



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

14. december 2015

Indholdsfortegnelse

Kapitel 1 Kravspecifikation	1
1.1 Indledning	1
1.2 Systembeskrivelse	1
1.3 Funktionelle krav	1
1.3.1 Aktør-kontekstdiagram	2
1.3.2 Aktørbeskrivelse	3
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	19
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	30
3.2.3 Test af Signalbehandlingsblok	33

Kapitel 4 SW implementering og test	37
4.1 Implementering	37
4.1.1 Datalag	38
4.1.2 Logiklag	39
4.1.3 Præsentationslag	39
4.1.4 Datamodel Laget	40
4.1.5 Metodebeskrivelser	42
4.2 Test	48
4.2.1 Use Case 1: Kalibrer	48
4.2.2 Use Case 2 Vis måling	51
4.2.3 Use Case 3: Nulpunktsjuster	54
4.2.4 Use Case 4: Deaktiver filter	55
4.2.5 Use Case 5: Aktiver filter	57
4.2.6 Use Case 6: Gem måling	59
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	67
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 funktionerne 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 funktionerne kun dækker over de opstillede krav til projektet.
1.2	30-09-2015	Alle	Små ændringer 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

1.1 Indledning

Kravspecifikationen vil gennem seks Use Cases beskrive blodtryksmålerens 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 lokal database. Systemet er udviklet som en prototype, der er mulig at teste udfra de givne rammer.

1.3 Funktionelle krav

De funktionelle krav vil nedenstående 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. Det er selve systemet.

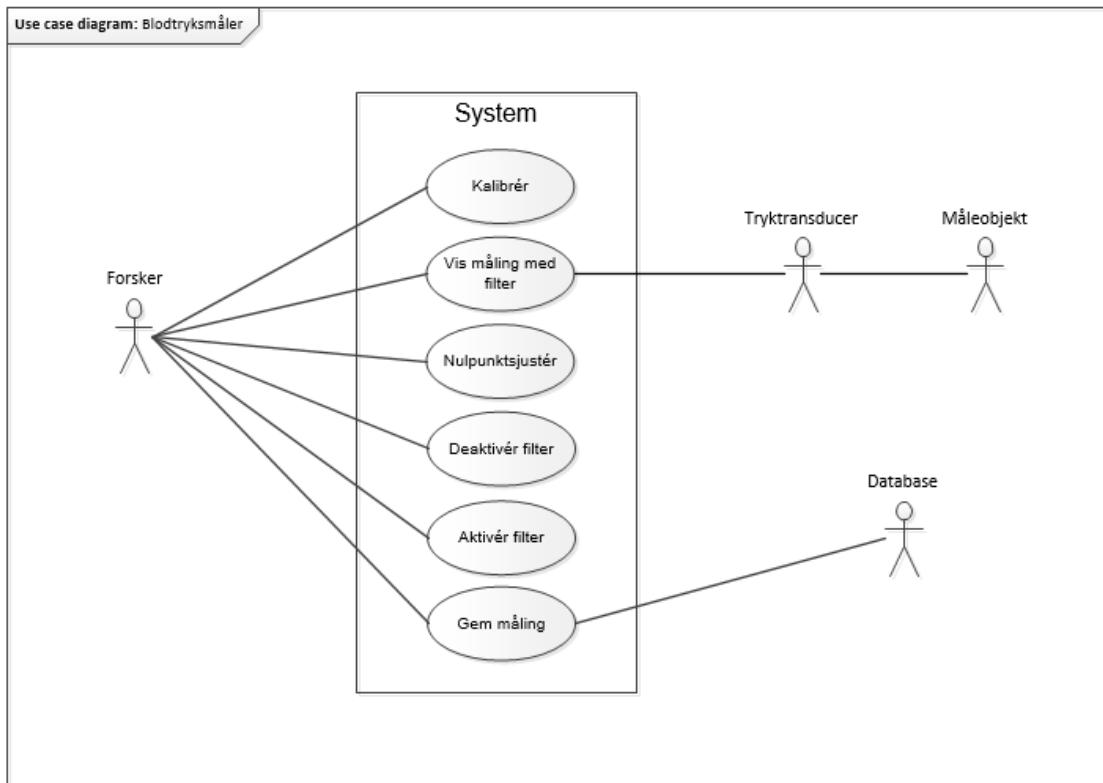
Primæraktøren i dette projekt er en Forsker. Sekundære aktører er Database, Tryktransducer og Måleobjekt. Måleobjekt er en package af Physionet og Analog Discovery, som er eksterne aktører.

1.3.2 Aktørbeskrivelse

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
Atørnavn	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. Det er Forskeren, der 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 tidligere 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 |
|-------------|--|

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-vindet

- | | |
|-------------|---|
| Hovedforløb | <ol style="list-style-type: none"> 1. Forsker vælger at aktivere digitalt filter 2. System udskriver det filtrerede blodtrykssignal |
|-------------|---|

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 <ul style="list-style-type: none"> [3.a <i>Forsker stopper optagelsen af måling inden optagelseslængden er nået</i>] 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

