

AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI 3. SEMESTERPROJEKT



Rapport

Gruppe 1

Lise Skytte Brodersen (201407432)
Nina Brkovic (201406458)
Jakob Degn Christensen (201408532)
Toke Tobias Aaris (201407321)
Annsofie Randrup Wagner (201406360)
Anders Wiggers Birkelund (201404118)

Vejleder

Studentervejleder
Peter Johansen
Aarhus Universitet

15. december 2015

Resumé

Projektet tager udgangspunkt i problemstillingen om at kunne monitorere et blodtryks-signal invasivt. Invasive blodtryksmålinger bliver til hverdag benyttet under operationer samt hos svært syge patienter på intensiv afdelinger.

I dette projekt skal der udvikles et blodtryksmålesystem, der kontinuerligt kan vise en blodtrykskurve på en monitor. Blodtryksmålesystemet skal uddover den kontinuerlige visningen kunne kalibreres, nulpunktjusteres, kunne anvende digitalt filter samt gemme data om målingen i en database.

Systemet består af en softwaredel, der skal løse de funktionelle krav for Blodtryksmålesystemet. Før blodtrykssignalet kommer i kontakt med softwaren, skal signalet behandles af forskellige hardwarelementer. Der er blevet udleveret en A/D konverter og en tryktransducer ved projektets start. Signalet skal ydermere forstærkes og filtreres i en Signalbehandlingsblok, som gruppen selv designer og realiserer på veroboard.

Signalbehandlingsblokken behandler det analoge signal fra tryktransduceren som ønsket. Softwaren er blevet udviklet i Visual Studio og opfylder de opstillede krav for Blodtryksmålesystemet.

Det udviklede Blodtryksmålesystem gennem hele processen er blevet testet. Hardware og software er blevet testet hver for sig via unittest og integrationstest. Systemet er som enhed blevet testet ved udførsel af accepttest. Blodtryksmålesystemet er blevet afprøvet på forskellige måleobjekter, både via simulerede blodtryksmålinger fra PhysioNet, via en In Vitro hjertemodel og til slut på en levende gris.

Abstract

This paper studies the challenges of invasive measurement of blood pressure. This procedure is used daily in surgery of patients suffering from severe illness, such as those in intensive care units.

The goal of this assignment is to develop a system, which can measure and display blood pressure continuously on a screen. The system requires functions such as: calibration, zero adjustment, digital filtration and saving measured data on a database.

The system consists of a software module with the purpose of solving the functional requirements. Before being processed by the software, the blood pressure signal must be amplified and filtered by dedicated hardware elements. These elements will be designed and built by the group. A handed out pressure transducer and an A/D converter is used to produce and digitize the blood pressure signal.

It is concluded that the final system of hardware and software meets the requirements

specified in the paper.

The system has been tested continuously during the work process. Unit tests have been used to test software and hardware independently. Integration tests and acceptance tests have been performed to test the system in its entirety. Furthermore the real life application of the system has been tested on a In Vitro heart simulator, simulated APB signal acquired from PhysioNet, and eventually on a live pig.

Ordliste

Forkortelse Forklaring

bdd	Block Definition Diagram
ibd	Internal Block Diagram
DTO	Data Transfer Object
GUI	Graphical User Interface
UC	Use Case
DAQ	Data Acquisition USB-device fra National Instruments
FURPS+	Functionality, Usability, Reliability, Performance, Supportability and others
UML	Unified Modelling Language
SysML	Systems Modelling Language
SW	Software
HW	Hardware

Tabel 1: Ordliste

Indholdsfortegnelse

Resumé	i
Ordliste	iii
Kapitel 1 Indledning	1
Kapitel 2 Projektformulering og afgrænsning	2
Kapitel 3 Baggrund	4
3.1 Kroppens kredsløb	4
3.2 Invasiv blodtryksmåling	8
3.2.1 Dynamiske egenskaber	11
Kapitel 4 Systembeskrivelse	15
Kapitel 5 Krav	16
Kapitel 6 Projektbeskrivelse	19
6.1 Projektgennemførelse	19
6.1.1 Deadlines	19
6.1.2 Mødestruktur	20
6.1.3 ASE-modellen	20
6.1.4 V-modellen	21
6.1.5 Vandfaltsmodellen	22
6.2 Metode	22
6.3 Specifikation og analyse	24
6.3.1 Hardware	24
6.3.2 Software	25
6.4 Arkitektur	25
6.4.1 Hardwarearkitektur	25
6.4.2 Softwarearkitektur	27
6.5 Design, implementering og test	27
6.5.1 Hardware	28
6.5.2 Software	30
6.6 Resultater og diskussion	31
6.6.1 In Vitro hjertemodel	31
6.6.2 Blodtryksmåling på en gris	33
6.7 Opnåede erfaringer	35
6.8 Fremtidigt arbejde	35
Kapitel 7 Konklusion	37

Litteratur	39
Bilag	40

Indledning 1

Måling af det arterielle blodtryk er en af de hyppigst udførte kliniske undersøgelser og et uundværligt led i den kliniske vurdering af patienter. Der findes forskellige metoder til måling af blodtrykket, både invasive og non-invasive metoder, hvor der ved begge tilfælde bestemmes en værdi for det systoliske og diastoliske blodtryk samt puls.

De allerførste målinger af blodtryk blev udført på dyr. Den engelske præst, Stephen Hales (f. 1677), var den første person til at lave invasive målinger af blodtrykket. Han introducerede en ny teknik, der gik ud på at indsætte den ene ende af et messingrør i venstre arterie på en hest og derefter tilslutte den anden ende af røret til et lodret glasrør således, at han kunne måle blodsøjlens højde og iagttage, hvordan denne steg og faldt i takt med hestens puls. Herved observerede han også, at hestens blodtryk var lavere, når den var i hvile, end når hesten var ophidset.

Den non-invasive metode, vi kender til i dag, hvor der anvendes et kviksølvmanometer, blev introduceret i 1896 af den italienske læge Scipione Riva-Rocci (f. 1863), som med en okklusionsmanchet gjorde det muligt at måle det systoliske blodtryk. Få år efter, i år 1905, blev korotkoff-lydene for første gang beskrevet af den russiske kirurg Nikolai S. Korotkov. Det viste sig ligeledes at få stor indflydelse på nutidens metode til at måle et pålideligt indirekte blodtryk ved den auskultatoriske metode. Dog er den elektroniske blodtryksmåling væsentlig mere udbredt til non-invasive målinger, hvor der anvendes et almindelig blodtryksapparat [9].

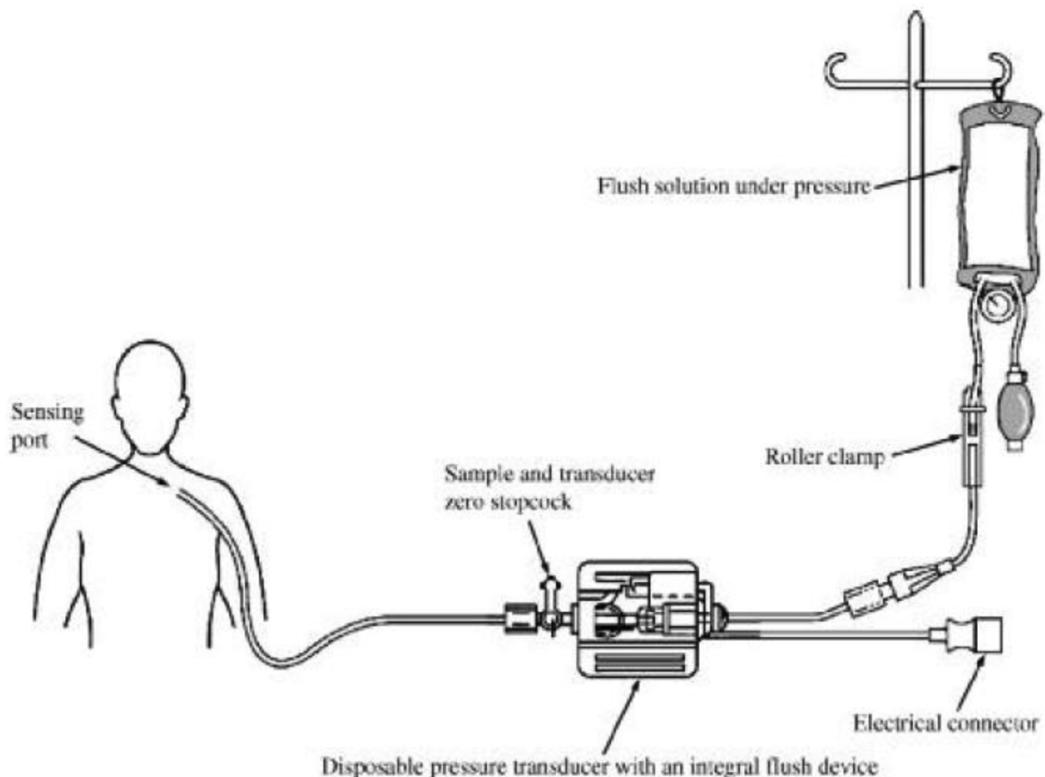
Derudover er det i daglig klinisk praksis ofte meget anvendeligt at kunne monitorere en patients blodtryk kontinuert, eksempelvis under operationer eller på intensive afdelinger med svært syge patienter. Her bliver blodtrykket målt invasivt, hvor Blodtryksmålesystemet er tilsluttet patientens arterier via et væskefyldt kateter. Således kan de hæmodynamiske data monitoreres. Denne metode giver et præcist og kontinuert, grafisk overblik over hjertets arbejde.

Denne rapport tager udgangspunkt i ovenstående metode, hvor formålet med projektetet har været at lave et monitoreringssystem, der kan behandle og udskrive de hæmodynamiske data.

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 et vigtigt parameter til monitorering af deres helbredstilstand.

Blodtrykket måles invasivt, det vil sige, at Blodtryksmålesystemet er tilsluttet patienternes arterier via et væskefyldt kateter, som afbildet i Figur 2.1.



Figur 2.1: Måleopstilling, [1, s. 296]

I dette projekt bliver der fokuseret på at udvikle et blodtryksmålesystem, som kan benyttes til forskning, hvor det typisk vil være interessant at observere blodtrykket kontinuerligt. Systemet realiseres ved udvikling af to elementer, hvoraf det ene består af et elektronisk hardwarekredsløb, hvis formål er at forstærke signalet fra en tryktransducer samt at filtrere signalet med et indbygget analogt filter.

Det andet element realiseres ved et C#-program, der skal udskrive blodtrykket som funk-

tion af tiden. Programmet skal derudover leve op til en række krav for blandt andet kalibrering, nulpunktsjustering, digital filtrering samt mulighed for at gemme interessante målinger af blodtrykssignalet til efterfølgende forskning i en database. Ydermere er der valgt at afbildede systolisk og diastolisk blodtryk med tal, samt angivelse af pulsværdi.

I forhold til udvikling af softwaren, har det været nødvendigt at lave antagelser, hvad angår design af brugergrænsefladen, da der ikke har været kommunikation eller mulighed for test af systemet med slutbrugerne.

Ansvarsområder

De forskellige opgaver i forhold til udvikling af Blodtryksmålesystemet er blevet uddelt mellem gruppemedlemmerne. Der er en gruppe, der har haft ansvaret for udviklingen, dokumentationen samt rapportelementer for hardwaren, mens den anden gruppe har haft ansvaret for udviklingen, dokumentationen samt rapportelementer for softwaren. Alle gruppens medlemmer har haft ansvaret for de resterende rapportelementer. Opdelingen kan ses i Tabel 2.1.

HW udvikling og dokumentation	SW udvikling og dokumentation
Lise Skytte Brodersen (LSB)	Anders Wiggers Birkelund (AWB)
Nina Brkovic (NB)	Toke Tobias Aaris (TTA)
Annsofie Randrup Wagner (ARW)	
Jakob Degn Christensen (JDC)	

Lise Skytte Brodersen (LSB)	Anders Wiggers Birkelund (AWB)
Nina Brkovic (NB)	Toke Tobias Aaris (TTA)
Annsofie Randrup Wagner (ARW)	
Jakob Degn Christensen (JDC)	

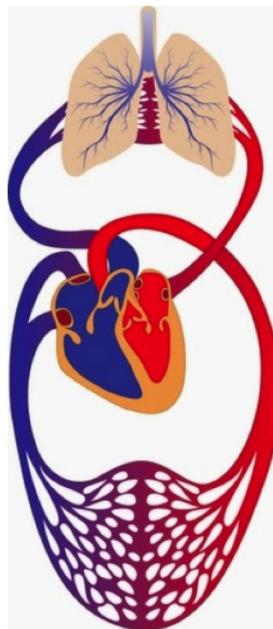
Tabel 2.1: Ansvarsområder

Baggrund 3

3.1 Kroppens kredsløb

Det menneskelige kredsløb har til hovedfunktion at transportere de forskellige livsnødvendige stoffer rundt i kroppen. Kredsløbet transporterer også kroppens affaldsstoffer ud til de organer, hvor de kan udskilles. Kredsløbet fungerer som kroppens transportmiddel.

Kredsløbet består af hjertet samt et system af blodkar. Hjertet er opdelt i to halvdeler, som begge fungerer som en pumpe. Den højre side af hjertet pumper blodet gennem det lille kredsløb, lungekredsløbet, mens venstre side pumper blodet gennem det store kredsløb, legemskredsløbet.



Figur 3.1: Illustration af kroppens kredsløb, [2]

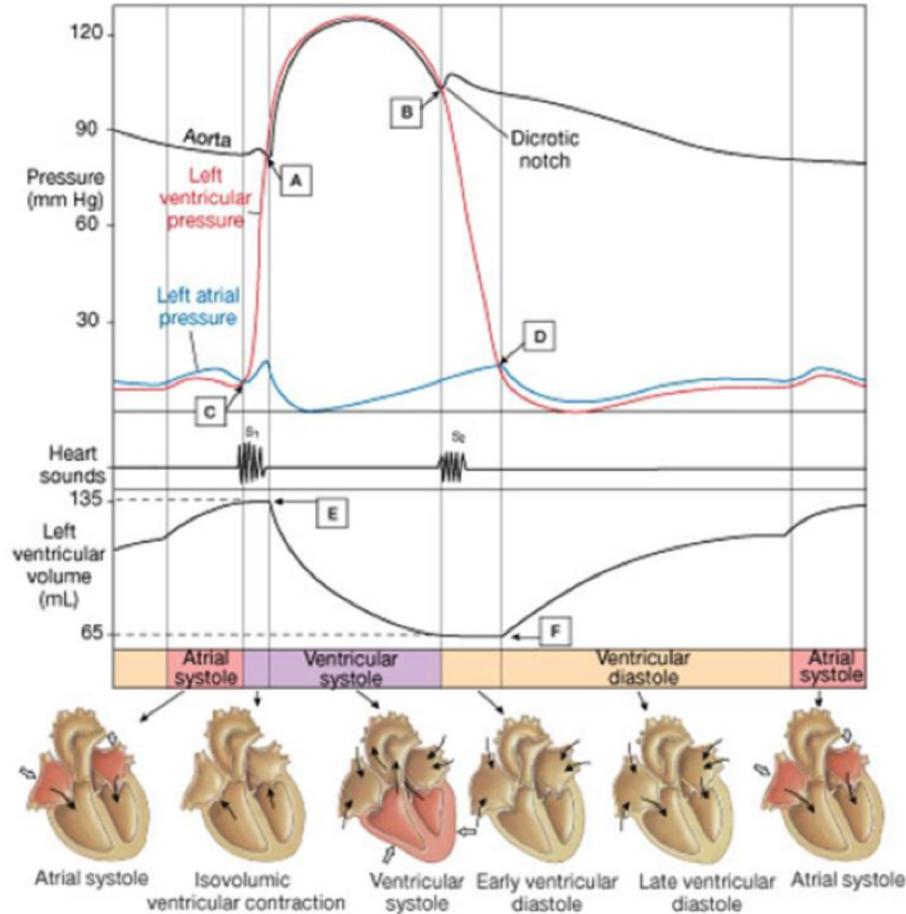
På Figur 3.1 ses det lille og det store kredsløb. Hjertet er placeret i midten, hvor det er delt op i højre og venstre side. Højre atrium modtager iltfattigt veneblod, som har været igennem det store kredsløb. Fra atrium strømmer blodet til højre ventrikel, som pumper det gennem lungearterien til lungerne, hvor blodet får tilført ilt og afgiver kuldioxid. Blodet skal nu retur til hjertet gennem lungerne til venstre atrium. Fra venstre atrium strømmer blodet til venstre ventrikel, hvor det efterfølgende pumpes ud gennem aorta til alle krop-

pens organer, som har brug for ilt. Blodet bliver igen iltfattigt og skal tilbage til højre side af hjertet, før det kan benyttes igen. Mennesket har cirka fem liter blod i kroppen, som hele tiden genbruges. Den menneskelige krop kan prioritere fordelingen af det iltrige blod i forhold til, hvor der mest er brug for det.

Systemet af blodkar indeholder overordnet tre forskellige typer: Arterier, vene og kapillærer. Arterier leder blodet fra ventrikler og ud til kroppens organer. Vene sørger for at returnere langt det meste til atrierne, og kapillærerne forbinder arterier og vene.

Arterierne er som sagt de blodkar, der modtager blodet fra hjertet. Disse skal være elastiske, da de udsættes for et stort tryk, når hjertet trækker sig sammen og pumper blodet ud i det store kredsløb. Arterierne forgrener sig til arterioler, der igen forgrener sig til kapillærer. Det er i disse blodkar udvekslingen af næringsstoffer, luftarter og affaldsstoffer til cellerne sker. Venelerne samler blodet fra kapillærerne og fører det over i vene, som returnerer det til hjertets højre side.

Hjertet har i alt fire hjerteklapper. To af klapperne betegnes som AV-klapperne, atrioventrikulærklapperne. Disse adskiller atrium og ventrikel i henholdsvis højre- og venstre side. Den i højre side kaldes tricuspidalkappen, mens den i venstre side kaldes bicuspidalkappen. Den tredje kaldes pulmonalkappen og sidder mellem lungarterien og højre atrium. Den fjerde kaldes aortakappen og sidder mellem aorta og venstre ventrikkel. Alle hjerteklapperne er designet således, at blodet kun kan løbe den ene vej igennem. Åbningen og lukningen af disse er en passiv proces, som bestemmes af forskelle i væsketrykket på de to sider af klapperne.



Figur 3.2: Illustration af hjertets cyklus, [3, s. 33]

På Figur 3.2 illustreres hjertets cyklus ud fra trykket (mmHg) i henholdsvis venstre atrium og ventrikkel, hjertelyde, lukning og åbning af hjerteklapperne samt volumen i venstre ventrikkel. Der er i alt illustreret fem faser for hjertet.

Den første fase kaldes atrium systole. Her er trykket højere i atrium end i ventrikkel, og dermed er bicuspidalkappen åben, så blodet kan strømme fra atrium til ventrikkel. Det ses også, at volumen i ventriklen stiger.

Når trykket i ventriklen overstiger trykket i atrium, lukker bicuspidalkappen, og første hjertelyd forekommer på grund af vibrationer. Dette kaldes hjertets anden fase. Her stiger trykket svagt i atrium over en kort periode, mens trykket i ventriklen stiger voldsomt. Det ses, at volumen forbliver konstant i denne fase.

I tredje fase overstiger trykket i ventriklen trykket i aorta, og aortaklappen åbner sig – denne fase kaldes ventrikel systole. Volumen i ventriklen falder i takt med, at ventriklerne trækker sig sammen. Det ses i denne fase, at trykket i ventriklen og aorta følger hinanden. Forklaringen på dette er, at aorta har en høj compliance, som gør, at den kan udvide sig og på den måde opretholde trykket. Samtidigt modtager atrium blod fra lungevenerne, hvilket gør, at trykket i atrium stiger.

Fjerde fase forekommer, når trykket i ventriklen er under trykket i aorta, hvilket forårsager, at aortaklappen lukkes, og anden hjertelyd kan høres. Denne fase kaldes tidlig ventrikel diastole. Her begynder ventriklen at slappe af.

Femte fase begynder, når ventrikel slapper så meget af, at trykket kommer under trykket i atrium. Her åbner bicuspidalklappen, og fyldningen af ventriklen begynder stille og rolig igen. Det ses, at volumen langsomt stiger. Denne fase kaldes sen ventrikkel diastole. Efter denne fase begynder første fase forfra, hvor trykket i atrium stadigvæk er større end i ventriklen.

Væskestrømningen er det væskevolumen, der fragtes gennem et rør per tidsenhed. Væskestrømningen, Q , stiger med trykforskellen i begyndelsen og slutning af røret, ΔP , samt aftager med rørets modstand mod væskestrømmen, R .

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

Trykforskellen udgør drivkraften for væskestrømmen gennem røret. Denne strøm går fra højt til lavt tryk, hvilket hjertets kontraktion fremkalder. Modstanden i røret er et udtryk for gnidningsmodstanden mellem den væske, der bevæger sig og rørvæggen, som er i ro. Denne modstand har indflydelse på væskestrømmen i et rør. Efterhånden som væsken strømmer gennem røret, falder trykket i væsken. Ved stigende modstand mod væskestrømmen, stiger dette trykfald. Når modstand mod væskestrømmen i røret stiger, bliver væskestrømmen mindre under forudsætning af, at trykforskellen ikke stiger tilsvarende. Denne modstand bestemmes af tre faktorer: rørets længde, indre diameter og væskens viskositet. Modstanden stiger med øget rørlængde, reduceret diameter og øget viskositet.

Væskestrømmen i blodkar kaldes blodstrømmen. Hjertets kontraktion sætter væsken i rørene under tryk – jo kraftigere hjertet pumper, jo større bliver trykforskellen og dermed blodstrømningen. Diameteren af blodkarret har stor betydning for modstanden mod blodstrømmen. Modstanden er mindre, og blodstrømmen er større i de store kar end i de små.

Når blodet går fra hjertet til kapillærerne forgrener blodkarrene sig til mange små og mindre kar. Herved stiger det samlede tværsnitsareal kraftigt, og der sker automatisk en aftagning af blodets strømningshastighed. Blodets lave strømningshastighed i kapillærerne er vigtig for stofudvekslingen, der sker gennem kapillærvæggene. Ved blodets vej tilbage fra kapillærerne til hjertet stiger strømningshastigheden igen, idet det samlede tværsnitsareal aftager.

Væskestrømmens tryk gennem et rør er som nævnt trykforskellen mellem rørets begyndelse og slutning. I det store kredsløb er trykket lig med forskellen mellem trykket i aorta og trykket i højre atrium. Det tryk, der er i højre atrium, er ved normale tilstande meget tæt på nul, og derved kan man betragte det gennemsnitlige tryk i aorta som trykforskellen. Det arterielle blodtryk er det tryk, der er i de store arterier, som er ligeså højt som trykket i aorta. Modstanden i hele det store kredsløb kaldes den totale perifere modstand. Formlen for sammenhængen mellem væskestrøm, trykforskel og modstand kan også overføres til det store kredsløb, hvor minutvolumen, MV, stiger med blodtrykket, BT, og aftager med den

totale perifere modtand, TPM.

$$MV = \frac{BT}{TPM} \quad (3.2)$$

Dette betyder, at ændringerne i det arterielle blodtryk skyldes forandringer i enten hjertets minutvolumen eller blodkarrenes modstand mod væskestrømmen. Arteriosklerose ændrer modstanden i blodkarrene, da der sker en indsnævring af arterierne og arterioleerne. Dette medfører en belastning for hjertet, da den samme mængde blod ønskes transporteret rundt i kroppen på samme tid, da man gerne vil opretholde minutvolumen. Hvis denne sygdom ikke behandles, vil indsnævringen blive så slem, at minutvolumen ikke kan opretholdes, og patienten vil være i en livstruende tilstand. Patientens arterielle blodtryk vil stige i takt med indsnævringen.

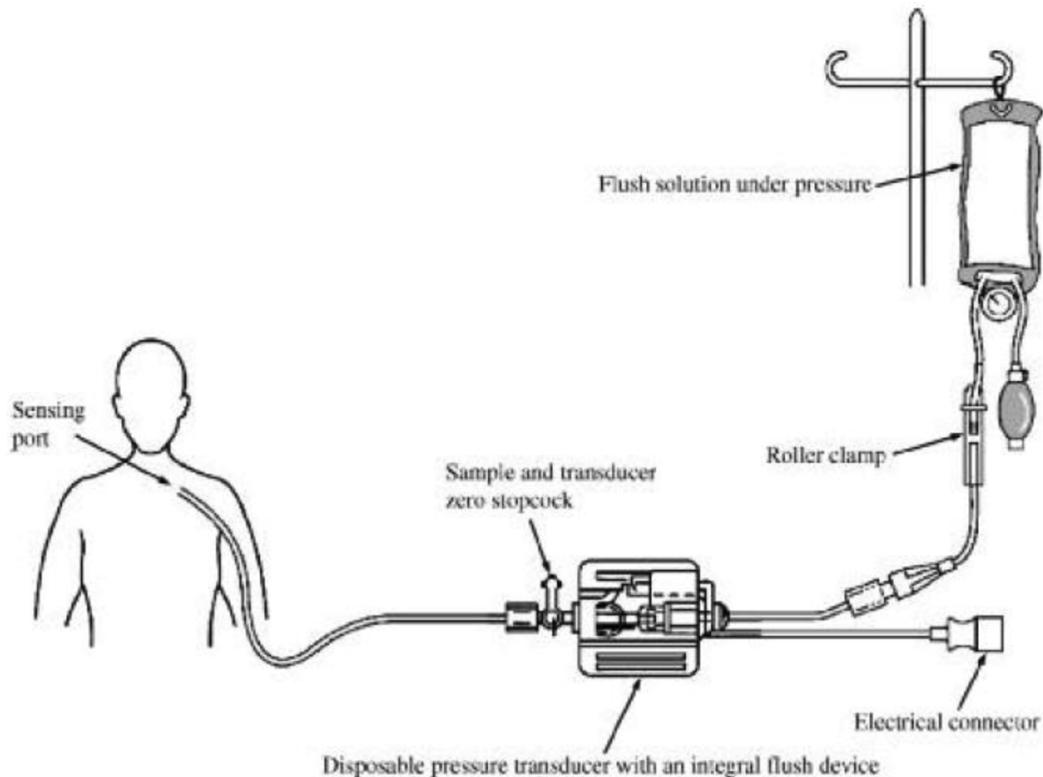
Det arterielle blodtryk (systole og diastole) angiver, hvor hårdt hjertet arbejder. Hvis man har et arterielt blodtryk, hvor systole er over 140, og diastole er over 90, siges det, at man har et forhøjet blodtryk. Et forhøjet blodtryk kan have anledning til forskellige sygdomme, eksempelvis arteriosklerose.

Der findes flere forskellige metoder til at måle blodtryk – både invasivt og non-invasivt. I begge tilfælde bestemmes en værdi for det systoliske og diastoliske blodtryk. Non-invasivt kan blodtrykket måles med ultralyd eller ved den oscillometriske metode med manchet, stetoskop og kviksølvsmannometer. Derudover findes der den automatiserede teknik, der erstatter stetoskopet med en mikrofon.

I dette projekt er der blevet designet et monitoreringssystem til at vise en invasiv blodtryksmåling.

3.2 Invasiv blodtryksmåling

Invasive blodtryksmålinger bruges ofte til monitorering af hæmodynamiske data og anvendes som regel hos svært syge patienter, på eksempelvis intensiv afdeling, samt ved større operationer. Ved den invasive blodtryksmåling måles blodtrykket via et kateter, der lægges ind i arterien. Med denne metode får man et mere præcist og kontinuerligt billede af, hvordan hjertet arbejder.



Figur 3.3: Måleopstilling, [1, s. 296]

På Figur 3.3 ses måleopstillingen for den invasive blodtryksmåling. Der føres et væskefyldt kateter ind i personens arterie. Trykposen på Figur 3.3 fyldes med natriumklorid og fungerer som en del af flushsystemet, som benyttes til at fjerne bobler fra det væskefyldte kateter. Trykket i posen pumpes op til mellem 180 og 300 mmHg, hvilket er over det systoliske blodtryk.

Slangen fra trykposen går ned til tryktransduceren og flushsystemet. Når flushsystemet startes, vil man på monitoren kunne se, at trykket stiger til det tryk, der er i trykposen. Når dette stoppes, er systemet udviklet således, at der altid vil løbe en dråbe natriumklorid ind i kateteret. Dette forhindrer blod i at strømme ud i kateteret. Trykket i det væskefyldte kateter er lig med trykket i personens arterie. Det er dette tryk, som tryktransduceren transformerer om til et elektrisk signal, som kontinuert vises på monitoren.

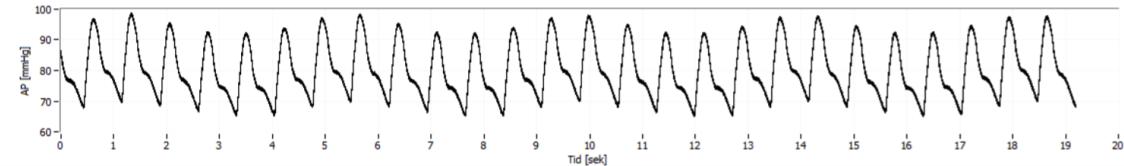
Tryktransduceren består af fire strain gauges, i dette tilfælde kaldet en Wheatstone bro. Strain gauges er en modstandståd, der kan foldes og limes på et hvilket som helst materiale. Modstanden, R bestemmes ud fra længden, l , resistiviteten, ρ og tværsnitsarealet, A .

$$R = \frac{l \cdot \rho}{A} \quad (3.3)$$

Ændres en eller flere af disse, vil modstandsværdien ændres efterfølgende. Materialet, som strain gauges er limet fast på kan ændre form, hvilket strain gauges registrerer. Det er dette, tryktransduceren benytter sig af, hvor de fire strain gauges er limet fast på en membran, som sidder for enden af det væskefyldte kateter. Når blodtrykket ændrer sig,

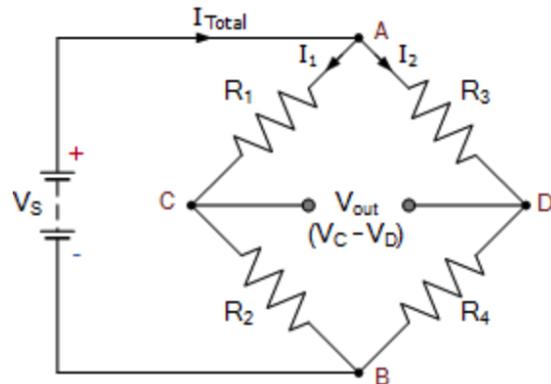
påvirkes membranen, da den har en meget højere compliance end selve det væskefyldte kateter. Denne forskel i compliance er en af grundene til, at tryktransduceren kan registrere selv de mindste trykændringer.

Trykændringer registreres af de fire strain gauges, som transformerer dem om til et elektrisk signal, hvilket afdiges grafisk. På Figur 3.4 ses afdelingen af en invasiv blodtryksmåling, hvor det også er muligt at aflæse det systoliske og diastoliske blodtryk. Det systoliske tryk svarer til toppunkterne, hvor minimumspunkterne svarer til det diastoliske tryk.



Figur 3.4: Grafisk afdeling af en invasiv blodtryksmåling, [3, s. 34]

Strain gauges benyttes i dette tilfælde, da man ønsker, at selv den mindste trykændring skal kunne registreres. Da disse modstandsændringer er små, ønskes disse forstærket op. For at kunne gøre dette bedst muligt, skal der benyttes et måle setup, som er centraliseret omkring nul. Til dette bruges en Wheatstone bro, nærmere en helbro, se Figur 3.5.

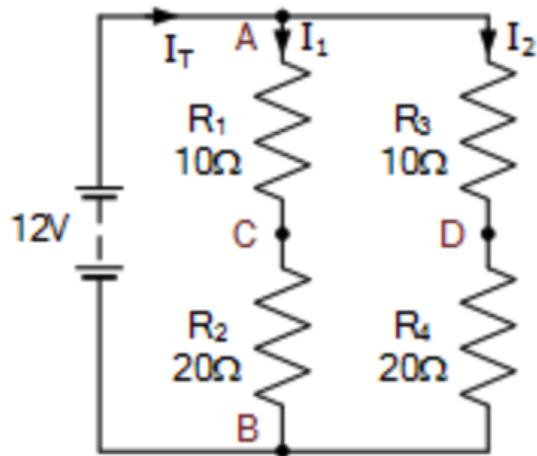


Figur 3.5: Wheatstone bro, [4]

En Wheatstone bro er et kredsløb, der består af to serieforbindelser af to modstande, som sidder i parallelforbindelse. I mellem serieforbindelserne kan V_{out} beregnes ud fra spændingen ved punkt C og punkt D.

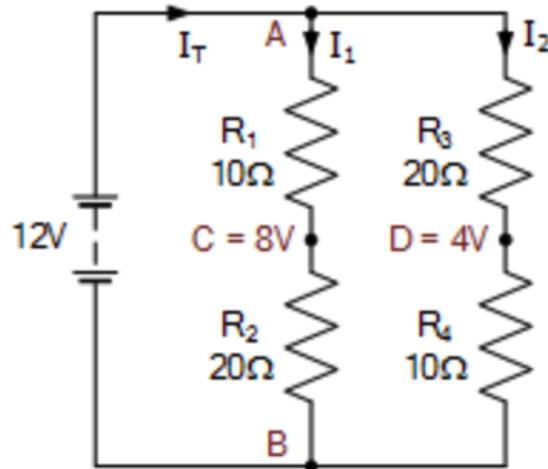
$$V_{out} = V_C - V_D \quad (3.4)$$

Hvis modstandene R_1 og R_3 samt R_2 og R_4 er ens, vil V_{out} være lig med nul, og broen vil være i balance, se Figur 3.6.



Figur 3.6: Wheatstone bro i balance, [4]

Hvis modstandene R_1 og R_3 samt R_2 og R_4 er forskellige af hinanden, vil V_{out} være forskellig fra nul, og broen vil være i ubalance, se Figur 3.7.



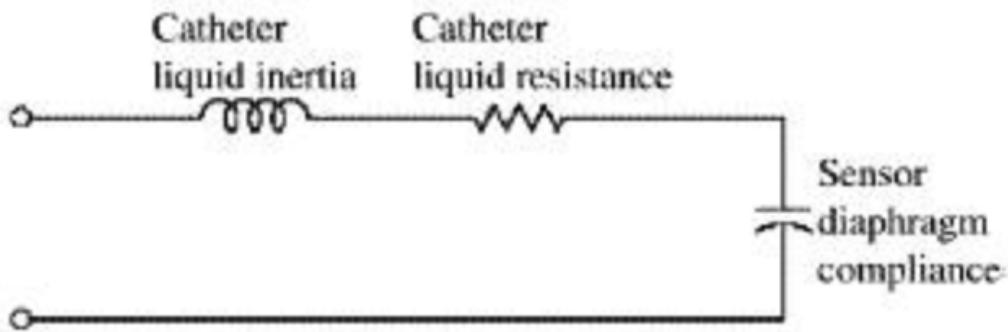
Figur 3.7: Wheatstone bro i ubalance, [4]

Når trykændringerne påvirker membranen, vil de fire strain gauges ændre form og dermed ændre modstandsværdi. Dette skaber ubalance i Wheatstone broen, og spændingen ændrer sig i takt med dette. Spændingsændringen kan beregnes ud fra nedenstående formel (3.5).

$$\frac{v_0}{v_{in}} = \frac{\Delta R}{R} \quad (3.5)$$

3.2.1 Dynamiske egenskaber

Et væskefyldt kateter har inertি, friktion og elastiske egenskaber, der kan betegnes som inertans, modstand og compliance (eftergivelse). Disse egenskaber kan beskrives som elektriske komponenter i form af en induktor, resistor og kapacitor, se Figur 3.8.



Figur 3.8: Anden ordens lavpassystem, [1, s. 303]

De forskellige egenskaber kan udtrykkes ved ligninger (3.6), (3.7) og (3.8).

$$L = \frac{l \cdot p}{\pi \cdot r^2} \quad (3.6)$$

$$R = \frac{8 \cdot \mu \cdot l}{\pi \cdot r^4} \quad (3.7)$$

$$C = \frac{\Delta V}{\Delta P} = \frac{1}{E_d} \quad (3.8)$$

Figur 3.8 viser et anden ordens lavpassystem, der kan beskrives med følgende to overføringsfunktioner:

$$T(s) = \frac{Ks}{s^2 + 2 \cdot \zeta \cdot \omega_0 \cdot s + \omega_0^2} \quad (3.9)$$

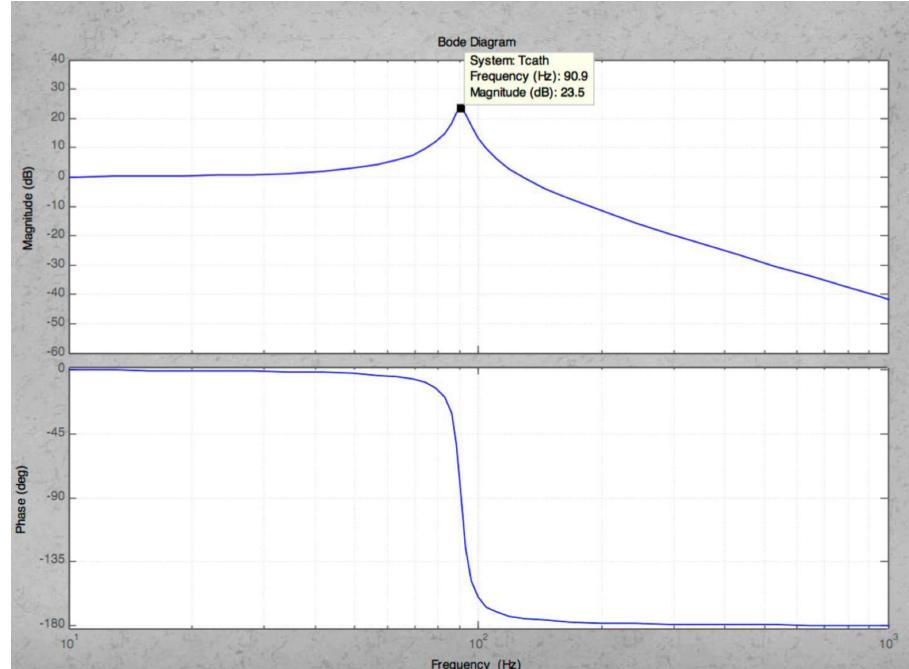
$$T(s) = \frac{k}{s^2 + \frac{\omega_0}{Q} \cdot s + \omega_0^2} \quad (3.10)$$

Ud fra disse overføringsfunktioner kan cutoff frekvensen, f_0 , og flowet, Q, findes:

$$f_0 = \frac{r}{2} \cdot \sqrt{\frac{1}{l \cdot p \cdot \pi} \cdot \frac{\Delta P}{\Delta V}} \quad (3.11)$$

$$Q = \frac{r^3}{8 \cdot \mu} \cdot \sqrt{\frac{\rho \cdot \pi}{l} \cdot \frac{\Delta P}{\Delta V}} \quad (3.12)$$

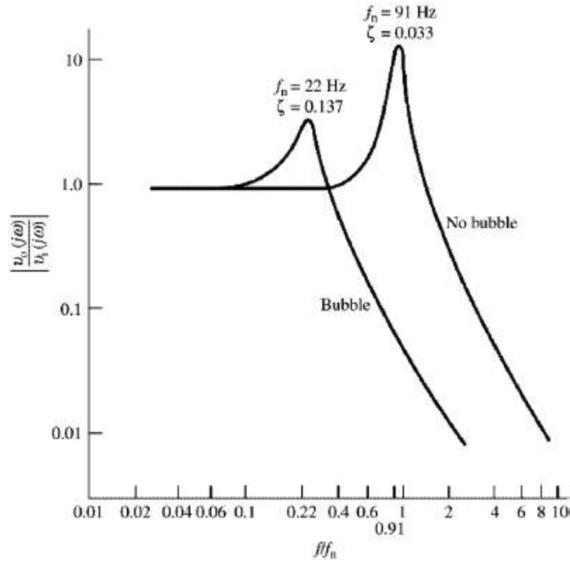
Figur 3.9 viser et bodeplot for amplitude og fasen for lavpasfilteret.



Figur 3.9: Bodeplot, [5, slide 18]

Det interessante er, at cutoff frekvens ligger ved 90 Hz. Dette betyder, at systemet medtager frekvenser op til de 90 Hz, hvilket er mere end, hvad systemet skal kunne klare i forhold til en blodtryksmåling, hvis frekvenser ikke kommer over 50 Hz.

På Figur 3.10 kan man se, hvorledes bobler i systemet påvirker systemets cutoff frekvens og dermed dens båndbredde og dæmpningsfaktor.



Figur 3.10: Diagram over cutoff frekvensen og dæmpningsfaktoren med og uden påvirkning af bobler, [1, s. 306]

Hvis der er bobler tilstede i kateteret, bliver der samtidig tilført ekstra compliance. Dette medvirker til, at cutoff frekvensen forekommer tidligere, hvilket betyder, at båndbredden bliver mindre. Som det ses i Figur 3.10, er cutoff frekvensen mindre og dæmpningsfaktoren større, når der er bobler tilstede i kateteret.

En mindre cutoff frekvens forårsager, at systemet kun tillader udæmpet passage for frekvenser fra 0 til 22 Hz, hvilket ikke er tilstrækkeligt til at repræsentere en optimal blodtryksmåling. Dæmpningsfaktoren øges, når compliance stiger. Det vil sige, at jo større dæmpningsfaktor og compliance, jo mindre quality, jo større tab over system. Det ses ved ligningen for dæmpningsfaktoren:

$$\zeta = \frac{1}{2 \cdot Q} \quad (3.13)$$

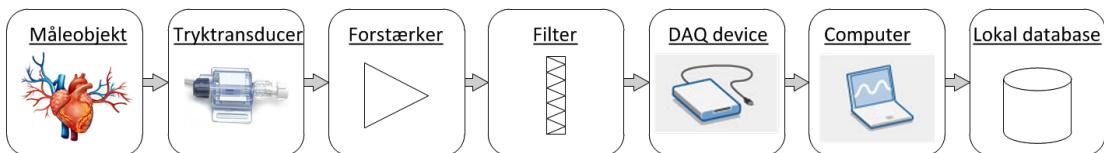
For at undgå bobler i væsken, kan man benytte føromtalte flushsystem, hvor man ved hjælp af trykposen kan erstatte luftbobler med natriumklorid.

