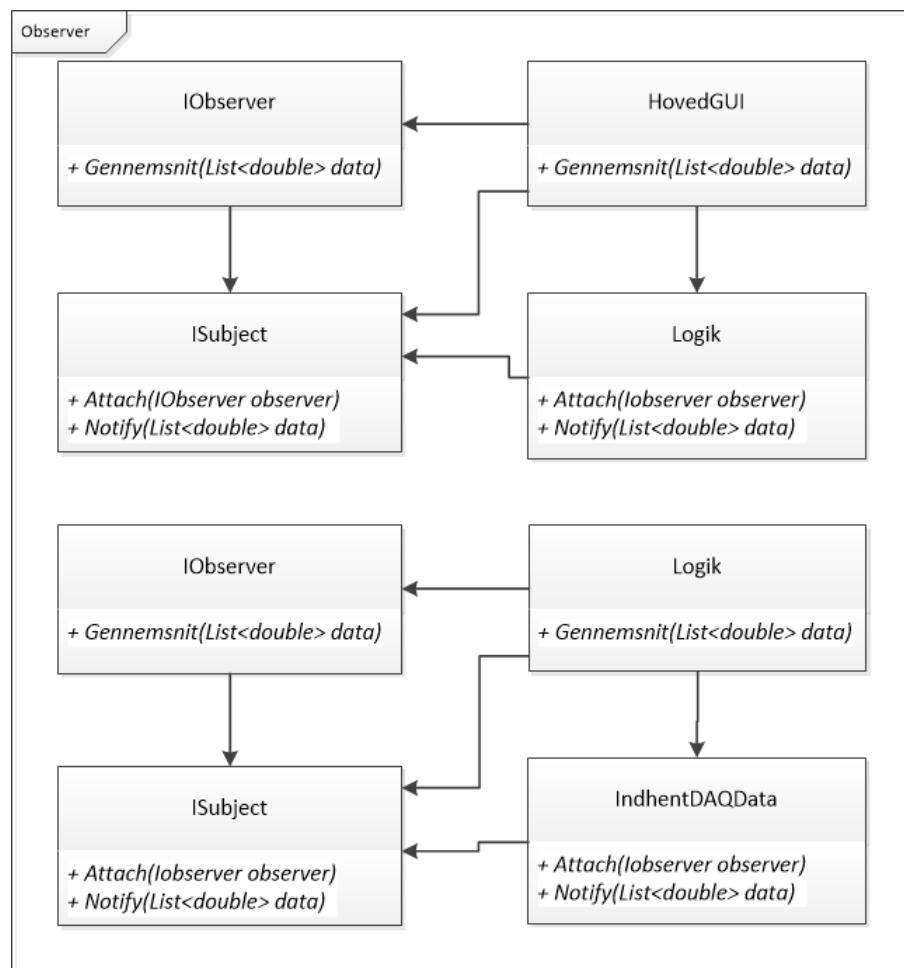
Af figur xx ses det at grafen er en væsentlig del af display's brugergrænseflade. Grafen implementeres som en Windows Form komponent. Det vælges at få vist signalet som en kurve, og førsteaksen indstilles til tid i sekunder fra 0 til 7 sekunder, og andenaksten til en minimums værdi på 0 mmHg og en maksimum værdi på 250 mmHg, hvilket er givet i kravspecifikationen.

Observer - Strategy

Observer og strategy er to programmeringsmønstre. Der i samarbejde med hinanden er gode til at håndtere at sende data fra et lag til et andet lag. Det er valgt at bygge softwarekoden op efter disse to mønstre. Observer definerer et en til mange forhold mellem objekter således at en ændring i et objekts tilstand medfører at de mange objekter informeres om ændringer og dermed opdateres automatisk.

Dette implementeres ved at oprette to interfaces IObserver og ISubject. Disse interfaces placeres i deres eget namespace, som alle lag kan tilgå, samt gør det muligt for alle nødvendige klasser arve fra disse interfaces. I ISubject placeres de generelle metoder Notify() og Attach(), hvis ansvar er at informere og flytte data fra en klasse til en anden klasse når de kaldes i Subject-klassen. IObserver indeholder metoden der kaldes i Observer når en Notify() fra Subject og ISubject modtages.

Mønstret opbygges som en push, hvilket vil sige at når Subject har ny data klar til at sende op til Observer, kaldes metoden Notify() indeholdende data'en som parametre og dette sendes op til Observer, via ISubject og IObserver. Således fortsætter koden med at arbejde så længe ny data ønskes flyttes op. Mønstret benyttes både mellem data-laget og logik-laget, og mellem logik-laget og præsentations-laget. Skematisk er det i dette projekt givet ved, hvor de relevante metoder i forhold til mønstret er medtaget, se figur 3.23.