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

Brugssituitioner

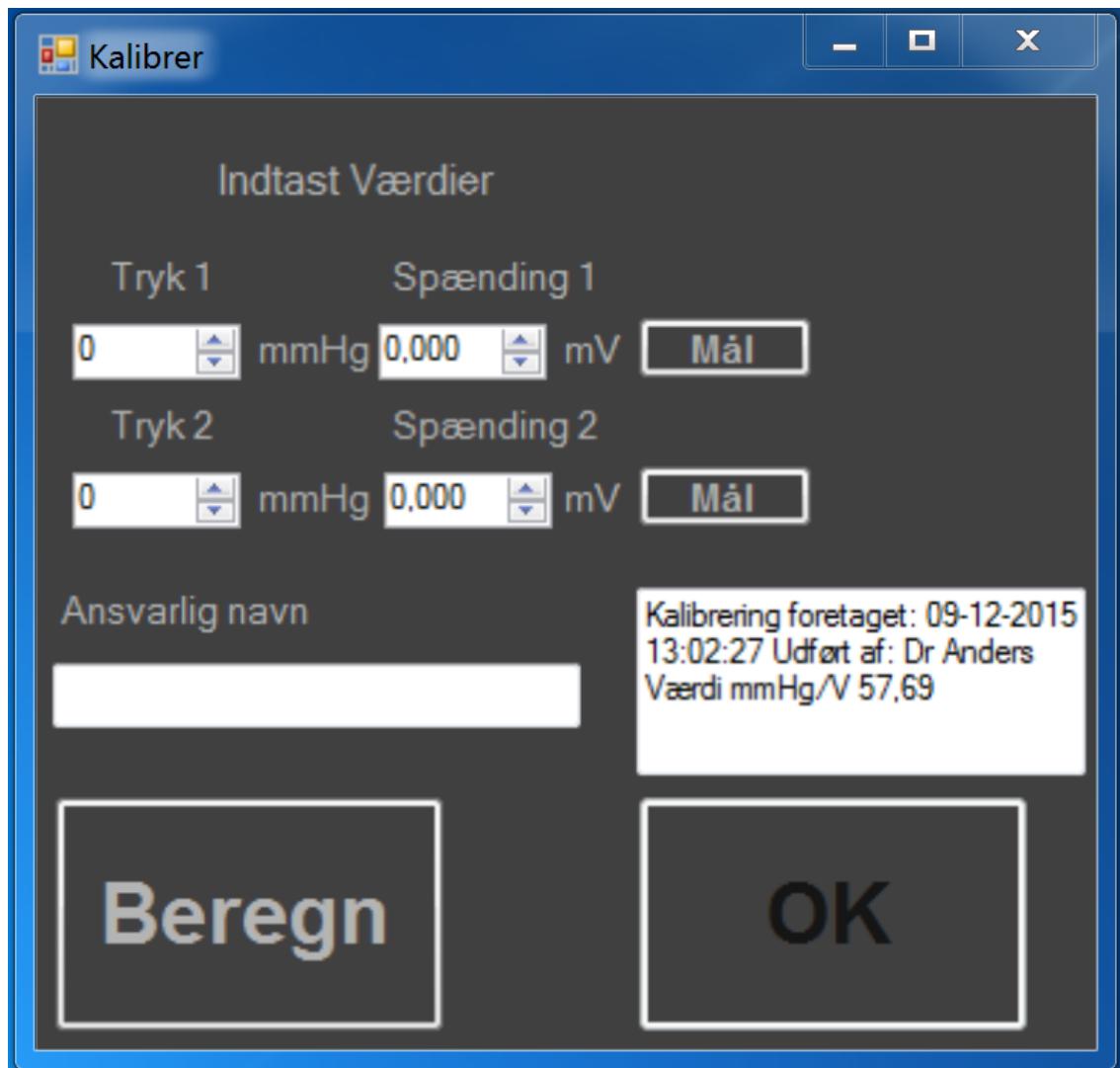
- Der skal være adgang til en computer med Windows 7 eller nyere – computeren skal have minimum 4 GB RAM.
- Der skal være adgang til en computer, hvor National Instruments er installeret.

Interface

- Blodtryksdiagrammet 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 ved større skriftstørrelse end andre værdier i Monitor-vinduet eksempelvis værdier på akserne.

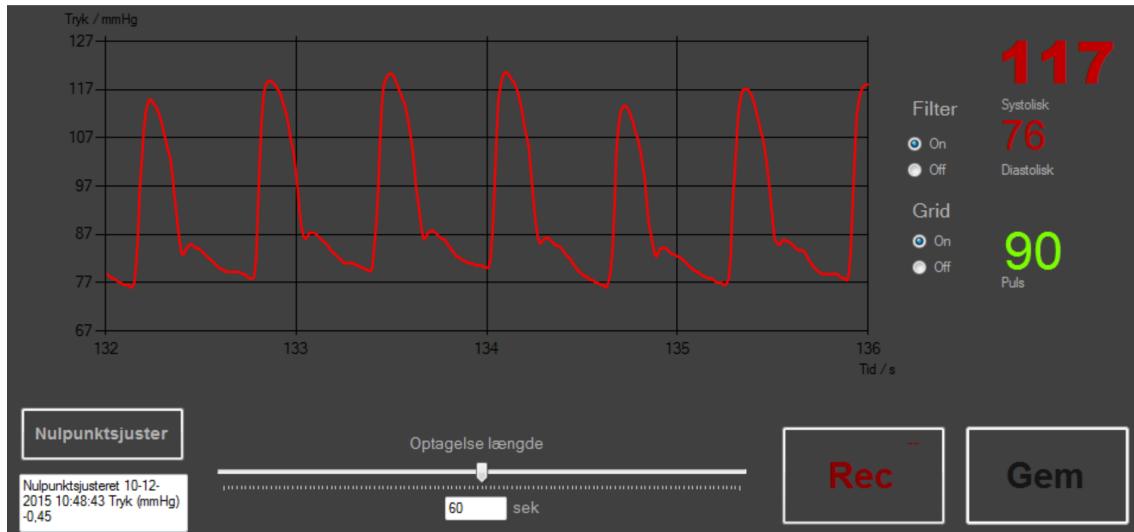
1.4.7 GUI

I det følgende afsnit vil der være indsatt figurer af, hvorledes det ønskes, at 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 (X.X).