Referancer

For baggrundsafsnittet er der blevet benyttet følgende referencer, [1], [2], [3], [4], [5], [6],

Systembeskrivelse 4

Det udviklede system består af en biomedicinsk måleopstilling med en hardware- og tilhørende softwaredel. Systemet er en invasiv blodtryksmåler udviklet til forskning. Blodtryksmålesystemet skal kunne tilsluttes et måleobjekt og kontinuerligt monitorere blodtrykket. På Figur 4.1 ses illustrationen af systemets opsætning.



Figur 4.1: Illustration af systemet

Den udviklede hardwaredel er opbygget som et elektrisk kredsløb bestående af en forstærker og et filter, der behandler signalet. Det tryk, der ønskes behandlet kommer fra et Måleobjekt og transformeres via en Tryktransducer. Tryktransduceren har til formål at konvertere Måleobjektets fysiske tryk til et analogt signal. Efter konverteringen fra tryk til et analogt signal forstærkes signalet i forstærkeren. Forstærkningen er nødvendig, da Tryktransducerens spændingssignal er meget svagt, og skal derfor forstærkes op til en spænding, der udnytter DAQ’ens måleområde bedst muligt. Derudover skal signalet filterses i det indbyggede analoge filter, hvor signalet frasorterer frekvenser, der er højere end 50 Hz, da disse frekvenser er irrelavante for blodtrykssignalet. Når signalet har passeret Forstærker- og Filterblokken, konverteres det i DAQ’en fra det analoge signal til et digitalt signal, hvorefter signalet kan behandles af softwaren.

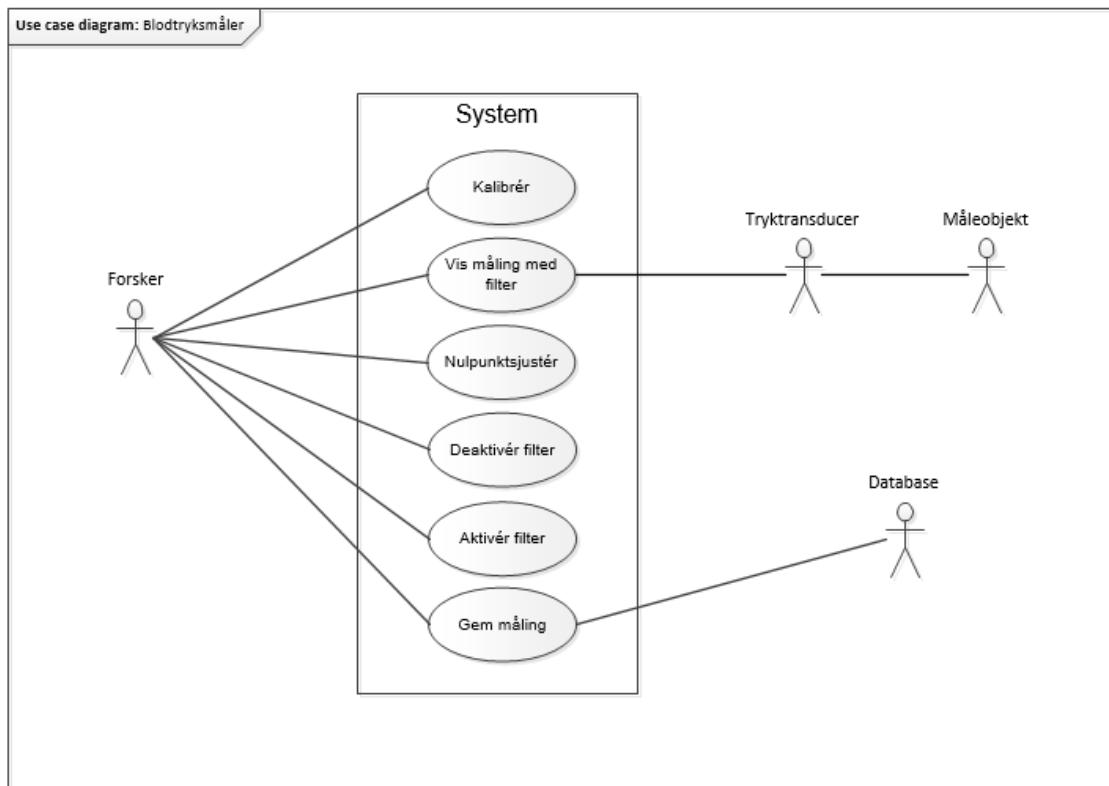
Blodtryksmålesystemets softwaredel er udviklet til at monitorere blodtrykssignalet grafisk som funktion af tiden. Softwaren er programmeret i Visual Studio C#, og er udviklet til at være anvendelig i forskningssammenhænge. Brugergrænsefladen skal grafisk vise blodtrykssignalet kontinuert samt opdatere og udskrive de hæmodynamiske data. Systemet skal ligeledes kunne kalibreres, nulpunktsjusteres, anvende digitalt filter samt gemme målinger på en lokal database.

Krav 5

Dette afsnit har til formål at beskrive hvilke krav, der er til systemet og for at skabe et overordnet overblik.

Med udgangspunkt i projektformuleringen er der udarbejdet seks Use Cases, som repræsenterer de funktionelle krav for systemet. Hver Use Case dækker over et givet krav og beskriver Forskerens interaktion med systemet. Samlet dækker de seks Use Cases over systemets funktionalitet.

Systemets primære funktionalitet beskrives ved Use Case 2, Use Case 6 samt enkelte ikke-funktionelle krav, der tydeliggør muligheden for at monitorere et kontinuerligt blodtrykssignal og gemme en måling til efterfølgende forskning.



Figur 5.1: Use Case diagram

På Figur 5.1 ses de funktionelle krav beskrevet ud fra et Use Case diagram, der samtidig viser, hvilke aktører, der har betydning for hvilke Use Cases. Aktører beskrives herunder:

Forsker er den primære aktør for samtlige seks Use Cases og styrer dermed programmet og dets funktioner.

Måleobjekt er sekundær aktør i Use Case 2 og genererer trykket, der måles på. Måleobjektet kan enten realiseres in vivo eller in vitro.

Tryktransducer er sekundær aktør i Use Case 2 og fungerer som et bindeled mellem de to aktører, Forsker og Måleobjekt. Tryktransduceren sørger for, at konvertere det målte tryk fra Måleobjektet om til et elektrisk analogt signal.

Database er sekundær aktør i Use Case 6 og er ansvarlig for, at de gemte blodtryksmålinger kan opbevares.

Nedenfor ses en kort beskrivelse af de seks Use Cases og aktørernes roller i disse. For yderligere detaljer af Use Cases henvises til kravspecifikationen under Funktionelle krav 1.3 i Projektdokumentationen.

Use Case 1: Kalibrér

Forskeren skal kunne kalibrere systemet via en grafisk brugergrænseflade.

Use Case 2: Vis måling med filter

Forskeren skal kunne se en kontinuerlig blodtrykskurve med aktivt filter for et givet Måleobjekt på en grafisk brugergrænseflade.

Use Case 3: Nulpunktsjustér

Forskeren skal kunne nulpunktsjustere systemet via en grafisk brugergrænseflade.

Use Case 4: Deaktivér filter

Forskeren skal kunne deaktivere det digitale filter og udskrive det ufiltrerede blodtrykssignal på en grafisk brugergrænseflade.

Use Case 5: Aktivér filter

Forskeren skal kunne aktivere det digitale filter og udskrive det filtrerede blodtrykssignal på en grafisk brugergrænseflade.

Use Case 6: Gem måling

Forskeren skal kunne optage blodtryksmålinger og derefter gemme dem i en Database.

Udover de funktionelle krav er der ved hjælp af FURPS+ opstillet en række ikke-funktionelle krav med henblik på at klarlægge design og kvalitetsgrad for systemet.

Herunder bliver der beskrevet krav til blandt andet design af den grafiske brugergrænseflade, systemets præstation og diverse krav til udstyr og generel udvikling af systemet.

Det er gældende for hele den grafiske brugergrænseflade, at alle knapper skal være selv-

forklarende i forhold til den funktion, de varetager. Det gælder blandt andet knapperne ”Beregn”, ”Nulpunktsjuster”, ”Rec”, ”Stop” og ”Gem”.

Ydermere er der forsøgt, at overholde diverse standarder for design af en grafisk brugergrænseflade, hvad angår placering, størrelse og farve af knapper, værdier, grafer med videre. For yderligere detaljer af ikke-funktionelle krav henvises der til kravspecifikationen under Ikke-funktionelle krav 1.4 i Projektdokumentationen.

Projektbeskrivelse 6

6.1 Projektgennemførelse

Ud fra den givne projektformulering og tilgang til projektet er der udformet en tidsplan, som indeholder de overordnede deadlines for review og tests givet fra AU.

Uger	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
P = Planlagt Arbejde															
R = Review															
✓ = Udført arbejde															
Projektopstart	✓						(✓)								
Kravsspecifikation		P	P/R	R			(✓)							✓	
Accepttest		P	P/R	R			(✓)							✓	
Design							P	P	P/R	R/(✓)				✓	
Udarbejdelse af Software		P	P	P			P	P	P	P	P	P	P/(✓)	✓	
Udarbejdelse af Hardware		P	P	P		P/(✓)								✓	
Rapportskrivning								P	P		P	P	P/(✓)		✓
Korrekturlæsning														P	P/✓
Accepttest med vejleder														✓	
Rapport aflevering															✓

Figur 6.1: Overordnet tidsplan

Udviklingsprocessen Scrum er blevet brugt til at nedbryde opgaven i delelementer, hvor de vigtigste delopgaver er prioriteret først. Via Scrum inddeltes arbejdet i sprints, hvilket tidsplanen ligeledes er blevet. Sprints er forskellige faser, som eksempelvis projektopstart, kravsspecifikation, accepttest, design osv.

I tidsplanen er der markeret et P for det planlagte arbejde, hvor reviews er markeret med bogstavet R. Derudover er der markeret flueben i tidsplanen for udført arbejde. Flueben i parentes repræsenterer, hvornår arbejdet skulle have været færdigt. I uge 42, 47 og 48 var der ikke planlagt arbejde grundet efterårsferie og eksamensforberedelse.

6.1.1 Deadlines

Der er fra projektets start blevet stillet deadlines til forskellige dele af projektet.

Dato	Deadlines
02.10.2015	Aflevering af kravsspecifikation og accepttest til review-gruppen
09.10.2015	Review af kravsspecifikation og accepttest færdiggjort med review-gruppen
06.11.2015	Aflevering af design til review-gruppen

13.11.2015 Review af design færdiggjort med review-gruppen

11.12.2015 Accepttest med vejleder

16.12.2015 Aflevering af projekt

Tabel 6.1: Deadlines

De første to deadlines i Tabel 6.1 har omhandlet kravspecifikation og accepttest, hvor kravspecifikationen er blevet udarbejdet først efterfulgt af accepttesten. Derudover har der været deadlines til design, samt accepttest med vejleder og projektaflevering.

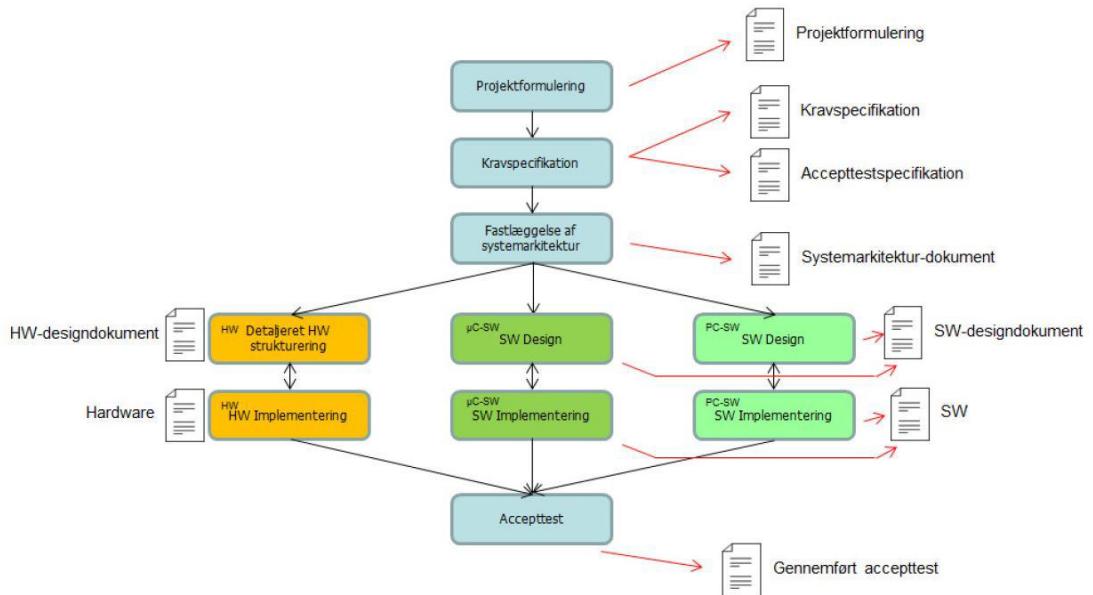
6.1.2 Mødestruktur

Der er blevet fastlagt mødestruktur ved projektets start, hvor alle gruppens medlemmer har udarbejdet en samarbejdsaftale. Der er blevet afholdt et ugentligt møde med vejleder efterfulgt af gruppermøde hver onsdag. Derudover er der blevet holdt gruppearbejde/gruppermøde efter behov. Hvert møde er blevet indledt med en opsamling af projektet og en dagsorden for mødet. Til kommunikation omkring møder, er der benyttet Facebook og en fælles kalender over iCloud.

Der er blevet ført mødereferat for samtlige møder med vejleder samt logbog ved møder gruppemedlemmerne imellem.

6.1.3 ASE-modellen

Projektets udviklingsproces har taget udgangspunkt i ASE-modellen, som ses på Figur 6.2. Denne afspejles desuden også i tidsplanen på Figur 6.1.



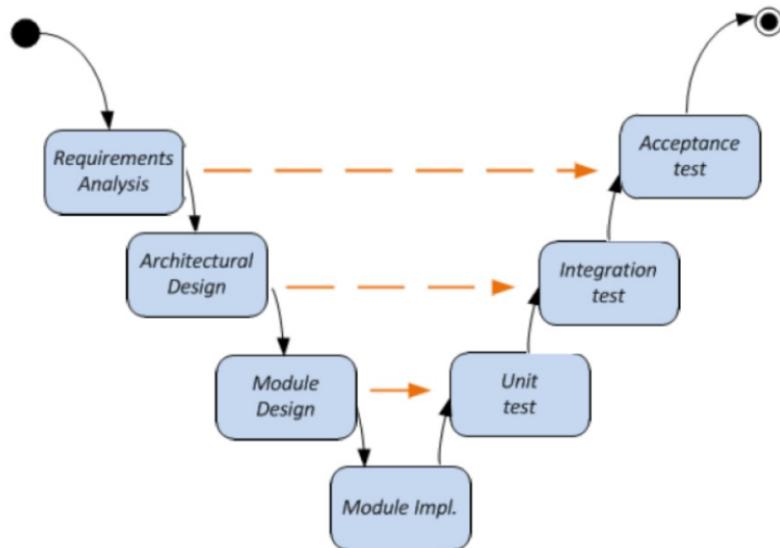
Figur 6.2: Udviklingsmodel: ASE-model [7, s. 6]

ASE-modellen er en udviklingsmodel, der tager udgangspunkt i Use Cases, som defineres i kravspecifikationen i starten af projektarbejdet. ASE-modellen er inspireret af vandfalgsmodellen, hvor projektarbejdet opdeles i faser. Der fastlægges en opgaveformulering, kravspecifikation og systemarkitektur, før derefter at designe, implementere og teste de enkelte moduler i iterationer. Ud fra kravspecifikationen bliver systemets accepttest udarbejdet. Efter kravspecifikationen er fastlagt, udarbejdes systemarkitekturen. Systemarkitekturen designer systemet ved at nedbryde det efter funktionalitet, som kan bindes til både software og hardware.

De første step i udviklingen af projektet er blevet udarbejdet af alle gruppens medlemmer. Alle har bidraget til projektformuleringen, kravspecifikationen og accepttesten. I begyndelsen var der primært fokus på at få dette færdiggjort, men der blev allerede her arbejdet på hardwaren. I starten af projektarbejdet var det udtænkt, at alle gruppens medlemmer skulle arbejde med alle områder i projektet. Dette kunne ikke udføres rent tidsmæssigt, og derfor var en opdeling af arbejdsopgaver i mindre grupper nødvendig. Projektarbejdet blev delt op i software- og hardwarehold, dog var en del af hardwaren allerede beregnet og implementeret, da denne opdeling fandt sted. Tilbage af hardwaren var at implementere det på et veroboard og derefter teste det.

6.1.4 V-modellen

V-modellen er en udviklingsmodel opdelt i forskellige faser, der beskriver udviklingsfaserne og testfaserne i projektet sideløbende. Denne model er blevet benyttet sideløbende med ASE-modellen, og fungerer således, at specifikationen af tests foregår sideløbende med udviklingen af selve systemet. Hver fase skal færdiggøres inden næste fase påbegyndes, hvilket også var tiltænkt i projektet. Dette blev dog ikke helt opfyldt i projektet, da der ofte blev rettet i tidlige faser, for at opnå en overenstemmelse med det færdige ønskede produkt.

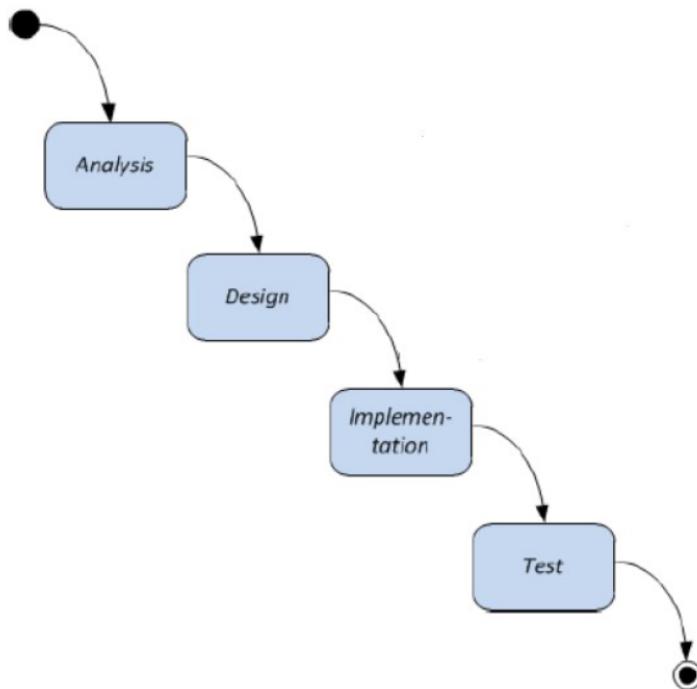


Figur 6.3: V-modellen, [7, s. 4], [8]

På ovenstående figur ses V-modellen, hvor den første fase er udviklingen af kravspecifikationen. Hertil udvikles en tilhørende accepttest, som gør det muligt, at tjekke om systemet opfylder de stillede krav. Næste fase er systemarkitekturen, hvor der ligeledes udvikles en tilhørende test. Testen skal undersøge integrationen mellem de implementerede moduler. De to sidste faser er design og implementering, som der udføres løbende tests af.

6.1.5 Vandfaldsmodellen

Vandfaldsmodellen er en model, der ofte benyttes for udviklingen af software. Udviklingen af software foregår på den måde, at en ny fase af modulen først påbegyndes, når den foranliggende fase er færdiggjort, som kan ses i Figur 6.4. nedenfor.



Figur 6.4: Vandfaldsmodellen, [8]

De tre udviklingsmodeller hænger alle sammen på den måde, at der arbejdes i kronologisk rækkefølge. De har suppleret hinanden alle tre, hvor ASE-modellen har skabt det overordnede overblik. Vandfaldsmodellen er den udviklingsproces, som der er blevet arbejdet efter i alle projektets facetter. V-modellen har sikret, at de nødvendige tests af systemet og de enkelte moduler har fundet sted.

6.2 Metode

Dette afsnit har til formål at beskrive hvilke metoder og arbejdsredskaber, der er blevet benyttet til udarbejdelsen af software, hardware, rapport og dokumentation for projektet.

Der er blevet anvendt metoder og viden fra forskellige kurser over 1.- 2.- og 3.semester.

Fra kurset ST3KVI er der blevet benyttet viden om blodtryksmåling og transducerprincippet. Fra kurserne ST2ITS2 og ST3ITS3 har viden om databasestruktur, principperne om trelagsmodellen og publisher/subscriber pattern været grundlaget for opbygningen af softwaren. Et krav til projektet var, at der skulle kunne aktiveres og deaktiveres et digitalt filter. Til at designe dette filter blev der anvendt viden fra kurset E3DSB. Kurset ST1SUN1 har bidraget med viden om blodtryk og den anatomiske opbygning af hjertet. Til beregning af komponentværdier, design og implementering af hardwaren, er der blevet benyttet metoder og viden fra kurset E2ASB. Principperne om design diagrammer fra kurset I2ISE har bidraget til designafsnittet for hardware og software i dokumentationen. Alle kurserne har tilsammen været nødvendige for, at kunne udvikle Blodtryksmålesystemet.

Ved hjælp af SysML er software- og hardwaredesign blevet specifieret. Dette er en metode, der anvender forskellige diagrammer til at beskrive opbygning og kommunikation for både software og hardware. Selve systemet vil også blive beskrevet ved brug af SysML. Hensigten er at give læseren det store overblik over hvilke aktører, der interagerer med systemet samt hvilken funktionalitet, der tillægges det.

Softwareen er også beskrevet ved hjælp af UML - helt specifikt ved et UML klassediagram. Klassediagrammet viser, hvilke klasser og metoder softwaren består af, samt hvordan systemet er bygget op efter trelagsmodellen.

For at forstå det grundlæggende om blodtryk og blodtryksmåling, er der blevet anvendt en redegørende metode, hvor der er blevet indsamlet viden gennem læsning af hjemmesider og sundheds- og tekniskfaglige bøger. Ud fra den viden har det været muligt, at redegøre for principperne, samt at analysere de resultater systemet har givet.

Efter hardwarens funktionalitet var bestemt, blev den matematiske metode benyttet til at beregne de nødvendige komponentværdier, der skulle til for at realisere den ønskede hardware.

Arbejdsredskaber

Af benyttede arbejdsredskaber er der først og fremmest brugt en fælles arbejdsplatform, GitHub. GitHub er en online filopbevaringsplatform, hvor der er mulighed for at foretage ændringer samtidigt, og gemme i en fælles mappe. Yderligere er der mulighed for en detaljeret versionshistorik.

Alle SysML- og UML-diagrammer er udarbejdet i programmet Visio. Koden er skrevet i sproget C# i programmet Visual Studio. Visual Studio spiller også sammen med programmet WaveForms Generator, i forbindelse med simulering af blodtrykssignalet. Selve rapporten, mødereferater, logbog og dokumentationen er udformet i tekstoprogrammet LaTeX. MatLab, som er et matematik- og signalbehandlingsprogram, er blevet benyttet til at udarbejdelse af hardwaren. Yderligere er Facebook og en fælles iCloud kalender blevet brugt til mødeindkaldelse og generel kommunikation.

6.3 Specifikation og analyse

Dette afsnit har til formål at beskrive de løsninger, der benyttes i forhold til de valgte hardware og software specifikationer.

6.3.1 Hardware

Den udviklede hardware, som omtales Signalbehandlingsblok, består af en Forstærker og et Filter. Forstærkeren har til opgave at forstærke det elektriske signal, som Tryktransduceren har transformeret ud fra en given trykændring. Filterets opgave er at filtrere unødige frekvenser fra for, at få det mest optimale signal.

Tryktransducer

Eksitationsspændingen for Forstærkeren og Filteret leveres af to 9 V's batterier. Der er valgt, at sætte en 5 V's regulator i forbindelse med Tryktransduceren for at sikre, at strømmen gennem Wheatstone broen ikke bliver for stor. En for stor strøm kan give anledning til, at Wheatstone broen vil drive og eventuelt brande de fire strain gauges af.

For at kunne beregne de tilstrækkelige komponentværdier for Forstærkeren og Filteret bestemmes en maksimal trykændring, den udviklede hardware skal kunne klare. Den maksimale trykændring for dette system er sat til at være i intervallet 0-300 mmHg. Da det er blodtryk, systemet skal kunne monitorere, er denne trykændring tilstrækkelig, da der ikke forventes, at et blodtryk vil komme over 300 mmHg.

Forstærker

Tryktransduceren vil transformere et tryk på 300 mmHg om til 7,5 mV - se Projektdokumentationen under grænseflader for udregningen (2.2). Denne spænding er forholdsvis lille, og vil ikke udnytte DAQ'ens måleområde optimalt, som er valgt til +/- 5 V. For at få den bedste konvertering af det analoge signal til det digitale signal, skal spændingen forstærkes op til 5 V.

Til dette benyttes den ikke-inverterende operationsforstærker, INA114 samt et potentiometer, der fungere som gain-modstanden. Gain-modstanden er en variabel modstand, der kan variere i forhold til, hvilken forstærkning, der ønskes. Udregningen for forstærknin-gen ses i Projektdokumentationen under Hardware arkitektur 2.3 under ligningerne (2.4) og (2.5). Udregningen for gain-modstanden kan ses i Projektdokumentationen under HW implementering og test 3 under lancing (3.3).

Filteret

Det forstærkerede analoge signal skal videre filtreres, før det konverteres til det digitale signal. De relevante frekvenser, der udgør en blodtryksmåling, ligger mellem 0-50 Hz [1, s. 10]. Derfor ønskes alle frekvenser over 50 Hz dæmpet.

Til dette designes et Sallen Key anden ordens lavpasfilter med unity gain, hvor cutoff

frekvensen er 50 Hz. Udregningerne for komponentværdierne til denne realisering, ses i HW implementering og test 3 i Projektdokumentationen under Filterblok 3.2.

6.3.2 Software

Struktur

Beslutningen om at anvende trelagsmodellen blev overvejende taget på baggrund af de krav, der var stillet til projektet om, at der skulle ske en dataopsamling, databehandling og visualisering. De tre lag repræsenterer hver især disse opgaver. For at overskueliggøre og effektivisere softwaren var en vigtig overvejelse navngivning af klasser og metoder efter deres pågældende funktion. Dette skaber høj samhørighed og lav kobling klasserne imellem. Datalaget er et eksempel herpå, med en opdeling i tre klasser, med hver deres specifikke funktion (DataHent, DataGem, DAQ). På denne måde er fejlfinding overskueligt, da fejlen sker i en indkapslet funktion. Se i Projektdokumentationen under SW implementering og test 4 under Figur 4.2.

Dataindhentning

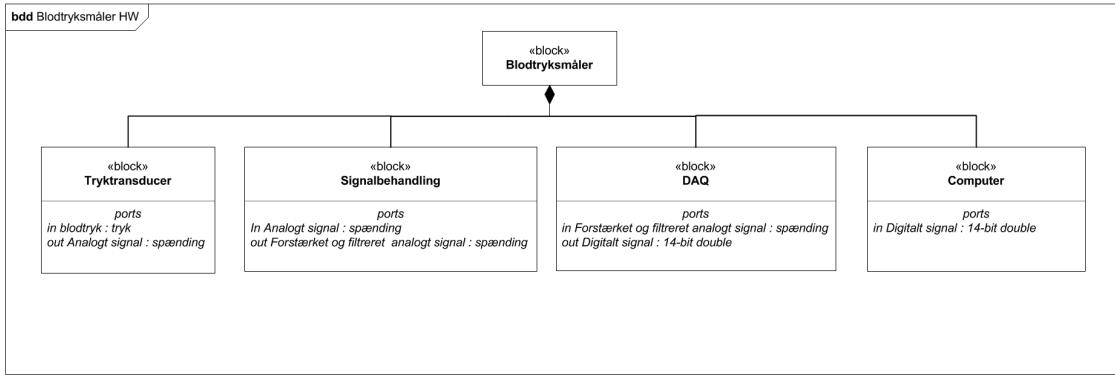
I overvejelserne omkring udskrivning af målt blodtryk i Monitor-vinduet var en prioritet, at signalet blev udskrevet rettidigt, altså med en minimal forsinkelse fra, at målingen var foretaget af den fysiske DAQ. I den forbindelse var det muligt enten at requeste data fra logiklaget jf. trelagsmodellen eller at anvende subscriber/publisher pattern med DAQ klassen som publisher. Ved den første metode ville behandling af måledata være styret eksternt fra, hvor data blev produceret, altså fra logiklaget. Med den anden mulighed ville databehandlingen være styret af den fysiske DAQ. Dette var at foretrække, da data ville blive behandlet som det blev produceret, altså mest rettidigt.

6.4 Arkitektur

I følgende afsnit beskrives systemarkitekturen for Blodtryksmålesystemet. Systemarkitekturen fungerer som en udviklingsramme for videreudviklingen af design og implementering. Her bliver systemets funktionalitet nedbrudt til overordnede moduler. Gennem dette afsnit ønskes der at skabe et overordnet overblik over systemet. Der benyttes diagrammer med tilhørende beskrivelser til at specificere og klarlægge systemkrav.

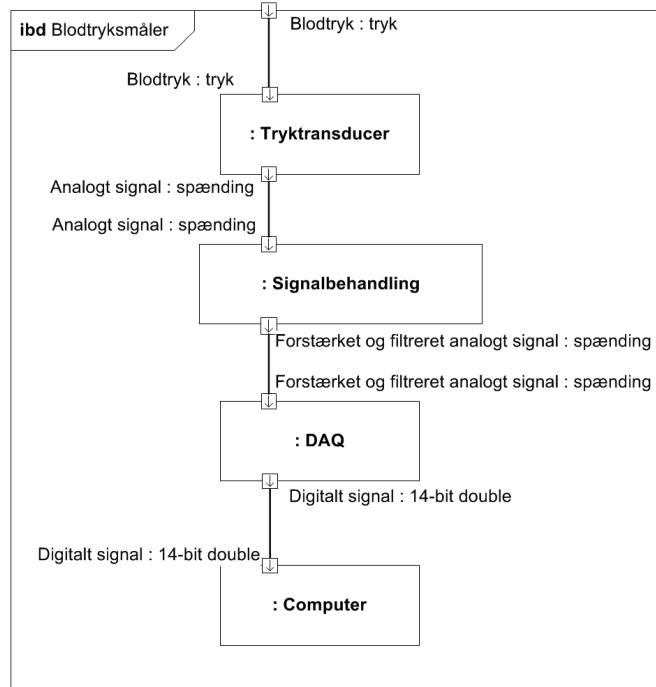
6.4.1 Hardwarearkitektur

Systemets hardware kan illustreres i et bdd og ibd, der beskriver det overordnede system, og hvordan de forskellige hardwareblokke interagerer med hinanden. På Figur 6.5 ses et bdd, der viser Blodtryksmålesystemet bestående af fire hardwareblokke: Tryktransducer, Signalbehandling, DAQ og Computer. Hertil ses også hvilke porte, blokkene består af.



Figur 6.5: bdd af blodtrykmålesystemet

Tryktransduceren registrerer en fysisk størrelse i form af en trykændring. Dennes opgave er at transformere det fysiske tryk til en elektrisk spænding, som derefter viderebehandles. Viderebehandlingen foregår i Signalbehandlingsblokken, som består af to dele: en Forstærker og et Filter. Her bliver det elektriske signal forstærket og filtreret således, at det er klar til at blive konverteret i DAQ'en fra et analogt til et digitalt signal. Computeren indeholder software til systemet, som kan vise det digitale signal grafisk, samt kalibrere, nulpunktjustere og gemme målinger. Derudover kan software aktivere og deaktivere filteret. Bdd benyttes til at definere relationen mellem de forskellige blokke, som i dette tilfælde er en komposition.



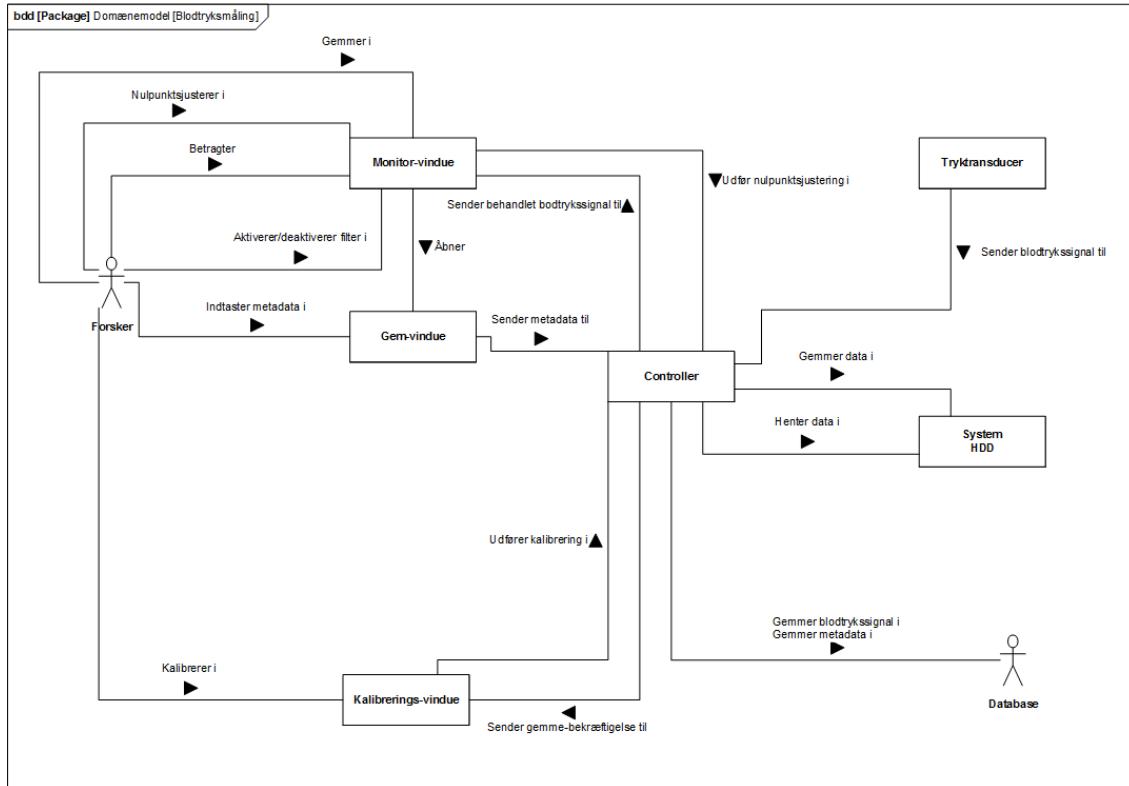
Figur 6.6: ibd af blodtrykmålesystemet

På Figur 6.6 ses et ibd for systemet, der viser signalets behandling gennem systemet. Signalet transformeres fra et målt fysisk tryk til et digitalt signal, som softwaren kan

viderebehandle og vise grafisk. For yderligere informationer om grænseflader henvises der til Projektdokumentationen under Design 2.2.

6.4.2 Softwarearkitektur

Systemets softwaredesign beskrives i dette afsnit via domænemodellen, som ses på Figur 6.7. Domænemodellen er skabt på baggrund af kravspecifikationens seks Use Cases, og viser et samlet overblik over systemet.



Figur 6.7: Domænemodel af blodtrykmålesystemet

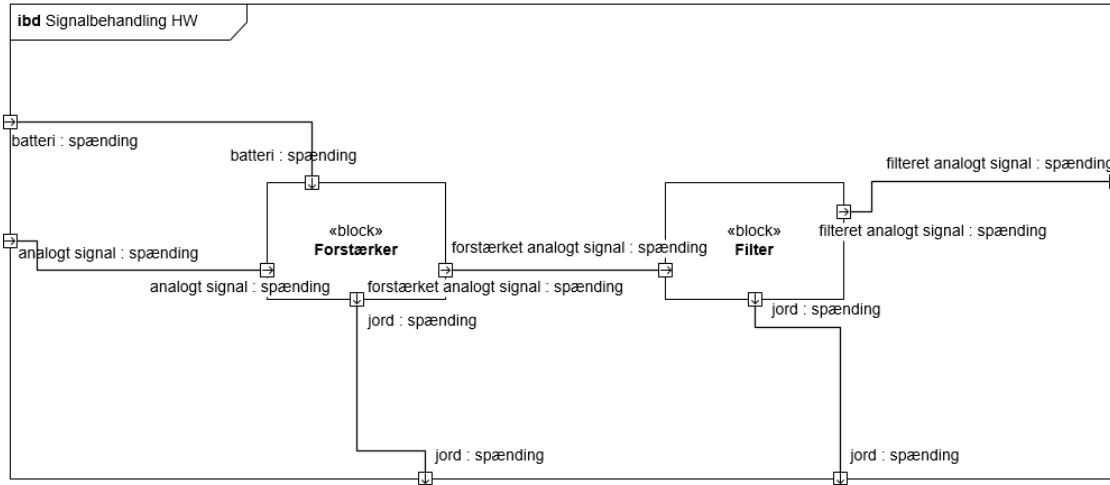
I domænemodellen ses interaktionen mellem de konceptuelle klasser og aktører. Controlleren udfører kommandoer og sørger for, at systemet fungerer optimalt. Funktionaliteten af den enkelte kommando kan variere fra Use Case til Use Case. Domænemodellen viser desuden Forskerens interaktion med systemet. Forsker udfører en handling, der medfører igangsættelse af en række processer i systemet. For yderligere information om forløbene i de seks Use Cases henvises der til Projektdokumentationen under design 2.4.2.

6.5 Design, implementering og test

I følgende afsnit vil design- og implementeringsprocessen for den udviklede hardware og software beskrives. Der vil være en beskrivelse af, hvordan de valgte løsninger er blevet testet.

6.5.1 Hardware

Under afsnittet Specifikation og analyse 6.3.1 står specifikationer for Signalbehandlingsblokken beskrevet. Signalbehandlingsblokken består af en Forstærker og et Filter. Kommunikationen internt for Signalbehandlingsblokken, samt kommunikationen ud ad til, ses i et ibd for Signalbehandlingsblokken i Figur 6.8.



Figur 6.8: idb for Signalbehandlingsblokken

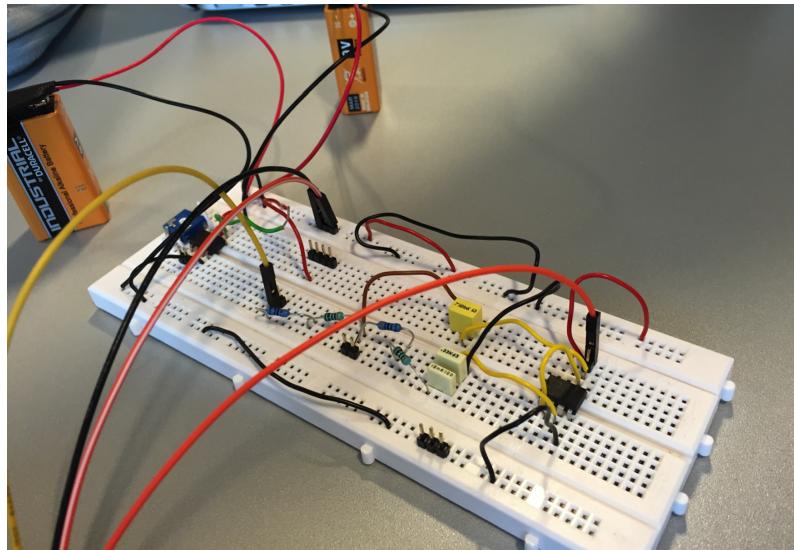
Til realisering af specifikationen for Forstærkeren er der blevet beregnet den ønskede forstærkning og gain-modstand, så et tryk på 300 mmHg forstærkes op til 5 V.

Udregningen for forstærkningen ses i Projektdokumentationen under Hardware arkitektur 2.3 under ligningerne (2.4) og (2.5). Udregningen for gain-modstanden kan ses i Projektdokumentationen under HW implementering og test 3 under lancing (3.3).

Til realisering af specifikationen for Filteret er der blevet beregnet de forskellige komponentværdier, så filteret får en cutoff frekvens ved 50 Hz.

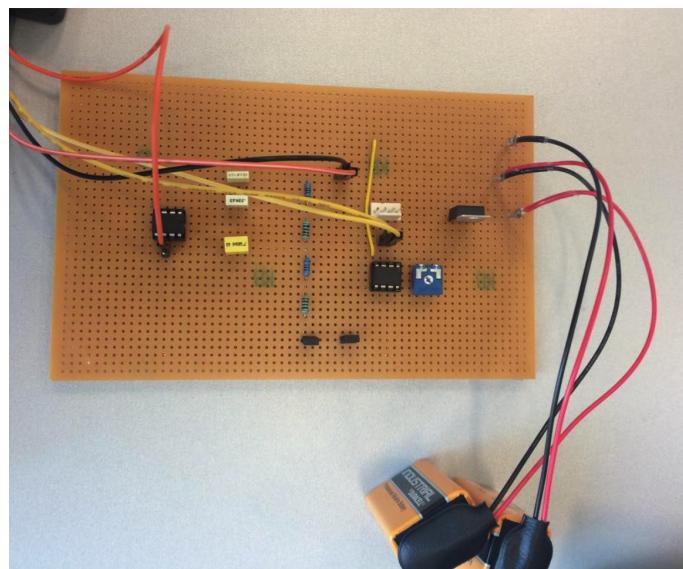
Udregninger for komponentværdierne ses under HW implementering og test 3 i Projektdokumentationen under Filterblok 3.2.

Signalbehandlingsblokken blev i starten af projektet implementeret på et fumlebræt. På Figur 6.9 ses denne opstillingen.