Figur 3.23: Observer mønstre

Strategy mønstret indkapsler algoritmer og gør dem udskiftelige med hinanden. Det vil sige at en metode oprettes i et interface. Klasser vil så arve fra dette interface, afhængig af hvem der bruger metoden vil metoden så blive overskrevet i klassen og den nødvendige funktion tilføjet. I samarbejde med Observer-mønstret bruges det ved at Subject arver fra ISubject, og Observer arver fra IObserver. I projektet blev mønstrene i første omgang først benyttet fra logik-laget til præsentations-laget i forbindelse med at sende data til visning i graf. Men undervejs viste det sig nødvendigt også at implementere mønstrene fra data-laget til logik-laget, således at det kan kontrolleres hvor stor en mængde data der sendes op ad gangen.

Samplefrekvens

Samplefrekvensen er som krav givet til 1000 Hertz. Hvilket svarer til at systemet modtager 1000 samples i sekunder. Varigheden af en sample er givet ved:

$$\frac{1}{f_s} = \frac{1}{1000} = 0.001 \text{ sek} \quad (3.14)$$

Det har vist sig under arbejdet med softwaren, at systemet ikke kan følge med til at modtage så mange målinger i sekundet. Derfor er det valgt at skære i antallet af målinger pr. sekund der skal videre bearbejdes i logik-laget og udskrives i præsentations-laget. Antallet skæres ned til 50 målinger pr. sekund. Dette gøres ved at gennemsnittet af 20 målinger efter hinanden bestemmes, hvorefter gennemsnitsværdien returneres og gemmes i listen der sendes videre i systemet. Herefter findes så gennemsnittet af de næste 20 målinger og således fortsættende.

Nulpunktsjustering

Formålet med en nulpunktsjustering er at flytte signalets offset enten op eller ned, så det atmosfæriske tryk altid er placeret ved 0 volt på outputsignalet. Dette gøres ved at åbne for den tilsluttede transducer til systemet, så det atmosfæriske tryk måles. Ud fra denne værdi kan justeringsfaktoren så bestemmes ved, hvor x er det målte atmosfæriske tryk i volt modtaget gennem DAQ'en:

$$faktor_{jus} = 0 - (x) \quad (3.15)$$

Af ligningen ses det at justeringsfaktoren både vil kunne blive positiv og negativ, afhængig af om offset værdien skal rykkes op eller ned for at blive placeret i nul. Optimalt set vil det atmosfæriske tryk være en konstant værdi ved den samme måling, men det opleves at der er en smule støj på signalet og derfor vil den målte værdi være en tilnærmelse af det atmosfæriske tryk. Systemet ønskes nulpunktsjusteret for at sikre at alle de målte blodtrykssignaler har samme udgangspunkt. Hvilket gør at målingerne kan sammenlignes. Systemet foretager automatisk nulpunktsjusteringen når systemet startes ved at retunerer den første værdi fra DAQ'en, når der trykkes på knappen FORETAGET. Denne værdi er justeringsfaktoren der lægges til samtlige samples i det indhentede blodtrykssignal.

Kalibrering

Ved kalibrering ønskes det at bestemme hardwarens visningsfejl. I dette projekt betyder det at kalibreringsfaktoren fra volt til millimeter kviksolv bestemmes. Denne bestemmes

ved at tilkoble en væskesøjle til systemet. Væskesøjlen fyldes med vand til den vil give et kendt mængde tryk på systemet angivet i mmHg. Herefter kan output i volt fra hardwaren måles. kalibreringsfaktor er givet ved:

$$faktor = \frac{x[\text{mmHg}]}{y[\text{Volt}]} \quad (3.16)$$

x angiver trykket fra væskesøjlen, denne hardcodes til 50 mmHg. y angiver den målte spændingsoutput på hardwaren. Optimalt set er kalibreringsfaktoren givet ved:

$$\frac{250[\text{mmHg}]}{5[\text{V}]} = 50 \quad (3.17)$$

hvor 250 mmHg er det maksimale blodtryk systemet kan måle og 5 Volt er maks spændingen i volt. Grafisk vil det se ud som vist på figur 3.8 under hardware modultest. Af figur 3.8 kan det aflæses at den optimale outputspænding ved 50 mmHg er 1 Volt. Kalibreringsfaktoren skal ganges på samtlige sample-værdier der kommer fra DAQ'en og som ønskes udskrevet på graf i display. Kaliberingen implementeres i softwaren ved brug af konfiguration. Forskeren beregner omsætningsværdien udfra ligning 3.16. Resultatet af denne beregning indtaster forsker i konfigurations xml-filen under App.settings. XML-filen kan tilgås uden opstart af systemet, derfor bliver kaliberingen uafhængig af hvornår systemet kører og kaliberingen kan dermed foretages på et vilkårligt tidspunkt. Værdien der ændres i XML-filen er den tilhørende "Value" til "KalibreringsKoefficient". Den er markeret med grøn firkant omkring på figur 3.24.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3      <startup>
4          <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
5      </startup>
6      <appSettings>
7          <add key="KalibreringsKoefficient" value="50"/>
8      </appSettings>
9  </configuration>
```

Figur 3.24: Konfigurations XML-fil

Metoden Kalibrering() i Kalibreringsklassen, som er en del af logik-laget læser så "Kalibreringskoefficienten" fra konfigurations-filen hver gang kalibreringsfaktoren skal ganges på et signal.

Det er vigtigt at pointere at nulpunktsjusteringsfaktoren lægges til samtlige værdier i signalet førend at kalibreringsfaktoren ganges på. Dette udføres i kodens logik-lag.

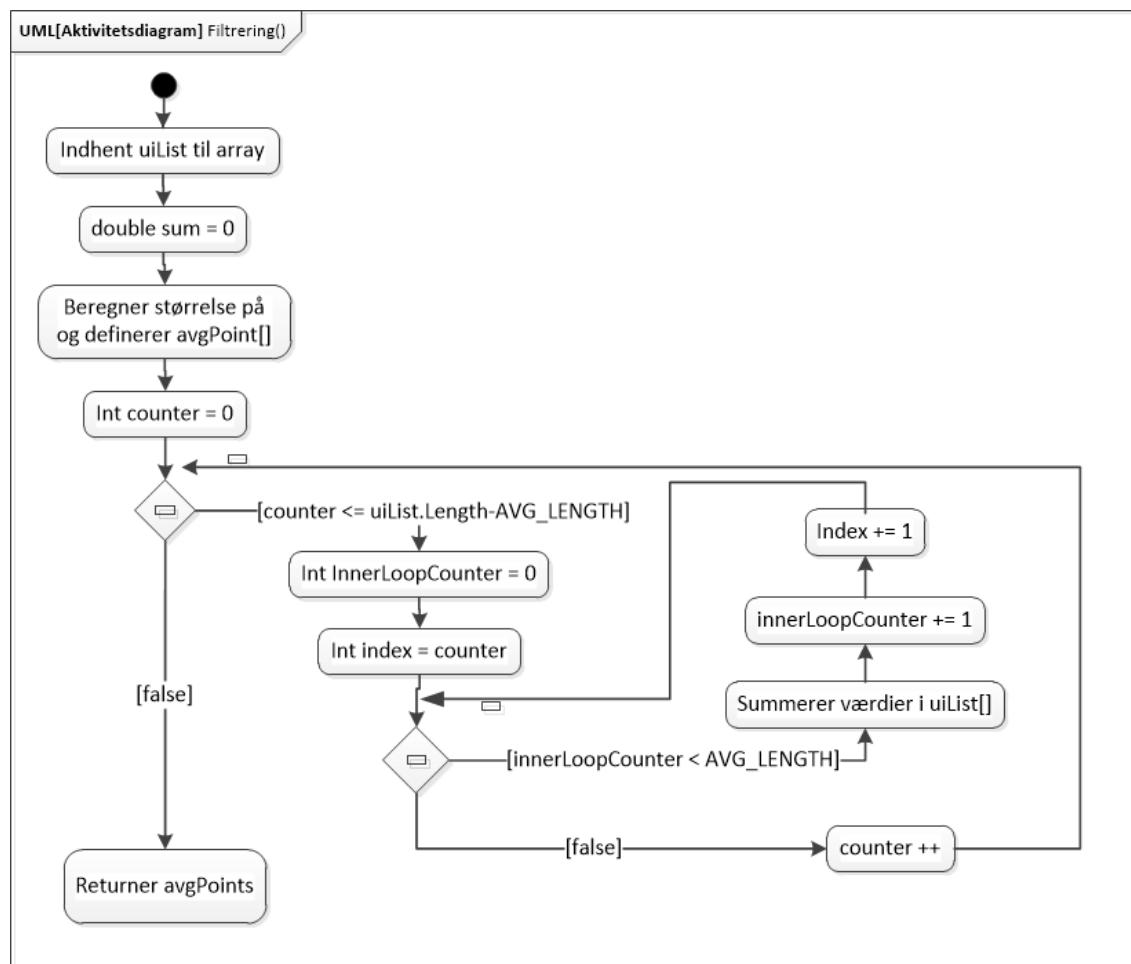
Digitalt Filter

[?] Formålet med implementering af et digitalt filter er at fjerne støj fra det indhentede signal. Dette gøres ved at udglatte signalet. Til dette kan en række forskellige filtre benyttes. Vi har valgt at implementere et glidende middelværdifilter (moving average filter). Fordelen ved dette filter er at det er simpelt at forstå og at det er optimalt at bruge på signaler i tidsdomænet. Skulle signalet være vist i frekvensdomænet ville valget have faldet på et helt andet filter.

Det glidende middelværdifilter fungerer ved midling af en række punkter fra inputsignalet for at frembringe hvert punkt i outputsignalet. Hvilke punkter der tages fra inputsignalet vil flytte sig en plads for hvert beregnet outputsignal punkt, heraf kommer den glidende effekt. Matematisk er filtret givet ved:

$$y[i] = \frac{1}{M} \cdot \sum_{j=0}^{M-1} x[i+j] \quad (3.18)$$

Hvor $x[]$ er inputsignalet, $y[]$ er outputsignalet og M er antallet af punkter der benyttes i det glidende middelværdifilter. Denne beregning benytter sig udelukkende af punkter placeres på den samme side af output sample nummeret, hvilket vil føre til en relativ forskydning mellem input og output. M sættes til 5. Implementeringen af filtret er vist i et aktivitetsdiagram på figur 3.25.



Figur 3.25: Aktivitetsdiagram af metoden *Filtrering()*

Måden hvorpå filtret er implementeret gør at der ikke sker en filtrering af de første fire samples, det ses af følgende i koden. `AVG_LENGTH` er defineret til 5, og mængden af punkter der benyttes i filtret i ligning 3.18 svarer dette til M .