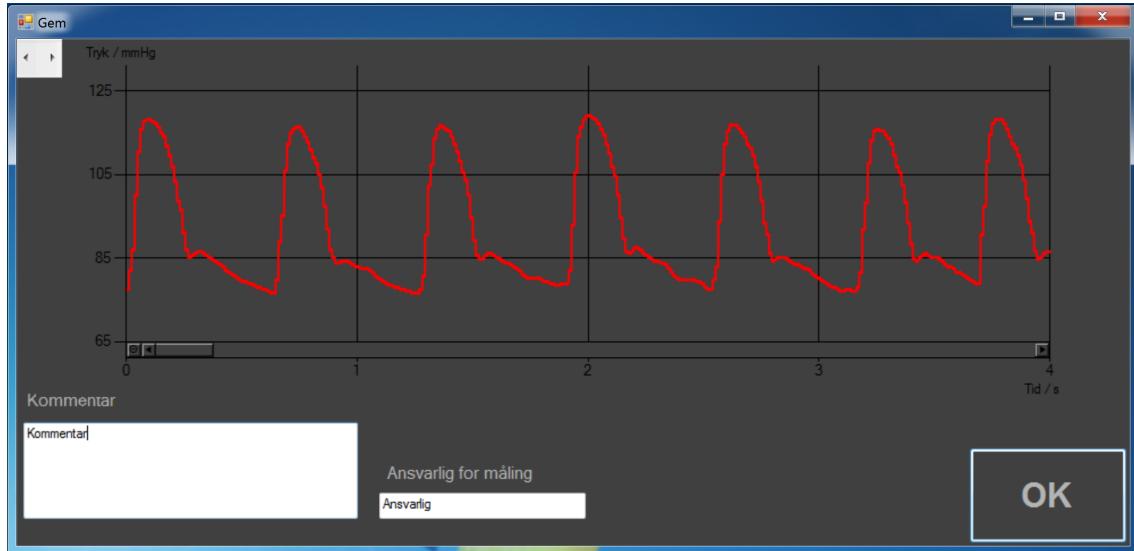
Figur 1.3: GUI: Kalibrerings-vindue

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

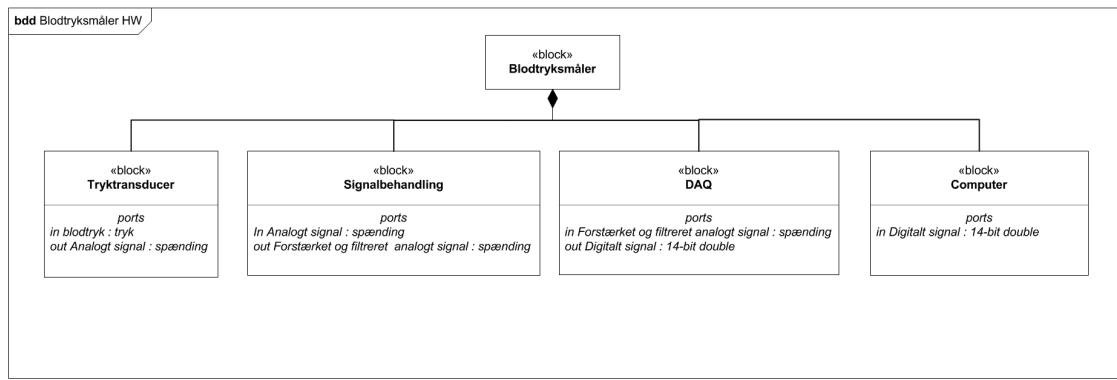
Version	Dato	Ansvarlig	Beskrivelse
1.0	20-10-2015	Alle	Første udkast til domænemodel, bdd, ibd og sekvensdiagrammer
1.1	21-10-2015	Alle	Små ændringer i bdd og ibd efter møde med vejleder
1.2	27-10-2015	Alle	Ændring af bdd og ibd efter møde med Kim; blokkene Filter og Forstærker er blevet lagt sammen 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	Alle	Skrevet hardwaredesign afsnittet. Små rettelser i de andre afsnit i Design således, det er klar til review

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.



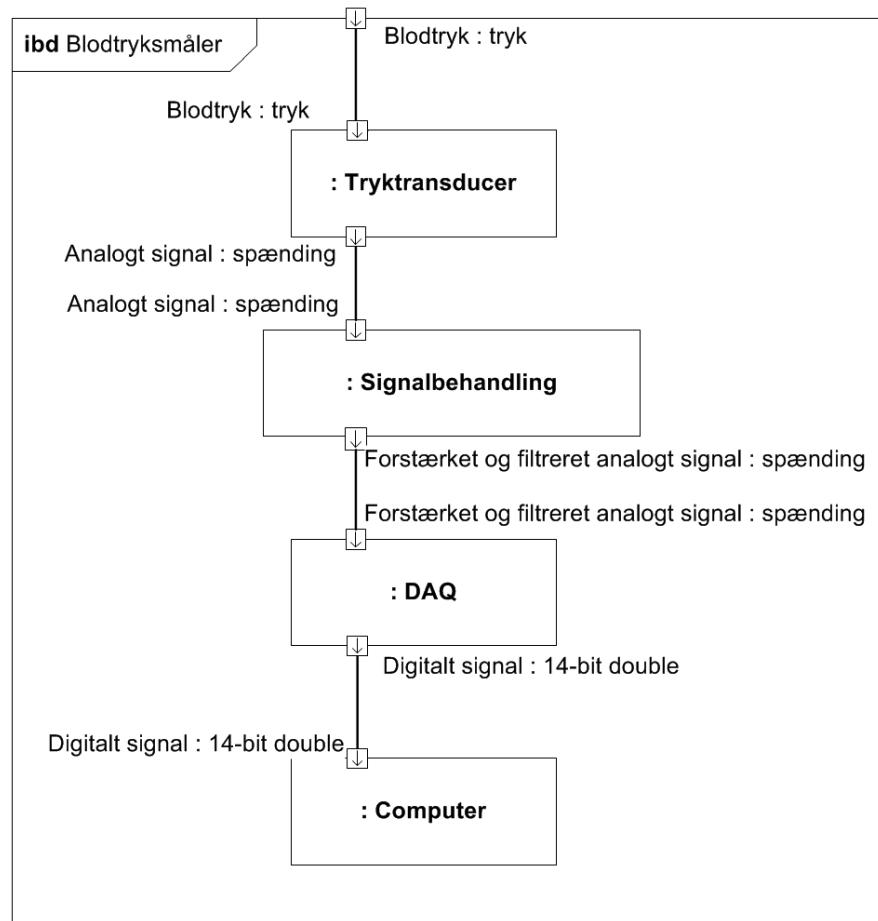
Figur 2.1: bdd

Blok	Beskrivelse
Blodtryksmåler	Det overordnede system, som indeholder Tryktransducer, Signalbehandling, DAQ og Computer.
Tryktransducer	Registrarer en fysisk størrelse i form af en trykændring. Tryktransducenten har til opgave at transformere den fysiske størrelse til en elektrisk spænding, som viderebehandles gennem de resterne hardware blokke.
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 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 målbare område skal ligge i intervallet 0-300 mmHg. Den maksimale outputspænding bliver derfor udtrykt fra det maksimale tryk på 300 mmHg, sensitiviteten på 5μ og inputspændingen på 5 V. Inputspændingen kommer reelt fra 9 V batterier, men der er valgt at indsætte en 5 V regulator, så man er sikker på,

hvad inputsspændingen er, da batterier er 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. Tryktransducerens 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 meget svage spændingssignal fra Tryktransduceren bliver forstærket til en spænding, der ligger indenfor det måleinterval, der svarer til DAQ'en, som er valgt til +/- 5 V.