Figur 6.9: Signalbehandlingsblokken implementeret på et fulmebræt

Senere i implementeringsprocessen blevet Signalbehandlingsblokken implementeret på et veroboard. Dette blev valgt grundet bedre stabilitet og udseende. På Figur 6.10 ses veroboardet. Det var tiltænkt, at Signalbehandlingsblokken skulle have været implementeret på en printboard, hvor forbindelsen mellem komponenterne er lavet på forhånd. Printboardets design skulle laves digitalt - først i programmet Multisim og videre i programmet Ultiboard. Dette blev ikke realiseret på grund af mangel på tid.



Figur 6.10: Signalbehandlingsblokken implementeret på et veroboard

Signalbehandlingsblokken er blevet testet via Analog Discovery & Waveforms. Først blev der lavet unittest, hvor Forstærkeren og Filteret blev testet hver for sig, og derefter blev der lavet en integrationstest, hvor Signalbehandlingsblokken blev testet som en enhed.

Specifikationen for forstærkeren er, at den skal forstærke en spænding på 7,5 mV, der svarer til et tryk på 300 mmHg, op til 5 V.

Signalgeneratoren leverer inputspændingen, hvor oscilloskopet måler outputspændingen. Inputspændingen kan via Waveforms bestemmes. Til denne unittest af Forstærkeren sættes inputspændingen til 7 mV. Reelt burde den sættes til 7,5 mV, men dette er ikke muligt. Den forventede outputspænding ved en inputspænding på 7 mV, er 4,7 V - se Projektdokumentationen under Test af forstærkerblok 3.2.1 for udregningen, samt dokumentationen af denne test. Resultatet af testen stemte overens med det forventede.

Specifikationen for filteret er, at den skal dæmpe frekvenser over 50 Hz. Denne specifikation realiseres via et anden ordens lavpasfilter.

Til denne unittest af Filteret sættes inputspændingen til 5 V. For at teste om det implementerede Filter har en cutoff frekvens ved 50 Hz ændres inputspændingens frekvenser. Der laves i alt tre tests. Først testes Filteret, hvor frekvensen er 1 Hz. Her forventes det, at outputspændingen er lig med inputspændingen. Derefter testes Filteret, hvor frekvensen er 50 Hz. Her forventes det, at outputspændingen er blevet dæmpet med 3 dB. Til sidst testes Filteret, hvor frekvensen er 500 Hz. Her forventes det, at outputspændingen er blevet dæmpet med 40 dB. I Projektdokumentationen under Test af filterblok 3.2.2 og i dokumentationen af denne unittest, ses de forventede outputspændinger ved de forskellige frekvenser. Resultaterne af testen stemte overens med det forventede.

6.5.2 Software

Softwareen er designet efter trelagsmodellen samt anvendelse af publisher/subscriber mønstret.

Dette er gjort i henhold til de givne krav til systemet, som er dataopsamling, databehandling og visualisering. Trelagsmodellen sørger for at håndteringen af disse opgaver sker i hvert sit lag. Det visuelle design er lavet, så det assimilerer eksisterende blodtryksmonitorer.

Design af software er konkretiseret i sekvensdiagrammer med konceptuelle klasser. Se Projektdokumentationen under Applikationsmodel 2.4.2.

Præsentationslaget er implementeret med tre GUI vinduer: Monitor, Gem og Kalibrer. Monitor-vinduet udskriver blodtrykssignalet, giver Forskeren mulighed for at udføre nulpunktsjustering og optage det viste signal. Gem-vinduet fremviser det optagede signal til revision før gemning i Databasen. Kalibrerings-vinduet giver Forskeren mulighed for kalibrering af systemet.

Logiklaget er implementeret med en enkelt klasse, Beregninger, som udfører opgaver i specificerede metoder, såsom digital filtrering, kalibrering, nulpunktsjustering og behandling af rådata.

Datalaget er implementeret med klasser med specifikke opgaver, der sørger for lav kobling og høj samhørighed. Dataoverførslen af rådata sker jf. publisher/subscriber mønstret.

Implementeringens af software er beskrevet med sekvensdiagrammer med konkrete softwarklasser og metoder i Projektdokumentationen under Metodebeskrivelser 4.1.5.

Softwareen er testet i forhold til Use Cases. Analog Discovery & Waveform Generator er

brugt som input til systemet, og resultater er kontrolleret med debugging af kode. Se Projektdokumentationen under SW implementering og test 4.2.

6.6 Resultater og diskussion

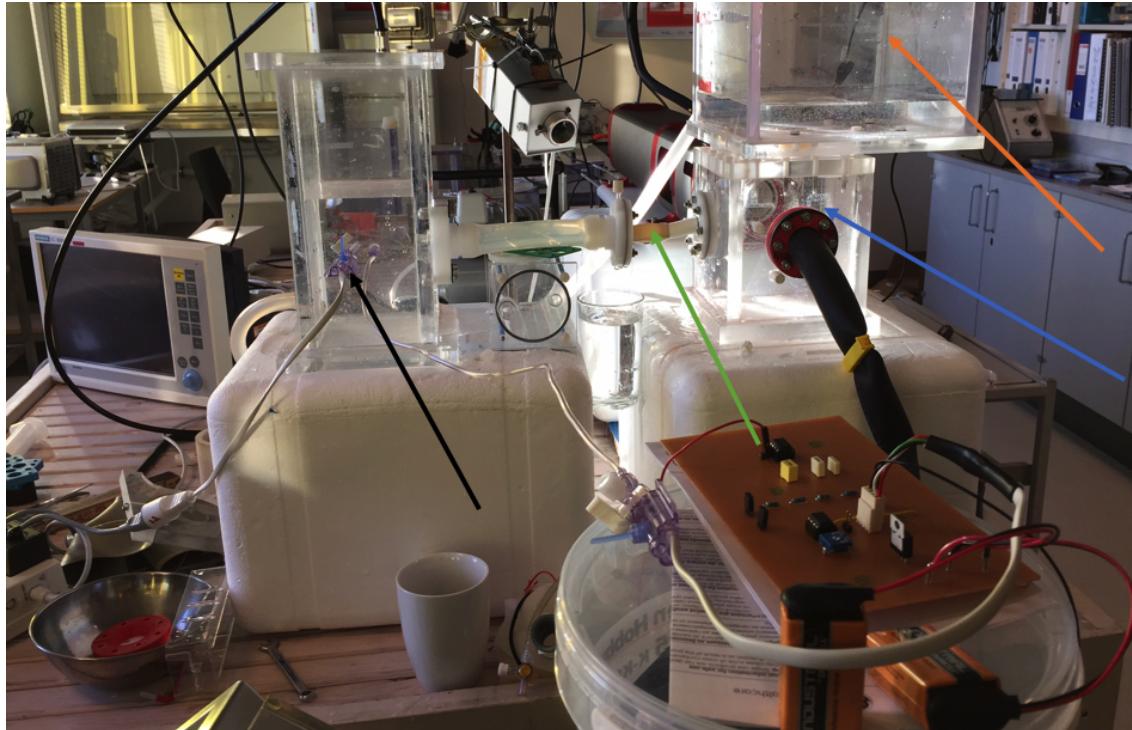
Accepttest blev den 9. december 2015 gennemført af vejleder Peter Johansen. Til denne test af Blodtryksmålesystemet blev et blodtrykssignal fra PhysioNet [10] simuleret ved hjælp af Analog Discovery & Waveforms. Accepttesten blev gennemført uden problemer, og der kunne sættes flueben ved alle funktionelle og ikke-funktionelle krav.

6.6.1 In Vitro hjertemodel

Samme dag var det muligt, at teste Blodtryksmålesystemet via en In Vitro hjertemodel. In Vitro betyder ”i glas”, hvilket er en betegnelse, man benytter, når man laver forskningsforsøg uden for den levende organisme.

Den benyttede In Vitro hjertemodel til testen af Blodtryksmålesystemet genskaber hjertets venstre sides funktion. Trykket i modellen bliver skabt via stempelpumpen ViVitro SuperDup'r. Hjertets venstre side genskabes via et ventrikulær kammer, en arteriel reservoir samt en aortarod. Mellem det ventrikulære kammer og det arterielle reservoir sidder der en mekanisk hjerteklap, der efterligner funktionaliteten af mitralklappen. I aortaroden er aortaklappen også genskabt. Filmen ”Aortaklap”, se bilag, viser, hvordan aortaklappen fungerer.

Via denne In Vitro hjertemodel kan hjertes venstre sides blodflow simuleres, hvilket gør det muligt at måle trykket efter aortaroden, der kan sammenlignes med det arterielle blodtryk.



Figur 6.11: Opstilling af In Vitro hjertemodel

På Figur 6.11 ses opstillingen for In Vitro hjertemodel. Den orange pil peger på den arterielle reservoir, den blå pil peger på det ventrikulære kammer og den grønne pil peger på aortaroden. Tryktransduceren fastsættes, hvor den sorte pil peger. På billedet ses, at der er fastsat to tryktransducerer. Den ene er tilkoblet et monitoreringssystem fra Siemens, mens den anden er tilkoblet Blodtryksmålesystemet.

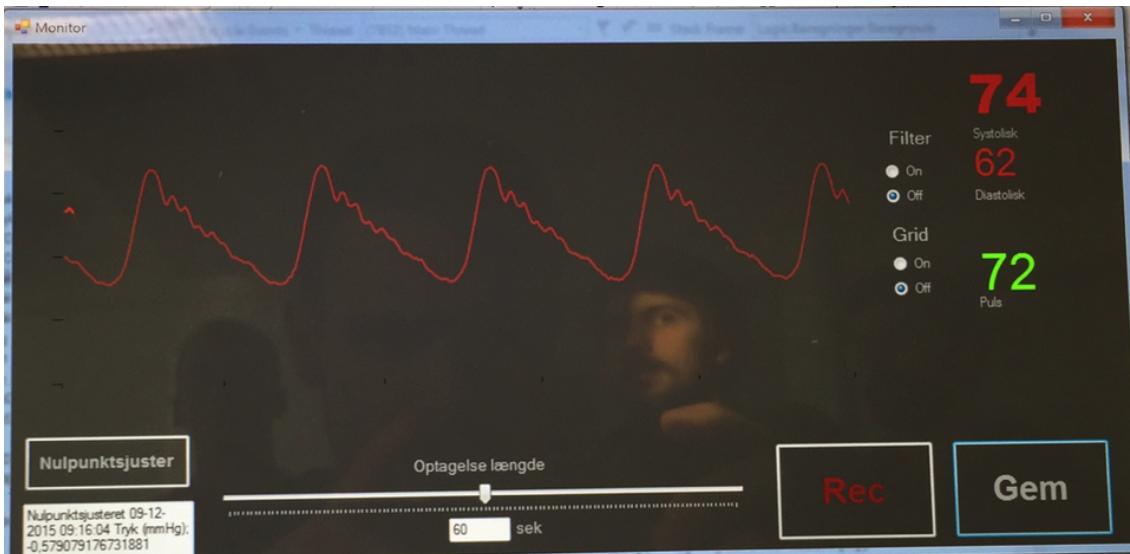
Før målingen kan foretages skal Blodtryksmålesystemet kalibreres. Til dette benyttes en vandsøjle. På vandsøjlen er tre punkter markeret, hvor trykket i mmHg er kendt. Når Blodtryksmålesystemet skal kalibreres, skal der bestemmes en kalibreringskonstant, som systemet skal korrigeres efter.

Der måles først en spænding ved eksempelvis 10 mmHg og derefter en spænding ved eksempelvis 50 mmHg. Kalibreringskonstanten beregnes ud fra følgende formel.

$$K_{konstant} = \frac{tryk_2 - tryk_1}{spnding_2 - spnding_1} \quad (6.1)$$

Efter kalibreringen er foretaget, kan målingen påbegyndes.

Resultaterne af målingen ses på Figur 6.12 og Figur 6.13. Figur 6.12 viser resultatet af Blodtryksmålesystemet, mens Figur 6.13 viser resultatet af Siemens' monitoreringssystem.



Figur 6.12: Blodtryksmålesystemets Monitor-vindue



Figur 6.13: Siemens' monitoreringssystem

Der ses, at der er en lille forskel i værdier for de to monitoreringssystemer. Årsagen hertil kan være, at billederne ikke blev taget på nøjagtig samme tid, da dette ikke var muligt. Forskellen er dog så lille, at det accepteres.

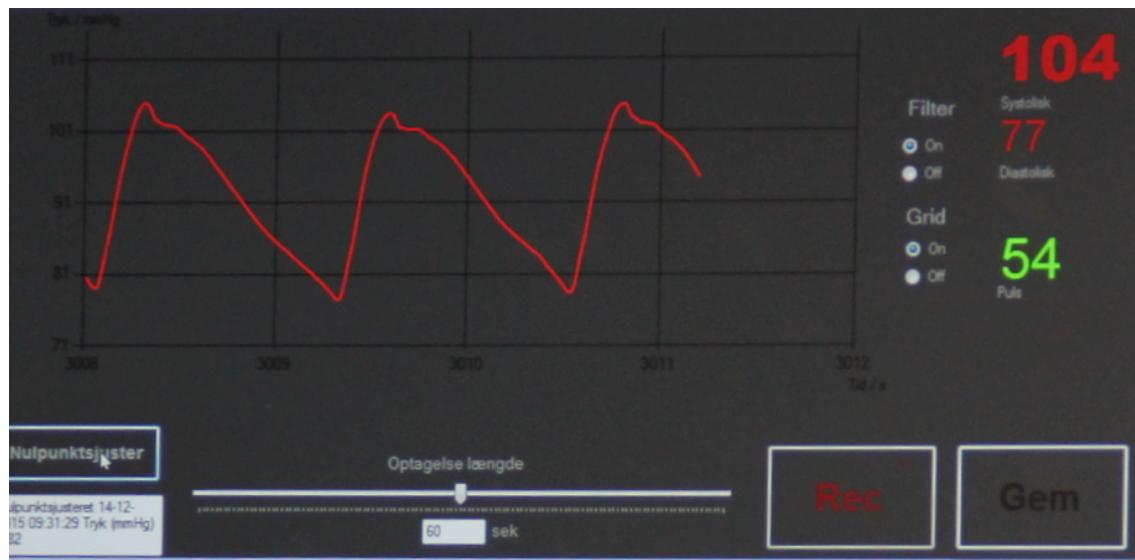
Værdierne er heller ikke realistiske i forhold til det arterielle blodtryk – systolen er meget lavere end det forventede. Årsagen til dette er opstillingen af In Vitro hjertemodel, hvor der var nogle problemer med pakningerne mellem det ventrikulære kammer og aortaroden samt at der var et lille hul i aortaroden, hvor vand ville sprøjte ud, hvis trykket fra stempelpumpen blev for stort. Filmen ”Utæt pakning mellem ventrikulær kammer og aortaroden” i bilag viser dette.

Ud fra denne test via In Vitro hjertemodel konkluderes det, at Blodtryksmålesystemet fungerer som ønsket, da de to monitoreringssystemer følges ad og viser tilnærmelsesvis samme værdier.

6.6.2 Blodtryksmåling på en gris

Den 14. december 2015 var det muligt at teste Blodtryksmålesystemet på en gris på AUH. Formålet med denne test var at undersøge, om Blodtryksmålesystemet kunne monitorere et blodtryk, der kan sammenlignes med det menneskelige.

Målingen foregik på en gris, der blev benyttet til klinisk forskning i dette tilfælde en hjerteoperation udført af medicinstuderende. Operationen foregik samtidig med monitorenningen af blodtrykket. I en arterie tæt på sternum blev et kateter påsat, som gjorde det muligt at fastsætte Tryktransduceren fra Blodtryksmålesystemet. På Figur 6.14 ses resultaterne fra målingen. I det tidsrum, hvor Blodtryksmålesystemet stod for monitorenningen, var det nødvendigt for de medicinstuderende at vide blodtryksværdierne. Disse kunne Blodtryksmålesystemet levere og medicinstuderende accepterede disse.



Figur 6.14: Vores monitoreringssystem

Efter målingen blev sygehusets allerede implementerede system påsat. På Figur 6.15 ses resultaterne fra dette monitoreringssystem.



Figur 6.15: Sygehusets monitoreringssystem

Igen ved denne test ses der en lille forskel i værdierne. Årsagen kan være, at billederne ikke blev taget på nøjagtig samme tid. Filmen "Griseforsøg", se bilag, viser blodtryksmålingen på grisen.

Ud fra denne test af Blodtryksmålesystemet konkluderes det, at vores monitoreringssystem fungere som ønsket, og kan monitorere reelle blodtryksværdier.

6.7 Opnåede erfaringer

På baggrund af den generelle viden om hjertet, blodtrykket og de forskellige måletyper er der designet og udviklet et system til at måle blodtrykket invasivt. Med udgangspunkt i problemformuleringen og opsatte rammer er der arbejdet efter vandfalldsmetoden, hvor vi undervejs i udviklingsprocessen gennem løsninger og erfaringer har opnået bedre kendskab til det ønskede produkt. Vi har styrket vores tværfaglige kompetencer og formået at udvikle en teknisk løsning samt at benytte teori i praksis.

Et væsentligt fokusområde i udviklingen af Blodtryksmålesystemet har været integreringen af hardware og software. I udarbejdelsen af hardware startede vi med at beregne komponentværdierne og derefter designe det elektriske kredsløb for at kunne opstille det på et fumlebraet. Herefter var hensigten at designe kredsløbet i Multisim, og derefter overføre det til printboard via Ultiboard. Dette blev ikke gennemført grundet tidspres. Derfor designede vi kredsløbet på et veroboard, hvor komponenterne og forbindelserne blev loddet. I selve processen med design af kredsløb opstod der problemer, da vi ved en fejl havde placeret de to kondensatorerne omvendt af hinanden, så de ikke stemte overens med teorien.

I udviklingen af det ønskede software har vi tildelt os ny viden i undervisningen i ST3ITS3, men primært har softwaren været en viderebygning på erfaringer fra 2. semesterprojektet. Processen har været uden svære komplikationer, dog blev vi bevidste om et problem i forhold til udskrivning af samples. Da vi har målt med 1000 samples i sekundet, erfarede vi at programmet ikke kunne udskrive dette. Dette løste vi ved i stedet kun at udskrive hvert 10. sample, hvilket dog indebar, at signalet ikke blev lige så præcist.

Selve projektarbejdet har i høj grad været struktureret af faste møder og dagsordener. Derudover har der været en fælles opsamling på opgaver, der har været fordelt mellem gruppemedlemmerne. På denne måde har det været muligt for alle at have et overblik over de forskellige arbejdsopgaver. Denne strukturering af projektarbejdet har styrket vores færdigheder i projektstyring og tilrettelægning af arbejdsopgaver.

6.8 Fremtidigt arbejde

Gennem udviklingsforløbet af Blodtryksmålesystemet er der blevet arbejdet ud fra de retningslinjer og obligatoriske krav, der er blevet givet fra AU's side. Første prioritet har været at udvikle et produkt, der lever op til disse krav, men samtidig har der været fokus på at skabe det bedst mulige produkt for Forskeren. Som resultat heraf er der blevet implementeret forbedringer i Blodtryksmålesystemet i form af afbildning af systolisk/diastolisk

blodtryk samt puls med tal.

Yderligere forbedringer hertil kunne være angivelse af pulsslag med svage bip-lyde, således Forskeren ikke behøver at kigge på monitoren for at få besked om Måleobjektets puls. Denne funktion vil hjælpe Forskeren til at holde overblik over Måleobjektets tilstand, samtidigt med, at der kan foretages indgreb, justeringer med videre.

Ligeledes har der været overvejelser om at udvikle en alarmfunktion, der undervejs angiver særligt høje eller lave blodtryk hos Måleobjektet. Denne funktion vil have samme virkning, som beskrevet ved angivelse af bip-lyde for pulsslag, og vil derfor være behjælpelig med, at gøre Forskerens arbejde lettere.

Blodtryksmålesystemet, der er udviklet i dette projekt, har til formål at hjælpe Forskere med kontinuert at monitorere Måleobjekters blodtryk, og derfor vil inddragelse af fagprofessionelle til design af brugergrænsefladen og klarlæggelse af funktioner være en naturlig del af det fremtidige arbejde.

Via usabilitytesting kan Blodtryksmålesystemet evalueres af dets slutbrugere, og derefter tilpasses til deres behov, således brugergrænsefladens design og systemets funktioner møder slutbrugerens ønsker.

Ved monitorering af Måleobjekters helbredstilstand kan det, udover blodtryk, være relevant at observere deres saturation og EKG-signal.

En fremtidig udvidelse kunne derfor være at udvikle en funktion til at vise Måleobjektets saturation og EKG-signal på samme måde, som blodtrykket bliver vist.

Denne udvidelse vil samtidig give anledning til bredere anvendelsesmuligheder, idet systemet dækker flere af de funktioner, der blandt andet er brug for under forskning og ved almindelig klinisk benyttelse på sygehuse.

Hardwaren i Blodtryksmålesystemet er realiseret på et veroboard, men til fremtidigt arbejde kunne der udvikles et printboard for at sikre endnu bedre stabilitet og udseende. Hardwaren vil desuden blive markant mindre med et printboard fremfor på et veroboard.

Samlet set er der i fremtiden flere muligheder for at ændre og forbedre brugergrænsefladen. Blandt andet gennem samarbejde med fagprofessionelle slutbrugere, men især i takt med videreudvikling af flere funktioner til systemet.

Specielt implementering af monitorering af saturation og EKG-signal kan være med til at løfte produktet til bredere anvendelsesmuligheder samt et generelt bedre udgangspunkt for monitorering af Måleobjekters helbredstilstand.

Konklusion 7

Formålet med dette projekt var at udvikle et blodtryksmålesystem, der via et væskefyldt kateter invasivt kunne måle blodtrykket og udskrive dette samt de hæmodynamiske data på en brugergrænseflade. Ud fra problemformuleringen og de opstillede krav er Blodtryksmålesystemet udviklet til at kunne nulpunktjustere, kalibrere samt filtrere blodtrykket via et digitalt filter, der kan aktiveres og deaktiveres. Det var et ønske, at Blodtryksmålesystemet skulle være udviklet til forskningsmæssige sammenhænge. Derfor blev dette prioriteret i designfasen, så det færdige system skulle være mere anvendeligt for en forsker, som ønskede at foretage og observere forskellige blodtryksmålinger.

I den udviklede hardwaredel er det lykkedes at opbygge et elektronisk kredsløb, der behandler et signal, der er transformert fra et tryk via den udleverede Tryktransducer. Når trykket er blevet transformert til et analogt signal skal det viderebehandles gennem en Signalbehandlingsblok, som består af en Forstærker og et Filter. Når signalet har passeret Forstærker og Filter bliver det konverteret i den udleverede DAQ. Her konverteres signalet fra et analogt til et digitalt signal, hvorefter signalet er klar til at blive behandlet i den udviklede software.

Softwareen er ud fra kravene udviklet til grafisk at kunne udskrive blodtrykssignalet kontinuert som funktion af tiden. Systemet indeholder et C#-program udviklet i Visual Studio, som var et af de obligatoriske krav. Ud over kalibrering, nulpunktjustering og muligheden for at aktivere og deaktivere det digitale filter, er der udviklet en funktion til at optage en sekvens af det målte blodtrykssignal i en bestemt optagelseslængde. Herefter er det muligt, at gemme det optagede blodtryksmåling. Dette var ligeledes et krav til softwaren, som er implementeret. Ud over de obligatoriske krav, er det valgt, at systemet ligeledes skulle afbilde værdierne for det systoliske og diastoliske blodtryk med tal samt detektering af puls og udskrivelse af denne. Dette er også succesfuldt implementeret.

Selve Blodtryksmålesystemets hardware- og softwaredel er løbende testet i udviklingsfasen. Til de endelige test af systemet erfares det, at Blodtryksmålesystemet kan måle og udskrive et blodtrykssignal simuleret fra PhysioNet. Derudover er der også udført test på en In Vitro hjertemodel, hvor systemet ligeledes var i stand til at måle det korrekte tryk, In Vitro modellen genererede. Dette kunne vi konkludere, da der ligeledes var tilkoblet et blodtryksmonitoreringssystem fra Siemens, som målte de samme data. Den sidste test, der blev foretaget på Blodtryksmålesystemet var In Vivo, hvor vi fik mulighed for at teste på en levende gris på AUH. Ud fra denne test af kan det konkluderes, at vores monitorerings-system fungerer som ønsket og kan monitorere reelle blodtryksværdier.

Det endelige Blodtryksmålesystem er et vellykket produkt, der succesfuldt kan udskrive en blodtrykskurve på en grafisk brugergrænseflade. Systemet er udviklet fra problemformuleringen og de opstillede krav, som alle er opfyldt.

Litteratur

- [1] Webster, John G., m.fl, *Medical Instrumentering*, John Wiley & Sons, INC, ISBN-13 978-0471-67600-3.
- [2] Ukendt, http://www.biologifaget.dk/uploads/tx_cliopolaroidphotoflex/MarynaMelnyk.Shutterstock._02.jpg, Besøgt d. 07-12-2015
- [3] Johansen, Peter, *Hæmodynamik og hjerte-kar systemet*, Note fra KVI undervisning, Se bilag
- [4] Wayne Storr, <http://www.electronics-tutorials.ws/blog/wheatstone-bridge.html>, Besøgt 7. December 2015, Sidst opdateret 7. December 2015, Se bilag
- [5] Johansen, Peter, *PowerPoint, Dynamics of Pressure Measurement Systems*, PowerPoint fra KVI undervisning, Se bilag
- [6] Sand, Olav m.fl. & Bjålie, Jan G., *Menneskets anatomi og fysiologi*, Gad Forlag, 18. August 2008
- [7] IHA, *Vejledning for gennemførelse af projekt 2*, 20. Februar 2015, Se bilag
- [8] Bjerre, Kim, *PowerPonit, Development Processes*, Se bilag
- [9] A. Roguin, *Scipione Riva-Rocci and the men behind the mercury sphygmomanometer*, Marts 2005, Se bilag
- [10] <https://www.physionet.org>, Besøgt gennem hele projektet

Bilag

Alle bilagsdokumenter vil være at finde i mappen kaldet Bilag. I Tabel 1 ses bilagslisten.

Type	Bilagsnavn
Film	Aortaklap
PowerPoint	Development Processes
Film	Griseforsøg
Note	Hæmodynamik og hjertekarsystemet (ver 5)
Datablad	INA 114
Datablad	NI USB 6008
Datablad	OP27G datasheet
PowerPoint	Pressure dynamic systems III
Datablad	Sallen_key_low_pass_filter
Note	Scipione Riva-Rocci and the men behind the mercury sphygmomanometer
Datablad	Tryktransducer datablad
Film	Utaet pakning mellem ventrikulær kammer og aortaroden
Note	Vejledning for gennemførelse af projekt 2
Note	Vejledning til 3. semestersprojekt
Note	Wheatstone Bridge Circuit and Theory of Operation

Tabel 1: Bilagsliste

Logbog, Mødereferater, Detaljeret tidsplan samt Samarbejdsaftale finde ligeledes i bilags mappen.

Softwaren for Blodtryksmålesystemet er også at finde i bilagsmappen under navnet Projekt_1.0.

AARHUS SCHOOL OF ENGINEERING

SUNDHEDSTEKNOLOGI 3. SEMESTERPROJEKT



Dokumentation

Gruppe 1

Lise Skytte Brodersen (201407432)
Nina Brkovic (201406458)
Jakob Degn Christensen (201408532)
Toke Tobias Aaris (201407321)
Annsofie Randrup Wagner (201406360)
Anders Wiggers Birkelund (201404118)

Vejleder

Studentervejleder
Peter Johansen
Aarhus Universitet

15. december 2015

Indholdsfortegnelse

Kapitel 1 Kravspecifikation	1
1.1 Indledning	1
1.2 Systembeskrivelse	1
1.3 Funktionelle krav	2
1.3.1 Aktør-kontekstdiagram	2
1.3.2 Aktørbeskrivelse	2
1.3.3 Use casediagram	4
1.3.4 Use Cases	4
1.4 Ikke-funktionelle krav	8
1.4.1 Functionality	8
1.4.2 Usability	9
1.4.3 Reliability	9
1.4.4 Performance	9
1.4.5 Supportability	10
1.4.6 Andre(+)	10
1.4.7 GUI	10
Kapitel 2 Design	13
2.1 Systemarkitektur	13
2.1.1 bdd	13
2.1.2 ibd	14
2.2 Grænseflader	15
2.3 Hardware arkitektur	16
2.3.1 bdd	17
2.3.2 Ibd	17
2.3.3 Grænseflader	17
2.3.4 Forstærkerblok	18
2.3.5 Filterblok	18
2.4 Software arkitektur	19
2.4.1 Domænemodel	19
2.4.2 Applikationsmodel	20
Kapitel 3 HW implementering og test	25
3.1 Implementering	25
3.1.1 Forstærkerblok	25
3.1.2 Filterblok	27
3.2 Test	29
3.2.1 Test af Forstærkerblok	29
3.2.2 Test af Filterblok	31

3.2.3	Test af Signalbehandlingsblok	34
Kapitel 4	SW implementering og test	38
4.1	Implementering	38
4.1.1	Datalag	39
4.1.2	Logiklag	40
4.1.3	Præsentationslag	40
4.1.4	Datamodellaget	42
4.1.5	Metodebeskrivelser	43
4.2	Test	49
4.2.1	Use Case 1: Kalibrer	49
4.2.2	Use Case 2 Vis måling	52
4.2.3	Use Case 3: Nulpunktsjuster	55
4.2.4	Use Case 4: Deaktiver filter	57
4.2.5	Use Case 5: Aktiver filter	58
4.2.6	Use Case 6: Gem måling	60
Kapitel 5	Accepttest	65
5.1	Accepttest af Use Cases	65
5.1.1	Use Case 1	65
5.1.2	Use Case 2	66
5.1.3	Use Case 3	66
5.1.4	Use Case 4	67
5.1.5	Use Case 5	68
5.1.6	Use Case 6	68
5.2	Accepttest af ikke-funktionelle krav	69

Kravspecifikation

1

Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse
1.0	23-09-2015	Alle	Første udkast til Use Cases. I alt 4, hvor en af funktionaliteterne var, at man kunne optage en lydsekvens
1.1	29-09-2015	Alle	Ændring af Use Cases efter møde med Peter. I alt fem, hvor funktionaliteterne kun dækker over de opstillede krav til projektet
1.2	30-09-2015	Alle	Små ændring af formuleringerne samt byttet om på Use Case 1 og Use Case 2 og tilføjet en Use Case 6. De ikke-funktionelle krav er blevet tilføjet. Klar til Review
2.0	08-10-2015	Alle	Rettelser efter review møde
2.1	04-11-2015	Alle	Tilføjet Tryktransducer som en sekundær aktør
2.2	08-12-2015	Alle	Små rettelser i forhold til udførelse af accepttest
3.0	15-12-2015	Alle	Gennem retning inden projektaflevering

1.1 Indledning

Kravspecifikationen vil gennem seks Use Cases beskrive Blodtryksmålesystemets funktionelle krav. Systemets ikke-funktionelle krav er udarbejdet på baggrund af FURPS+. Dertil vil der være aktør-kontekstdiagram og Use Case diagram samt beskrivelse af de forskellige aktører, der interagerer med systemet.

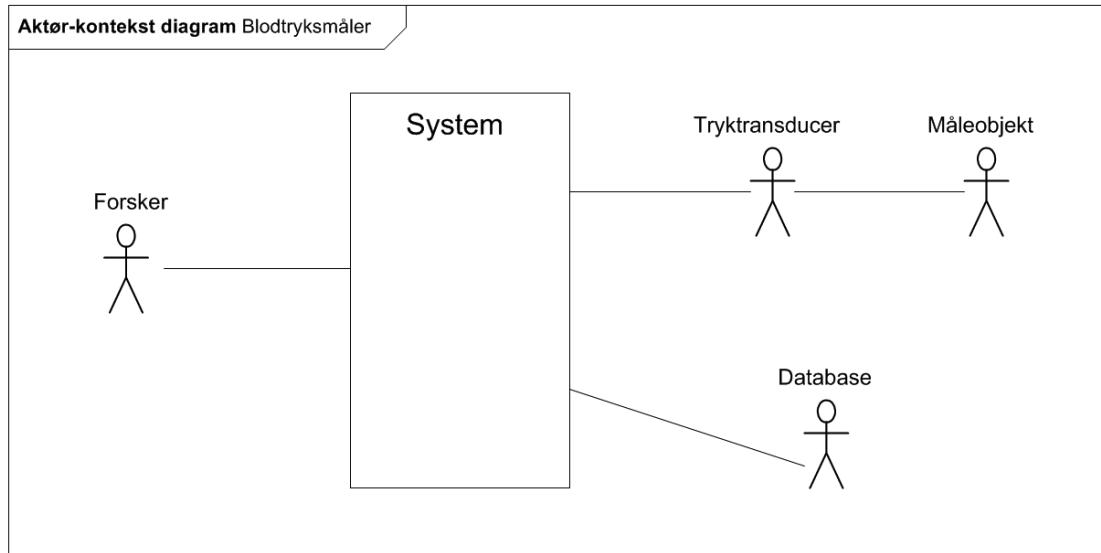
1.2 Systembeskrivelse

Systemet skal kunne vise et blodtrykssignal kontinuert i en graf. Derudover skal systemet kunne kalibrere, nulpunktsjustere samt gemme data for målingen i en database. Systemet er udviklet som et færdigt produkt, der er muligt at teste udfra de givne rammer.

1.3 Funktionelle krav

De funktionelle krav vil beskrives ud fra aktør-kontekstdiagram, aktørbeskrivelse, Use Cases samt Use Case diagram.

1.3.1 Aktør-kontekstdiagram



Figur 1.1: Aktør-kontekstdiagram

Systemet består af en software- og en hardwaredel. Softwaredelen er udarbejdet i Visual Studio C#. Hardwaredelen består af flere komponenter sat sammen: Tryktransducer, instrumentationforstærker, et aktivt anden ordens lavpasfilter af typen Sallen-Key med unity gain og en DAQ. Dette er selve systemet.

Primæraktøren i dette projekt er en Forsker. Sekundæraktørerne er Database, Tryktransducer og Måleobjekt. Måleobjekt kan være en package af PhysioNet og Analog Discovery.

1.3.2 Aktørbeskrivelse

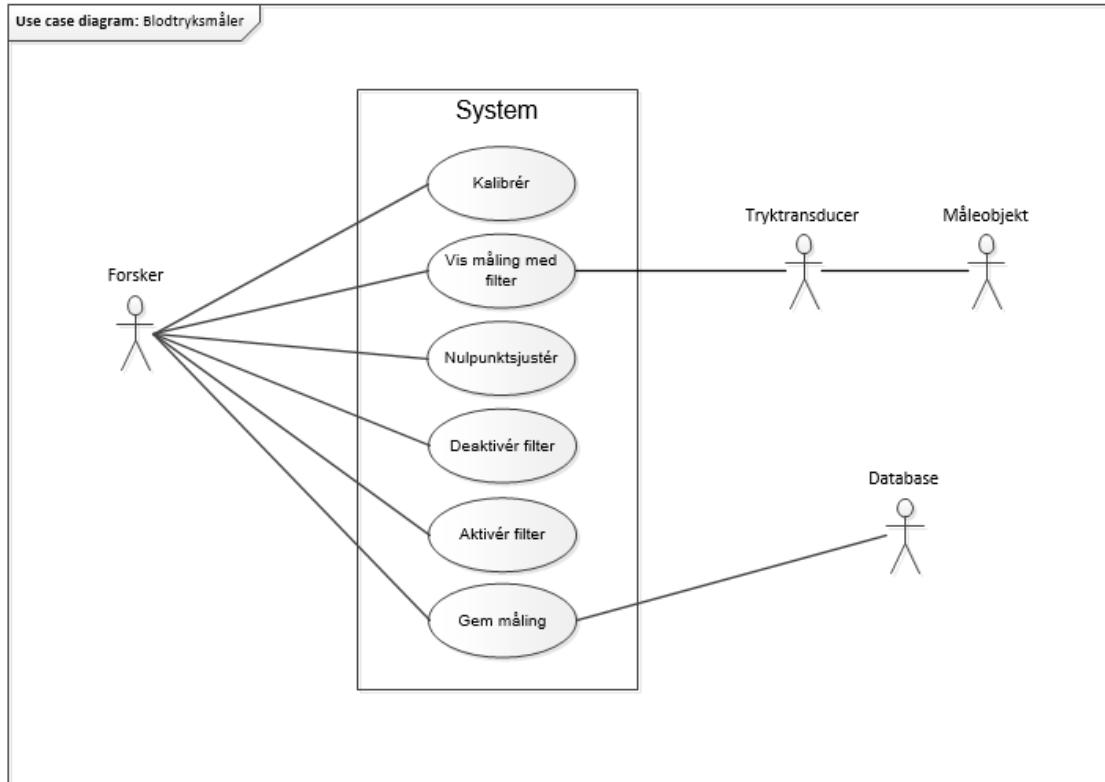
I Tabel 1.2 vil de forskellige aktører beskrives.

Aktørnavn	Forsker
Type	Primær
Beskrivelse	Person med relevant baggrundsviden inden for blodtryksanalyse
Aktørnavn	Tryktransducer
Type	Sekundær
Beskrivelse	Tryktransducer måler og omformer trykket fra Måleobjekt til et analogt signal

Aktørnavn	Måleobjekt
Type	Sekundær
Beskrivelse	Måleobjekt i det færdigudviklede produkt er et signal genereret enten in vitro eller in vivo. I prototypen er Måleobjekt en kombination af Physionet og Analog Discovery. Måleobjekt repræsenterer data fra Physionet leveret til blodtryksmålesystem igennem Analog Discovery
Aktørnavn	Database
Type	Sekundær
Beskrivelse	Database bruges i blodtrykmålesystemet til at gemme data
Aktørnavn	Physionet
Type	Ekstern
Beskrivelse	Physionet er en ekstern database, som indeholder blodtrykssignalet fra forskellige patienter
Aktørnavn	Analog Discovery
Type	Ekstern
Beskrivelse	Analog Discovery omdanner data fra Physionet til et analogt signal

Tabel 1.2: Aktørbeskrivelse

1.3.3 Use casediagram



Figur 1.2: Use case-diagram

Forskeren af systemet er den primære aktør i alle seks Use Cases. Forskeren sætter alle Use Cases igang og styrer, hvad der skal ske og hvornår. Tryktransducer, som er en af de sekundære aktører, interagerer i Use Case 2. Tryktransduceren behandler tryk fra den anden sekundære aktør, Måleobjekt, og omformer det til et analogt signal. Blodtryksmålingen skal vises i Use Case 2. For at få gemt data interagerer den sekundære aktør, Database, med Use Case 6.

1.3.4 Use Cases

Use Case 1

Navn	Kalibrér
Use case ID	1
Samtidige forløb	1
Primær aktør	Forsker
Sekundære aktør	
Mål	Forsker ønsker at kalibrere systemet
Initiering	Startes af Forsker

Forudsætninger	System er aktivt og tilgængeligt
Resultat	System er kalibreret
Hovedforløb	<ol style="list-style-type: none"> 1. Kalibrerings-vinduet vises ved opstart og tidlige kalibreringsdata vises 2. Forsker indtaster målte kalibreringsdata <ul style="list-style-type: none"> [2.a <i>Forsker vælger ikke at kalibrere</i>] [2.b <i>Indtastede kalibreringsdata er ugyldige</i>] 3. System kalibrerer 4. Det fremgår i Kalibrerings-vinduet, at kalibrering er udført
Undtagelser	<ol style="list-style-type: none"> 2.a Forsker ønsker ingen kalibrering. Use Case 1 afsluttes og Kalibrerings-vinduet lukkes 2.b Det fremgår i Kalibrerings-vinduet, at indtastede kalibreringsdata er ugyldige. Forsætter ved Use Case 1 punkt 2

Tabel 1.3: Fully dressed Use Case 1.

Use Case 2

Navn	Vis Måling med digitalt filter
Use case ID	2
Samtidige forløb	1
Primær aktør	Forsker
Sekundære aktør	Måleobjekt og Tryktransducer
Mål	Forsker ønsker at vise blodtrykssignal med digitalt filter
Initiering	Startes efter afsluttet Use Case 1
Forudsætninger	System er aktivt og tilgængeligt. Digitalt filter er aktivt. Tryktransducer er tilsluttet systemet. Måleobjekt er tilsluttet Tryktransducer
Resultat	Det filtrerede blodtrykssignal udskrives
Hovedforløb	<ol style="list-style-type: none"> 1. Det filtrerede blodtrykssignal vises i en graf i Monitor-vinduet
Undtagelser	

*Tabel 1.4: Fully dressed Use Case 2.***Use Case 3**

Navn	Nulpunktsjustér
Use case ID	3
Samtidige forløb	1
Primær aktør	Forsker
Sekundære aktør	
Mål	Forsker ønsker at nulpunktsjustere system
Initiering	Startes af Forsker
Forudsætninger	System er aktivt og tilgængeligt. Use Case 2 kører. Tryktransduceren skal befinde sig i atmosfærisk luft
Resultat	System er nulpunktsjusteret
Hovedforløb	<ol style="list-style-type: none"> 1. Forsker vælger at udføre en nulpunktsjustering 2. System nulpunktsjusterer 3. Det fremgår i Monitor-vinduet, at nulpunktsjustering er udført
Undtagelser	

*Tabel 1.5: Fully dressed Use Case 3.***Use Case 4**

Navn	Deaktivér filter
Use case ID	4
Samtidige forløb	1
Primær aktør	Forsker
Sekundære aktør	
Mål	Forsker ønsker at deaktivere det digitale filter
Initiering	Startes af Forsker
Forudsætninger	System er aktivt og tilgængeligt. Use Case 2 kører

Resultat	Ufiltreret blodtrykssignal vises i Monitor-vinduet
Hovedforløb	<ol style="list-style-type: none"> 1. Forsker vælger at deaktivere digitalt filter 2. System udskriver det ufiltrerede blodtrykssignal
<hr/>	
Undtagelser	

*Tabel 1.6: Fully dressed Use Case 4.***Use Case 5**

Navn	Aktivér filter
Use case ID	5
Samtidige forløb	1
Primær aktør	Forsker
Sekundære aktør	
Mål	Forsker ønsker at aktivere det digitale filter
Initiering	Startes af Forsker
Forudsætninger	System er aktivt og tilgængeligt. Use Case 4 er afsluttet
Resultat	Filtreret blodtrykssignal vises i Monitor-vinduet
Hovedforløb	<ol style="list-style-type: none"> 1. Forsker vælger at aktivere digitalt filter 2. System udskriver det filtrerede blodtrykssignal
<hr/>	
Undtagelser	

*Tabel 1.7: Fully dressed Use Case 5.***Use Case 6**

Navn	Gem måling
Use case ID	6
Samtidige forløb	1...*
Primær aktør	Forsker

Sekundære aktør	Database
Mål	Forsker ønsker at gemme data i Database
Initiering	Startes af Forsker
Forudsætninger	System er aktivt og tilgængeligt. Use Case 4 eller Use Case 5 kører
Resultat	Data er gemt i Database
Hovedforløb	<ol style="list-style-type: none"> 1. Forsker bestemmer optagelseslængde for målingen i Monitor-vinduet 2. Forsker igangsætter optagelse af målingen 3. Målingen stopper <p>[3.a <i>Forsker stopper optagelsen af måling inden optagelseslængden er nået</i>]</p> 4. Forsker åbner Gem-vinduet 5. Forsker indtaster oplysninger om målingen 6. Forsker trykker på "OK"-knappen og system gemmer målingen i Database 7. Det fremgår af Gem-vinduet, at målingen er gemt
3.a	Forsker stopper optagelse af målingen. Forsætter ved punkt 4 i Use Case 6

Tabel 1.8: Fully dressed Use Case 6.