```

1 reference
public List<double> Filtrering(List<double> data)
{
    double sum = 0;
    List<double> avgPoints = new List<double>();
    for (int i = 0; i < data.Count() - AVG_LENGTH + 1; i++)
    {

```

Figur 3.26: Udsnit af koden til det glidende middelværdifilter

Det ses at der skal være minimum 5 samples i data.Count først at listen avgPoints oprettes. Det er en begrænsning vi er opmærksom på, men som accepteres da de første fire samples ved visning i graf er kørt så hurtigt igennem, at det ikke skaber en begrænsning for brugen af systemet for forsker. Optimalt set vil der sættes en begrænsning på filtret således når første måling modtages vil gennemsnittet findes af en sample, dernæst af to samples, tre samples osv. Indtil der er fem samples og gennemsnittet vil så altid bestemmes af de fem seneste samples.

Systemet gør det muligt for forsker selv at vælge om signalet ønskes vist filtreret eller ufiltreret. Dette vælges på brugergrænsefladen. Vælges visning af det ufiltrede signal sendes det indhentede signal naturligvis ikke gennem det digitale filter. Det er muligt at skifte mellem filtreret og ufiltreret signal, mens systemet kører. I det tilfælde skifter hele det viste signal til det valgte, da alt data i listen der indhentes dermed skifter. Filtreringen vil dermed ikke vise sig som en løbende kurve grafisk.

Analyse

Analyse dækker over indhentningen af de systoliske-, diastoliske- og puls-værdi ud fra blodtrykssignalet. Dette er implementeret i en klasse kaldet Analyse. Heri er placeret metoder for henholdsvis systole og diastole. I en blodtrykskurve er den systoliske værdi givet ved maximum på kurven og den diastoliske er givet ved minimums værdien på kurven. Metoderne bestemmer derfor den maksimale værdi og den mindste værdi i listen, der medtages som parametre til metoderne. Listen der bruges som parametre er UILIST indeholdende 350 tal ad gangen. UILIST er listen der sendes fra logik-laget til præsentations-laget med de behandlede data, som vises i grafen. I præsentations-laget er implementeret en timer, der håndterer at de systoliske- og diastoliske værdier i display kun opdateres hvert 3 sekund. I løbet af 3 sekunder vil der være gennemløbet 3-5 blodtryksperioder, afhængig af pulsfrekvensen. Dermed vil samtlige systoliske og diastoliske værdier ikke blive udskrevet. Intervallet på 3 sekunder er valgt da det er passende tid til at kunne nå og aflæse den pågældende værdi.

I forhold til implementering af puls er der gjort en række overvejelser og mulige løsninger. Puls er defineret ved slag pr. minut og på en puls vil der være en systole og diastole. Pulsen må derfor kunne bestemmes ved at tælle antallet af systoliske værdier på 6 sekunder, antallet ganges så med 10 for at få den rette enhed. Udfordringer er dog opstået i forhold til at kunne bestemme præcist hvornår der er gået 6 sekunder i programmet. En anden mulighed er også at bestemme pulsen ved at finde antallet af samples mellem to systoliske værdier. Omregnes samples så til sekunder og ganges op til et minut, må dette være lig med måleobjektets øjeblikkelige puls. Det er dog ikke lykkedes at omsætte overvejelserne til

kode, og pulsen er dermed ikke blevet implementeret ved projekt aflevering.

Database

I systemet er der implementeret en lokal database. Databasen er oprettet gennem host webhotel10.ihb.dk. Formålet med databasen er at lagre det målte blodtrykssignals rådata. Det er valgt at implementere databasen som typen SQL, da denne database-type indeholder de funktioner som er nødvendige for dette system. Data gemmes i denne type database i tabeller. Indledningsvis for at oprette den nødvendige tabel defineres en type til hver værdi. SQL-koden til oprettelse af tabel er vist på figur xx.