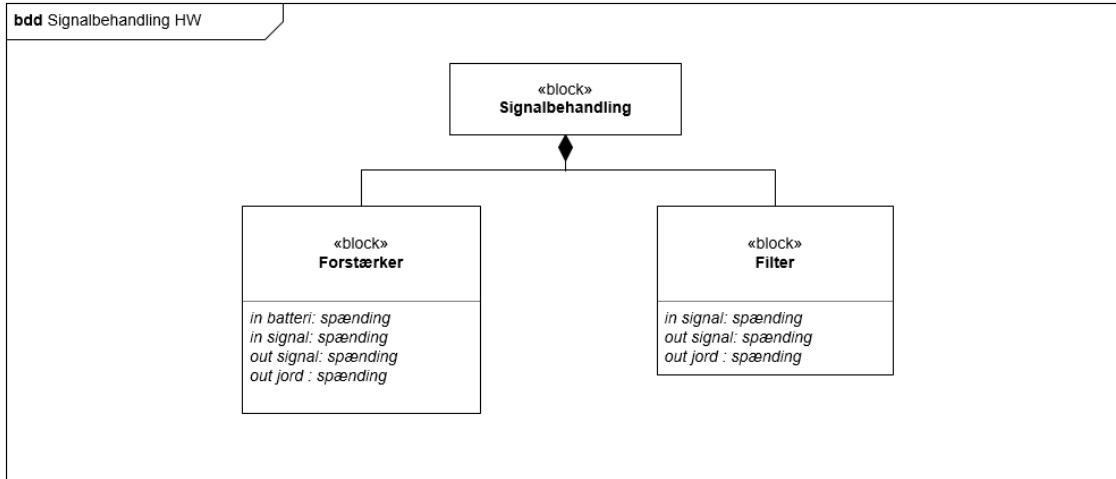
¹Se datablad for Tryktransduceren i bilag

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

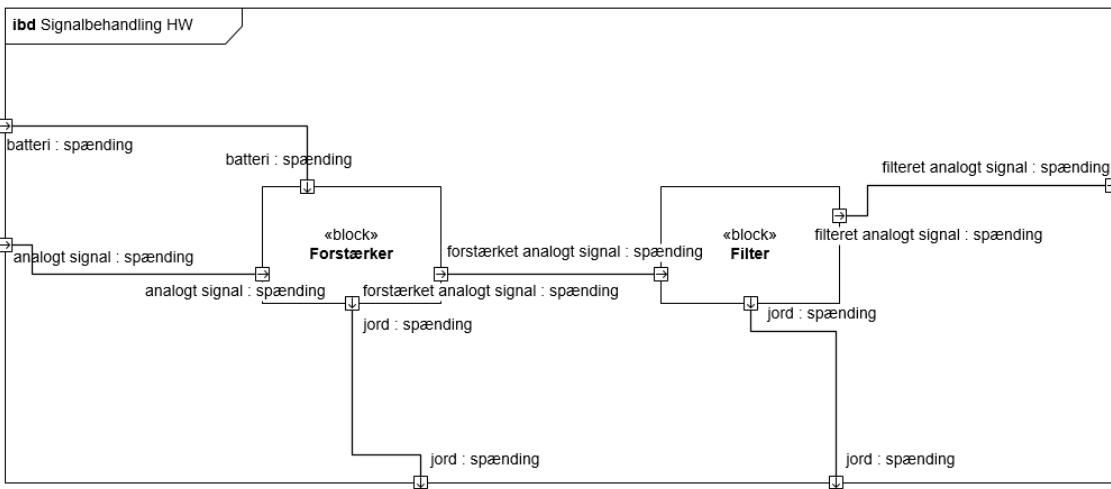
På Figur 2.3 ses bdd for Signalbehandlingsblokken, hvor Forstærkerblokkens og Filterblokkens input- og outputporte er vist.



Figur 2.3: bdd for Signalbehandling HW

2.3.2 Ibd

På Figur 2.4 ses 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

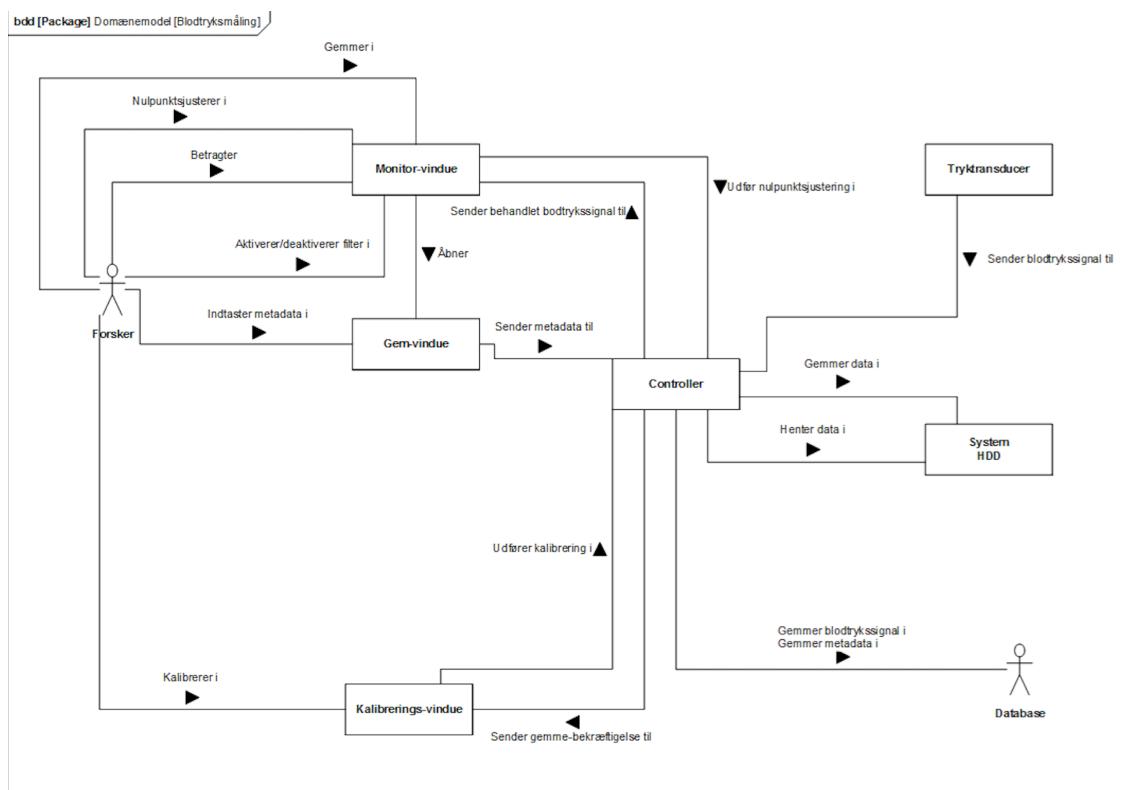
- Anden ordens lavpasfilter
- Cutoff frekvens ved 50 Hz
- Unity gain (ingen forstærkning)
- -40 dB ved 500 Hz