1.4 Ikke-funktionelle krav

De ikke-funktionelle krav er specificeret ved brug af redskabet FURPS+, der står for henholdsvis Functionality, Usability, Reliability, Performance, Supportability og andre krav til eksempelvis brugssituationer og interface.

1.4.1 Functionality

- Systemet skal kunne kalibreres.
- Systemet skal kunne vise et kontinuert blodtrykssignal i Monitor-vinduet.
- Systemet skal kunne vise systole-, diastole- og pulsværdier med op til tre cifre.
- Systemet skal kunne vise et blodtrykssignal med og uden digitalt filter.
- Systemet skal kunne nulpunktsjustere blodtrykssignalet.

- Systemet skal kunne gemme en blodtryksmåling i en database.

1.4.2 Usability

- Kalibrerings-vinduet skal indeholde tekstbokse til indtastning af kalibreringsdata.
- Kalibrerings-vinduet skal indeholde en "Beregn"-knap.
- Kalibrerings-vinduet skal indeholde en "OK"-knap.
- Kalibrerings-vinduet skal indeholde et tidsstempel for seneste kalibrering.
- Monitor-vinduet skal indeholde en "Gem"-knap.
- Monitor-vinduet skal indeholde en "Nulpunktsjustér"-knap.
- Monitor-vinduet skal indeholde et tidsstempel for seneste nulpunktsjustering.
- Monitor-vinduet skal indeholde to radiobuttons til aktivering og deaktivering af digitalt filter.
- Monitor-vinduet skal indeholde en tekstboks til ændring af optagelseslængden.
- Monitor-vinduet skal indeholde en "Rec"-knap.
- Monitor-vinduet skal indeholde en "Stop"-knap.
- Gem-vinduet skal indeholde tekstbokse til indtastning af oplysninger om målingen.
- Gem-vinduet skal indeholde en "OK"-knap.
- Det skal være muligt at aflæse systole-, diastole- og pulsværdier på Monitor-vinduet fra to meters afstand med normalt syn.

1.4.3 Reliability

- Systemet skal have en effektiv MTBF (Mean Time Between Failure) på 99 timer og en MTTR (Mean Time To Restore) på 20 minutter (1/3 time).

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} = \frac{99}{99 + 1/3} = 0,997 = 99,7\% \quad (1.1)$$

1.4.4 Performance

- Blodtrykssignalet skal vises maksimalt fem sekunder efter Use Case 1 er afsluttet.
- Systemet skal vise en graf for blodtryksmålingen, hvor y-aksen er mmHg og x-aksen er tid i sekunder.
- Systemet skal kunne måle blodtryksværdier fra 0 til 300 mmHg.
- Blodtrykssignalet vises kontinuert over et interval af fire sekunder ad gangen.
- Blodtrykssignalet vises via en rød kurve.
- Y-aksen varierer efter blodtrykssignalets maksimum- og minimumværdier.

1.4.5 Supportability

- Softwaren skal opbygges efter trelagsmodellen.

1.4.6 Andre(+)

Brugssituationer

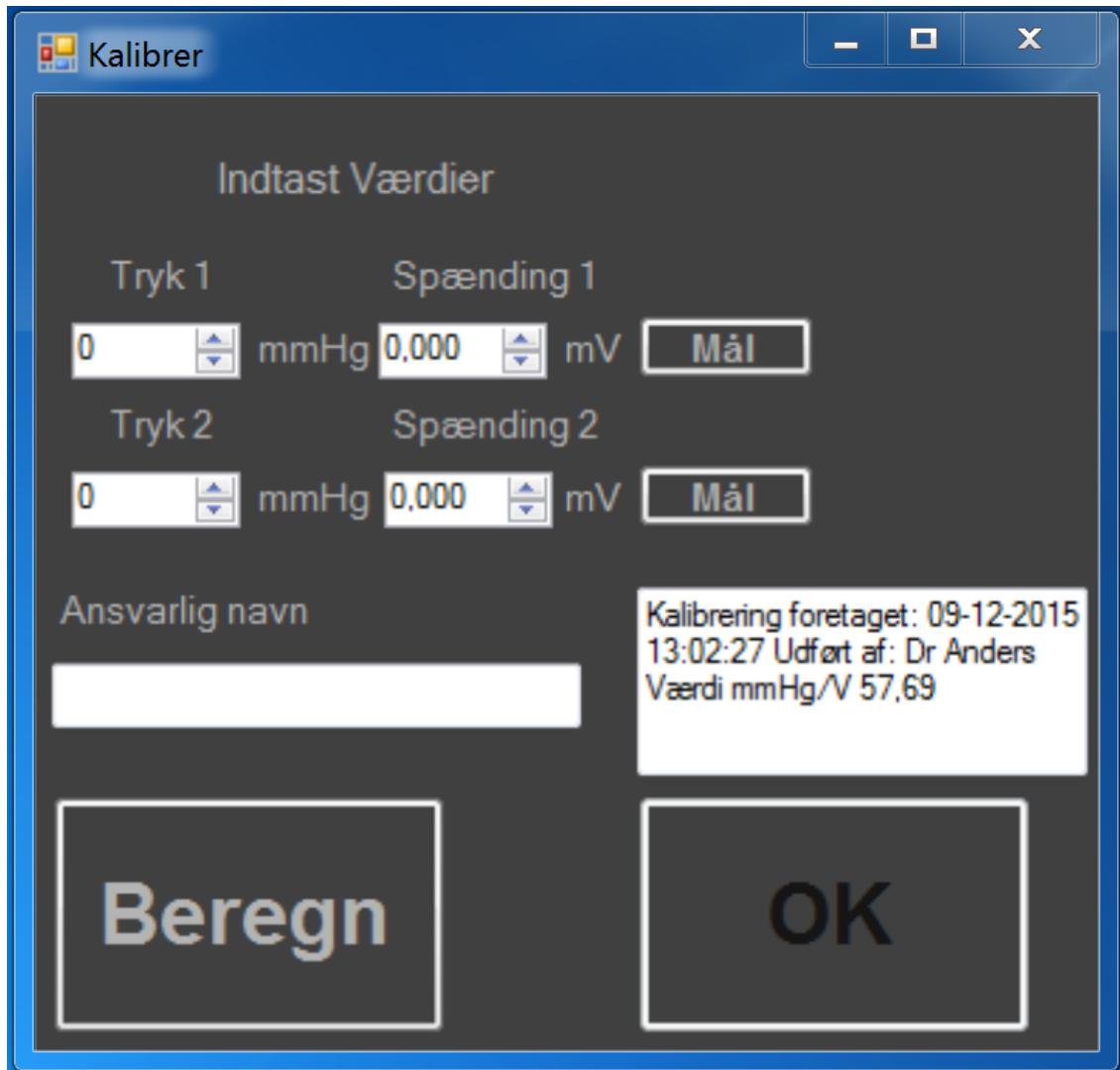
- Der skal være adgang til en computer med Windows 7 eller nyere – computeren skal have minimum 4 GB RAM.
- Der skal være adgang til en computer, hvor National Instruments er installeret.

Interface

- Blodtryksgrafen skal fylde minimum 1/3 af Monitor-vinduet.
- Baggrunden i Monitor-vinduet skal være mørk.
- Blodtrykssignal og -værdier (systole og diastole) skal være røde, og puls skal være grøn.
- Systolisk og diastolisk blodtryk skal fremhæves øverst i højre hjørne, med større skriftstørrelse end andre værdier i Monitor-vinduet, eksempelvis værdier på akserne.

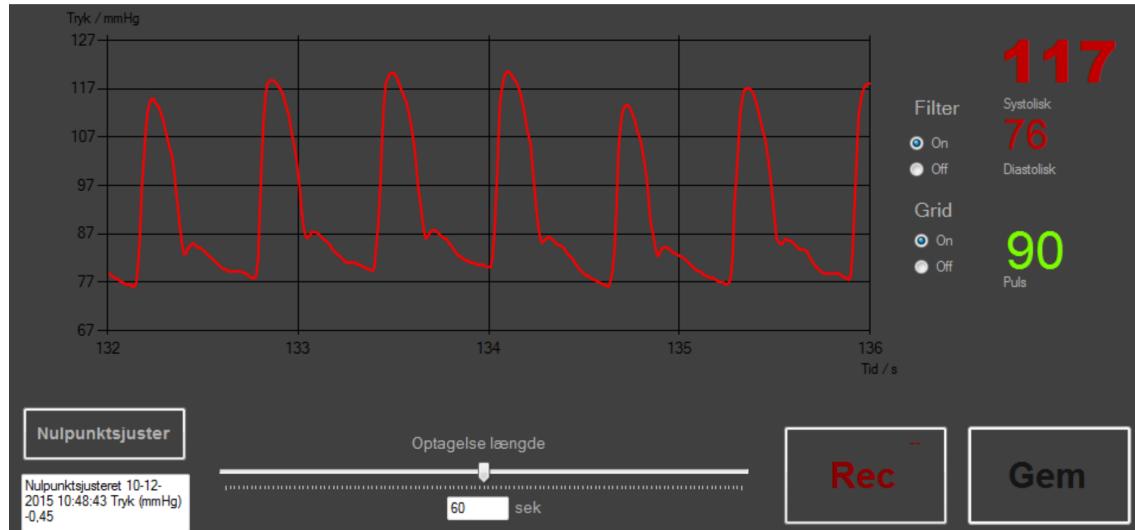
1.4.7 GUI

I det følgende afsnit vil der være indsatt figurer af, hvordan brugergrænsefladerne skal designes. Brugergrænsefladen er bindeleddet mellem systemet og dets bruger. Her stilles krav til, at programmet skal være let anvendeligt og designet med fokus på slutbrugeren. Yderligere specifikationer om designfasen af programmets brugergrænseflade kan findes i Projektdokumentationen under Design 2.



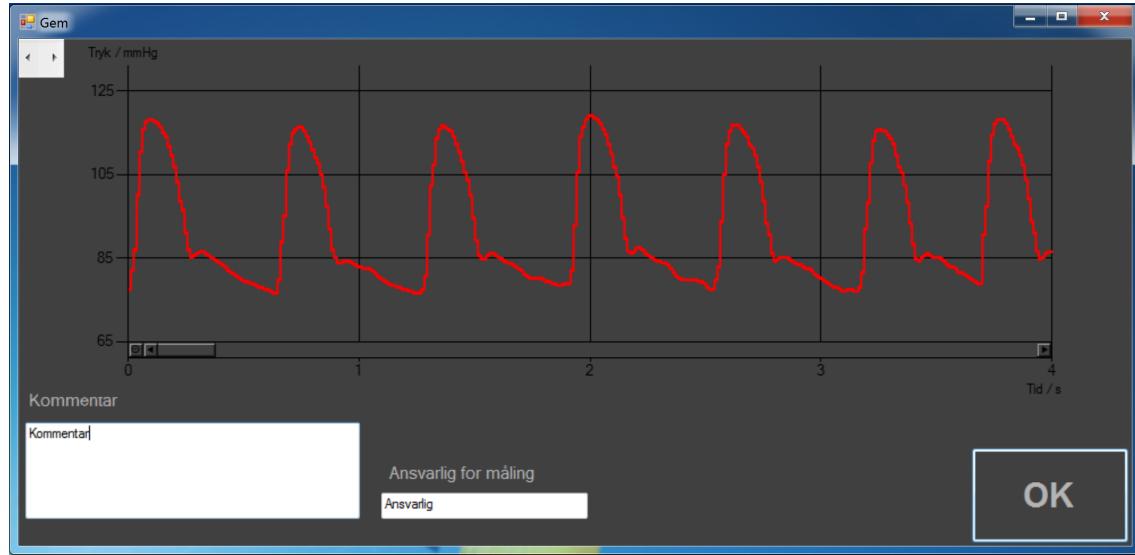
Figur 1.3: GUI: Kalibrerings-vindue

På Figur 1.3 ses en skitse over, hvorledes Kalibrerings-vinduets brugergrænseflade ønskes designet.



Figur 1.4: GUI: Monitor-vindue

På Figur 1.4 ses en skitse over, hvorledes Monitor-vinduets brugergrænseflade ønskes designet.



Figur 1.5: GUI Gem vindue

På Figur 1.5 ses en skitse over, hvorledes Gem-vinduets brugergrænseflade ønskes designet.

Design 2

Versionshistorik

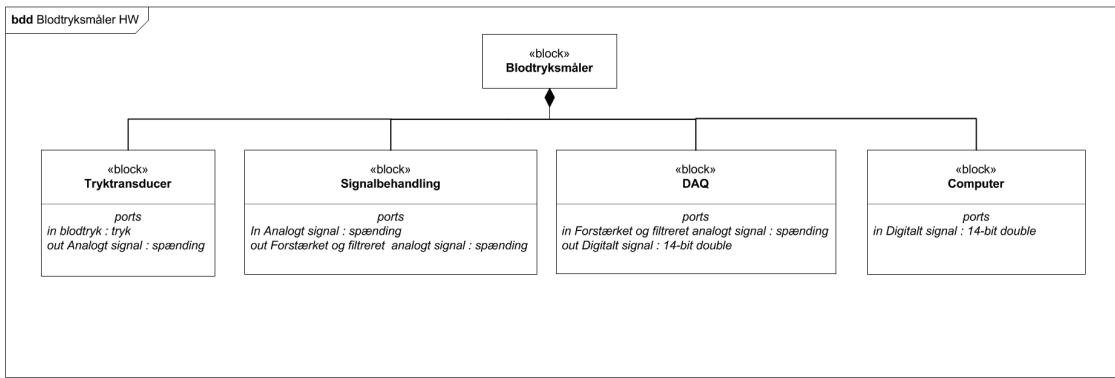
Version	Dato	Ansvarlig	Beskrivelse
1.0	20-10-2015	Alle	Første udkast til domænemodel, bdd, ibd og sekvensdiagrammer
1.1	21-10-2015	Alle	Små ændringer i bdd og ibd efter møde med vejleder
1.2	27-10-2015	Alle	Ændring af bdd og ibd efter møde med Kim; blokkene Filter og Forstærker er blevet lagt sammen under blokken Signalbehandling
1.3	02-11-2015	Alle	Begyndte at oprette Design-dokumentet. Udkast til klassediagrammer for Use Cases
1.4	04-11-2015	LSB	Skrevet hardwaredesign afsnittet. Små rettelser i de andre afsnit i Design således, det er klar til review
2.0	12-11-2015	Alle	Rettelser efter review
2.1	03-12-2015	LSB	Hardware dokumentation ændres en smule i forhold til 5 volts regulator
2.2	07-12-2015	TTA og AWB	Softwaresekvensdiagrammer
3.0	15-12-2015	Alle	Gennem retning inden projektaflevering

2.1 Systemarkitektur

Igennem bdd og ibd vil det overordnede Blodtryksmålesystem beskrives i forhold til hvilke hardwareblokke, systemet består af, og hvordan de interagerer med hinanden.

2.1.1 bdd

På Figur 2.1 ses bdd for systemet. Bdd viser de forskellige hardwareblokke for systemet og hvilke porte, de består af. I Tabel 2.2 ses en beskrivelse af blokkene.



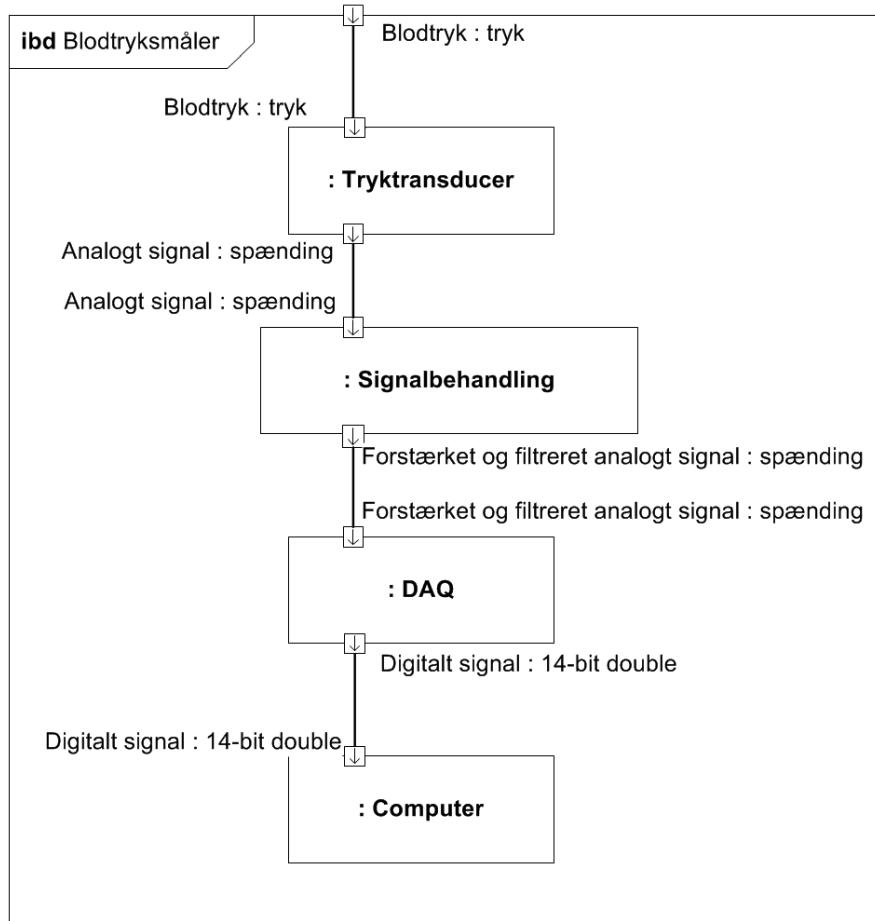
Figur 2.1: bdd

Blok	Beskrivelse
Blodtryksmåler	Det overordnede system, som indeholder Tryktransducer, Signalbehandling, DAQ og Computer.
Tryktransducer	Registrerer en fysisk størrelse i form af en trykændring. Tryktransduceren har til opgave at transformere den fysiske størrelse til en elektrisk spænding, som viderebehandles gennem de resterende hardwareblokke.
Signalbehandling	Består af to dele. En forstærkerdel og en filteringsdel. Det analoge signal fra Tryktransduceren bliver via denne blok forstærket og filteret.
DAQ	Konverterer det forstærkede og filterede analoge signal til et digitalt signal. Der er valgt et spændingsområde fra +/- 5 V.
Computer	Indeholder software til systemet, som er kodet i Visual Studio C#. Softwaren kan blandt andet vise det digitale signal grafisk. Softwaren kan ligeledes kalibrere, nulpunktsjustere og gemme målinger samt aktivere og deaktivere filter.

Tabel 2.2: Beskrivelse af blokkene for systemet

2.1.2 ibd

På Figur 2.2 ses et ibd for systemet. Ibd viser, hvordan de forskellige hardwareblokke interagerer med hinanden. Ibd fortæller signalets behandling gennem systemet - altså hvordan signalet transformeres fra et målt fysisk tryk til et digitalt signal, som softwaren kan viderebehandle og vise grafisk.



Figur 2.2: ibd

2.2 Grænseflader

Kommunikationsprotokol for hardwareblokkene ses i Tabel 2.3. Det er en beskrivelse og specifikation af hvilket input- og outputsignal, de forskellige hardwareblokke har.

Tryktransducerens maksimale output, V_{max} , bestemmes ud fra følgende ligning:

$$V_{max} = P \cdot K \cdot V_+ \quad (2.1)$$

P = Tryk

K = sensitivitet

V_+ = inputspænding

For systemet er der valgt, at Tryktransducerens målbarer område skal ligge i intervallet 0-300 mmHg. Den maksimale outputspænding bliver derfor udtrykt ved det maksimale tryk på 300 mmHg, sensitiviteten på 5 μ og inputspændingen på 5 V. Inputspændingen

kommer reelt fra 9 V's batterier, men der er valgt at indsætte en 5 V regulator, så man er sikker på, hvad inputspændingen er, da batterier kan være ustabile.

$$V_{max} = 300 \text{ mmHg} \cdot 5 \mu\text{V} \cdot 5 \text{ V} \quad (2.2)$$

$$V_{max} = 7,5 \text{ mV} \quad (2.3)$$

Grænseflade	Signal	Type	Format	Værdi
Tryktransducer	Blodtryk	in	Tryk	0 - 300 mmHg
	Analogt	out	Spænding	+/- 7,5 mV
Signalbehandling	Analogt	in	Spænding	+/- 7,5 mV
	Forstærket og filteret analogt	out	Spænding	+/- 5 V
DAQ	Forstærket og filteret analogt	in	Spænding	+/- 5 V
	Digitalt	out	14-bit double	0-5 V
Computer	Digitalt	in	14-bit double	0-5 V

Tabel 2.3: Kommunikationsprotokol

2.3 Hardware arkitektur

Herunder er de krævede specifikationer for Signalbehandlingsblokken beskrevet. Tryktransduceren beskrives ligeledes, da denne har indflydelse for Forstærkerblokkens specifikationer.

Tryktransducer

Tryktransduceren skal omsætte det fysiske tryk til en elektrisk spænding. Tryktransduceren har en meget høj common mode rejection samt en høj sensitivitet på $5 \mu\text{V}/\text{V/mmHg}$ $+/- 1\%$ ¹, da de forventede trykændringer er små. Enheden for sensitiviteten angiver, hvor mange μV output der kommer fra Tryktransduceren pr. antal V i Tryktransducerens eksitationsspænding pr. mmHg. For at være sikker på at strømmen igennem Wheatstone broen ikke bliver for høj, og dermed i sidst ende kan brænde strain gauges af, er der blevet valgt at indsætte en regulator, der omdanner eksitationsspænding fra 9 V batterier til 5 V.

Forstærkerblok

Forstærkerblokken sørger for at det svage spændingssignal fra Tryktransduceren bliver forstærket til en spænding, der ligger indenfor det måleinterval, der svarer til DAQ'ens, som er valgt til $+/- 5 \text{ V}$.

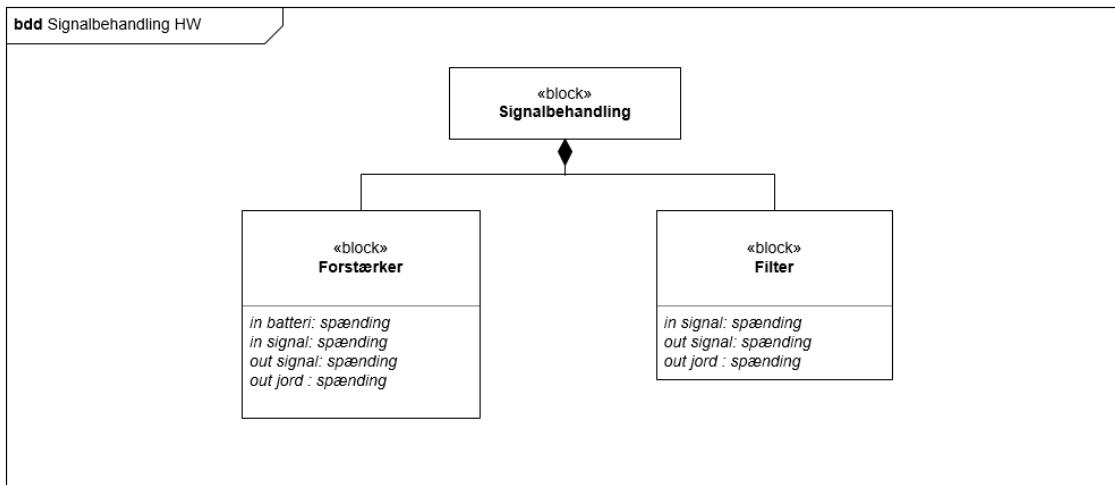
¹Se datablad for Tryktransduceren i bilag

Filterblok

Filterets formål er at frasortere blodtrykssignalets frekvenser, der er højere end 50 Hz, da disse ikke har nogen relevans for blodtrykssignalet. Dette skal realiseres ved et anden ordens lavpasfilter.

2.3.1 bdd

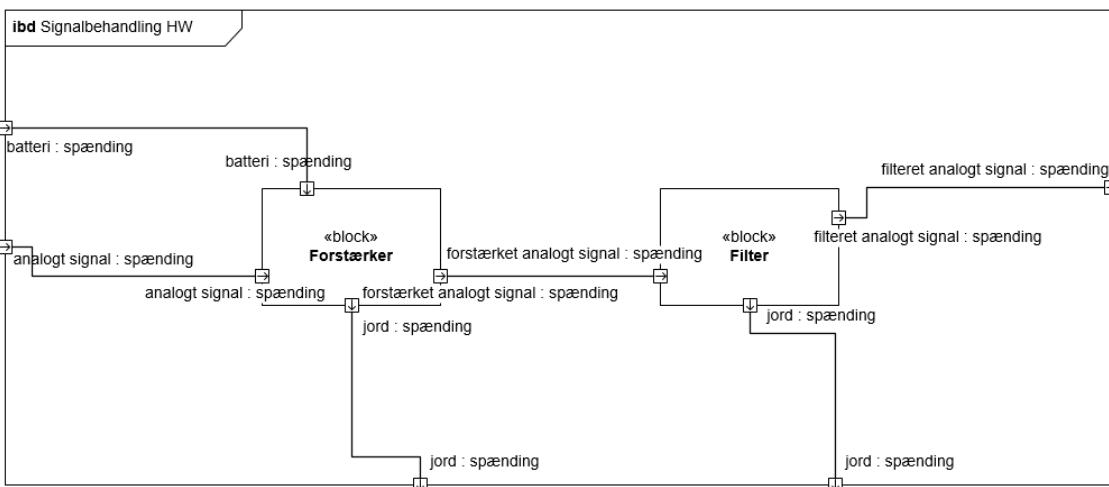
På Figur 2.3 ses et bdd for Signalbehandlingsblokken, hvor Forstærkerblokkens og Filterblokkens input- og outputporte er vist.



Figur 2.3: bdd for Signalbehandling HW

2.3.2 Ibd

På Figur 2.4 ses et ibd for Signalbehandlingsblokken. Ibd viser, hvordan Forstærkerblokken og Filterblokken interagerer med hinanden.



Figur 2.4: ibd for Signalbehandling HW

2.3.3 Grænseflader

Grænseflade	Signal	Type	Format	Værdi
Forstærker	Batteri	in	Spænding	+/- 9 V
	Analogt	in	Spænding	+/- 7,5 mV
	Jord	out	Spænding	0 V
	Forstærket analogt	out	Spænding	+/- 5 V
Filter	Forstærket analogt	in	Spænding	+/- 5 V
	Jord	out	Spænding	0 V
	Filteret analogt	out	Spænding	+/- 5 V

Tabel 2.4: Kommunikationsprotokol for Signalbehandlingsblok

2.3.4 Forstærkerblok

Specifikationer

- Forstærkningen, gain, skal forstærke den maksimale inputspænding, så outputtet bliver 10 V

$$Gain \cdot V_{in} = V_{out} \quad (2.4)$$

V_{out} og V_{in} er kendt, så gain isoleres

$$Gain = \frac{V_{out}}{V_{in}} \implies Gain = \frac{5000mV}{7,5mV} \implies Gain = 666,667 \quad (2.5)$$

- Inputspænding: +/- 7,5 mV
- Eksitationsspænding: +/- 9 V
- Outputspænding: +/- 5 V
- Minimum en båndbredde på 50 Hz
- Uendelig indgangsimpedans i teorien

2.3.5 Filterblok

Specifikationer

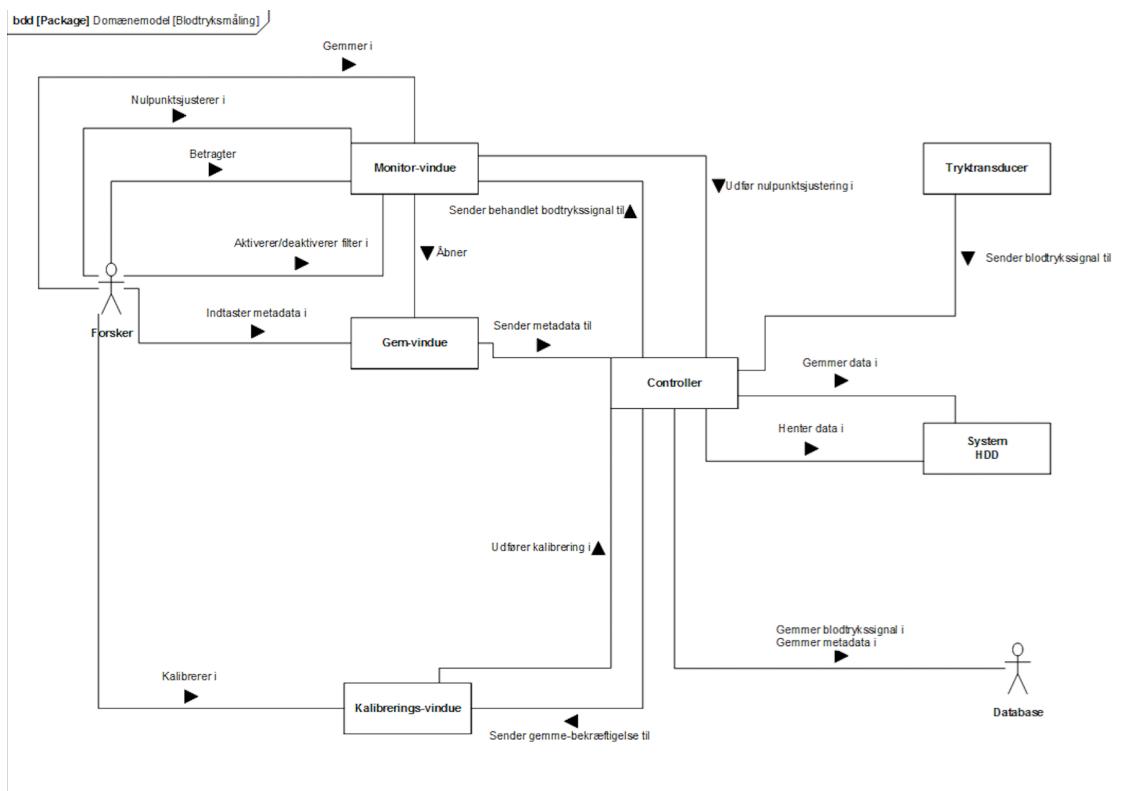
- Anden ordens lavpasfilter
- Cutoff frekvens ved 50 Hz
- Unity gain (ingen forstærkning)
- -40 dB ved 500 Hz

- Uendelig indgangsimpedans i teorien
 - Inputsspænding +/- 5 V
 - Eksitationsspænding +/- 9 V

2.4 Software arkitektur

2.4.1 Domænemodel

Domænemodellen er skabt på baggrund af de seks Use Cases og fungerer som et middel til at skabe et samlet overblik over systemet. Gennem navneordsanalyse er de konceptuelle klasser fundet. I modellen beskrives, hvordan de konceptuelle klasser og aktører interagerer med hinanden. Controlleren er ikke en konceptuel klasse, men den sørger for, at systemet fungerer optimalt og udfører kommandoer.



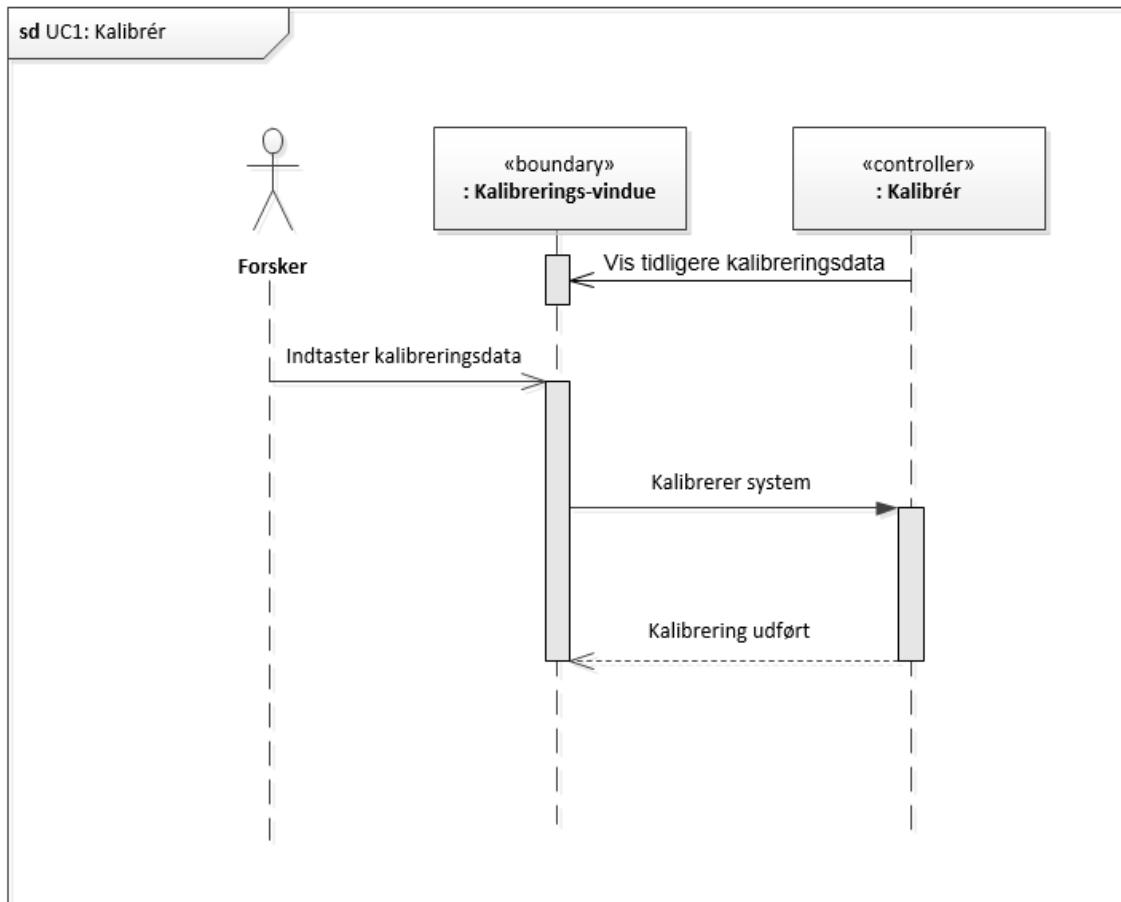
Figur 2.5: Domænemodel for blodtryksmålesystemet

I domænemodellen ses generelle principper om kommunikation mellem klasser og aktører. Kommunikationen kan derfor være af forskellig karakter, og funktionaliteten af den givne kommando kan variere fra Use Case til Use Case.

2.4.2 Applikationsmodel

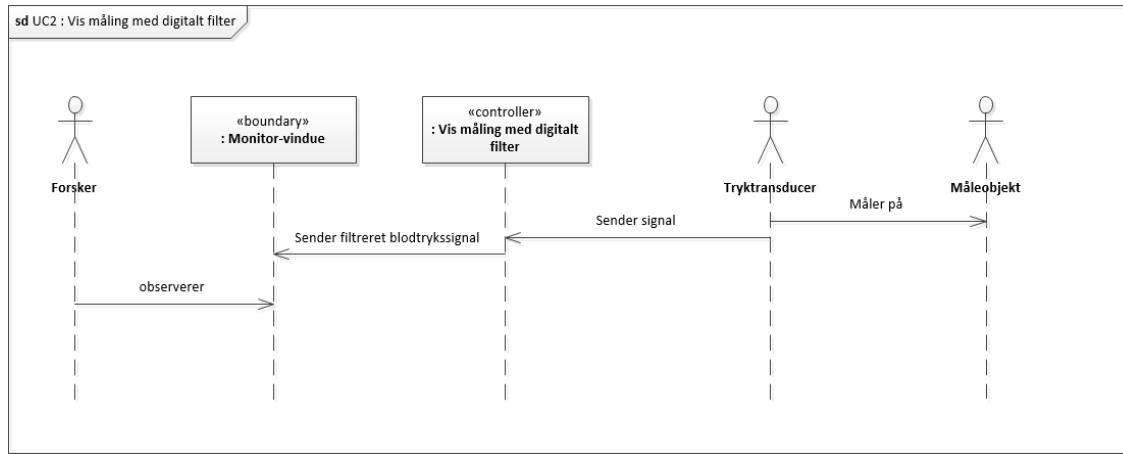
Sekvensdiagram

Sekvensdiagrammerne beskriver step-by-step forløbet i de forskellige Use Cases via metoder. Der er lavet et sekvensdiagram for hver Use Case for at gøre systemet mere overskueligt. Et sekvensdiagram består af boundary-klasserne og domain-klassen fra domænemodellen, samt en controller-klasse, med navn efter den specifikke Use Case.



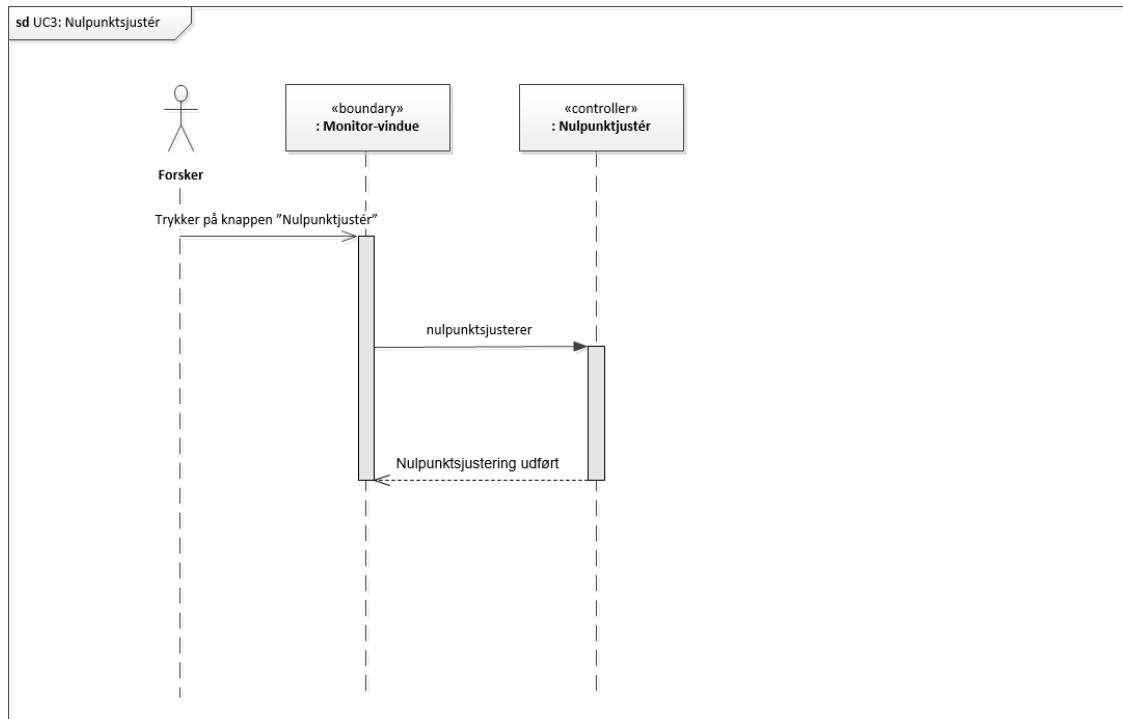
Figur 2.6: Sekvensdiagram for Use Case 1

Forsker interagerer med Monitor-vindue. Kalibreringsmetoden bliver kaldt, når Forsker trykker på knappen ”Ja”. Derefter igangsættes kalibreringen og kalibrerings-tidspunkt, og værdi sendes og gemmes i Databasen.