```

1  CREATE TABLE [db_owner].[SEMPRJ3] (
2      [Forsøgsnavn]      NVARCHAR (20)  NOT NULL,
3      [Id]                BIGINT        IDENTITY (1, 1) NOT NULL,
4      [Datostempel]       DATETIME     NOT NULL,
5      [Blodtryksmåling]  VARBINARY (MAX) NOT NULL,
6      PRIMARY KEY CLUSTERED ([Id] ASC)
7  );

```

Figur 3.27: SQL-kode til oprettelse af tabeller i database

Forsøgsnavnen referer til det forsøgsnavn der indtastes i GUI ved påbegyndelse af en ny måling. Dette er af typen NVARCHAR(20), hvilket betyder at forsøgsnavnet maksimalt kan være 20 tegn langt. Id er defineret som primær nøgle, det betyder at denne er unik for hver enkelt sekvens i database, og Id der vil referes til mellem tabeller i databasen, hvis flere tabeller var nødvendigt.

Et blodtrykssignal indeholder en stor mængde datapunkter, derfor gemmes signalet i en VARBINARY, hvor en række binære datapunkter gemmes som en enkelt enhed i databasen. Dette er valgt for at spare på data pladsen i databasen. Denne type besværliggør dog, at få vist hvilke værdier blodtrykssignalets datapunkter består af.

Databasen er implementeret således at flere sekvenser af den samme måling kan gemmes uden at systemet skal startes forfra. Dette er smart for forsker, hvis der testet flere ting på det samme signal.

3.2.3 Modultest

Accepttest 4

Version	Dato	Ansvarlig	Beskrivelse
0.1	28-09-2015	MHNK og MBA	Oprettelse og udfyldelse af Accepttest
0.2	30-09-2015	ABH	Tilrette accepttest
0.3	08-10-2015	Alle	Tilrette efter review med Grp. 1
0.4	15-10-2015	MBA	Indskrevet i LaTex
0.5	20-10-2015	MHNK	Tilretning
0.6	26-11-2015	MHNK	Retning af hele accepttesten. Konsekvent med stavemåder
0.7	10-12-2015	DHC, ABH, AJF	Rettelser i forhold til slutprodukt
0.8	13-12-2015	MHNK	Færdiggørelse efter accepttest. Indskrivning af godkendelser. Oprettelse og skrivning af problemrapport

4.1 Accepttest af Use Cases

4.2 Indledning

Accepttestene skal vise om produktet lever op til de standarder vi har sat op for, at den aktivt kan indgå i en forskningssituations. Accepttesten er en opfølgning af kravspecifikation, som har til formål at sikre at alle kravene er overholdt. Der vil blive testet både på hovedscenarier samt på undtagelser. Det er målsætningen, at disse test sikrer produktets kvalitet, idet produktet vil blive afprøvet før det tages i brug. Derfor er det accepttestens ansvarsfunktion, at godkende de opsatte delmål for produktet hvad angår både funktionalitet samt ikke-funktionelle krav.

Data der benyttes til målingerne fås fra In Vitro, der i form af tryk genererer et fysiologisk tryk. Brugergrænsefladen er det som forskeren initierer med, altså hvorfra systemet aktiveres. Brugergrænsefladen forkortes til GUI. Den benyttede Database er en lokal database. Når der i feltet Godkendt er et flueben, betyder det at testen er godkendt. Hvis der er et flueben i parenteser, betyder det at den er delvis godkendt.

4.2.1 Use Case 1

Indsæt beskrivelse og figurer med NI-DAQ, Analog discovery og transduceren. Det forventes for Use Case 1, at forskeren har fået påmonteret det væskefyldte kateter samt tændt for apparaturet.

Test af Use Case 1	Foretag nulpunktsjustering
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren og pop-up vindue for nulpunktsjustering er åbent

Handling	Forventet observa- tion/resultat	Faktisk observa- tion/resultat	Godkendt observa- tion/resultat
<i>Hovedscenarie</i>			
1. Forsker trykker på Foretag-knap	Systemet foretager nulpunktsjustering, hvorefter vinduet lukker	Som forventet foretager systemet nulpunktsjustering, hvorefter vinduet lukker, når forsker har trykket på Foretag-knap	✓

Tabel 4.3: Accepttest af Use Case 1

4.2.2 Use Case 2

Test af Use Case 2	Bestem kalibreringskoefficient
Scenarie	Hovedscenarie
Prækondition	Hardware er monteret ved 50 mmHg på væskesøjlen og er tilkoblet en computer med WaveForm.

Handling	Forventet observa- tion/resultat	Faktisk observa- tion/resultat	Godkendt observa- tion/resultat
<i>Hovedscenarie</i>			

1.	Output spænding fra hardware aflæses i WaveForm	Output aflæses til 1 V +/- 30%	I WaveForm aflæses output til 1 V +/- 30%	✓