- Uendelig indgangsimpedans i teorien
- Inputspænding +/- 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 det er den, der sørger for, at systemet fungerer optimalt og udfører kommandoer.



Figur 2.5: Domænemodel for blodtryksmålesystemet

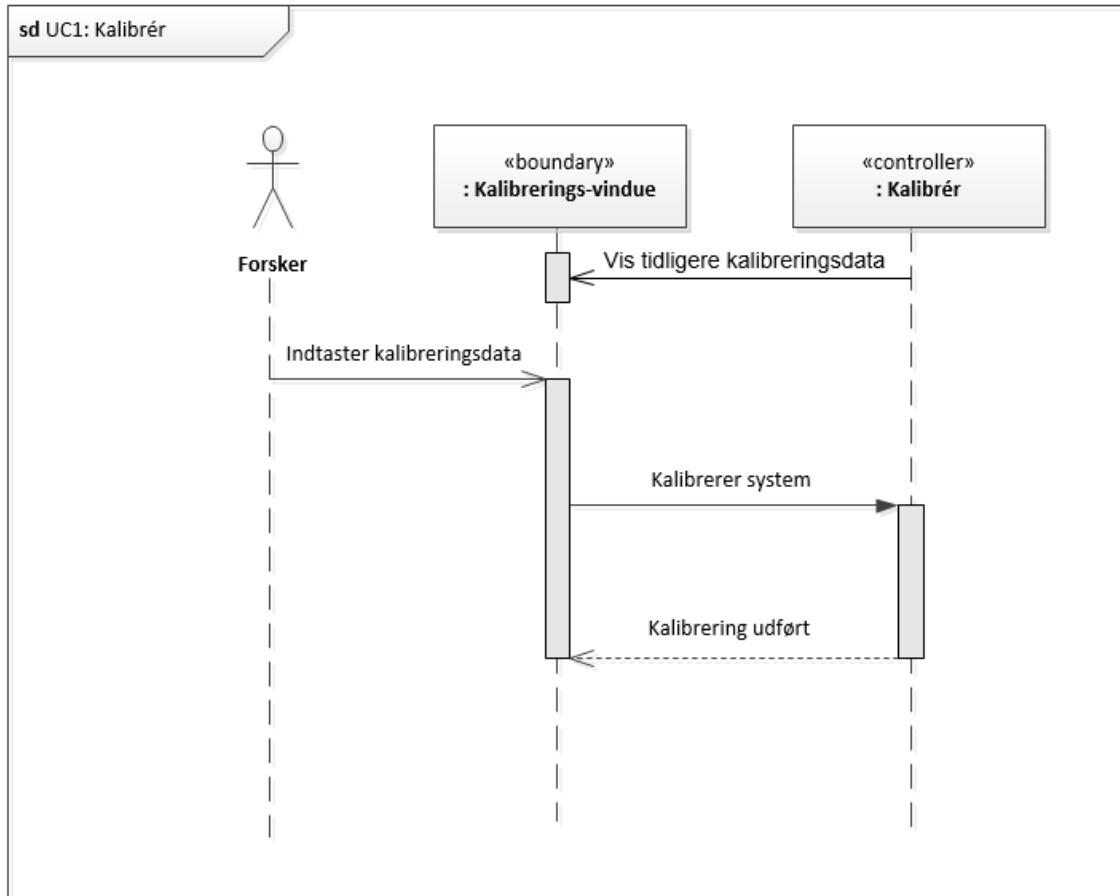
I domænemodellen ses generelle principper om kommunikation mellem klasser og aktører. Kommunikationen kan derfor være af forskellig karakter, og funktionaliteten af den givne kommando kan variere fra Use Case til Use Case.

2.4.2 Applikationsmodel

Sekvensdiagram

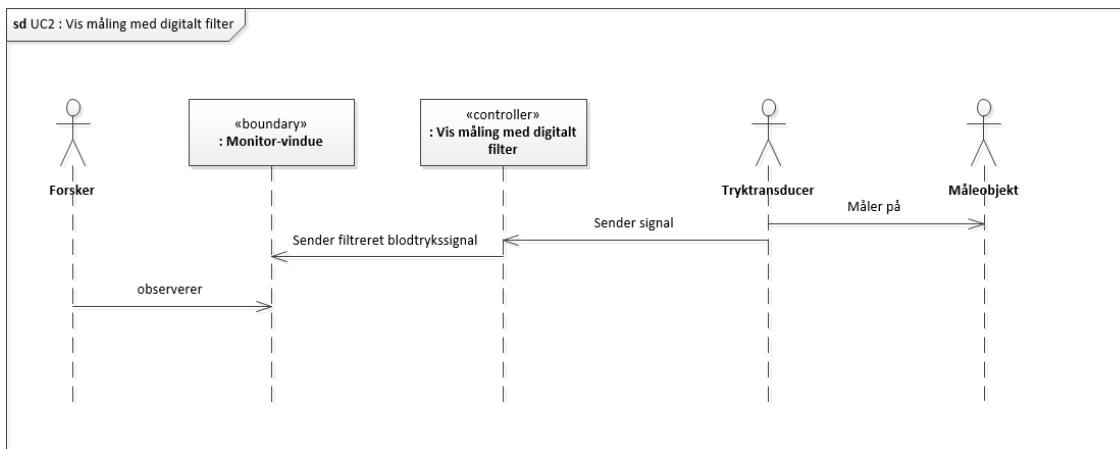
Sekvensdiagrammerne beskriver step-by-step, via metoder, forløbet i de forskellige Use Cases. Der er lavet et sekvensdiagram for hver Use Case for at gøre systemet mere