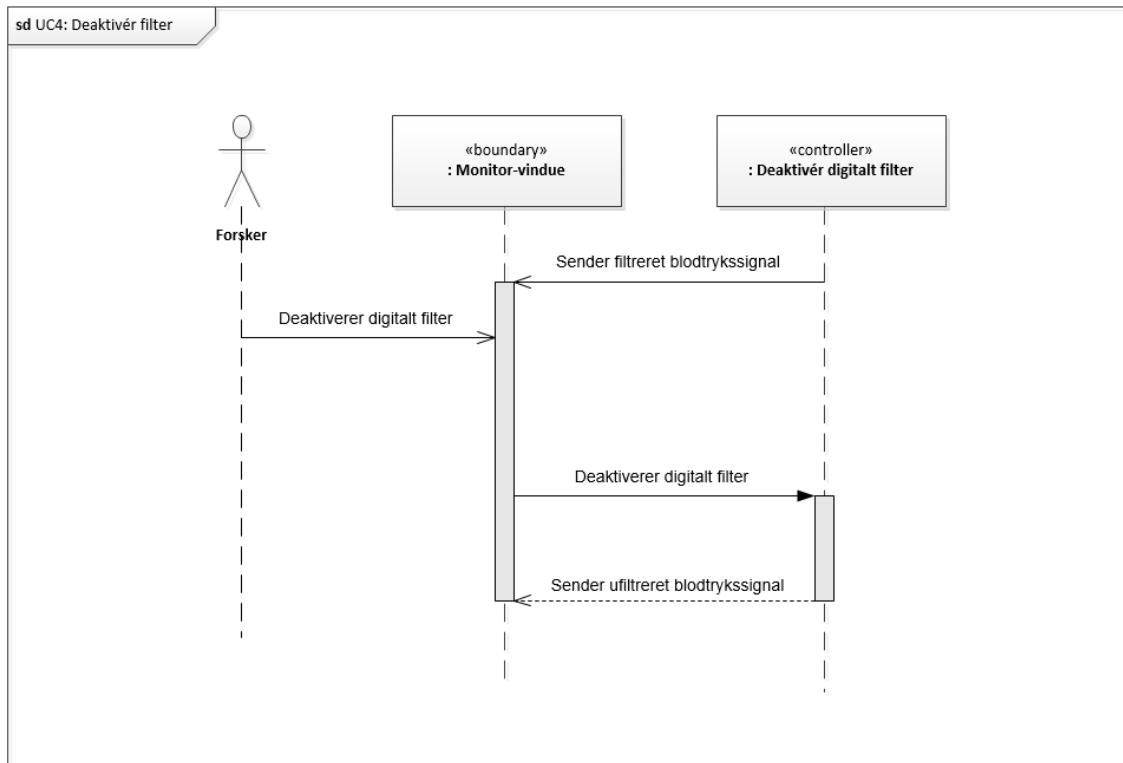
Figur 2.7: Sekvensdiagram for Use Case 2

Controller henter data fra Tryktransducer, som henter data i form af tryk fra Måleobjekt. Datafilerne sendes fra Måleobjekt via Tryktransducer tilbage til Controller, der kalder en metode. Monitor-vinduet opdateres, og herefter kan Forsker aflæse blodtryk.



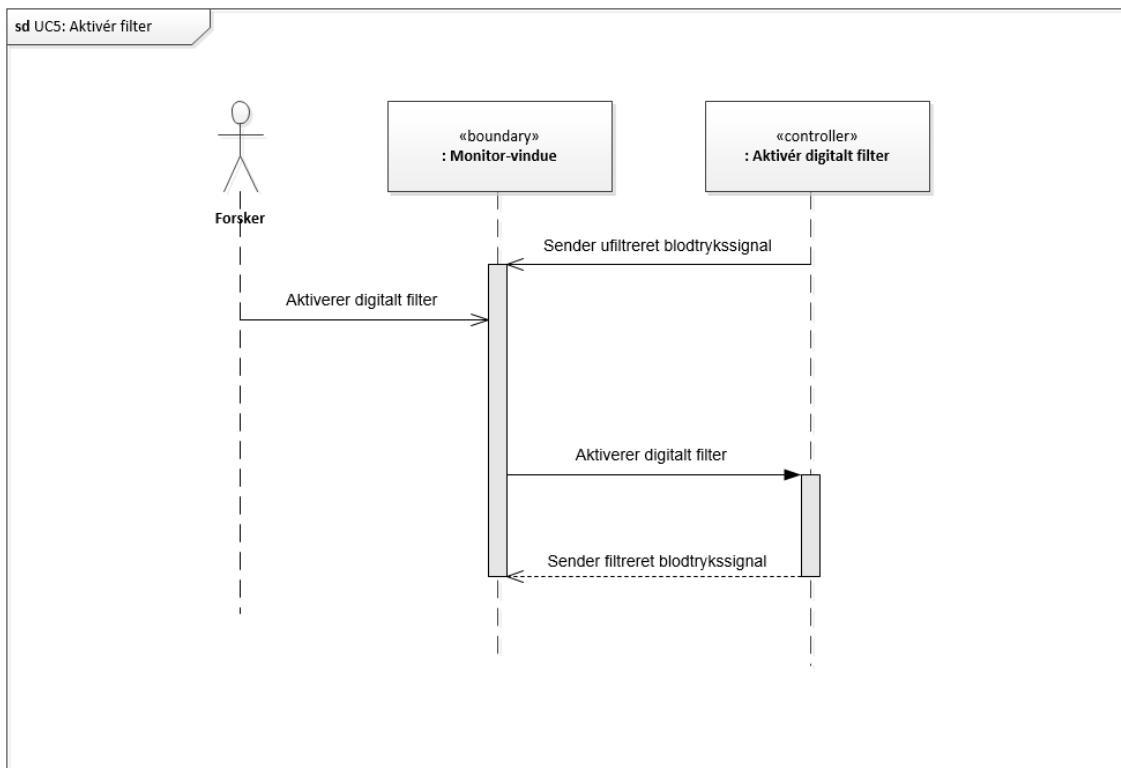
Figur 2.8: Sekvensdiagram for Use Case 3

Forsker interagerer med Monitor-vinduet ved at trykke på knappen ”Nulpunktsjustér”. Derefter kaldes en metode, og nulpunktsjusteringen startes. Tidspunktet og værdien for nulpunktsjusteringen sendes og gemmes i Databasen, hvorefter Forsker får besked om, at nulpunktsjusteringen er foretaget.



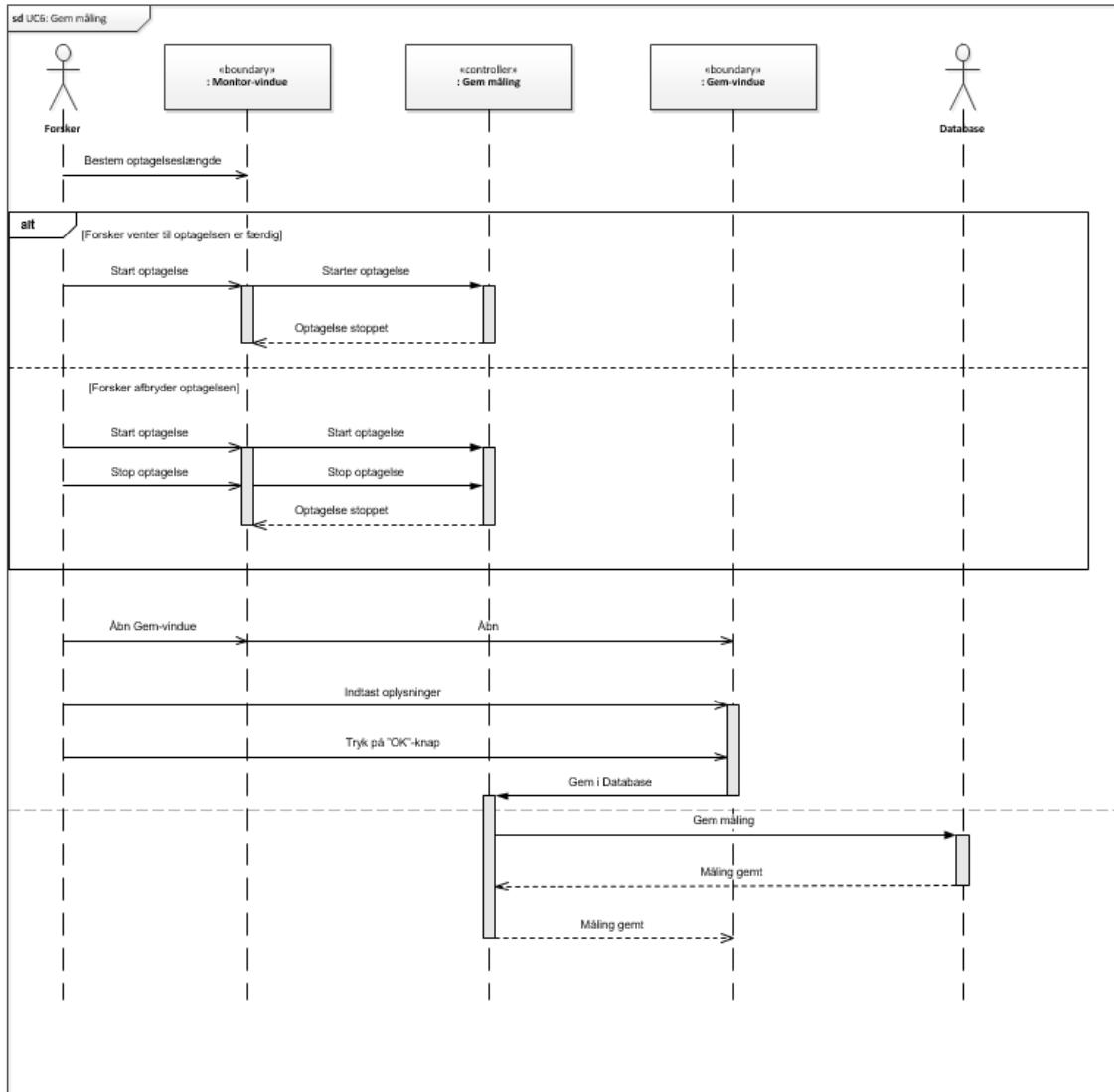
Figur 2.9: Sekvensdiagram for Use Case 4

Forsker interagerer med Monitor-vinduet ved at markere "Off"-radiobutton under "Filter". Derefter kaldes en metode, og filteret deaktiveres, hvorefter signalet bliver udskrevet ufiltreret.



Figur 2.10: Sekvensdiagram for Use Case 5

Forsker interagerer med Monitor-vinduet ved at markere "On"-radiobutton under "Filter". Derefter kaldes en metode, og filteret aktiveres, hvorefter signalet bliver udskrevet filtreret.



Figur 2.11: Sekvensdiagram for Use Case 6

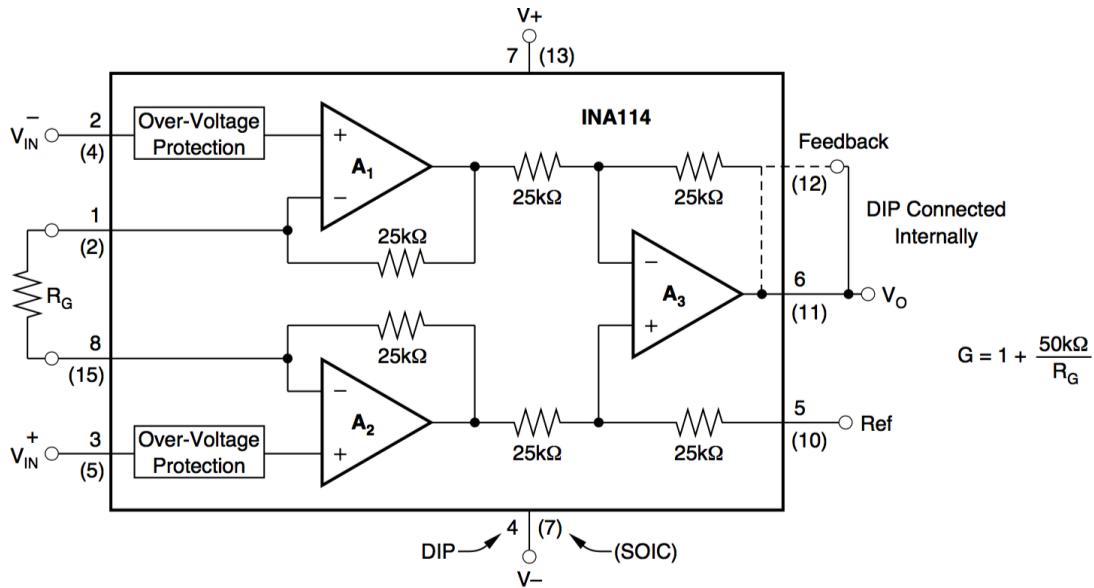
Forsker interagerer med Monitor-vinduet ved at trykke på knappen "Gem". Derefter kaldes en metode og Gem-vinduet åbnes. Når Forsker ønsker at gemme, indtastes data om målingen og der trykkes på knappen "OK". Kommando sendes og data gemmes. Controller bekræfter til Gem-vinduet, at data er gemt.

HW implementering og test 3

3.1 Implementering

3.1.1 Forstærkerblok

Som Forstærkerblok anvendes INA114, se Figur 3.1. Denne har den fordel, at gain kan kontrolleres af en variabel modstand (potentiometer), R_G .

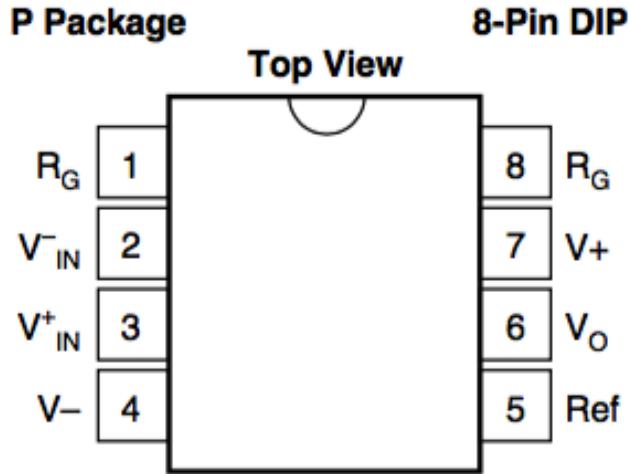


Figur 3.1: INA114

INA114 er blevet udleveret som én komponent. Den sorte boks på Figur 3.1 viser, hvilke delkomponenter INA114 indeholder. Det ses også, at R_G er placeret udenfor den sorte boks, hvilket betyder, at det er en regulerbar modstand, der skal tilkobles INA114.

På Figur 3.2 ses, hvordan INA114¹ skal realiseres i forhold til at implementere Forstærkeren.

¹Se datablad for INA114 i bilag



Figur 3.2: INA114

På ben et og otte forbindes INA114 med R_G . Ben fire forbindes med -9 V, ben syv forbindes med +9 V og ben fem forbindes til jord. Inputsignalet modtages ved ben to og tre, mens outputsignalet kommer fra ben seks, som forbindes med filteret, der efterfølgende viderebeandler signalet.

Forstærkning

Forstærkningen, gain, skal forstærke den maksimale inputspænding fra Tryktransduceren, så outputtet bliver 5 V, hvilket er det interval, der er valgt for DAQ'en. Udregningen for forstærkning ses i ligning (2.4) til (2.5).

$$Gain = 666,667 \quad (3.1)$$

Båndbredde

For INA114 er produktet af båndbredden og forstærkningen en konstant på $1.000.000^2$. Båndbredden for denne Forstærkerblok, hvor gain er 1333,33, er

$$Gain \cdot BW = 1.000.000 \implies BW = \frac{1.000.000}{666,667} \implies BW = 1500Hz \quad (3.2)$$

Båndbredden er større end 50 Hz, hvilket var det, den mindst måtte være.

Gainmodstand

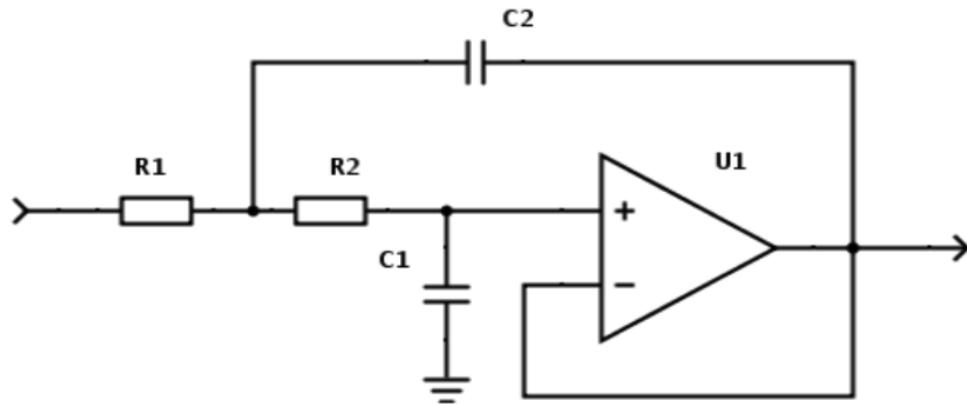
Gainmodstanden, R_G er en variabel modstand, der bestemmer forstærkningen i INA114. INA114 skal have en gain på 666,667. Ud fra ligning (3.3) kan R_G isoleres og beregnes.

²Se datablad for INA114 s. 2 i bilag

$$Gain = 1 + \frac{50k\Omega}{R_g} \Rightarrow R_G = \frac{50k\Omega}{Gain - 1} \Rightarrow R_G = \frac{50k\Omega}{666,667 - 1} \Rightarrow R_G = 75\Omega \quad (3.3)$$

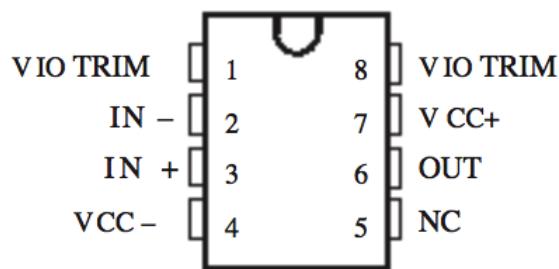
3.1.2 Filterblok

Som filter anvendes et Sallen Key lavpasfilter med unity gain.



Figur 3.3: Sallen Key lavpasfilter

Som ikke-inverterende operationsforstærker anvendes der OP27G³, se Figur 3.4.



Figur 3.4: OP27G

Ben tre modtager det forstærkede signal. Ben to forbindes med ben seks, hvor det behandlede signal kan måles. Ben fire forbindes med -9 V og ben syv forbindes med +9 V.

Beregning af komponentværdier

Ud fra overføringsfunktion for filteret kan de forskellige komponentværdier beregnes.

³Se datablad for OP27G i bilag

Overføringsfunktion

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{1}{R_1 R_2 C_1 C_2}}{s^2 + \frac{R_1 + R_2}{R_1 R_2 C_2} + \frac{1}{R_1 R_2 C_1 C_2}} \quad (3.4)$$

Standardform

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{w_n^2}{s^2 + 2\zeta\omega_n s + w_n^2} \quad (3.5)$$

$$\omega_n = \sqrt{\frac{1}{R_1 R_2 C_1 C_2}} \quad (3.6)$$

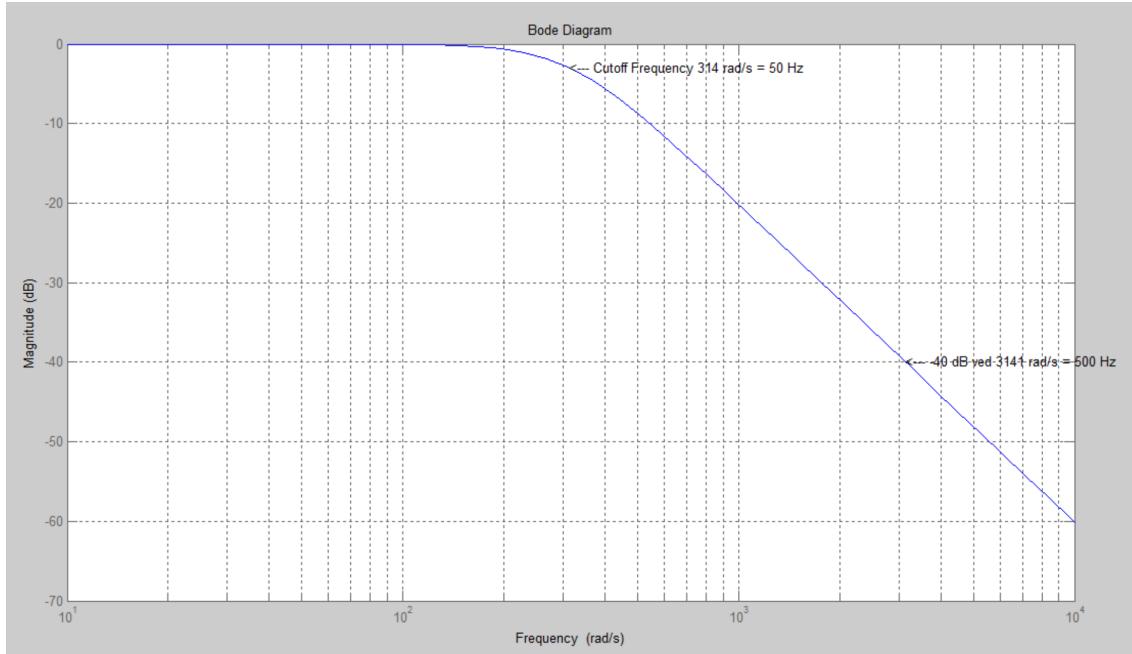
Kondensatoren, C2, skal være 680 nF og kondensatoren, C1, bestemmes til 340 nF, da denne kondensator kunne realiseres ud fra det, der var til rådighed. Der benyttes en kondensator på 330 nF og 10 nF til denne realisering. Modstandene R1 og R2 skal være samme værdi, $R = R_1 = R_2$

$$50 * 2\pi = \sqrt{\frac{1}{R * 340 * 10^{-9} * 680 * 10^{-9}}} \implies \text{Solve, } R \implies R = 6620\Omega \quad (3.7)$$

Ved realiseringen af de to modstande vælge $3k\Omega$ og $3.6k\Omega$.

Magnitude Bodeplot

Bodeplottet i Figur 3.5 viser, at ved -3 dB, hvor cutoff frekvensen befinder sig, er frekvensen 50 Hz, som var et krav. Det ses også, at grafen er faldet med 40 dB en dekade efter, 500 Hz, hvilket også var et krav.



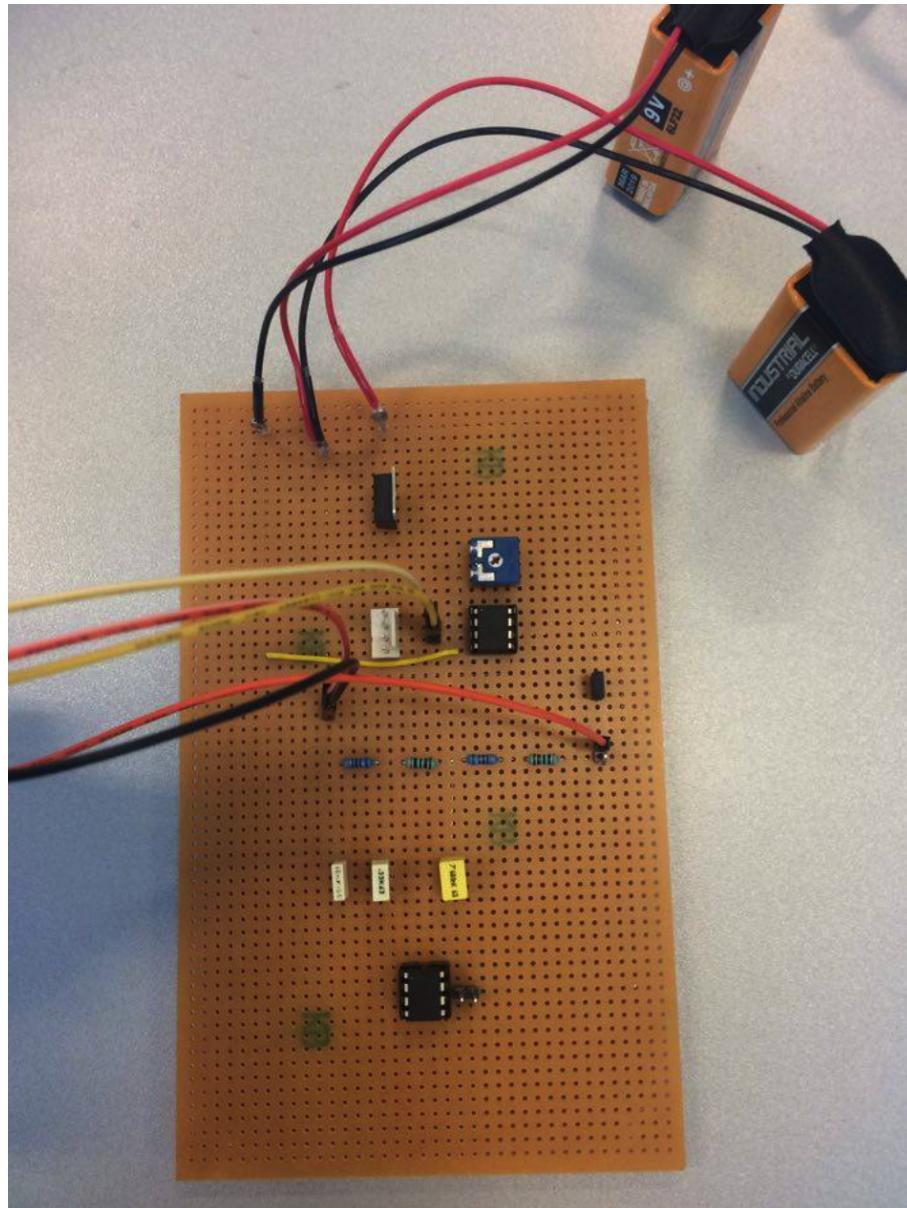
Figur 3.5: Magnitude Bodeplot Lavpasfilter

3.2 Test

Forstærkeren og filteret testes via Analog Discovery & Waveforms. Der foretages først unittest for hver af de to dele, og derefter bliver de sat sammen til en samlet Signalbehandlingsblok, som testes ved en integrationstest.

3.2.1 Test af Forstærkerblok

Testopstillingen kan ses på Figur 3.6.

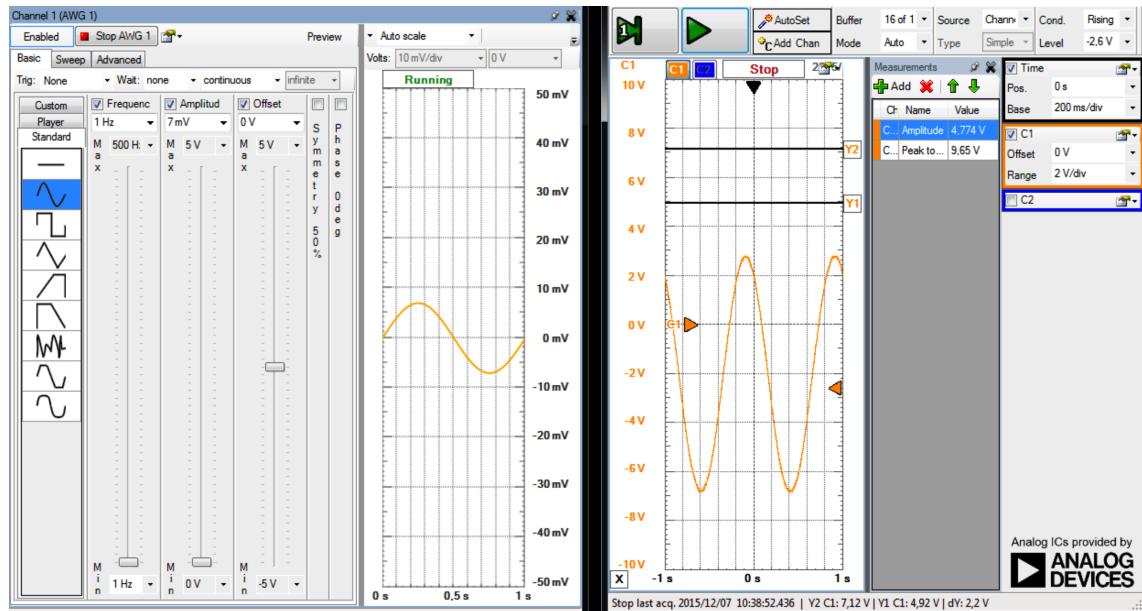


Figur 3.6: Testopstilling for Forstærkeren

Der ønskes en outputspænding på 5 V ved en inputspænding på 7,5 mV. Der kunne ikke påtrykkes en spænding på 7,5 mV, men istedet med 7 mV. Outputspændingen ved 7 mV er

$$7mV \cdot 666,667 = 4,7V \quad (3.8)$$

Det ønskede resultat ses, når man tester Forstærkeren, se Figur 3.7.

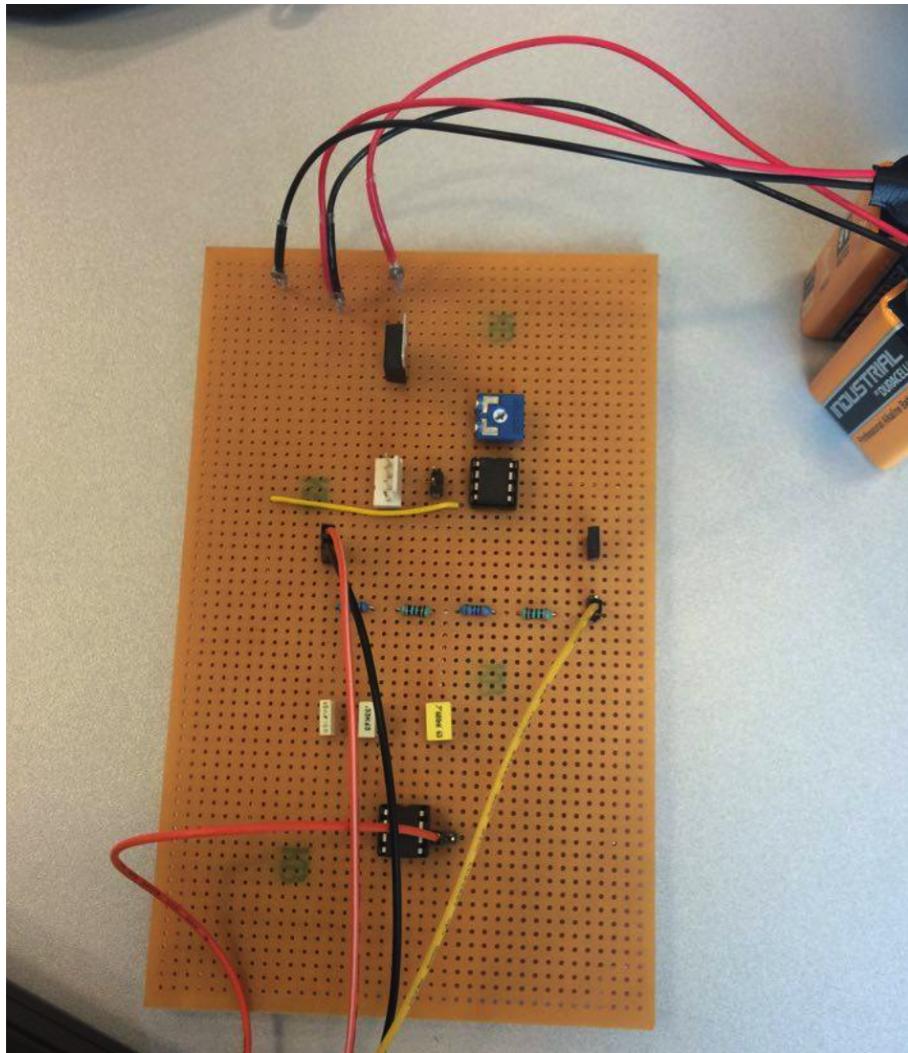


Figur 3.7: Forstærker respons ved 7 mV

3.2.2 Test af Filterblok

Testen af filteret er foretaget, hvor der påtrykkes en AC spænding på 5 V og frekvensen varieres imellem 1 Hz, 50 Hz og 500 Hz.

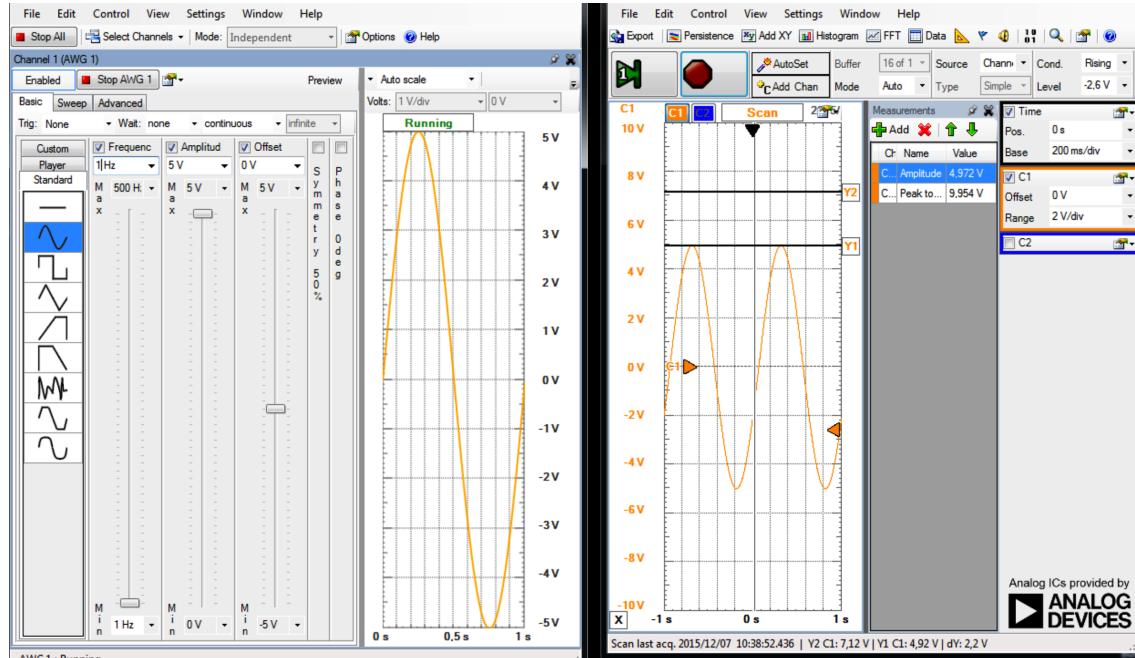
Testopstillingen ses på Figur 3.8.



Figur 3.8: Testopstilling af filteret

Channel 1 er inputspændingen og viser den varierende frekvens. Ændringen ses i venstre side på Figur 3.09, Figur 3.10 og Figur 3.11. Oscilloskopet mäter outputspændingen på filteret og resultatet vises i højre side af efterfølgende skærbilleder.

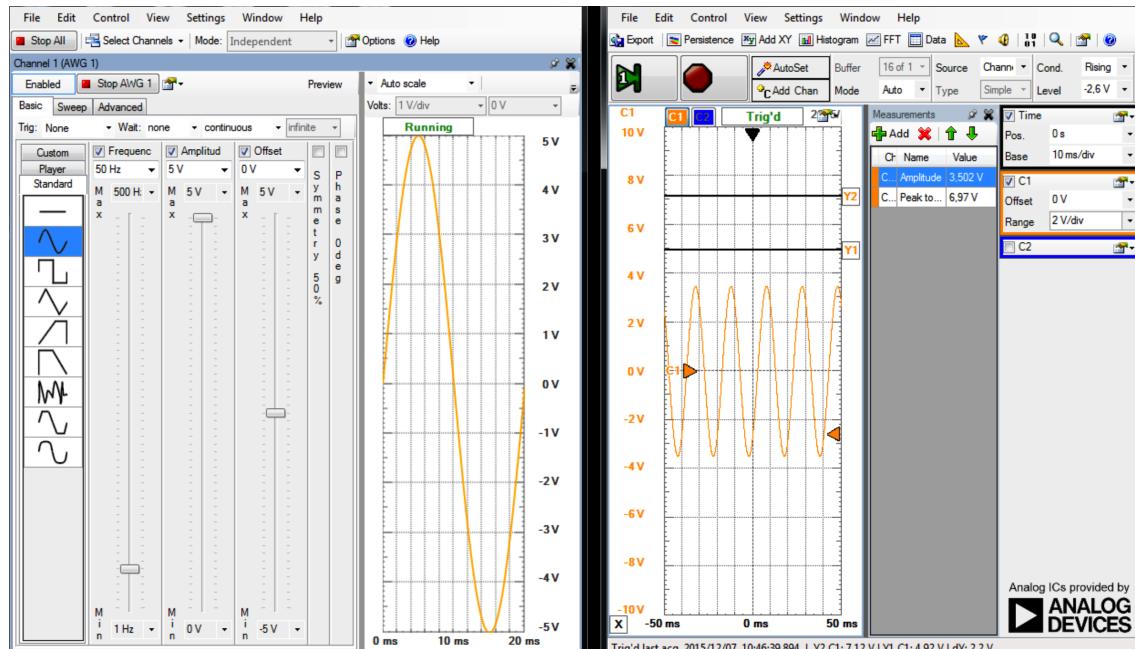
Filtering ved 1 Hz



Figur 3.9: Lavpasfilter respons 1 Hz

På Figur 3.9 ses det, at inputspændingen og outputspændingen er tilnærmelsesvis den samme. Dette er også det ønskede resultat, da filteret er et lavpasfilter, der er designet til at lade de lave frekvenser passere indtil 50 Hz.

Filtering ved 50 Hz



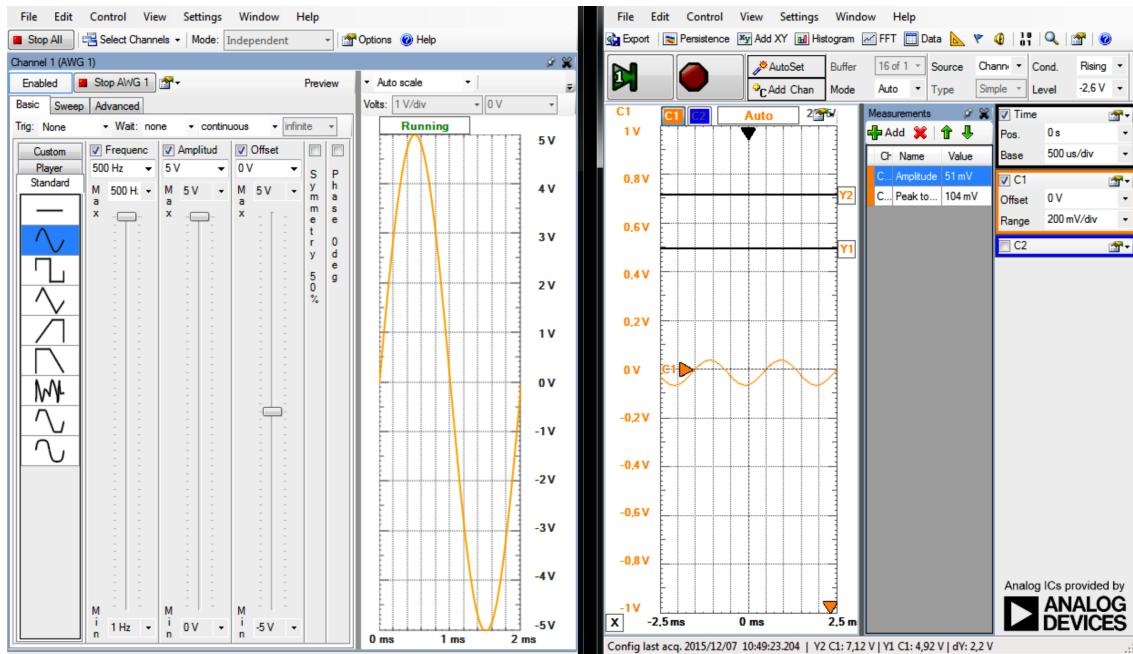
Figur 3.10: Lavpasfilter respons 50 Hz

Ved 50 Hz, som er filterets cutoff frekvens, skal inputspændingen være dæmpet med 3 dB efter filteringen. I ligningen (3.10) beregnes, hvad outputspændingen er efter en dæmpning med 3 dB.

$$3dB = 20 \cdot \log(x) \Rightarrow x = 0,707 \Rightarrow outputspnding = 5V \cdot 0,707 \Rightarrow outputspnding = 3,535V \quad (3.9)$$

På Figur 3.10 ses, at det i praksis er 3,502 V, hvilket der er acceptabelt.

Filtering ved 500 Hz



Figur 3.11: Lavpasfilter respons 500 Hz

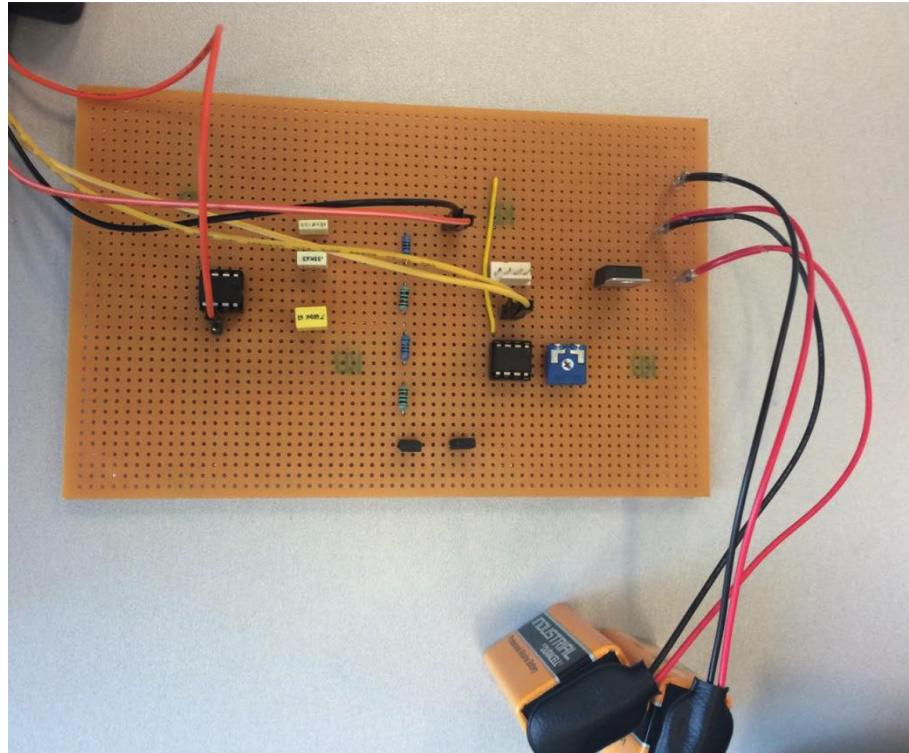
Her forventes der en dæmpning med 40 dB, da det er et anden ordens lavpasfilter, og 500 Hz er en dekade efter cutoff frekvensen. I ligning (3.10) ses beregningen for, hvad outputspændingen er efter en dæmpning på 40 dB.

$$40dB = 20 \cdot \log(x) \Rightarrow x = 0,01 \Rightarrow outputspnding = 5V \cdot 0,01 \Rightarrow outputspnding = 0,05V \quad (3.10)$$

På Figur 3.11 ses, at det i praksis er 51 mV. Det omregnes til 0,051 V, hvilket er acceptabelt i forhold til teorien.