2.	Beregning foretages ud fra formlen $\frac{50}{output} = coefficient$	Koefficenten beregnes til 50 +/- 30%	Som forventet beregnes koefficenten til 50 +/- 30	✓
3.	Forsker indtaster beregnet kalibreringskoefficient i konfigurations XML-fil	Koefficenten står i XML-fil	Koefficenten står i XML-fil	✓
4.	Kalibreringskoefficient kan tilgås af systemet	Sammenlign værdierne i listen råData (findes i IndhentDAQData) med den viste graf på GUI	Sammenlign værdierne i listen råData (findes i IndhentDAQData) med den viste graf på GUI	✓

Tabel 4.5: Accepttest af Use Case 2

4.2.3 Use Case 3

Test af Use Case 3	Start måling
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. UC1 er kørt succesfuldt

Handling	Forventet observasjon/resultat	Faktisk observasjon/resultat	Godkendt
<i>Hovedscenarie</i>			

1. Forsker indtaster Forsøgsnavn	Systemet tilgængeliggør Start Måling-knap	Når forsker indtaster Forsøgsnavn, så gør systemet Start Måling-knap tilgængelig	✓
2. Filteret signal er valgt per default af systemet	Radiobutton til filtret signal er checket af	Det ses at Radiobutton til filtret signal er checket af	✓
3. Forsker trykker på Start Måling-knap på GUI	Signal vises i graf på GUI	Når forsker trykker på Start Måling-knap, vises signalet i graf på GUI	✓
4. Systolisk og diastolisk blodtryk samt puls bliver vist i bokse på GUI	GUI udskriver systoliske, diastoliske og puls værdier på GUI	GUI udskriver systoliske og diastoliske værdier på GUI. GUI udskriver ikke puls ¹	(✓)

Udvidelse 1: Forsker vælger filtreret/ufiltreret signal

1. Forsker vælger ufiltreret signal	Grafen viser det ufiltreret signal	Forsker vælger ufiltreret signal og grafen viser det ufiltreret signal	✓
2. Forsker vælger filtreret signal	Grafen viser det filtreret signal	Forsker vælger filtreret signal og grafen viser det filtreret signal	✓

Tabel 4.7: Accepttest af Use Case 3

4.2.4 Use Case 4

Test af Use Case 4	Gem data
Scenarie	Hovedscenarie

¹Se problemrapport

Prækondition Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. Use Case 1 er kørt succesfuldt, Use Case 3 kører

Handling	Forventet observasjon/resultat	Faktisk observasjon/resultat	Godkendt
<i>Hovedscenarie</i>			
1. Forsker trykker på Start Gem-knap	Start Gem-knap bliver highlightet med blå kant	Når forsker trykker på Start Gem-knap bliver den highlightet med blå kant	✓
2. Forsker trykker på Stop Gem-knap for at stoppe med at gemme	Filnavnet(forsøgsnavn_Id) bliver vist i tekstboks GUI	Når forsker trykker på Stop Gem-knap for at gemme bliver filnavnet(forsøgsnavn_Id) vist i tekstboks i GUI	✓
3. Forsker trykker på Gem-knap for at stoppe med at gemme	Det fremgår af GUI at data er gemt i Database	Når forsker trykker på Gem-knap for at stoppe med at gemme, fremgår det af GUI at data er blevet gemt i Database	✓

Undtagelse 1: Forsker trykker på Stop Måling-knap

1. Forsker trykker på Stop Måling-knap til et givent tidspunkt	Grafen på GUI fastholdes og datostempel på seneste indlagte data aflæses i databasens tabel. Tiderne sammenlignes	Når forsker trykker på Stop Måling-knap til et givent tidspunkt, fastholdes grafen på GUI og datostempel på seneste indlagte data aflæses i databasens tabel, hvor efter tiderne sammenlignes	✓
----------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---

Tabel 4.9: Accepttest af Use Case 4

4.2.5 Use Case 5

Test af Use Case 4	Stop måling
Scenarie	Hovedscenarie
Prækondition	Blodtryksmålesystemet er monteret korrekt. Forskeren har tændt for Blodtryksmåleren. Use Case 1 er kørt succesfuldt, Use Case 3 kører

Handling	Forventet observa- tion/resultat	Faktisk observa- tion/resultat	Godkendt resultat
<i>Hovedscenarie</i>			
1. Forsker trykker på Stop Måling-knap	Målingen stoppes og blodtryksgrafen fastholdes	Forsker trykker på Stop Måling-knap og målingen stoppes og blodtryksgrafen fastholdes	✓

Tabel 4.11: Accepttest af Use Case 5

4.3 Accepttest af ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt
1.	Blodtryksmåleren skal indeholde en Start Måling-knap til at igangsætte målingerne.	Kør Use Case 1 og 3	Start knap er på GUI	Use Case 1 og 3 køres og der er en Start Måling-knap på GUI	✓