overskueligt. Et sekvensdiagram består af boundary-klasserne og domain-klasserne fra domænemodellen, samt en controller-klasse, med navn efter den specifikke Use Case.



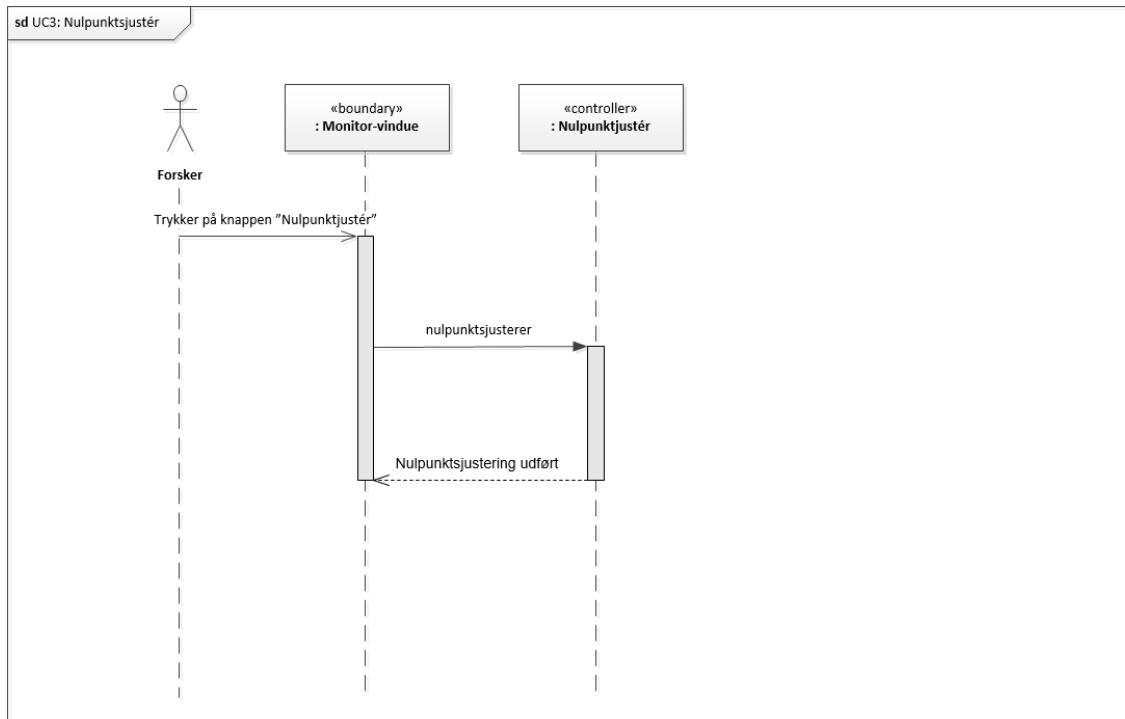
Figur 2.6: Sekvensdiagram for Use Case 1

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.



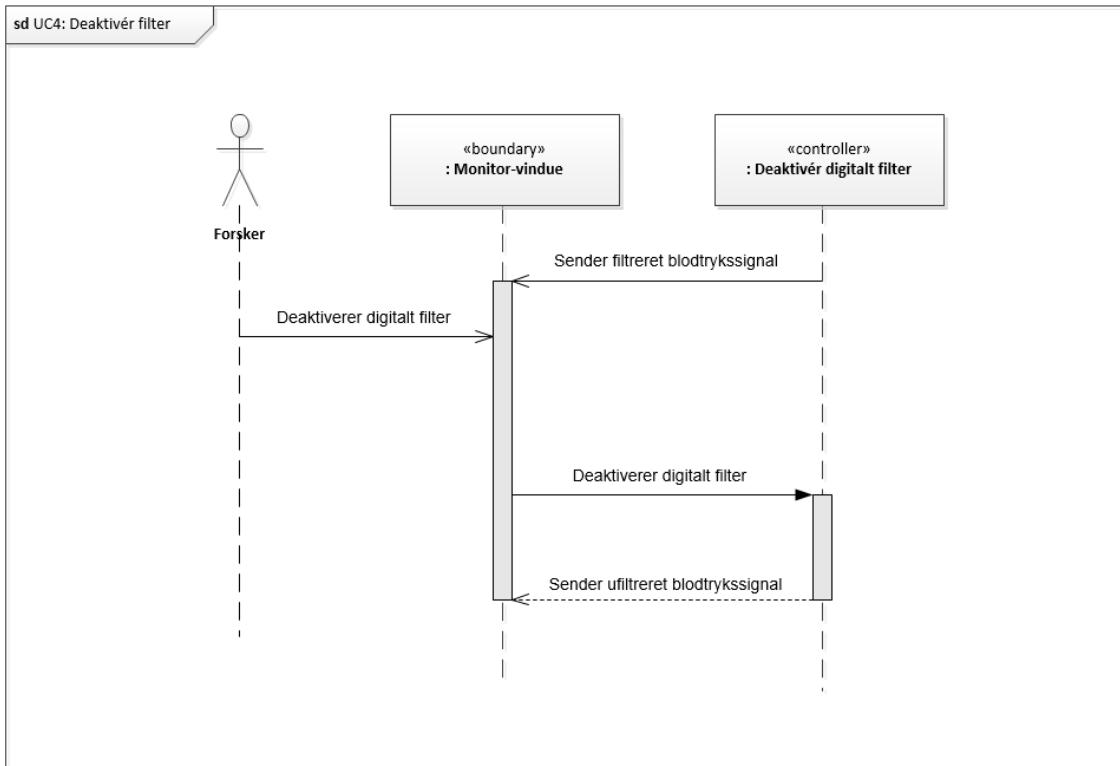
Figur 2.7: Sekvensdiagram for Use Case 2

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 metoden. Monitor-vindue opdateres, og herefter kan Forsker aflæse blodtryk.