3.2.3 Test af Signalbehandlingsblok

Testopstillingen ses på Figur 3.12.



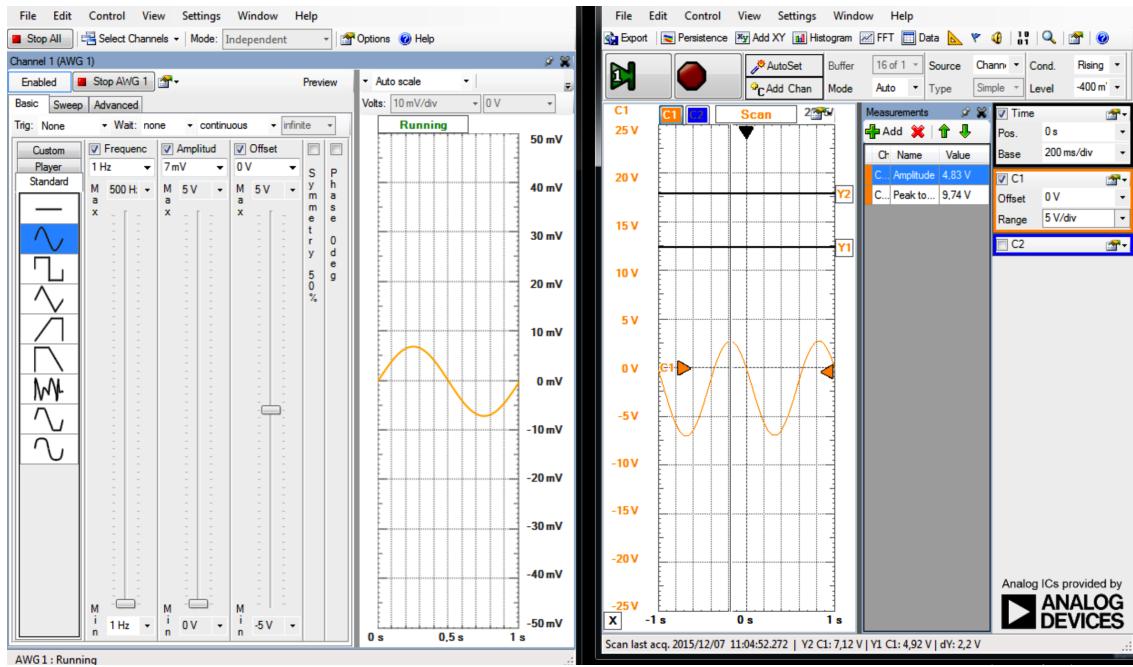
Figur 3.12: Testopstilling af Signalbehandlingsblok

Der laves tre forskellige tests, hvor frekvensen varieres mellem 1 Hz, 50 Hz og 500 Hz. Inputspændingen er i alle tests 13 mV.

Test ved 1 Hz

På Figur 3.13 ses resultatet ved 1 Hz. Der forventes, at Forstærkeren har forstærket signalen op til 5 V samt, at filteret ikke har påvirket outputtet, da frekvensen er under 50 Hz, som er filteret cutoff frekvens.

I praksis er det lig med 4,83 V, hvilket er acceptabelt.

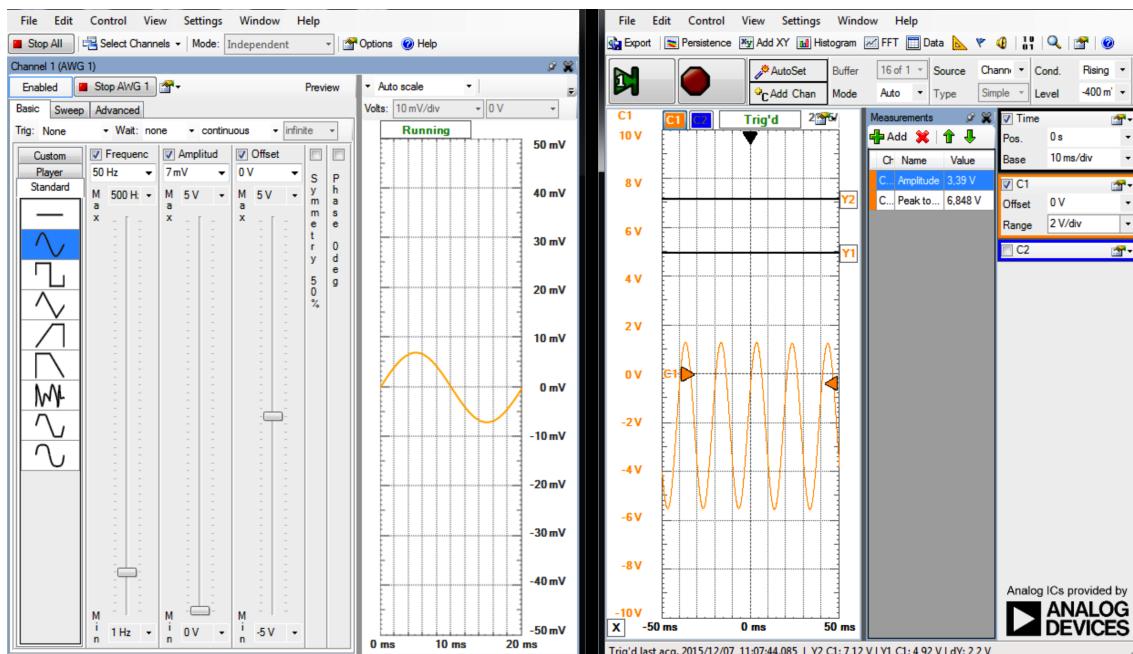


Figur 3.13: Signalbehanlding respons 1 Hz

Test ved 50 Hz

På Figur 3.14 ses resultatet ved 50 Hz. Der forventes, at Forstærkeren har forstærket signalet op til 5 V samt, at filteret har dæmpet outputspændingen med 3 dB, så outputspændingen er lig med 3,39 V.

I praksis er det lig med 3,39 V, hvilket er acceptabelt.

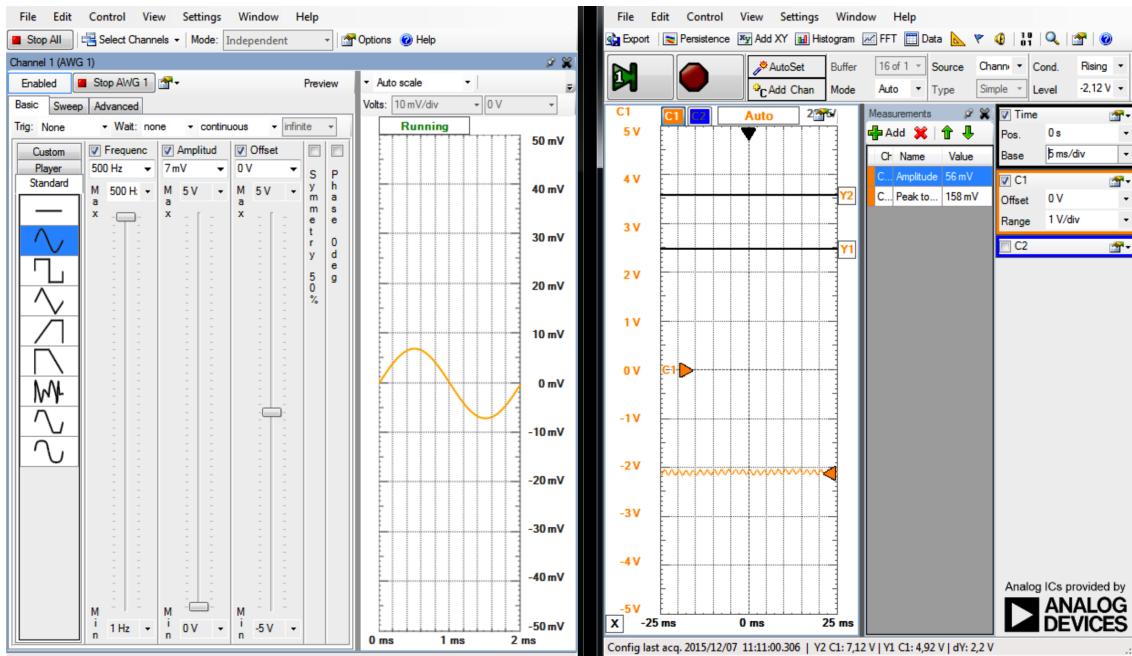


Figur 3.14: Signalbehanlding respons 50 Hz

Test ved 500 Hz

På Figur 3.15 ses resultatet ved 500 Hz. Der forventes, at Forstærkeren har forstærket signalet op til 5 V samt, at filteret har dæmpet outputspændingen med 40 dB, så outputspændingen er lig med 0,05 V.

I praksis er det lig med 0,056 V, hvilket er acceptabelt.

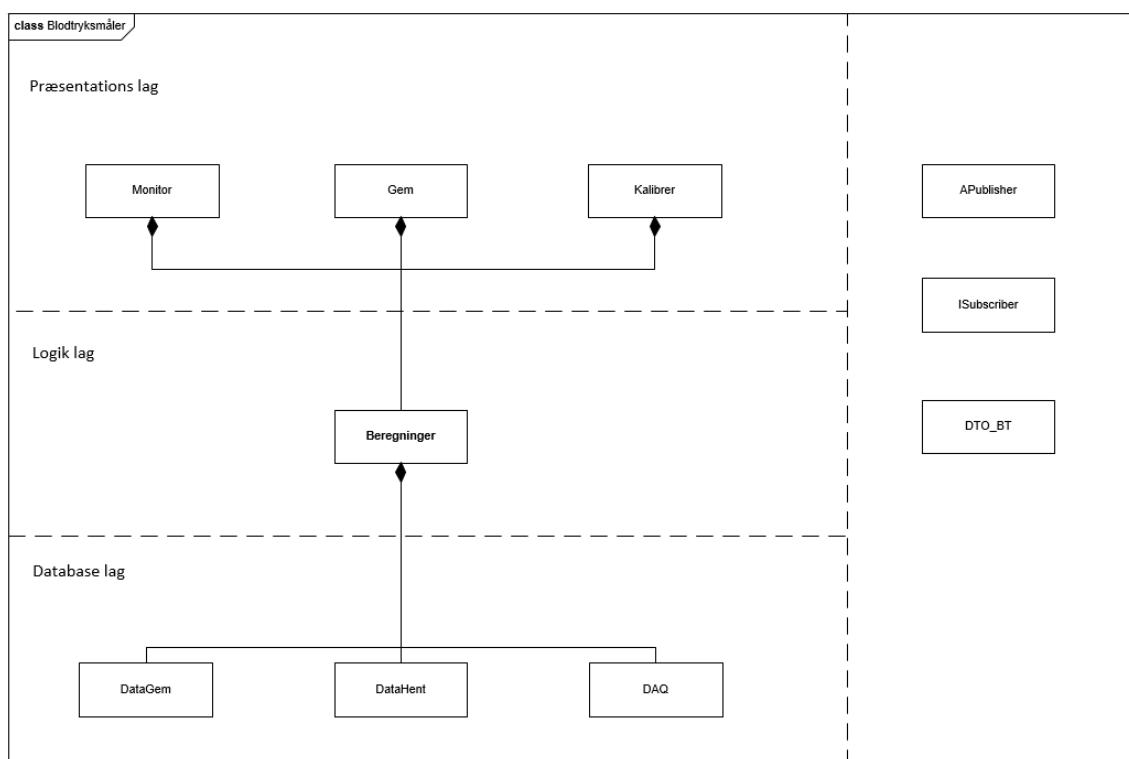


Figur 3.15: Signalbehanlding respons 500 Hz

SW implementering og test 4

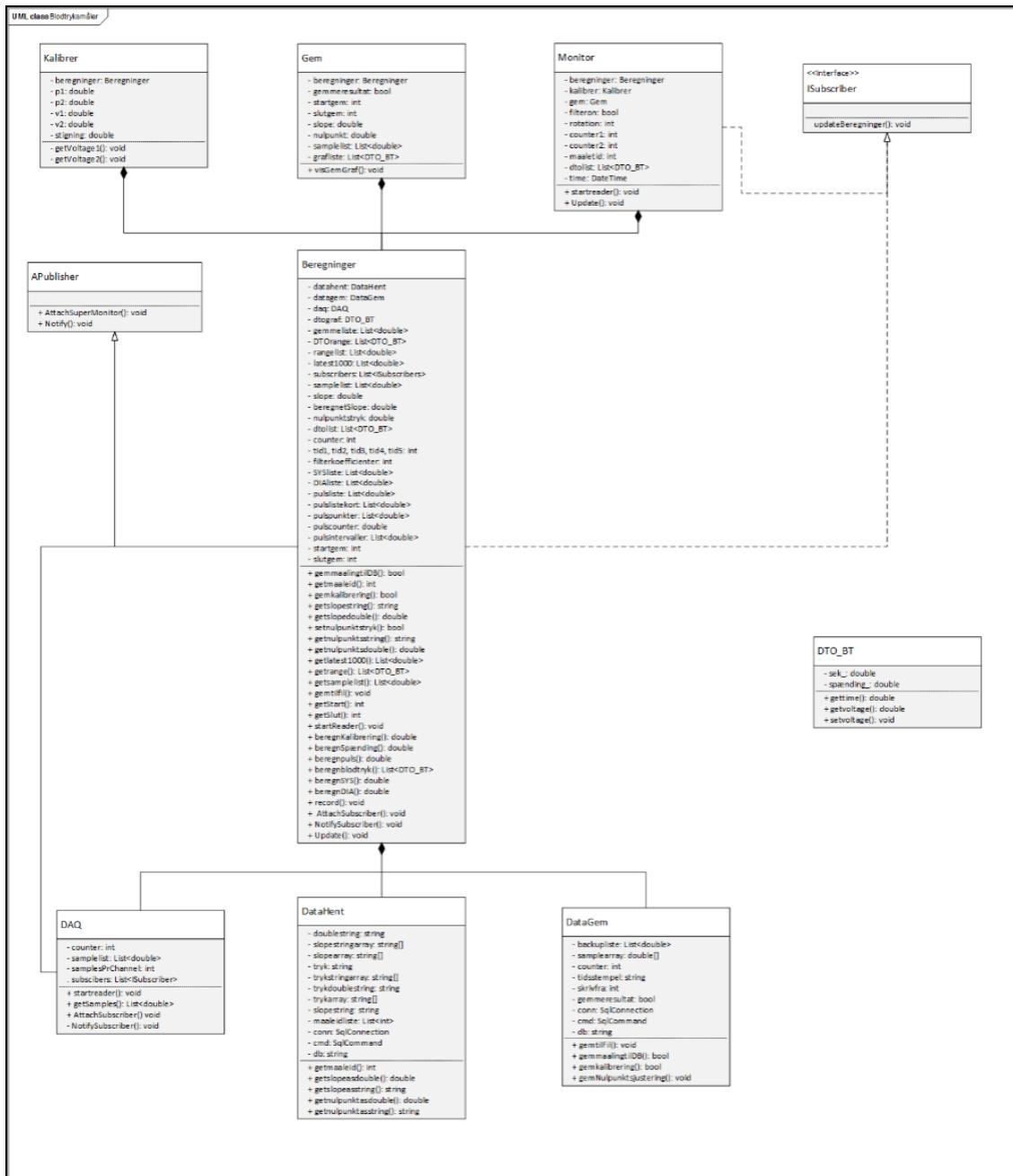
4.1 Implementering

Softwareen i blodtryksmålesystemet er opbygget omkring trelagsmodellen, se Figur 4.1.



Figur 4.1: UML kompositions klassediagram

For at danne et overblik over de forskellige klasser og deres metoder, er der udarbejdet et UML klassediagram. Klassediagrammet beskriver ligeledes relationerne mellem klasserne ved komposition, arv og interface, se Figur 4.2.



Figur 4.2: Klassediagram

4.1.1 Datalag

Datalaget kommunikerer med DAQ, Database og computerens harddisk.

DAQ

Klassen DAQ består hovedsageligt af software fra National Instruments, der kan udveksle data med den fysiske DAQ 6009. Denne kan betragtes som af typen "black box". Dette betyder, at udvikleren ikke har et begreb om den interne mekanisme i klassen, men udelukkende ser på outputtet. Som udgangspunkt er funktionen af klassen at levele data

omkring de målinger, der foretages i den fysiske DAQ.

De styrbare parametre i denne klasse er blandt andet samplingfrekvens og måleområde.

Datahent

Klassen henter data fra en Database og lokale tekstmæssige filer. Fra Databasen indhentes data om tidligere målinger, der nummereres således at de målinger, der skal gemmes i fremtiden, kan blive nummereret korrekt.

De lokale tekstmæssige filer indeholder data omkring tidligere kalibreringer og nulpunktsjusteringer.

DataGem

Klassen gemmer data på Database og lokale tekstmæssige filer.

4.1.2 Logiklag

Beregninger

Beregninger er hovedsageligt et lag, hvor logiske beregninger foretages.

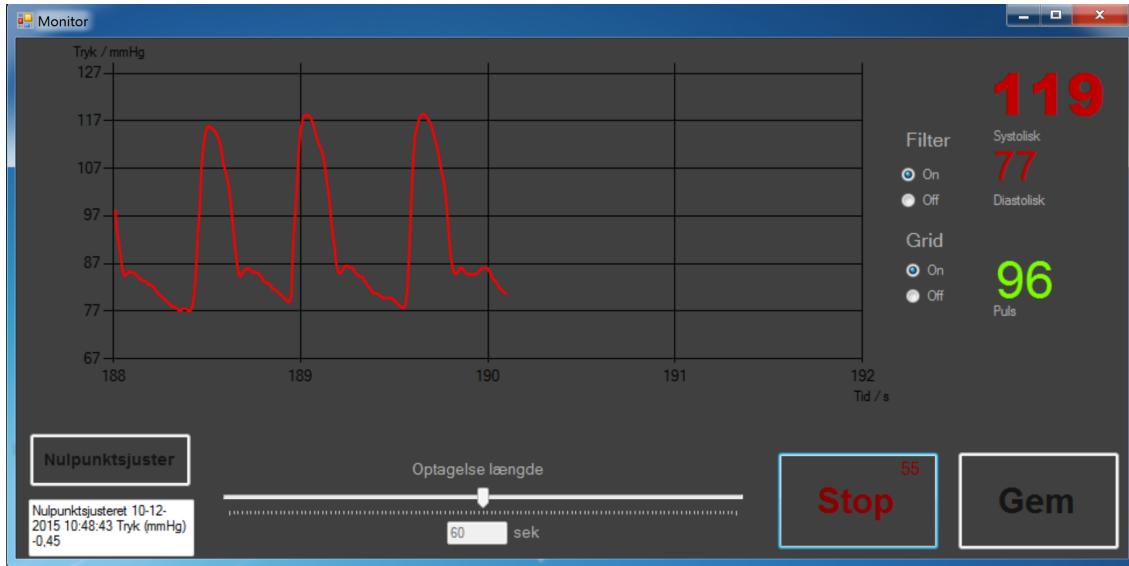
Herunder beregnes for eksempel systolisk og diastolisk blodtryk samt puls. Beregninger i forbindelse med kalibrering og nulpunktsjustering foregår ligeledes her. Ydermere skaber logiklaget forbindelse mellem datalaget og præsentationslaget, hvilket tillader, at metoder fra datalaget kan anvendes fra præsentationslaget.

4.1.3 Præsentationslag

Monitor

Monitorlaget er Forskerens generelle visuelle interaktionslag med systemet. Heri observeres udskrevne blodtrykssignaler over 4-sekunders intervaller. Værdierne puls, systolisk og diastolisk blodtryk udskrives ligeledes her. Designet er inspireret fra i forvejen eksisterende blodtryksmonitoreringssystemer.

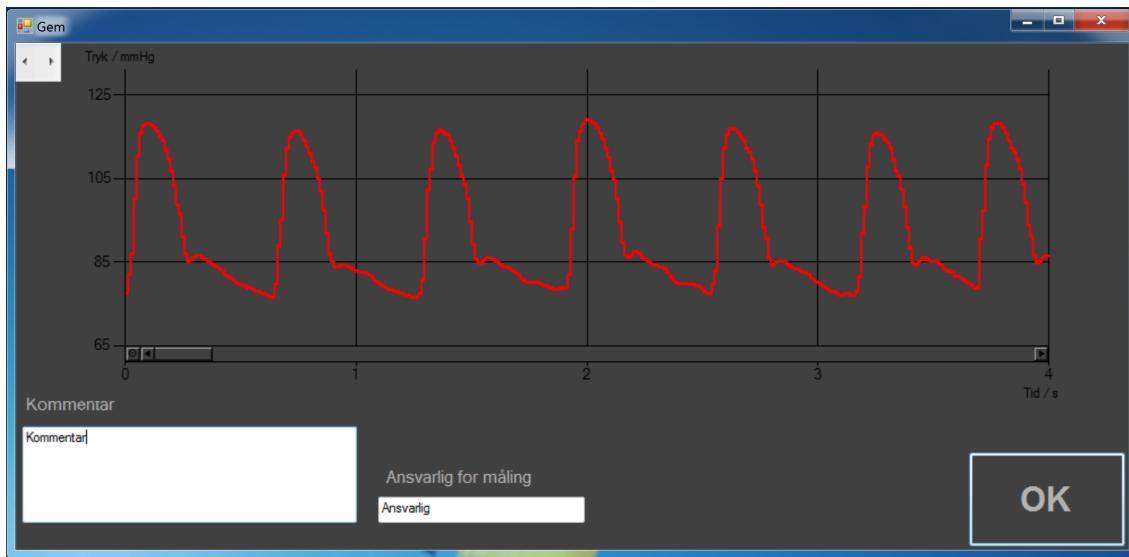
Der er mulighed for at vælge en optagelseslængde for målingen, samt starte og stoppe den via en "Rec"-knap/"Stop"-knap. Herefter kan Gem-vinduet åbnes, hvor den optagede sekvens kan gemmes. Derudover er der en knap til udførelse af nulpunktsjustering. Den kørende måling kan vises med eller uden et filter ved at ændre på de to radiobuttons, "On"/"Off". Y- og X-akse kan påsættes med to radiobuttons, "On"/"Off". Se Figur 4.3.



Figur 4.3: Monitor med igangsæt måling

Gem

Vinduet udskriver den optagede sekvens i et vindue, der visuelt ligner Monitor-vinduet. Der er to tekstmærker, hvor en kommentar og et navn for den ansvarlige af målingen, kan indtastes. Se Figur 4.4.



Figur 4.4: Gem-vindue med 10 sekunders måling

Kalibrér

Vinduet indeholder to tekstmærker til indtastning af tryk samt tekstmærke til dertilhørende spænding. Spændingerne kan måles automatisk ved tryk på "Mål"-knapperne. Kalibreringskonstanten kan udregnes med knappen "Beregn" og gemmes samt anvendes ved at trykke på "OK"-knappen.

4.1.4 Datamodellaget

Datamodellaget muliggør indirekte dataoverførsel mellem lag, der umiddelbart ikke har relation, der tillader dette.

DTO

DTO er et namespace, som indeholder data transfer object (DTO). Alle de andre lag har adgang til dette lag og kan oprette instanser af klassen DTO.

Denne klasse har attributterne tid og spænding. Formålet med denne klasse er at overføre sampleværdier fra DAQ til et format, der har en korrekt tidsværdi, samt et tryk korrigert for nulpunktstryk og kalibrering.

```
namespace DTO
{
    public class DTO_BT
    {
        private double sek_;
        private double spænding_;

        public DTO_BT(int sample, double spænding)
        {
            sek_ = Convert.ToDouble(sample) / 1000;
            spænding_ = spænding;
        }

        public double gettime()
        {
            return sek_;
        }

        public double getvoltage()
        {
            return spænding_;
        }

        public void setvoltage(double newpressure)
        {
            spænding_ = newpressure;
        }
    }
}
```

Figur 4.5: DTO klassen med metoder

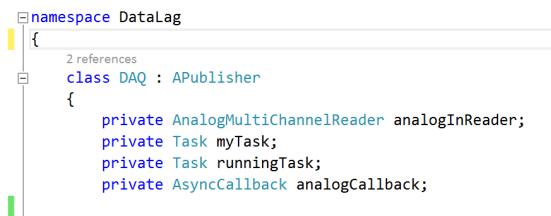
Publisher/subscriber pattern

Datastrømmen fra DAQ klassen er styrende for alle de andre klasser, blandt andet udskrivning af blodtrykket i Monitor-vinduet. Den fysiske DAQ mäter med 1000 Hz og returnerer data til DAQ klassen med 50 ms intervaller. Dette måleinterval indeholder 50 sampleværdier. Disse data sendes videre til Beregningerklassen, og fra Beregningerklassen til Monitor-klassen ved hjælp af publisher/subscriber mønstret. Når data befinner sig i Monitorklassen kan der laves logiske beregninger og manipulation på data i Beregningerklassen.

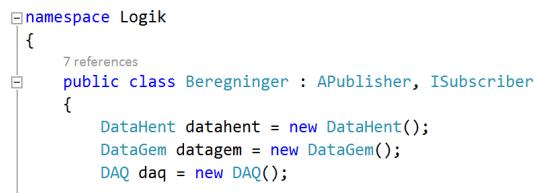
Forholdet mellem klasserne er således, at DAQklassen og Beregningerklassen har et publisher/subscriber forhold, og Beregninger- og Monitorklassen har et publisher/subscriber

forhold.

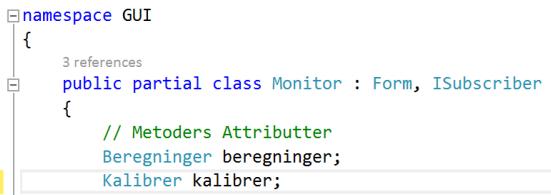
Hierakiet fremgår af Figur 4.6, Figur 4.7, og Figur 4.8.



Figur 4.6: DAQ klassens arvehierarki



Figur 4.7: Beregninger klassens arvehierarki



Figur 4.8: Monitor klassens arvehierarki

4.1.5 Metodebeskrivelser

De seks Use Cases er beskrevet ved hjælp af sekvensdiagrammer med korte beskrivelser herunder. Dette gøres for at se, hvilke klasser og metoder i softwaren der bruges i de forskellige Use Cases.

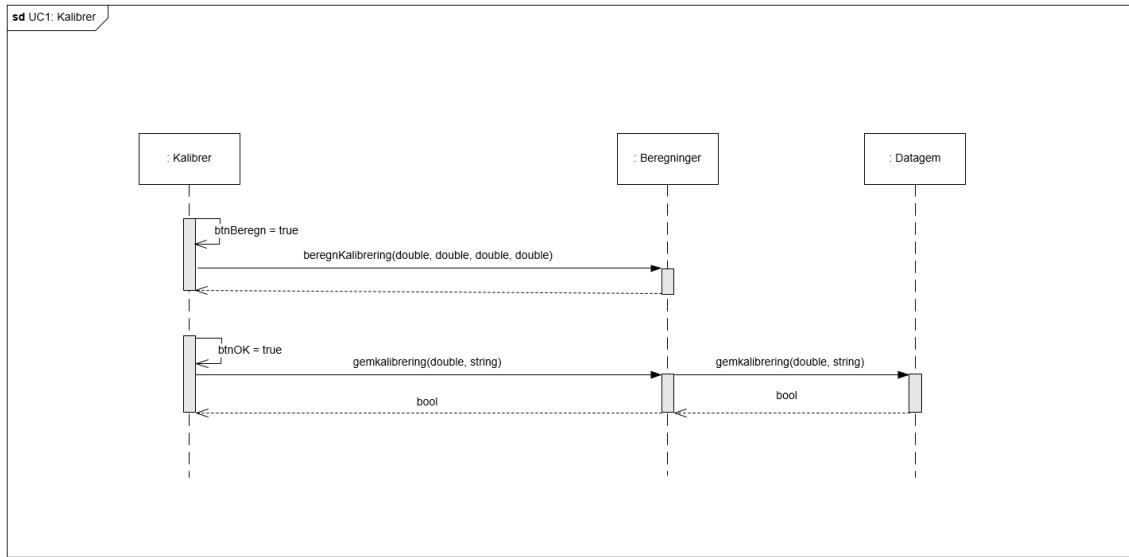
Sekvensdiagrammerne overskueliggør klassernes involvering i de forskellige Use Cases og beskriver kommunikationen herimellem.

Aktivitetsdiagrammer beskriver udførelsen af enkelte metoder step-by-step. Dette giver et fuldkommen indblik i metodens workflow.

Use Case 1: Kalibrer

Forsker trykker på knappen "Beregn" og anmoder Kalibreringsklassen om metoden 'beregn' i Beregningerklassen. Den udregnede kalibreringskonstant returneres.

Forsker trykker på knappen "OK" og metoden 'gemkalibrering' afvikles i Beregningerklassen. Herfra afvikles metoden 'gemkalibrering' i DataGemklassen. DataGem returnerer en bool, der beskriver om gemningen var succesfuld.

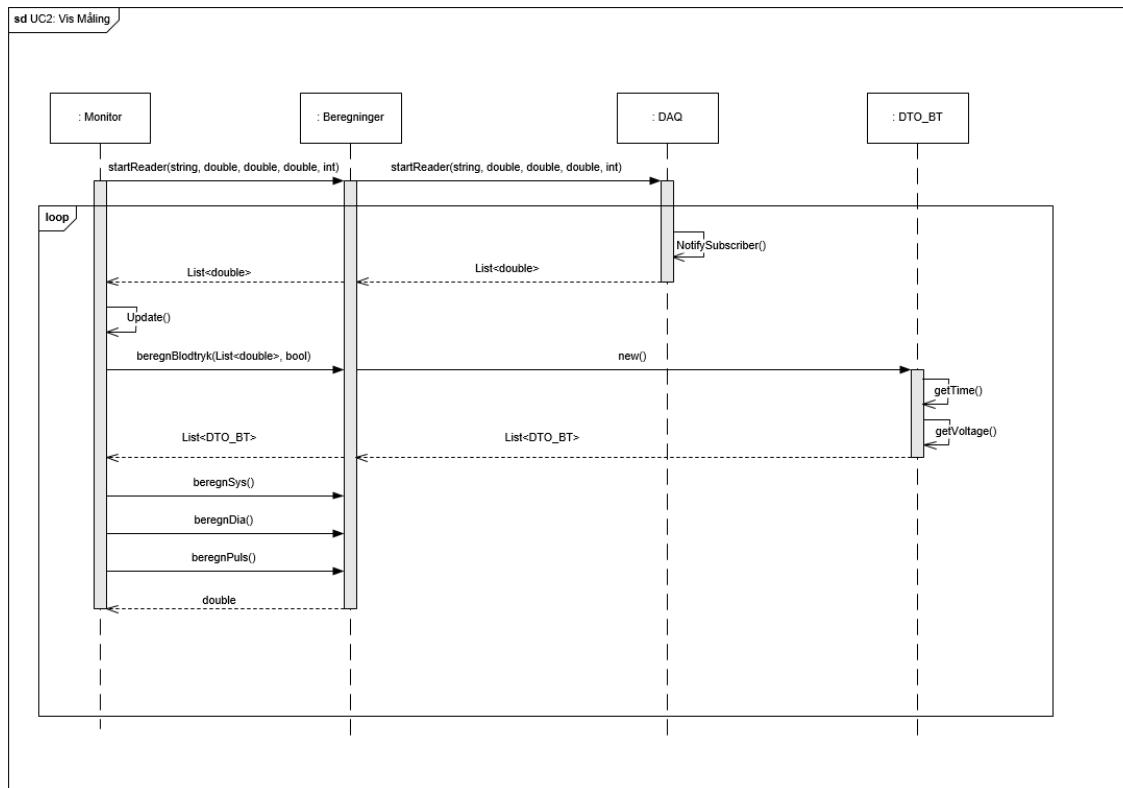


Figur 4.9: Sekvensdiagram for Use Case 1

Use Case 2: Vis måling med filter

Når Monitor-vinduet åbnes ved opstart af program afvikles metoden 'startreader' i DAQ klassen. Ved hjælp af publisher/subscriber metoden, 'NotifySubscriber', returneres rådata til Beregninger og videre til Monitor.

Rådata sendes til Beregninger og omregnes til tid og blodtryk ved hjælp af DTO klassen. 'BeregnSYS', 'beregnDIA' og 'beregnPuls' returnerer ligeledes double værdier med systolisk og diastolisk blodtryk samt puls.



Figur 4.10: Sekvensdiagram for Use Case 2

Metoden, der beregner puls, har to logiske grene til udregning af pulsen. Den første gren til venstre i Figur 4.11 beregner pulsen, når der er mellem 3000 og 10000 samples til rådighed - altså når den fysiske DAQ har målt mellem 3 og 10 sekunder.

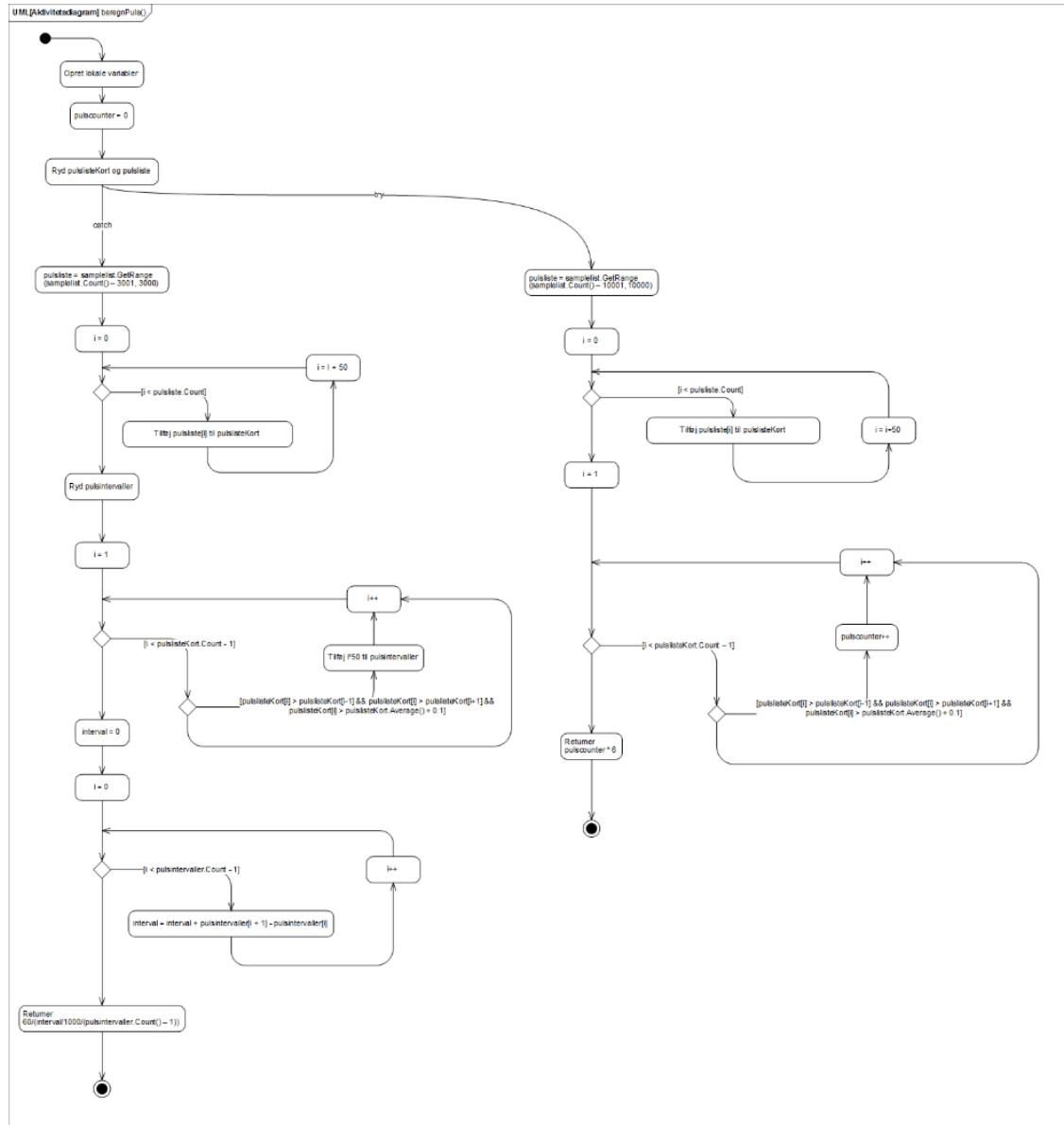
Metodikken består i at lokalisere toppunkter i signalet for at udregne det gennemsnitlige interval herimellem. Herudfra kan pulsen estimeres.

En for-løkke bruges til at evaluere alle sampleværdier i forhold til de førstliggende og efterfølgende samples. Hvis værdien på den pågældende sample har større værdi end den førstliggende og efterfølgende, gemmes tidspunktet for dette sample i en liste.

For at sikre at toppunktet ikke er et mindre toppunkt, fra eksempelvis støj eller højfrekvent signalindhold, evalueres der kun for hver 50. værdi, og tidspunktet gemmes kun, hvis værdien ligger 0,1 V over gennemsnittet af alle sampleværdier der evalueres. Ud fra gennemsnitsintervallet mellem toppunkterne beregnes pulsen.

Den anden logiske gren regner pulsen, når der er over 10000 samples til rådighed - altså efter den fysiske DAQ har målt i over 10 sekunder.

Metodikken er den samme som ovenfor, men i stedet for at gemme et tidspunkt for toppunktet, tælles en counter op. Når antallet af toppunkter for 10 sekunder er fundet, kan pulsen beregnes.

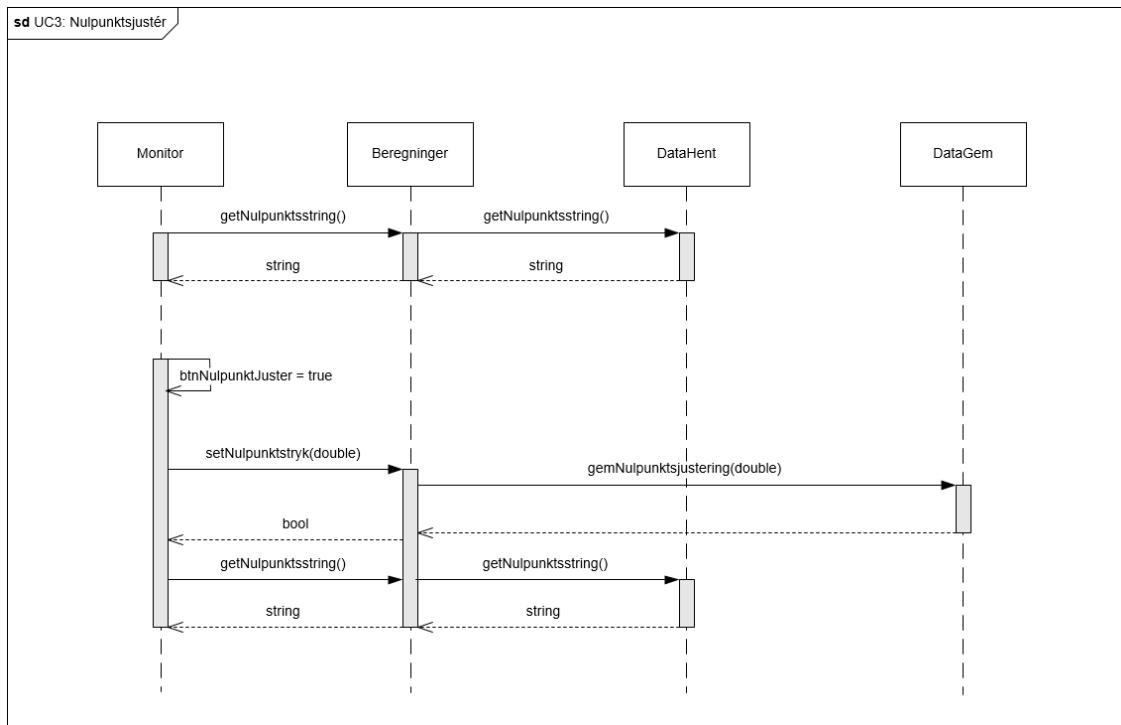


Figur 4.11: Aktivitetsdiagram over beregnpuls

Use Case 3: Nulpunktjuster

'GetNulpunktsstring' returnerer en string, der indeholder data om seneste nulpunktsjustering. Denne udskrives på Monitor.

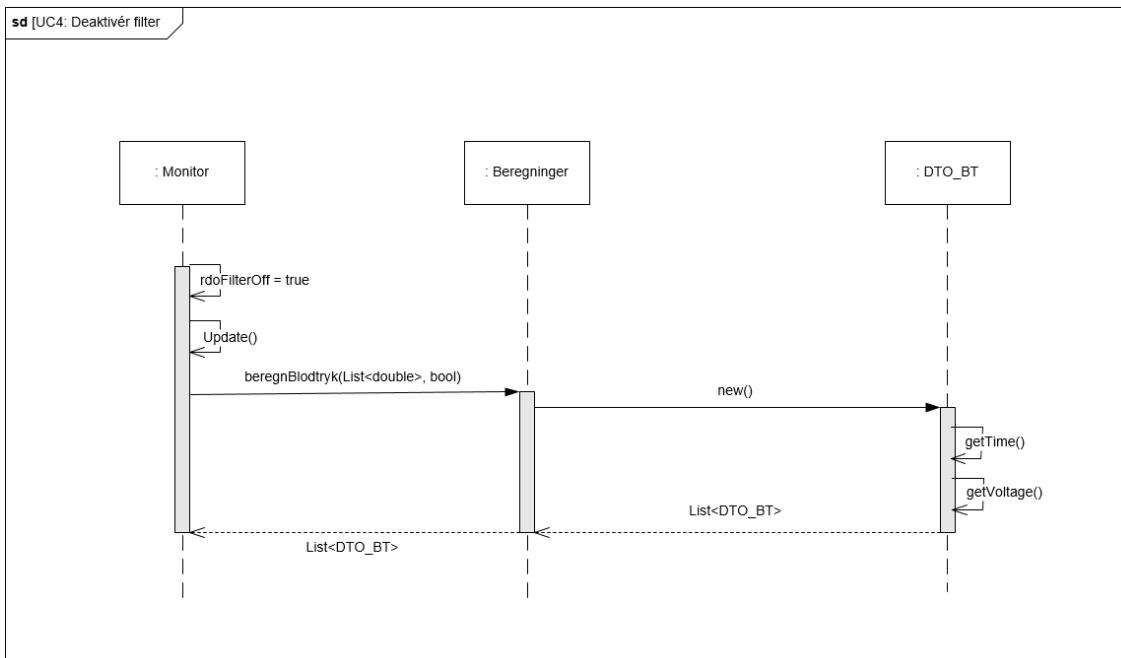
Forsker trykker på knappen 'Nulpunktjuster', og Beregninger udfører en nulpunktsjustering, gemmer data i DataGem og returnerer en ny string der beskriver den seneste nulpunktsjustering.