2.	Blodtryksmåleren skal indeholde en Stop Måling-knap, hvorfra måling kan stoppes.	Kør Use Case 1 og 3	Stop knap er på GUI	Use Case 1 og 3 køres og der er en Stop Måling-knap på GUI	✓
3.	Blodtryksmåleren skal indeholde en Start Gem-knap til påbegyndelses af at gemme måling i Database	Kør Use Case 1 og 3	Start knap er på GUI	Use Case 1 og 3 køres og der er en Start Gem-knap på GUI	✓
4.	Blodtryksmåleren skal indeholde en Stop Gem-knap til påbegyndelses af at gemme måling i Database	Kør Use Case 1 og 3	Stop knap er på GUI	Use Case 1 og 3 køres og der er en Stop Gem-knap på GUI	✓

5.	Blodtryksmåleren skal indeholde en tekstboks til forsøgsnavn, hvori forsker indtaster det pågældende forsøgsnavn	Kør Use Case 1 og 3	Tekstboks til forsøgsnavn er på GUI	Use Case 1 køres og der er en tekstboks til forsøgsnavn, hvori forsker indtaster det pågældende forsøgsnavn	✓
6.	Blodtryksmåleren skal indeholde radiobutton til filtreret signal, denne skal være default valget	Kør Use Case 1 og 3	Radiobutton til filtreret signal er på GUI	Use Case 1 og 3 køres og der er en radiobutton til filtreret signal, der er valgt per default	✓
7.	Blodtryksmåleren skal indeholde radiobutton til ufiltreret signal	Kør Use Case 1 og 3	Radiobutton til ufiltreret signal er på GUI	Use Case 1 og 3 køres og der er en radiobutton til ufiltreret signal	✓
8.	Blodtryksmåleren skal indeholde tekstbokse til puls, systolisk og diastolisk blodtryk, som vises med op til tre cifre	Kør Use Case 1 og 3	Systolisk-boks, diastolisk-boks og puls-boks er på GUI	Use Case 1 og 3 køres og der indeholder tekstbokse, der vises med op til tre cifre	✓

9.	Blodtryksmåleren skal indeholde en tekstboks, som viser filnavn(forsøgsnavn og id) på målingen, efter måling er gemt	Kør Use Case 1 og 3	Tekstboks til Filnavn er på GUI	Use Case 1 Filnavn er på og 3 køres og der er en tekstboks, der viser filnavn(forsøgsnavn og id) på målingen, efter måling er gemt	✓
10.	GUI'en skal se ud som på figur 2.3 i KS	GUI'en ser ud som figur 2.3 i KS	GUI'en ser ud som figur 2.3 i KS	GUI'en ser ud som figur 2.3 i KS	✓
11.	Forskeren skal kunne starte en default-måling maksimalt 30 sekunder efter systemet er startet	Systemet er åben samtidigt startes et stopur. Efter tryk på Start Måling-knap og målingen er startet stoppes uret	Måling er startet og stopuret viser mindre end 30 sekunder	Systemet er åbent og samtidigt med et stopur startes trykkes der på Start Måling-knap og når målingen er startet stoppes uret. Forskeren kunne altså indenfor 30 sekunder, efter systemet er startet, starte en default-måling	✓
12.	Det skal maksimalt tage 5 timer at gendanne systemet (MTTR - Mean Time To Restore)	Kan ikke testes på prototypen	Kan ikke testes på prototypen	Kan ikke testes på prototypen	✓

13.	Systemet skal have en oppetid uden nedbrud på minimum 1 måned (720 timer) (MTBF - Mean Time Between Failure)	Kan ikke testes på prototypen	Kan ikke testes på prototypen	Kan ikke testes på prototypen	✓
14.	Systemet skal have en oppetid/køretid på: $\frac{MTBF}{MTBF+MTTR} * 100 = 99,31\%$	Kan ikke testes på prototypen	Kan ikke testes på prototypen	Kan ikke testes på prototypen	✓
15.	Blodtryksmåleren skal, indenfor 3 sekunder, kunne vise systolisk og diastolisk blodtryk via graf. Dette accepteres med en tolerance på +/- 15 %	Kør Use Case 1 og 3. Der trykkes på Start Måling-knappen samtidig med at et stopur startes. Når måling vises i graf stoppes uret	Stopuret viser mellem 2.55 - 3.45 sekunder	Stopuret viser ved accepttesten 3.03 sekunder	(✓)

16. Blodtryksmåleren Kan ikke skal, indenfor testes på 5 sekunder fra prototypen der er trykket på Stop Gem-knap, have gemt målingerne i Databasen. Dette accepteres med en tolerance på +/- 15 %

Kan ikke testes på prototypen

Kan ikke testes på prototypen

✓

17. Grafen vises i ét vindue, hvor y-aksen måles i mmHg og x-aksen i tid i sekunder

Kør Use Case 1 og 3

På GUI er y-aksen målt i mmHg og x-aksen i tid pr. sekund

Use case 1 og 3 køres og grafen vises i ét vindue, hvor y-aksen måles i mmHg. x-aksen vises ikke i sekunder, men i antal samples²

✗