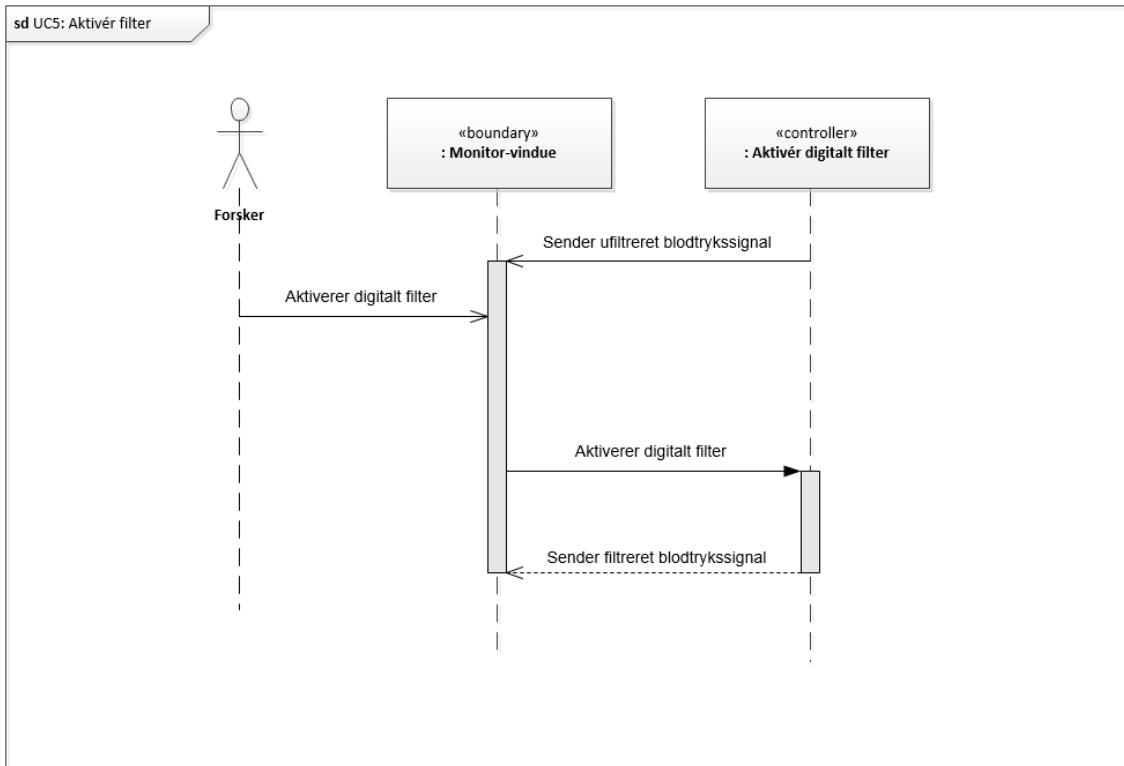
Figur 2.8: Sekvensdiagram for Use Case 3

Forsker interagerer med Monitor-vindue ved at trykke på knappen "Nulpunktsjustér". Derefter kaldes metoden, og nulpunktsjusteringen startes. Tidspunktet og værdien for nulpunktjusteringen sendes og gemmes i Databasen, hvorefter Forsker får besked om, at nulpunktsjusteringen er foretaget.



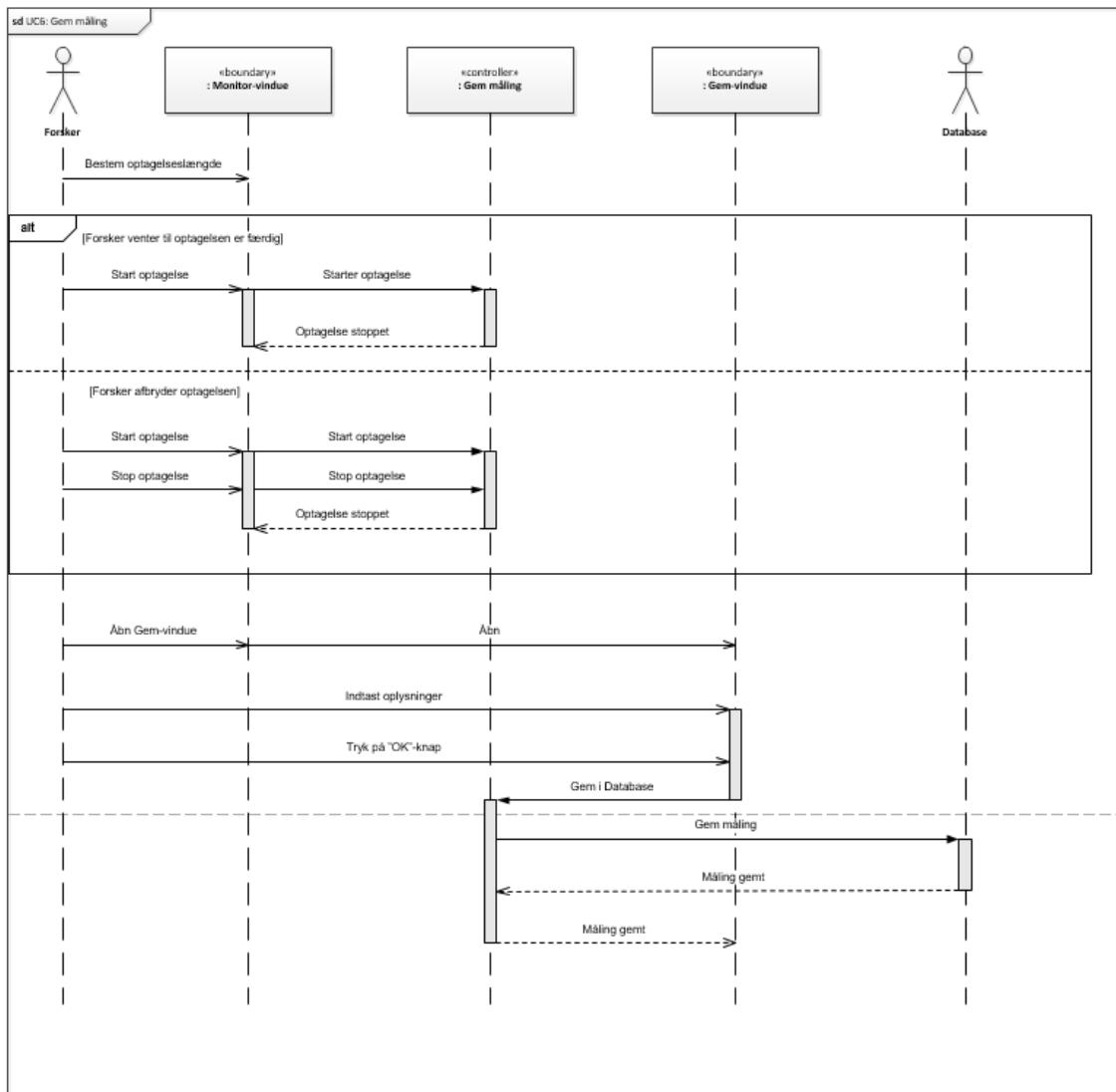
Figur 2.9: Sekvensdiagram for Use Case 4