Figur 4.12: Sekvensdiagram for Use Case 3

Use Case 4: Deaktivér filter

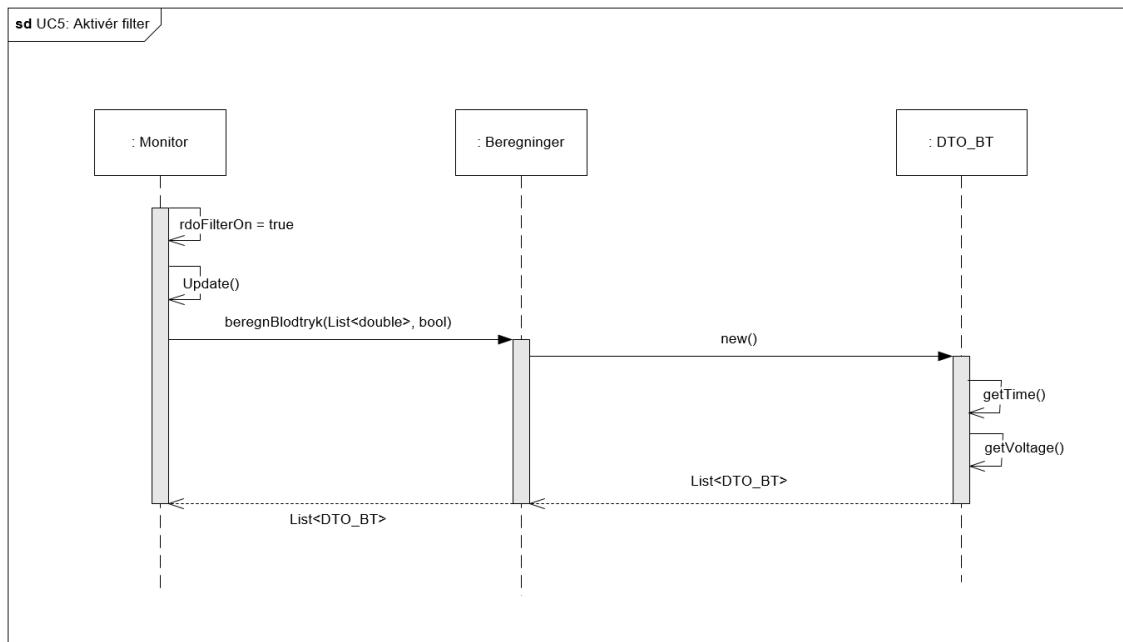
Forsker sætter radiobutton for filter til "Off". Boolen "filteron" sættes til 'false', og sendes med metoden, der udregner blodtryk 'beregnBlodtryk', i Beregningerklassen. Den returnerede liste af DTO'er, som udskrives i Monitor, er ufiltrerede.



Figur 4.13: Sekvensdiagram for Use Case 4

Use Case 5: Aktiver filter

Filteret er bygget op som et FIR-filter med 10 filterkoefficienter. Hvert udskrevet punkt på grafen er en gennemsnitsværdi af de foregående 10 værdier. Forsker sætter radiobutton for filter til "On". Boolen "filteron" sættes til 'true', og sendes med metoden, der udregner blodtryk 'beregnBlodtryk', i Beregningerklassen. Den returnerede liste af DTO'er, som udskrives i Monitor, er filtrerede.



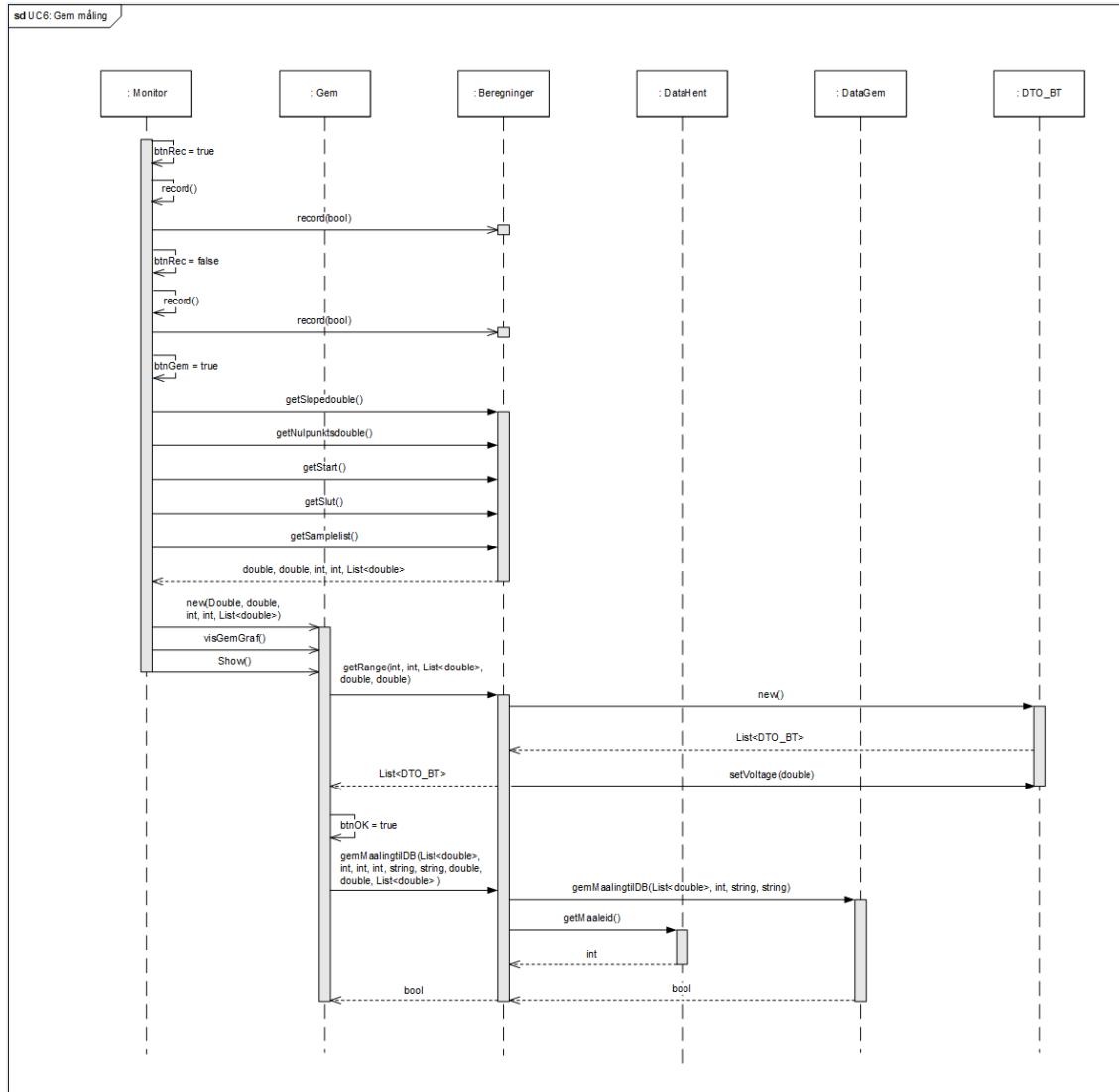
Figur 4.14: Sekvensdiagram for Use Case 5

Use Case 6: Gem måling

Forsker vælger en optagelseslængde og trykker på "Rec". Metoden record() køres i Beregningerklassen og der sættes en startværdi for målingen. Når tiden udløber, køres record() i Beregningerklassen igen og en slutværdi for målingen sættes. Forsker trykker på "Gem"-knappen i Monitor-vinduet og et Gem-vindue instantieres med data fra den optagede måling. Disse hentes via metoder i logiklaget.

Det optagede signal vises i Gem-vinduet. Dette gøres ved hjælp en liste af DTO'er, der oprettes.

Forsker trykker på "OK"-knappen og metoden gemmaalingtilDB() afvikles i DataGem klassen. Det målte signal og metadata gemmes.



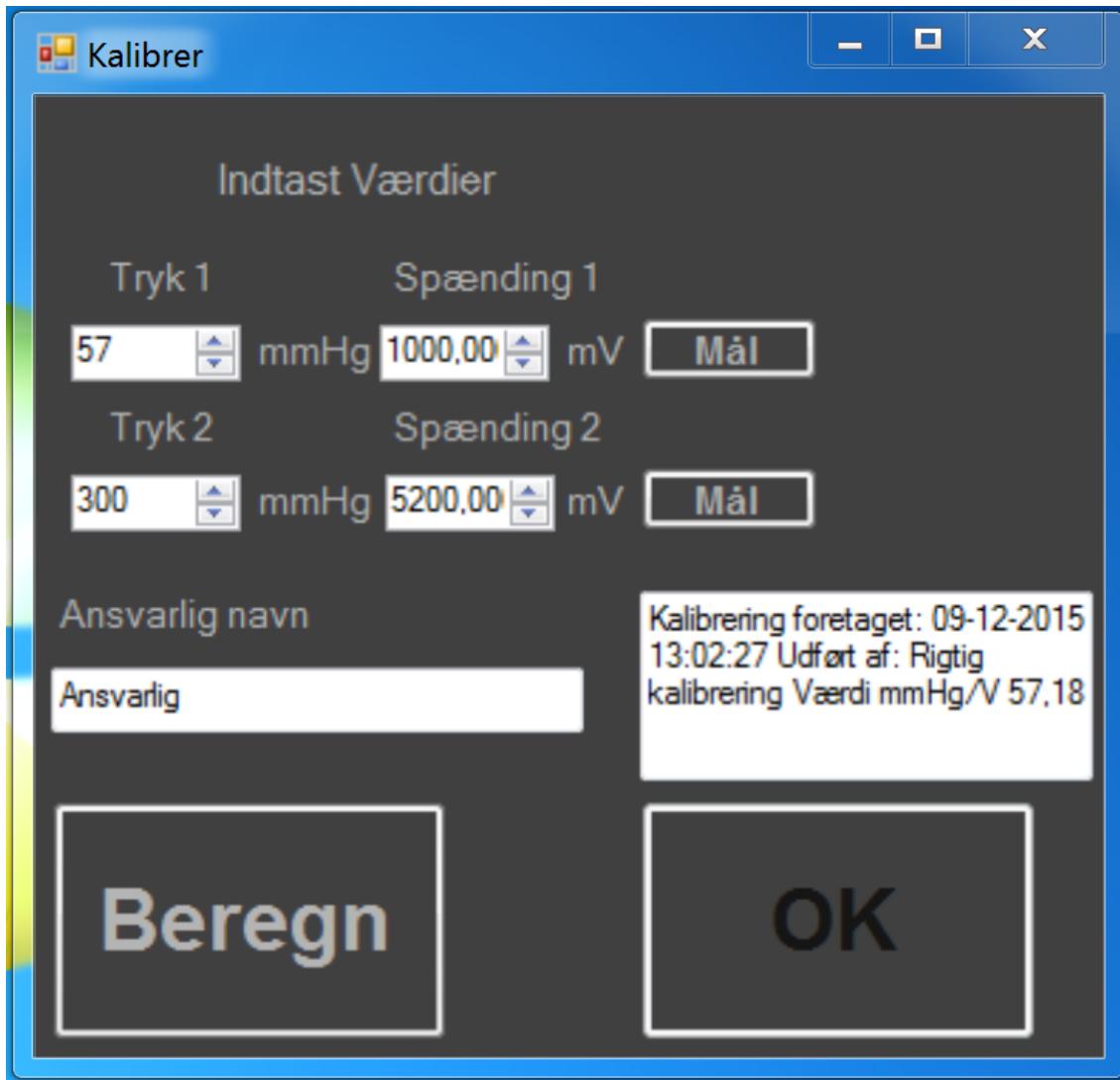
Figur 4.15: Sekvensdiagram for Use Case 6

4.2 Test

Til at teste de forskellige metoder i softwaren er der blevet benyttet debugging. Debugging gør det muligt at følge compilerens sekventielle udførelse af programmet. Når der udføres handlinger, og attributværdier sættes, kan dette følges og sættes op mod de forventede resultater. Såfremt der skulle opstå fejl, gør debugging det muligt at fokusere på enkelte metoder og undersøge, hvor fejlen opstår.

4.2.1 Use Case 1: Kalibrer

Forsker indtaster målte kalibreringsdata. Kalibreringsdata for seneste kalibrering vises. Se Figur 4.16.

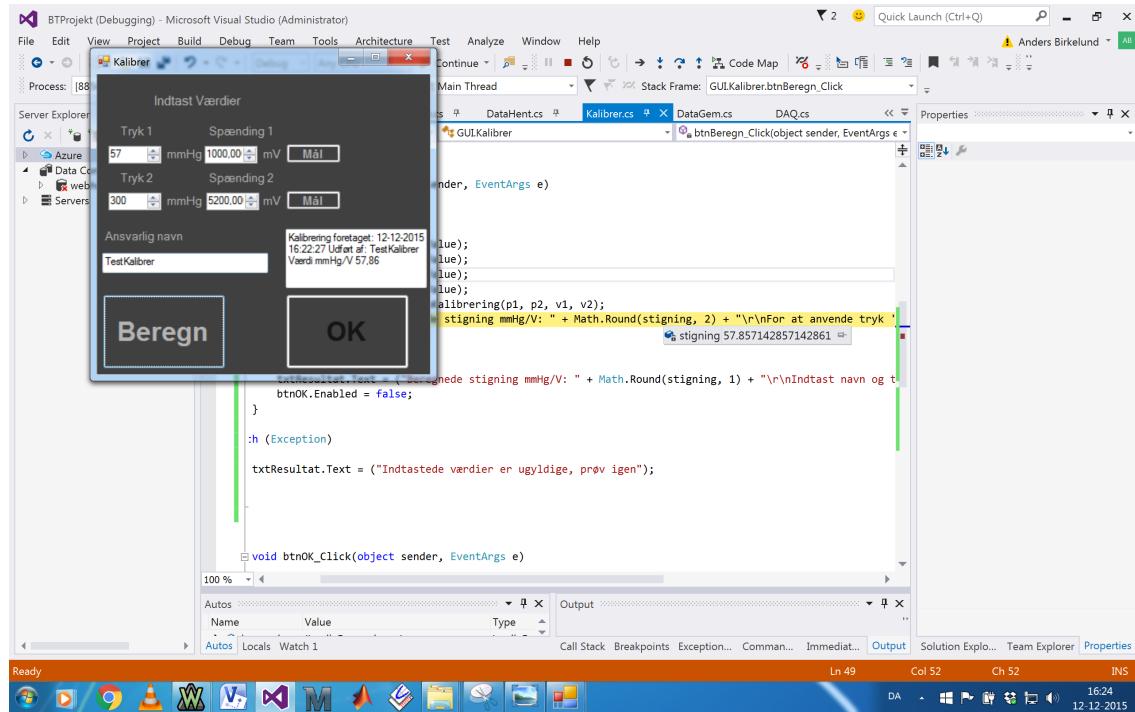


Figur 4.16: Kalibreringsdata indtastet

De målte værdier beregnes. Se Figur 4.17.

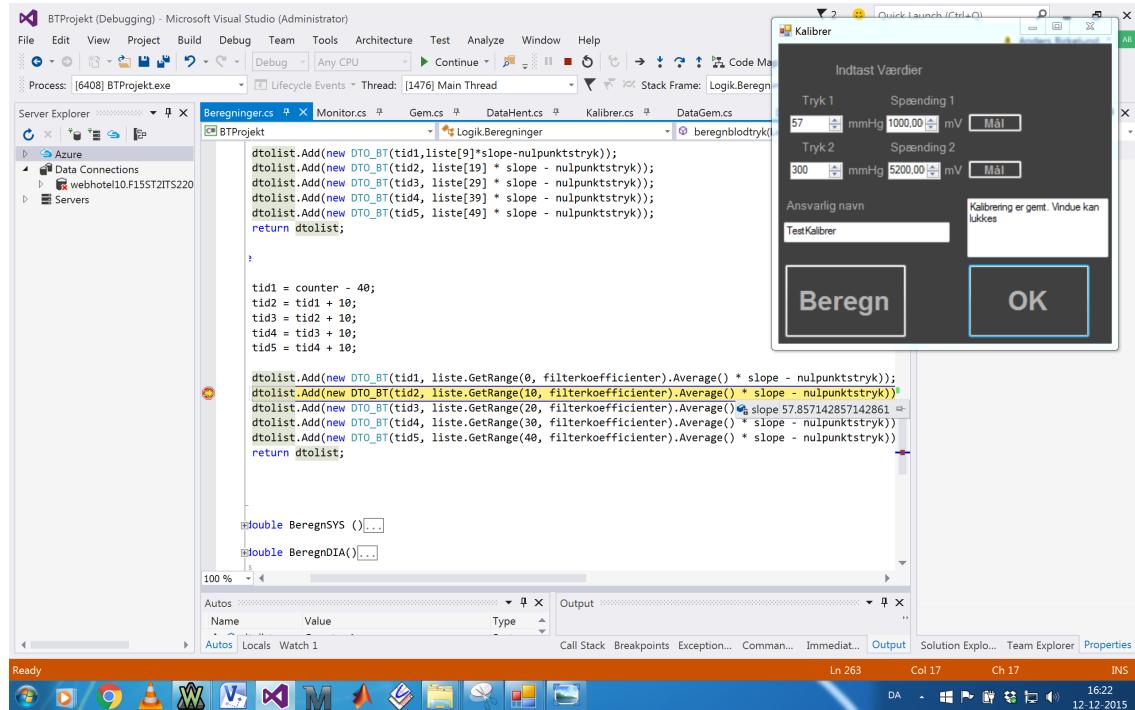
4.2. Test

ASE



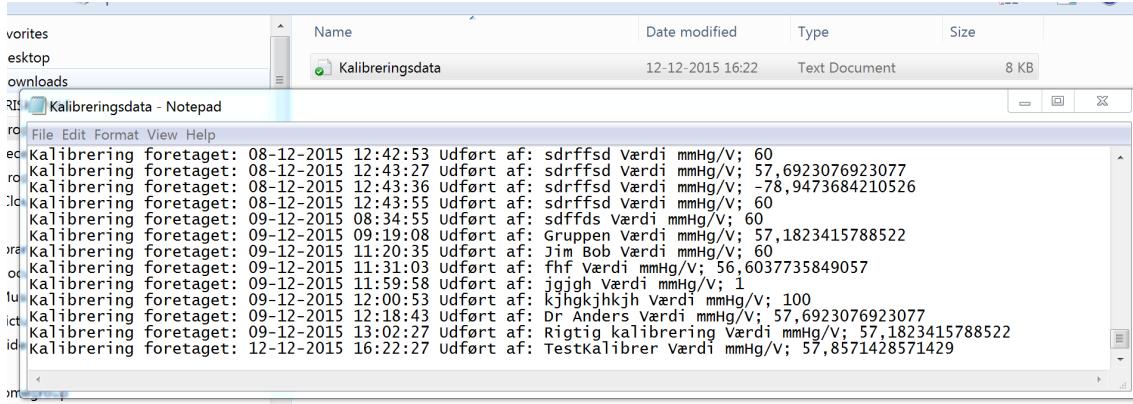
Figur 4.17: Kalibreringskonstant udregnes

Kalibreringskonstanten implementeres ved udskrivning på graf. Se Figur 4.18.



Figur 4.18: Kalibreringsdata anvendes

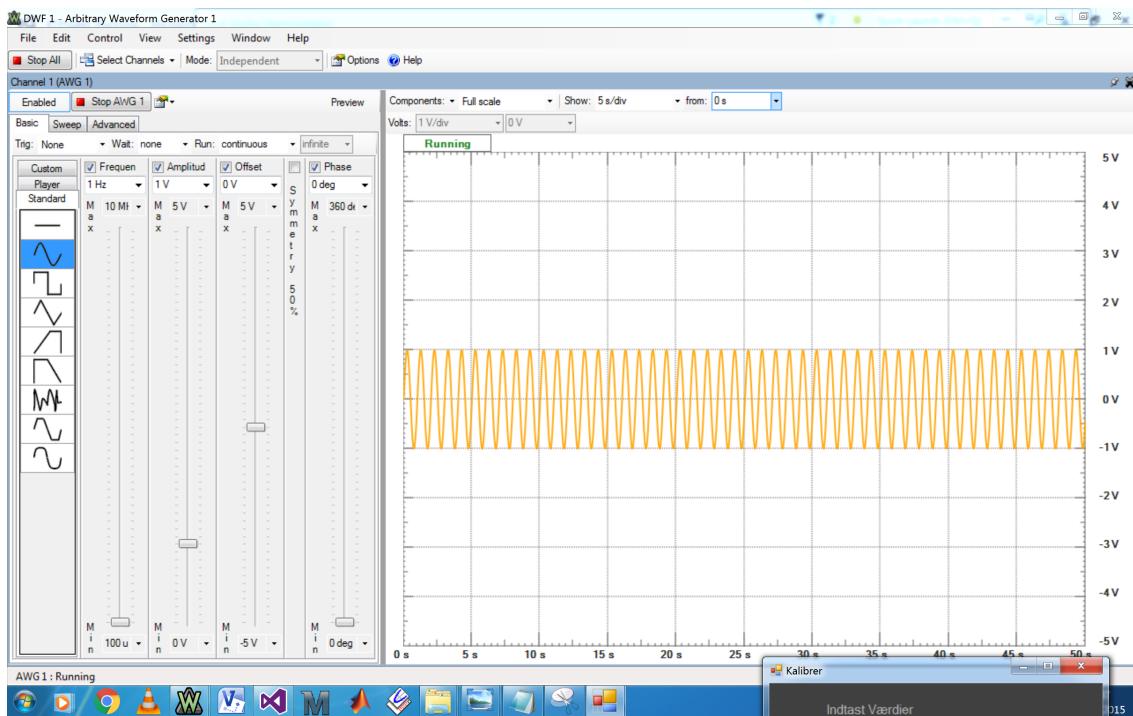
Det kontrolleres, at kalibreringsfilen er opdateret. Se Figur 4.19.



Figur 4.19: Kalibreringsdata er gemt til fil

4.2.2 Use Case 2 Vis måling

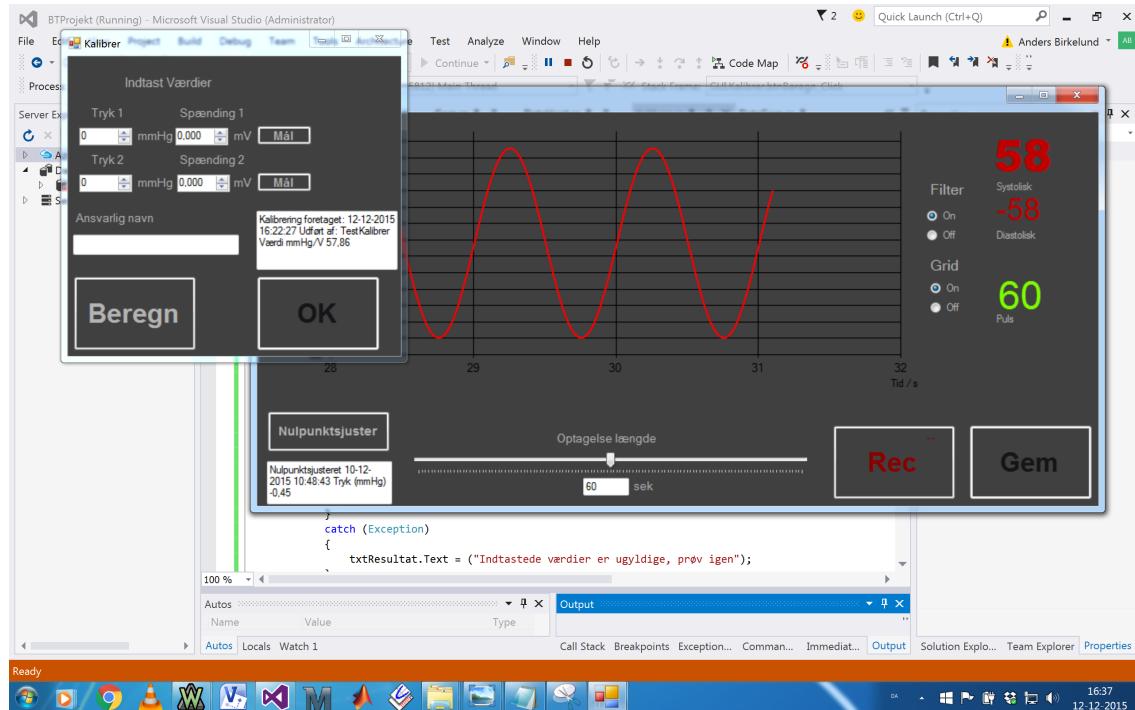
Et inputsignal på 1 Hz, 1 V amplitude og 0 V offset bruges til input på den fysiske DAQ. Se Figur 4.20.



Figur 4.20: Inputsignal til DAQ

Det ses, at kalibreringskonstanten (mmHg/V) er lig 57,9. Nulpunktsjusteringsværdien er tæt på 0. Maksimalværdien på signalet (systole) skal være lig 58 mmHg. Minimumsværdien på signalet skal være lig -58 mmHg. Pulsen skal være lig 60 (1 Hz). Se Figur 4.21.

Kalibreringskonstanten er lig 57,9. Det udskrevne blodtryk er lig 57,9 mmHg.



Figur 4.21: Output på Monitor-vindue

Puls

Til at teste pulsmetoden startes en optagelse fra starten af målingen. Målelængden sættes til 10 sekunder. Disse parametre sættes i Monitor-vinduets constructor, så optagelsen starter lige så snart måling startes. Se Figur 4.22.

```
public Monitor()
{
    InitializeComponent();
    filteron = new bool();
    beregninger = new Beregninger();
    kalibrer = new Kalibrer(beregninger);
    kalibrer.Show();
    kalibrer.TopMost = true;
    beregninger.AttachSubscriber(this);
    startreader("Dev1/aio0", -5, 5, 1000, 50);

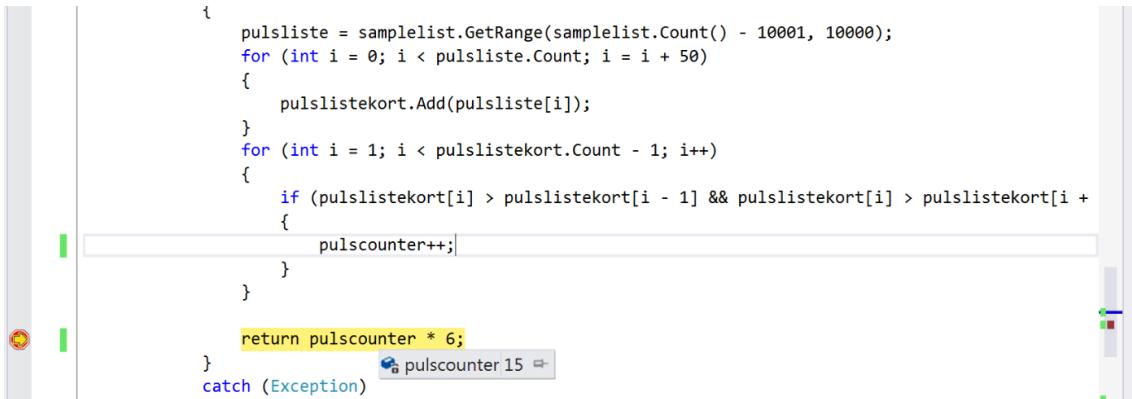
    //Textboxes
    textBoxNulpkt.Text = (beregninger.getnulpunktsstring());
    textBoxLængde.Text = "10";
    trackBar1.Value = 10;

    //GRAPH VISUALS

    chart1.ChartAreas[0].AxisX.Maximum = 4;
    chart1.ChartAreas[0].AxisX.Minimum = 0;
    chart1.ChartAreas[0].AxisX.Interval = 1;
    chart1.Series[0].BorderWidth = 2;
    btnGem.Enabled = false;
    gridButtonoff.Checked = false;
    gridButtonon.Checked = true;
    rdoFilterOff.Checked = false;
    rdoFilterOn.Checked = true;
    record();
}
```

Figur 4.22: Monitor-vinduets default constructor ved puls-test

Et break point sættes ved metoden, der returnerer en puls, baseret på 10000 samples. Metoden har talt 15 pulsslag. Se Figur 4.23.



```

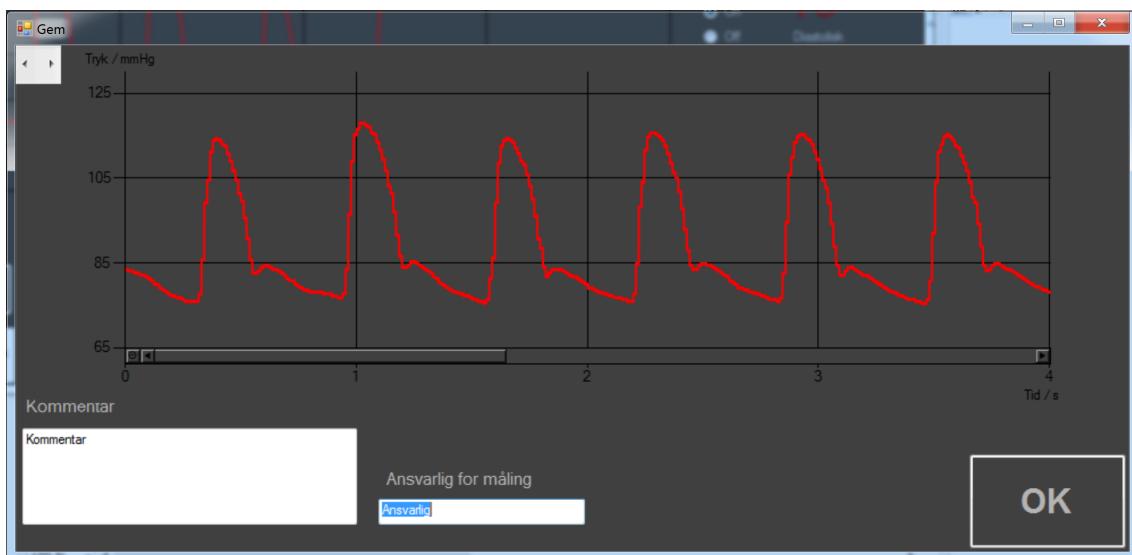
        pulsliste = samplelist.GetRange(samplelist.Count() - 10001, 10000);
        for (int i = 0; i < pulsliste.Count; i = i + 50)
        {
            pulslistekort.Add(pulsliste[i]);
        }
        for (int i = 1; i < pulslistekort.Count - 1; i++)
        {
            if (pulslistekort[i] > pulslistekort[i - 1] && pulslistekort[i] > pulslistekort[i + 1])
            {
                pulscounter++;
            }
        }

        return pulscounter * 6;
    } pulscounter 15
    catch (Exception)
}

```

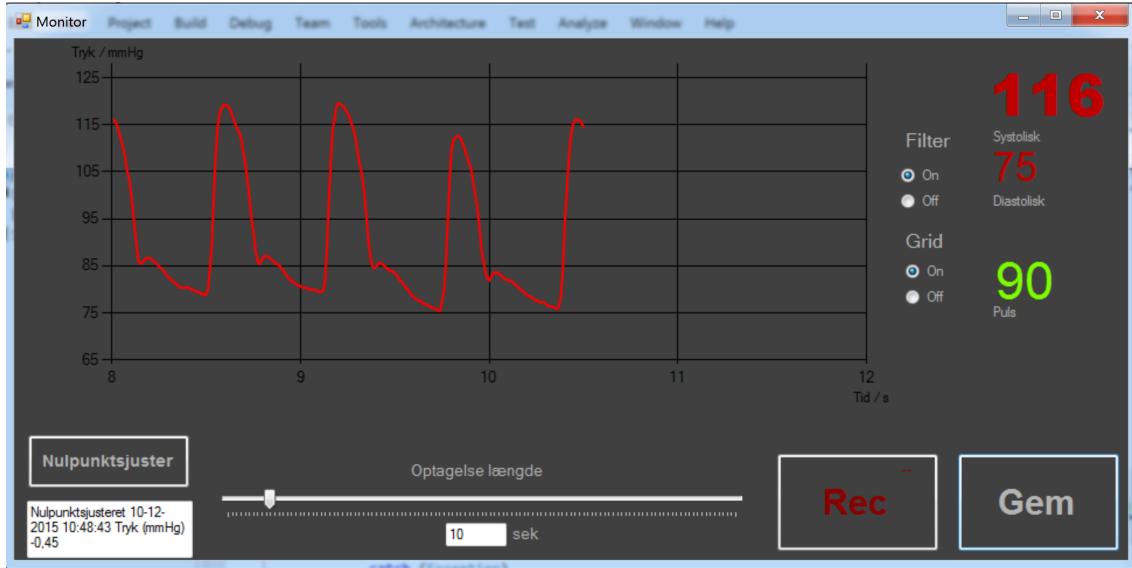
Figur 4.23: Puls efter 10 sekunders måling

Antallet af pulsslag talt på Gem-vinduet for de 10 sekunder. Her er der talt 15 pulsslag. Kun fire sekunder kan vises ad gangen. Se Figur 4.24.



Figur 4.24: Puls efter 10 sekunders måling

Det talte antal pulsslag holdes op mod dem, der er talt af metoden beregnpuls. Se pulsværdi i Figur 4.25.

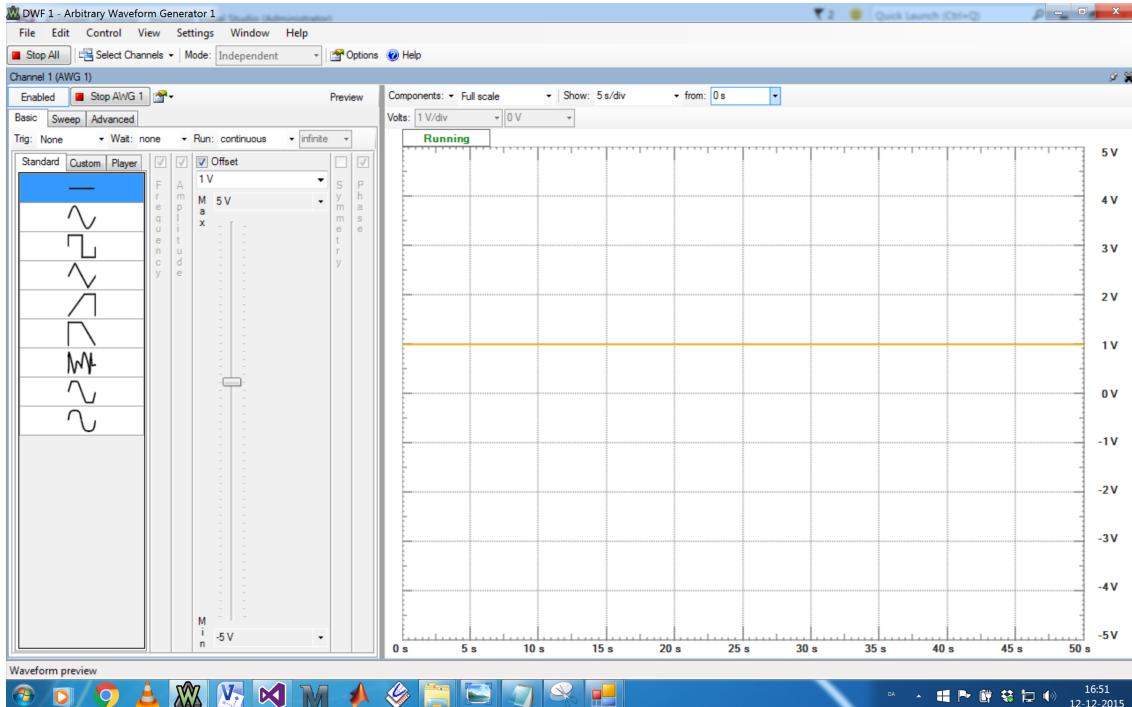


Figur 4.25: Puls efter 10 sekunders måling

Den beregnede puls er 90, hvilket stemmer overens med 15 pulsslag på 10 sekunder. Denne test kan bekræfte, at metoden fungerer efter hensigten.

4.2.3 Use Case 3: Nulpunktsjuster

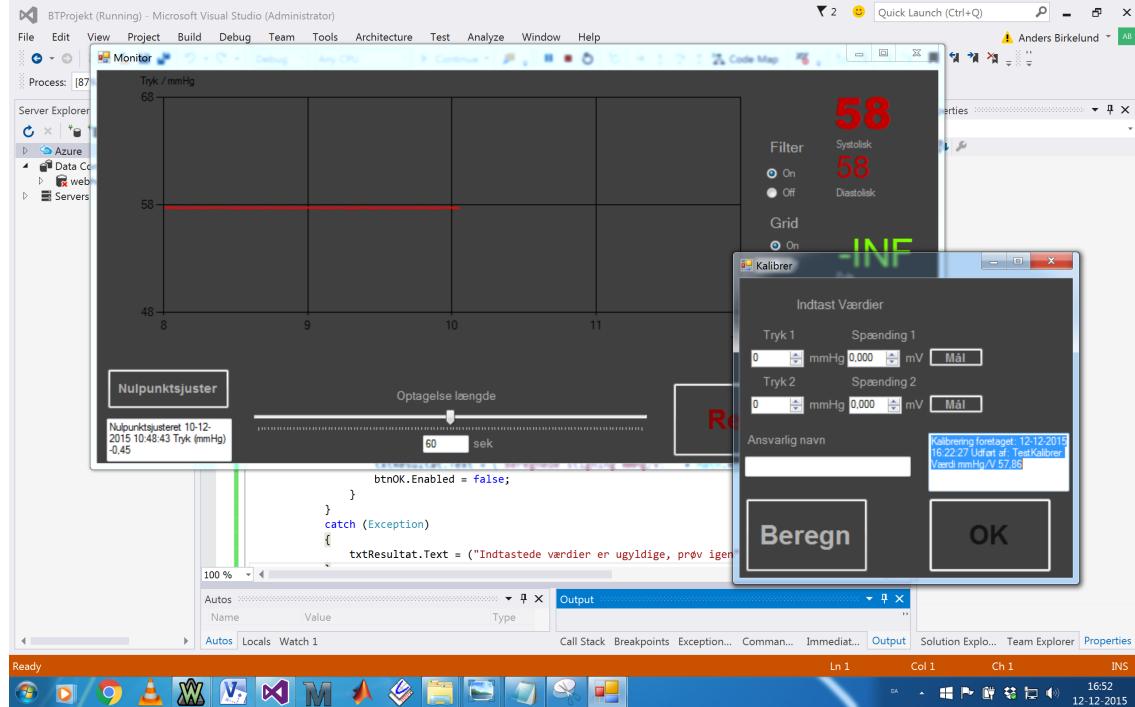
Et DC-signal på 1 V bruges til inputsignal til DAQ. Se Figur 4.26.



Figur 4.26: Inputsignal til DAQ

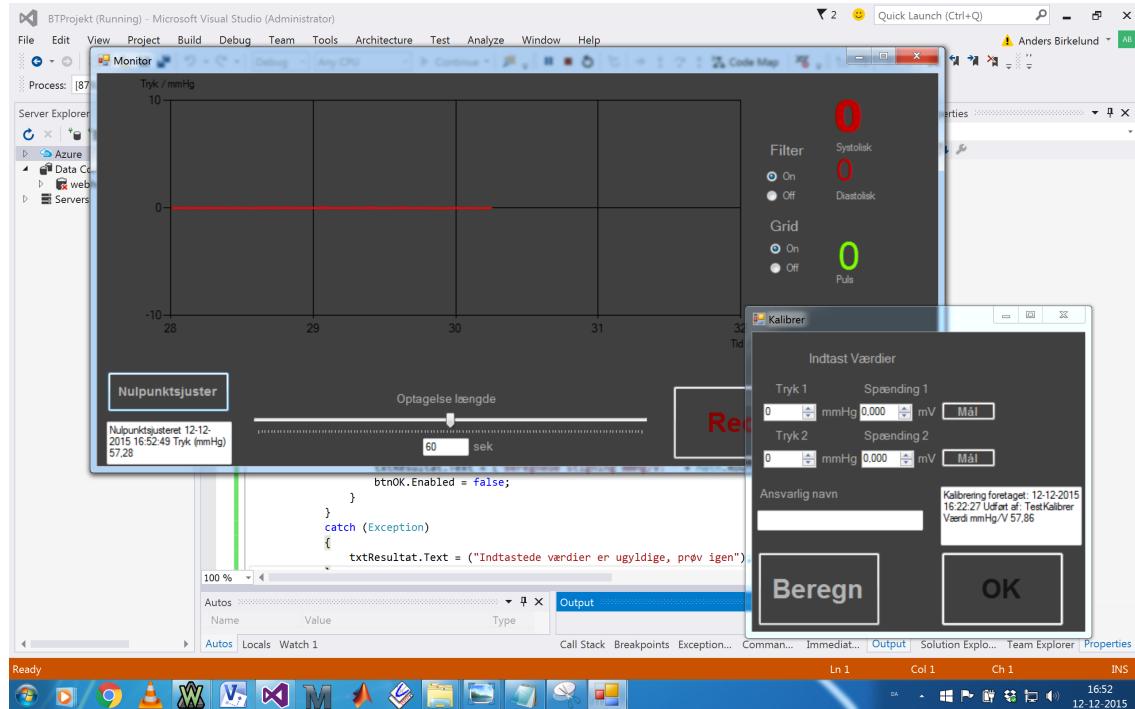
Det ses at kalibreringskonstanten ligger på 57,9 og inputsignalet genererer det korrekte

blodtryk. Se Figur 4.27.