18. Hvert 3. sekund skal værdier for systolisk og diastolisk blodtryk samt puls opdateres. Dette accepteres med en tolerance på +/- 15 %

Kør Use Case 1 og 3. Forsøgsnummer indtastes og der trykkes på Start Måling-knappen samtidig med at et stopur startes. Når værdier i bokse vises stoppes uret

Stopuret viser mellem 5.95 - 8.05 sekunder

Stopuret viste til accept-testen viste stopuret 5.01 sekunder

✓

²Se problemrapport

19.	Graf for blodtryk skal kører kontinuerligt i GUI efter principippet på figur 2.4	Kør Use Case 1 og 3	Grafen i GUI kører kontinuerligt efter principippet på figur 2.4	Use case 1 og 3 køres og grafen i GUI kører kontinuerligt efter principippet på figur 2.4	✓
20.	Når der trykkes på Stop Gem-knap gemmes signals rådata under det indtastede forsøgsnavn og et autogenereret id. " <i>forsøgsnavn_id</i> "	Kør Use case 1, 3 og 4	Data er blevet gemt i Databasen under filnavnet " <i>forsøgsnavn_id</i> "	Use case 1, 3 og 4 køres og når der trykkes på Stop Gem-knap gemmes signalets rådata under det indtastede forsøgsnavn og et autogenereret id. " <i>forsøgsnavn_id</i> "	✓
21.	Systemet skal kunne måle blodtryksværdier fra 0 til 250 mmHg	Kør Use Case 1 og 3	Det indhentede signals blodtryksværdier er indenfor 0 til 250 mmHg på grafens y-akse	Der er ingen begrænsning ³	(✓)
22.	Forskeren skal kunne udskifte batterierne til hardwaren på 2 minutter.	Udskiftning af batterier påbegyndes samtidig med at stopur startes. Når de er udskiftet stoppes uret	Stopuret viser mindre end 2 minutter	Forsker kan hurtigt skifte dette. Stopuret viser under 2 minutter	✓

³Se problemrapport

23. Softwaren skal Åbn systemets Koden er Koden er
opbygges med programkode opbygget med opbygget med
lav kobling lav kobling lav kobling ✓
-

Tabel 4.12: Accepttest af Ikke-funktionelle krav

4.4 Godkendelsesformular

Godkendes af Peter Johansen

Kunde IHA

Dato for test _____

Ved underskrivelse af dette dokument godkendes den kørte accepttest.

Sted

Dato

Kundens underskrift

Leverandørens underskrift

4.5 Problemrapport

Use Case 3, handling 4: Systolisk og diastolisk blodtryk samt puls bliver vist i bokse på GUI. Alle boksene er oprettet og vises i GUI, men kun systolisk og diastolisk bokse blive udfyldt. Puls er ikke kommet til at virke.

Ikke-funktionelle krav, handling 17: Grafen vises i ét vindue, hvor y-aksen måles i mmHg og x-aksen i tid i sekunder. y-aksen vises som tænkt i mmHg, men x-aksen vises ikke i tid, men i antal samples.

Ikke-funktionelle krav, handling 21: Systemet skal kunne måle blodtryksværdier fra 0 til 250 mmHg. Dette kan systemet også, det er blot blevet valgt ikke at sætte nogle begrænsninger på, så y-aksen med mmHg stopper ikke ved de 250 mmHg. Hvis dette ønskes kan det uden problemer tilføjes i koden.

Litteratur

- [1] Richard G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall, 2011.
- [2] Peter Johansen. *Instrumentationsforstærkeren*. IHA, 2014.
- [3] *INA114 Datasheet*.
- [4] https://en.wikipedia.org/wiki/Sallenkl_13.16.
- [5] Gregory J. Toussaint Roland E. Thomas, Albert J. Rose. *The Analysis and Design og Linear Circuits*. Wiley, 2012.

Figurer

2.1	Aktør-kontekstdiagram	4
2.2	Use Case-diagram	5
2.3	Skitse af GUI	10
2.4	Graf for blodtryk	11
3.1	Block Definition Diagram for hardware	13
3.2	Internal Block Diagram for hardware	13
3.3	Bodeplot	17
3.4	Diagram over HW	18
3.5	Forstærknings blok	18
3.6	Måling for 10 Hz	19
3.7	Måling for 50 Hz	19
3.8	Måling for 60 Hz	20
3.9	Graf til kalibrering, fra udregninger	20
3.10	Opstilling	21
3.11	Måling ved 50 mmHg	21
3.12	Måling ved 10mmHg	22
3.13	Måling ved 100mmHg	23
3.14	Overordnet sekvensdiagram for systemet	24
3.15	Domænemodel	24
3.16	Applikationsmodel for software	25
3.17	Sekvensdiagram for Use Case 1	26
3.18	Sekvensdiagram for Use Case 2	27
3.19	Sekvensdiagram for Use Case 3	27
3.20	Sekvensdiagram for Use Case 4	28
3.21	Sekvensdiagram for Use Case 5	28

3.22 Klassediagram	30
3.23 Observer mønstre	31
3.24 Konfigurations XML-fil	33
3.25 Aktivitetsdiagram af metoden Filtrering()	34
3.26 Udsnit af koden til det glidende middelværdifilter	35
3.27 SQL-kode til oprettelse af tabeller i database	36