Figur 4.27: Inputsignal til DAQ

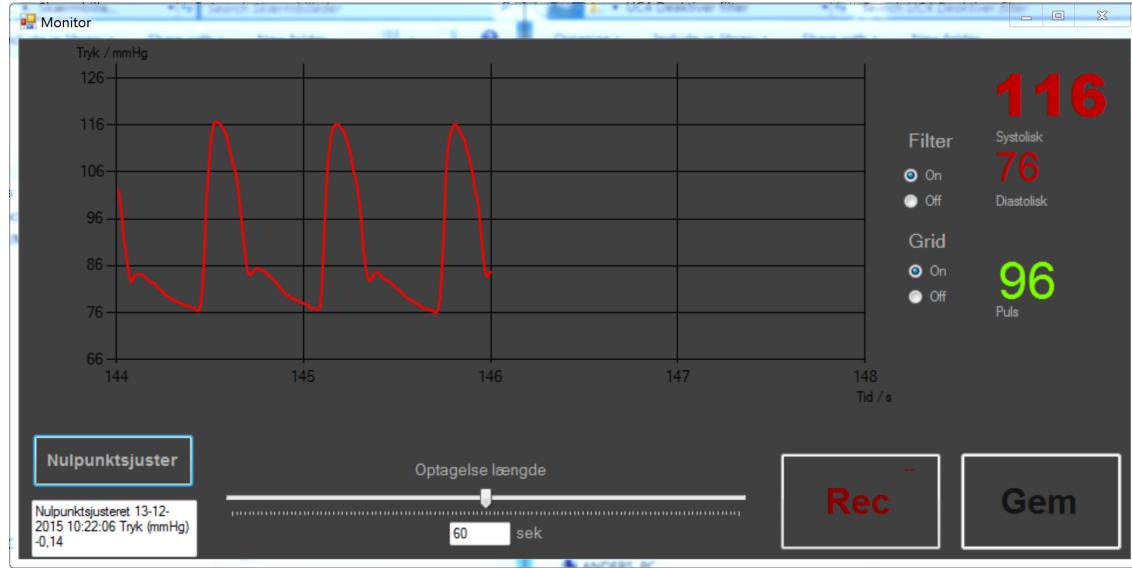
Der trykkes på knappen "Nulpunktsjuster", og grafens værdi falder til 0. Se Figur 4.28.



Figur 4.28: Inputsignal til DAQ

4.2.4 Use Case 4: Deaktiver filter

Filteret er aktiveret, og det filtrerede signal vises i graf. Se Figur 4.29.



Figur 4.29: Målingen vises med filter aktiveret

I Beregninger klassen beregnes målingen der udskrives på graf med filter, da bool'en 'filterOn' er 'true'. Se Figur 4.30.

```

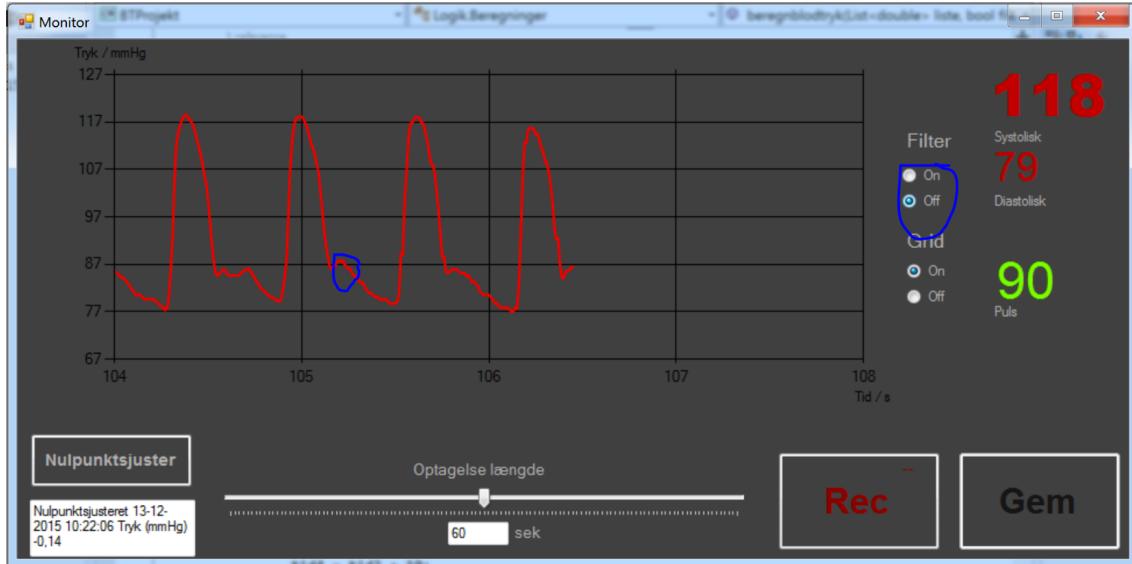
public List<DTO_BT> beregnBlodtryk(List<double> liste, bool filterOn)
{
    counter = counter + 50;
    dtolist.Clear();
    if (filterOn == false)
    {
        filterOn true
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

        dtolist.Add(new DTO_BT(tid1, liste[9] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid2, liste[19] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid3, liste[29] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid4, liste[39] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid5, liste[49] * slope - nulpunktstryk));
    }
    return dtolist;
}

```

Figur 4.30: Beregningerklassen opretter filtrerede DTO'er

Der trykkes på radiobutton "Off" under "Filter" og bool'en der sendes med metoden "beregnBlodtryk" sættes til 'false'. Metoden beregner blodtrykket der udskrives i Monitor vinduet, uden filter. Se Figur 4.31.



Figur 4.31: Blodtrykssignal før og efter deaktivering af filter. Den blå markering viser overgangen

Beregningerklassen kører metoden uden filter, da "filterOn" er 'false'. Se Figur 4.32.

```

public List<DTO_BT> beregnblodtryk(List<double> liste, bool filteron) // returnerer ETN DTO tra
{
    counter = counter + 50;
    dtolist.Clear();
    if (filterOn == false)
    {
        filterOn false
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

        dtolist.Add(new DTO_BT(tid1, liste[9]*slope-nulpunktstryk));
        dtolist.Add(new DTO_BT(tid2, liste[19] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid3, liste[29] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid4, liste[39] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid5, liste[49] * slope - nulpunktstryk));
        return dtolist;
    }
    else
    {
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

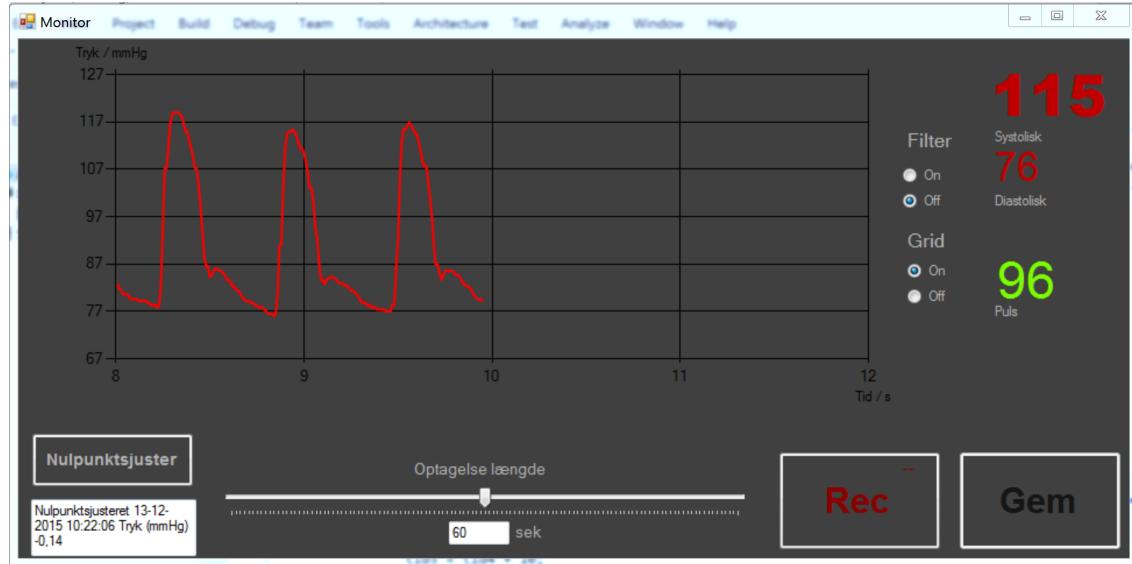
        dtolist.Add(new DTO_BT(tid1, liste.GetRange(0, filterkoefficienter).Average() * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid2, liste.GetRange(10, filterkoefficienter).Average() * slope - nulpunktstryk));
    }
}

```

Figur 4.32: Beregningerklassen opretter ufiltrerede DTO'er

4.2.5 Use Case 5: Aktiver filter

Filteret er ikke aktiveret, og det ufiltrerede signal udskrives. Se Figur 4.33.



Figur 4.33: Ufiltreret signal udskrives i graf

I Beregninger klassen beregnes målingen der udskrives på graf uden filter, da bool'en "filterOn" er 'false'. Se Figur 4.34.

```

Beregninger.cs  X  Monitor.cs  X  Gem.cs  X  DataHent.cs  X  Kalibrer.cs  X  DataGem.cs  DAQ.cs
BTProjekt      Logik.Beregninger
public List<DTO_BT> beregnblodtryk(List<double> liste, bool filterOn) // returnerer ETN DTO tra
{
    counter = counter + 50;
    dtolist.Clear();
    if (filterOn == false)
    {
        filterOn false
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

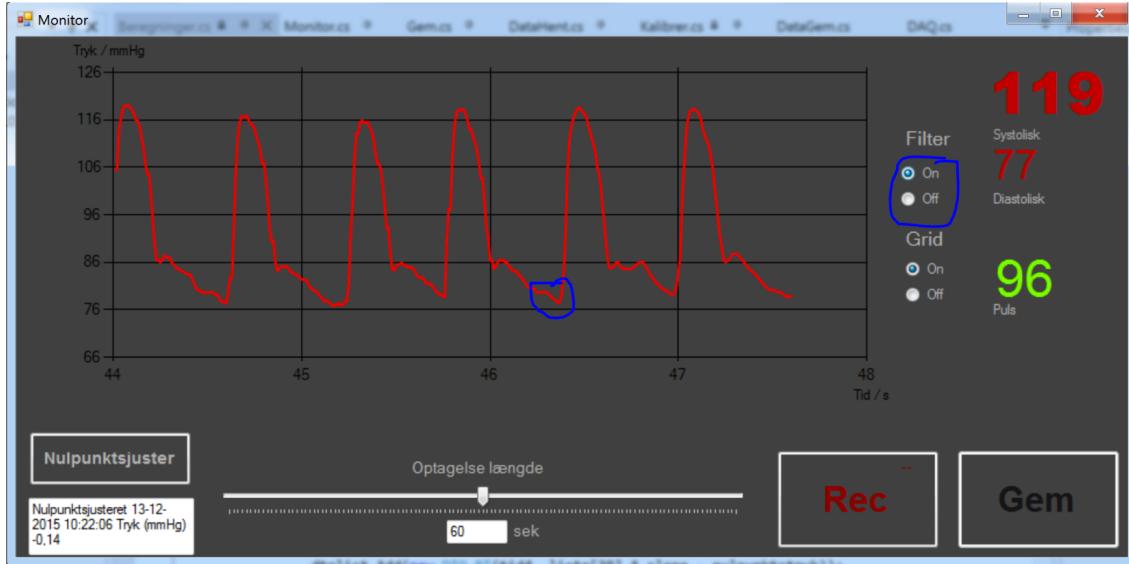
        dtolist.Add(new DTO_BT(tid1, liste[9]*slope-nulpunktstryk));
        dtolist.Add(new DTO_BT(tid2, liste[19] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid3, liste[29] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid4, liste[39] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid5, liste[49] * slope - nulpunktstryk));
        return dtolist;
    }
    else
    {
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

        dtolist.Add(new DTO_BT(tid1, liste.GetRange(0, filterkoefficienter).Average() * slope - filterkoefficienter));
        dtolist.Add(new DTO_BT(tid2, liste.GetRange(10, filterkoefficienter).Average() * slope - filterkoefficienter));
    }
}

```

Figur 4.34: Beregningerklassen opretter ufiltrerede

Der trykkes på "On" under "Filter" og det ses at det udskrevne signal filtreres. Se Figur 4.35.



Figur 4.35: Filtrerer signal udskrives i graf efter markering med blå cirkel

I beregninger klassen er bool'en der sendes med metoden "beregnBlodtryk" sat til 'true'. Metoden beregner blodtrykket, der udskrives i Monitor-vinduet, med filter. Se Figur 4.36.

```

Beregninger.cs  X  Monitor.cs  Gem.cs  DataHent.cs  Kalibrer.cs  DataGem.cs  DAQ.cs  ISubscriber.cs
BTProjekt      Logik.Beregninger
{
    tid1 = counter - 40;
    tid2 = tid1 + 10;
    tid3 = tid2 + 10;
    tid4 = tid3 + 10;
    tid5 = tid4 + 10;

    dtolist.Add(new DTO_BT(tid1, liste[9]*slope-nulpunktstryk));
    dtolist.Add(new DTO_BT(tid2, liste[19] * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid3, liste[29] * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid4, liste[39] * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid5, liste[49] * slope - nulpunktstryk));
    return dtolist;
}
else
{
    tid1 = counter - 40;
    tid2 = tid1 + 10;
    tid3 = tid2 + 10;
    tid4 = tid3 + 10;
    tid5 = tid4 + 10;

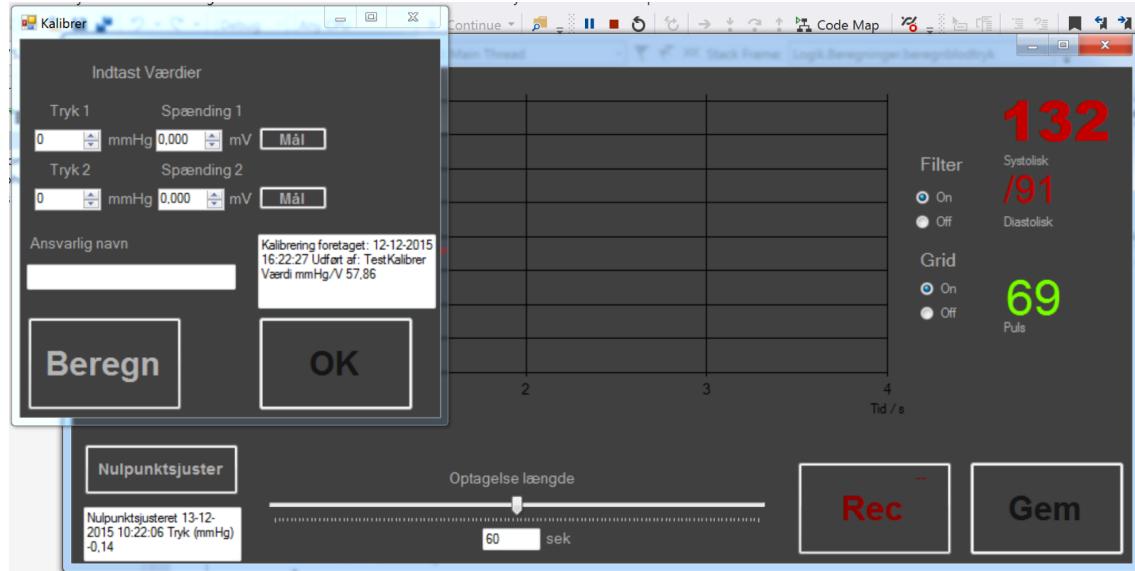
    dtolist.Add(new DTO_BT(tid1, liste.GetRange(0, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid2, liste.GetRange(10, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid3, liste.GetRange(20, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid4, liste.GetRange(30, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid5, liste.GetRange(40, filterkoefficienter).Average() * slope - nulpunktstryk));
    return dtolist;
}

```

Figur 4.36: Beregningeklassen opretter filtrerede DTO'er

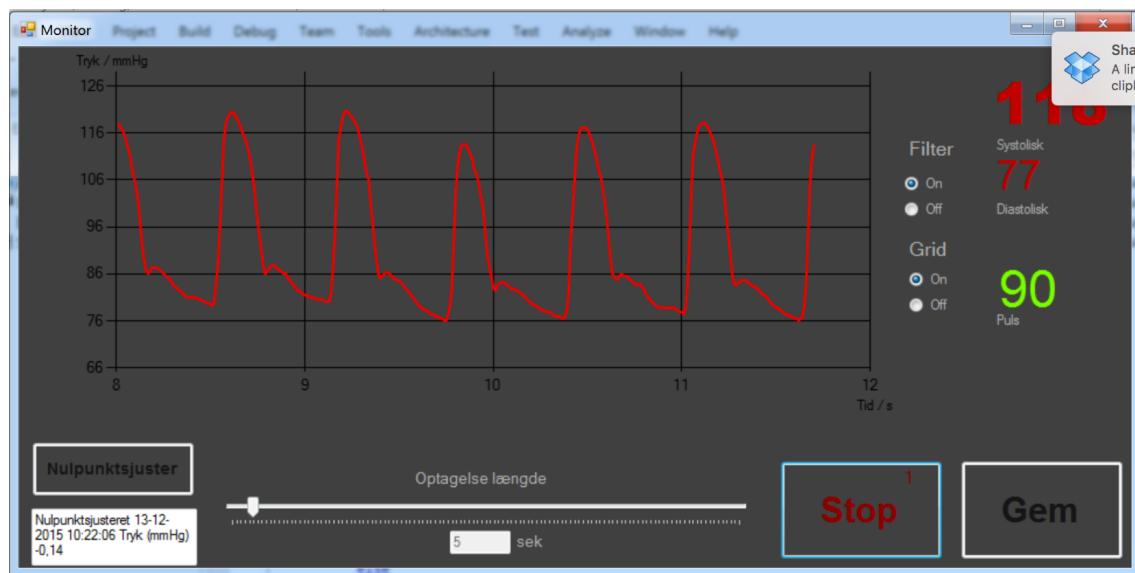
4.2.6 Use Case 6: Gem måling

Kalibreringskonstant (mmHg/V) og nulpunktsværdi (mmHg) for måling fremgår af Figur 4.37.



Figur 4.37: Monitor-vinduets initiale værdier (kalibreringskonstant og nulpunktstryk)

Forsker vælger målelængde til fem sekunder og trykker på "Rec". Se Figur 4.38.



Figur 4.38: Monitor-vindue efter tryk på "Rec"-knap. "Rec"-knappen skifter til en "Stop"-knap

Metoden record() køres i Beregningerklassen. Her sættes startværdien for måling. Se Figur 4.39.

```

Beregninger.cs  X  Monitor.cs  Gem.cs  DataHent.cs  Kalibrer.cs  DataGem.cs  DAQ.cs  ISubscriber.cs  Gem.cs [Design]
BTProjekt      Logik.Beregninger
public void Beregnin... (...) 2 references
public void record(bool recording)
{
    if (recording == false)
    {
        startgem = samplelist.Count();
    }
    else
    {
        slutgem = samplelist.Count();
    }
}
// PUBLISHER/SUBSCRIBER ...
public void AttachSubscriber(ISubscriber subscriber)...
private void NotifySubscriber(List<double> blodtryk)...
public void Update(List<double> sampleudsnit)...

```

Figur 4.39: Beregningerklassens metode `record()` køres, når optagelse starter, og startværdien sættes

Tiden udløber og optagelsen slutter. Metoden `record()` køres automatisk igen. Denne gang sættes slutværdien for målingen. Det observeres om differencen mellem startværdi og slutværdi er lig 5 sekunder, det vil sige 5000 samples. Se Figur 4.40.

```

Beregninger.cs  X  Monitor.cs  Gem.cs  DataHent.cs  Kalibrer.cs  DataGem.cs  DAQ.cs  ISubscriber.cs  Gem.cs [Design]
BTProjekt      Logik.Beregninger
public void Beregnin... (...) 2 references
public void record(bool recording)
{
    if (recording == false)
    {
        startgem = samplelist.Count();
    }
    else
    {
        slutgem = samplelist.Count();
    }
}
// PUBLISHER/SUBSCRIBER ...
public void AttachSubscriber(ISubscriber subscriber)...
private void NotifySubscriber(List<double> blodtryk)...
public void Update(List<double> sampleudsnit)...

```

Figur 4.40: Beregningerklassens `record()` køres når optagelse slutter, os slutværdien sættes

Forsker trykker på "Gem"-knap i Monitor-vinduet. Gem-vinduets constructor modtager værdier via metoder, der afvikles i beregninger. Metoderne `VisGemgraf()` og `Show()` afvikles. Se Figur 4.41.

```

Beregninger.cs  X  Monitor.cs  Gem.cs  DataHent.cs  Kalibrer.cs  DataGem.cs  DAQ.cs  ISubscriber.cs  Gem.cs [Design]
BTProjekt      GUI.Monitor
private void rdoFilterOff_CheckedChanged(object sender, EventArgs e)...
private void rdoFilterOn_CheckedChanged(object sender, EventArgs e)...
public void Update(List<double> liste)...
private void btnNulpunktJuster_Click(object sender, EventArgs e)...
private void btngem_Click(object sender, EventArgs e)
{
    abnGem();
}
private void abnGem()
{
    Gem gem = new Gem(beregninger, beregninger.getSlopeDouble(), beregninger.getNulpunktDouble(), beregninger.getStart(), beregninger.getSlut(), beregninger.getSampleList());
    gem.visGemgraf();
    gem.Show();
}

```

Figur 4.41: Oprettelse af nyt Gem-vinde

Det kontrolleres, om Gem-klassens værdier er korrekte. Følgende værdier tjekkes: slope_ = kalibreringskonstant, nulpunkt_ = nulpunktstryk, startgem_ = startværdi for måling,

slutgem_ = slutværdi for måling. Se Figur 4.42.

```

partial class Gem
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

Name	Type
this	GUI.Gem
beregn	Logik.Beregninger
slope_	double
nulpunkt_	double
startgem_	int
slutgem_	int
samplelist_	System.Collections.Generic.List<double>

Figur 4.42: Gem-vinduets værdier, efter oprettelse

Det kontrolleres, om længden af "samplearray" i metoden gemmaalingtilDB() stemmer overens med de valgte fem sekunder, det vil sige 5000 samples. Se Figur 4.43.

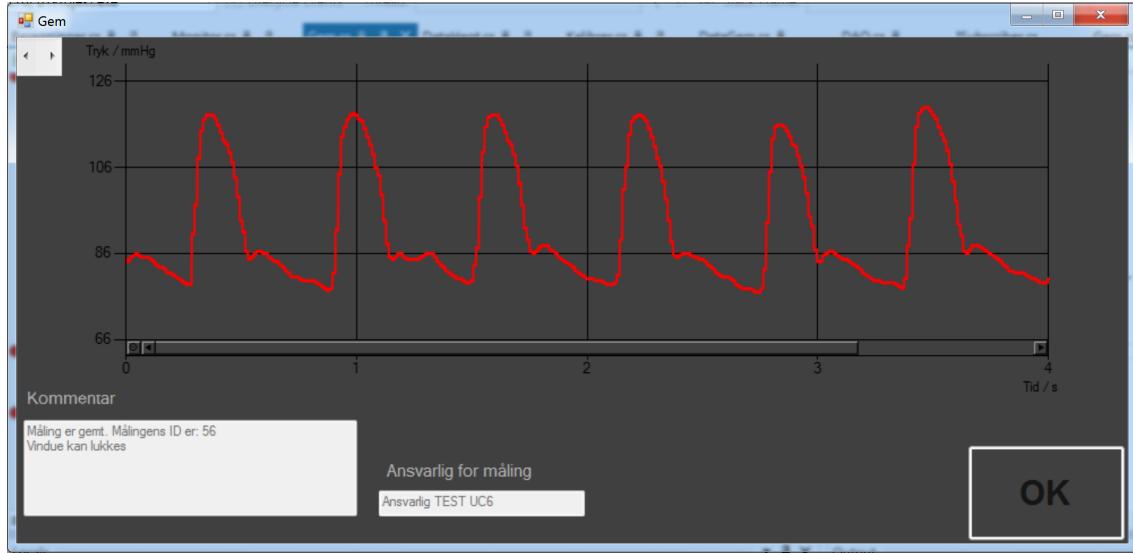
```

public void gemmaalingtilDB(List<double> samplelist, int maaleid)
{
    try
    {
        samplearray = samplelist.ToArray();
        conn = new SqlConnection("Data Source=10.ihb.dk;Initial Catalog=" + db + ";Persist Security Info=True;User ID=" + db + ";Password=" + db + "");
        //Ling - henter alle værdier ud én af gange og konverterer dem til Bytes og smider dem til sidste ud i et array
        byte[] data = samplearray.SelectMany(value => BitConverter.GetBytes(value)).ToArray();
        conn.Open();
    }
}

```

Figur 4.43: Array af blodtryksværdier, der gemmes til Database

Målingens ID noteres i Gem-vinduet. Se Figur 4.44.



Figur 4.44: Gem-vinduet efter gemt måling

Det kontrolleres, om den gemte måling befinner sig i Databasen. Se Figur 4.45.

maaleid	maaling	tid	ansvarlig	kommentar
53	0x4EE14B993B905A40...	09-12-2015 16:13:57	Ansvarlig	Kommentar
54	0x41E68F3A43635340...	12-12-2015 10:16:55	Ansvarlig	Kommentar
55	0x969E2A1C5C965A40...	13-12-2015 11:52:11	Ansvarlig	Kommentar
56	0x9FB5A33ED0265540...	13-12-2015 12:27:21	Ansvarlig TEST UC6	UC6 TEST
* NULL	NULL	NULL	NULL	NULL

Figur 4.45: Array af blodtryksværdier, der gemmes til Database

Accepttest 5

Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse
1.0	30-09-2015	Alle	Første udkast. Klar til review
2.0	08-10-2015	Alle	Rettelser efter reviewmøde
2.1	08-12-2015	Alle	Rettelse efter ændringer i Use Cases

5.1 Accepttest af Use Cases

5.1.1 Use Case 1

Kalibrér

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			
1. Start system	Kalibrerings-vinduet og seneste kalibreringstidspunkt vises	Kalibrerings-vinduet og seneste kalibreringstidspunkt vises	✓
2. Indtast kalibreringsdata og tryk på "Beregn"-knap	Der er indtastet kalibreringsdata i de forskellige tekstdokumenter og stigningenstabellen for, hvor mange mmHg/V udskrives i Kalibrerings-vinduet	Der er indtastet kalibreringsdata i de forskellige tekstdokumenter og stigningenstabellen for, hvor mange mmHg/V udskrives i Kalibrerings-vinduet	✓
3. Tryk på "OK"-knap for at udføre kalibrering	System kalibrerer og bekræfter fuldført kalibrering	System kalibrerer og bekræfter fuldført kalibrering	✓

Undtagelse

2.a Tryk på det røde kryds øverst i højre hjørne, hvis kalibrering ikke ønskes	Kalibrerings-vinduet lukkes ned	Kalibrerings-vinduet lukkes ned	✓
2.b Indtast ugyldige kalibreringsdata	System informerer om ugyldige kalibrerings-data	System informerer om ugyldige kalibrerings-data	✓

Tabel 5.2: Accepttest af Use Case 1.

5.1.2 Use Case 2

Vis måling med digitalt filter

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			
1. Udfør Use Case 1 og observér Monitor-vinduet	Det filtrerede blodtrykssignal samt systole-, diastole- og pulsværdier udskrives i Monitor-vinduet	Det filtrerede blodtrykssignal samt systole-, diastole- og pulsværdier udskrives i Monitor-vinduet	✓
<i>Undtagelse</i>			

Tabel 5.3: Accepttest af Use Case 2.

5.1.3 Use Case 3

Nulpunktsjustér

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			

1. Tryk på System nulpunktsjustér-knap	på System nulpunktsjustér	System nulpunktsjustér	(✓)
2. Observér kurven og Monitor-vinduet	Værdien af kurven er lig 0 samt tidsstempel for seneste nulpunktsjustering vises i Monitor-vinduet	Værdien af kurven er lig 0 samt tidsstempel for seneste nulpunktsjustering vises i Monitor-vinduet	(✓)
3. Observér Monitor-vinduet	Tidsstempel for nulpunktsjustering vises	Tidsstempel for nulpunktsjustering vises	(✓)

Undtagelser

Tabel 5.4: Accepttest af Use Case 3.

5.1.4 Use Case 4

Deaktivér filter

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			
1. Markér "Off"-knap under "Filter" i Monitor-vinduet	Digital filter deaktiveres	Digital filter deaktiveres	✓
2. Observér blodtrykssignalen	Det ufiltrerede blodtrykssignal udskrives i Monitor-vindue	Det ufiltrerede blodtrykssignal udskrives i Monitor-vindue	✓

Undtagelser

Tabel 5.5: Accepttest af Use Case 4.

5.1.5 Use Case 5

Aktivér filter

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			
1. Markér "On"-knap under "Filter" i Monitor-vinduet	Digitalt filter aktiveres	Digitalt filter aktiveres	✓
2. Observér blodtrykssignalen	Det filtrerede blodtrykssignal udskrives i Monitor-vinduet	Det filtrerede blodtrykssignal udskrives i Monitor-vinduet	✓
<i>Undtagelser</i>			

Tabel 5.6: Accepttest af Use Case 5.

5.1.6 Use Case 6

Gem måling

Test	Forventet resultat	Faktiske observationer	Godkendt
<i>Hovedforløb</i>			
1. Indtast ønsket optagelseslængde for målingen	Optagelseslængden ændres	Optagelseslængden ændres	✓
2. Tryk på "Rec"-knap	Optagelse af måling igangsættes og nedtælling vises	Optagelse af måling igangsættes og nedtælling vises	✓
3. Vent til optagelseslængden er nået	Optagelse af måling stoppes	Optagelse af måling stoppes	✓
4. Tryk på "Gem"-knap	Gem-vinduet åbnes	Gem-vinduet åbnes	✓

5.	Indtast oplysninger om målingen	Tekstboksene er blevet udfyldt	Tekstboksene er blevet udfyldt	✓	
6.	Tryk på "OK"-knap	System gemmer målingen, og det er ikke muligt at trykke på andre knapper end det røde kryds øverst i højre hjørne	System gemmer målingen, og det er ikke muligt at trykke på andre knapper end det røde kryds øverst i højre hjørne	✓	
7.	Observér vinduet	Gem-	Det fremgår af Gemvinduet, at målingen er gemt og der tjekkes i Databasen om målingen er gemt	Det fremgår af Gemvinduet, at målingen er gemt og der tjekkes i Databasen om målingen er gemt	✓

Undtagelser

3.a	Stop optagelse af måling inden optagelseslængden er nået	Optagelse stopper og kan gemmes efter ønske	Optagelse stopper og kan gemmes efter ønske	✓
-----	--	---	---	---

Tabel 5.7: Accepttest af Use Case 6.

5.2 Accepttest af ikke-funktionelle krav

Ikke-funktionelt krav	Test/handling	Forventet resultat	Faktiske observationer	Godkendt
<i>Functionality</i>				
System skal kunne kalibreres	Indtast gyldige kalibreringsdata. Tryk på "Beregn"-knap efterfulgt af "OK"-knap	System kalibrerer og udskriver tidsstempel i Kalibreringsvinduet	System kalibrerer og udskriver tidsstempel i Kalibreringsvinduet	✓

System skal kunne vise et kontinuert blodtryksignal i Monitor-vinduet	Observér Monitor-vinduet	I Monitor-vinduet vises et kontinuert blodtrykssignal	I Monitor-vinduet vises et kontinuert blodtrykssignal	✓
System skal kunne vise systole-, diastole- og puls-værdier med op til tre cifre	Observér Monitor-vinduet	Monitor-vinduet indeholder systole-, diastole- og pulsværdier med op til tre cifre	Monitor-vinduet indeholder systole-, diastole- og pulsværdier med op til tre cifre	✓
System skal kunne vise et blodtrykssignal med og uden et digitalt filter	Markér "On" og "Off"-knap og Observér Monitor-vinduet	Monitor-vinduet viser et blodtrykssignal med eller uden digitalt filter	Monitor-vinduet viser et blodtrykssignal med eller uden digitalt filter	✓
System skal kunne nulpunktsjustere kurven	Tryk på "Nulpunktsjustér"-lig -knap og observér Monitor-vinduet	Kurvens værdi er lig 0	Kurvens værdi er lig 0	✓
System skal kunne gemme en blodtryksmåling i en database	Der trykkes på "Gem"-knap i Monitor-vinduet, og indtastes gyldige værdier i Gem-vinduet og trykkes på "OK"-knap	System gemmer data i en Database og skriver stempel for gemt data i Gyldige værdier Gem-vinduet	System gemmer data i en Database og udskriver stempel for gemt data i Gem-vinduet	✓

Usability

Kalibrerings-vinduet skal indeholde tekster til indtastning af kalibreringsdata	Der ses i Kalibrerings-vinduet, om der er tekstbokse til indtastning af indtastning af kalibreringsdata	Der er tekstbokse til indtastning af kalibre-ringsdata i Kalibrerings-vinduet	Der er tekstbokse til indtastning af kalibre-ringsdata i Kalibrerings-vinduet	✓
Kalibrerings-vinduet skal indeholde en "Beregn"-knap	Der ses i Kalibrerings-vinduet, om der er en "Beregn"-knap	Der er en "Beregn"-knap i Kalibrerings-vinduet	Der er en "Beregn"-knap i Kalibrerings-vinduet	✓
Kalibrerings-vinduet skal indeholde en "OK"-knap	Der ses i Kalibrerings-vinduet, om der er en "OK"-knap	Der er en "OK"-knap i Kalibrerings-vinduet	Der er en "OK"-knap i Kalibrerings-vinduet	✓
Kalibrerings-vinduet skal indeholde et tidsstempel for seneste kalibrering	Der ses i Kalibrerings-vinduet, om der er et tidsstempel for seneste kalibrering	Der er et tidsstempel for seneste kalibrering i Kalibrerings-vinduet	Der er et tidsstempel for seneste kalibrering i Kalibrerings-vinduet	✓
Monitor-vinduet skal indeholde en "Gem"-knap	Der ses i Monitor-vinduet, om der er en "Gem"-knap	Der er en "Gem"-knap i Monitor-vinduet	Der er en "Gem"-knap i Monitor-vinduet	✓
Monitor-vinduet skal indeholde en "Nulpunktsjustér"-knap	Der ses i Monitor-vinduet, om der er en "Nulpunktsjustér"-knap	Der er en "Nulpunktsjustér"-knap i Monitor-vinduet	Der er en "Nulpunktsjustér"-knap i Monitor-vinduet	✓

Monitor-vinduet skal indeholde et tidsstempel for seneste nulpunktsjustering	Der ses i Monitor-vinduet, om der er et tidsstempel for seneste nulpunktsjustering	Der er et tidsstempel for seneste nulpunktsjustering i Monitor-vinduet	Der er et tidsstempel for seneste nulpunktsjustering i Monitor-vinduet	✓
Monitor-vinduet skal indeholde to radiobuttons til aktivering og deaktivering af digitalt filter	Der ses i Monitor-vinduet, om der er to radiobuttons til aktivering og deaktivering af digitalt filter	Der er to radio-buttons til aktivering og deaktivering af digitalt filter i Monitor-vinduet	Der er to radio-buttons til aktivering og deaktivering af digitalt filter i Monitor-vinduet	✓
Monitor-vinduet skal indeholde en tekstboks til ændring af optagelseslængde	Der ses i Monitor-vinduet, om der er en tekstboks til ændring af optagelseslængde	Der er en tekstboks til ændring af optagelseslængde i Monitor-vinduet	Der er en tekstboks til ændring af optagelseslængde i Monitor-vinduet	✓
Monitor-vinduet skal indeholde en "Rec"-knap/"Stop"-knap	Der ses i Monitor-vinduet, om der er en "Rec"-knap/"Stop"-knap	Der er en "Rec"-knap/"Stop"-knap i Monitor-vinduet	Der er en "Rec"-knap/"Stop"-knap i Monitor-vinduet	✓
Gem-vinduet skal indeholde tekstuoke til indtastning af oplysninger om målingen	Der ses i Gem-vinduet, om der er tekstuoke til indtastning af oplysninger om målingen	Der er tekstuoke til indtastning af oplysninger om målingen i Gem-vinduet	Der er tekstuoke til indtastning af oplysninger om målingen i Gem-vinduet	✓

Gem-vinduet skal indeholde en "OK"-knap	Der ses i Gem-vinduet, om der er en "OK"-knap	Der er en "OK"-knap i Gem-vinduet	Der er en "OK"-knap i Gem-vinduet	✓
Det skal være muligt at aflæse systole-, diastole- og puls værdier på Monitor-vinduet fra 2 meters afstand med normalt syn	Der testes på 5 personer af forskellige aldre med en synspuls værdier på +/- 0,25, som placeres 2 meter fra Monitor-vinduet	Det er muligt for de 5 personer at aflæse systole-, diastole- og pulsværdier på 2 meters afstand	Det er muligt for de 5 personer at aflæse systole-, diastole- og pulsværdier på 2 meters afstand	✓
<i>Reliability</i>				
Systemet skal have en effektiv MTBF på 99 timer og MTTR på 20 minutter	Ikke testbar			
<i>Performance</i>				
Blodtrykssignalet skal vises maksimalt 5 sekunder efter, Use Case 1 er afsluttet	Use Case 1 afsluttes samtidig med, at et stoppur startes på en iPhone 5s. Når blodtrykssignalet vises, stoppes uret	Blodtrykssignalet vises indenfor 5 sekunder	Blodtrykssignalet vises indenfor 5 sekunder	✓
Systemet skal vise en kurve for blodtryksmålingen, hvor y-aksen er mmHg og x-aksen er tid i sekunder	Der ses på Monitor-vinduet, om y-aksen er mmHg og x-aksen er tid i sekunder	Blodtrykssignalets y-akse er mmHg og x-aksen er tid i sekunder	Blodtrykssignalets y-akse er mmHg og x-aksen er tid i sekunder	✓

Systemet skal kunne måle blodtryksværdier fra 0 til 300 mmHg. Trykket simuleres op til 300 mmHg. Det ses i Monitor-vinduet at trykket er ca. 300 mmHg.

Det ses i Monitor-vinduet, at trykket er ca. 300 mmHg.

Blodtrykssignalet vises kontinuerligt over et interval af 4 sekunder af gangen. Der ses på Monitor-vinduet, om blodtrykssignalet vises kontinuerligt over et interval af fire sekunder af gangen.

Blodtrykssignalet vises kontinuerligt over et interval af fire sekunder af gangen.

Blodtrykssignalet vises via en rød kurve. Der ses på Monitor-vinduet, om blodtrykssignalet vises via en rød kurve.

Blodtrykssignalet vises via en rød kurve.

Y-aksen varierer efter blodtrykssignalets maksimum- og minimumsværdier. Der ses på Monitor-vinduet, om blodtrykssignalets maksimum- og minimumsværdier vises.

Blodtrykssignalets maksimum- og minimumsværdier vises.

Supportability

Softwareen skal opbygges efter trelagsmodellen. Der kigges i koden efter datalag, logik-lag og GUI-lag.

Data-lag, logik-lag og GUI-lag er at finde i koden.

Data-lag, logik-lag og GUI-lag er at finde i koden.

Andet(+)

Der skal være adgang til en computer med Windows 7 eller nyere – computeren skal have minimum 4 GB RAM	Der ses, om der er installeret Windows 7 eller nyere, og om der er minimum 4 GB RAM	Det ses, at der er installeret Windows 7 eller nyere, og om der er minimum 4 GB RAM	Det ses, at der er installeret Windows 7 eller nyere, og om der er minimum 4 GB RAM	✓
Der skal være adgang til en computer, hvor National Instruments er installeret	Der ses, om der er installeret National Instruments på computeren	Det ses, at der er installeret National Instruments på computeren	Det ses, at der er installeret National Instruments på computeren	✓
<i>Interface</i>				
Blodtryksdiagrammet skal fylde minimum 1/3 af Monitor-vinduet	Der ses, om blodtryksdiagrammet fylder minimum 1/3 af Monitor-vinduet	Blodtryksdiagrammet fylder minimum 1/3 af Monitor-vinduet	Blodtryksdiagrammet fylder minimum 1/3 af Monitor-vinduet	✓
Baggrunden i Monitor-vinduet skal være mørk	Der ses i Monitor-vinduet, om baggrunden er mørk	Baggrunden i Monitor-vinduet er mørk	Baggrunden i Monitor-vinduet er mørk	✓
Blodtrykssignal og -værdier(systole og diastole) skal være røde, og puls skal være grøn	Der ses på Monitor-vinduet om blodtrykssignal og -værdier(systole og diastole) er røde, og om puls er grøn	Blodtrykssignal og -værdier(systole og diastole) er røde, og puls er grøn	Blodtrykssignal og -værdier(systole og diastole) er røde, og puls er grøn	✓

Systolisk og dia-	Der ses i	Det ses i	Det ses i	✓
stolisk blodtryk	Monitor-	Monitor-	Monitor-	
skal fremhæves	vinduet, om	vinduet, at	vinduet, at	
øverst i ihøj-	det systoliske	det systoliske	det systoliske	
re hjørne ved	og det diasto-	og det diasto-	og det diasto-	
større skrift-	liske blodtryk	liske blodtryk	liske blodtryk	
størrelse end	er fremhævet	er fremhævet	er fremhævet	
andre værdier i	ved større skrift-	ved større skrift-	ved større skrift-	
Monitor-vinduet	størrelse end	størrelse end	størrelse end	
(fx værdier på	andre værdier i	andre værdier i	andre værdier i	
akserne)	Monitor-vinduet	Monitor-vinduet	Monitor-vinduet	

Tabel 5.8: Accepttest af Ikke-funktionelle krav