Forsker interagerer med Monitor-vindue ved at markere i radiobutton ”Deaktivér digitalt filter”. Derefter kaldes metoden, og filteret deaktiveres, hvorefter signalet bliver udskrevet ufiltreret.



Figur 2.10: Sekvensdiagram for Use Case 5

Forsker interagerer med Monitor-vindue ved at markere i radiobutton ”Aktivér digitalt filter”. Derefter kaldes metoden, og filteret aktiveres, hvorefter signalet bliver udskrevet filtreret.



Figur 2.11: Sekvensdiagram for Use Case 6

Forsker interagerer med Monitor-vindue ved at trykke på knappen "Gem". Derefter kaldes metoden og Gem-vinduet åbnes. Første gang Forsker ønsker at gemme, indtastes data om målingen og der trykkes på knappen "OK". Kommando sendes og data gemmes. De efterfølgende gange, der ønskes at gemme, er data udfyldt fra første gang, og der trykkes blot på "OK", hvorefter kommandoen sendes. Data gemmes og Gem-vinduet lukkes. Controller bekræfter til Monitor-vindue, at data er gemt.

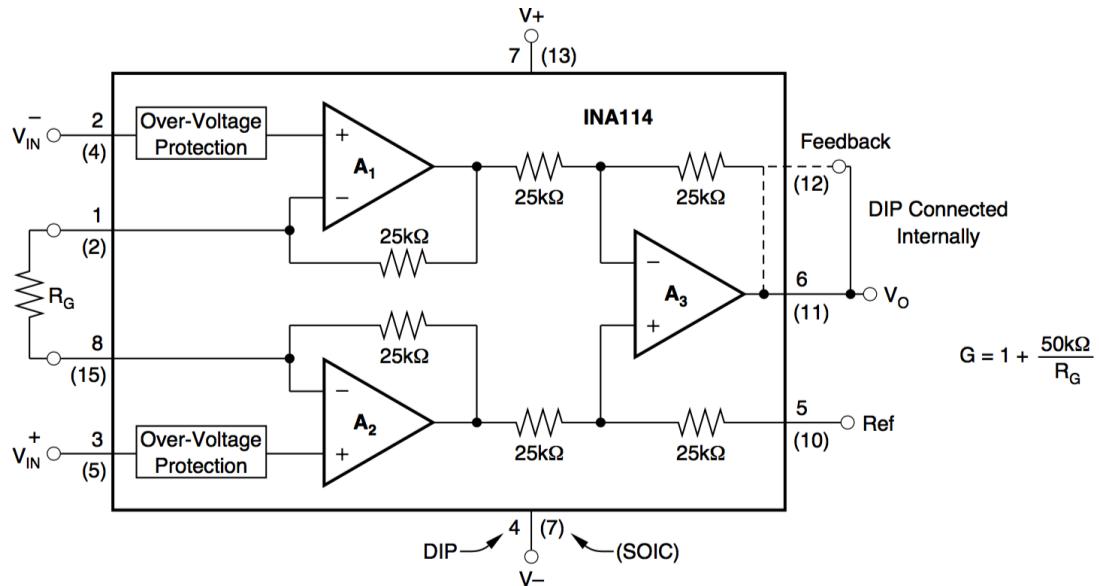
HW implementering og test

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 .

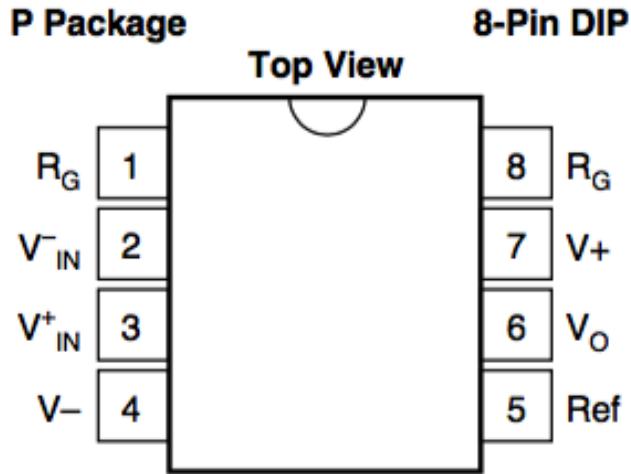


Figur 3.1: INA114

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 variabel 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 forbinderes INA114 med gainmodstanden, R_G . Ben fire forbinderes med -9 V. Ben syv forbinderes med +9 V. Ben fem forbinderes til jord. Inputsignalet modtages ved ben to og tre, mens outputsignalet kommer fra ben seks, som forbinderes 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 kan 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 skulle være.

Gainmodstand

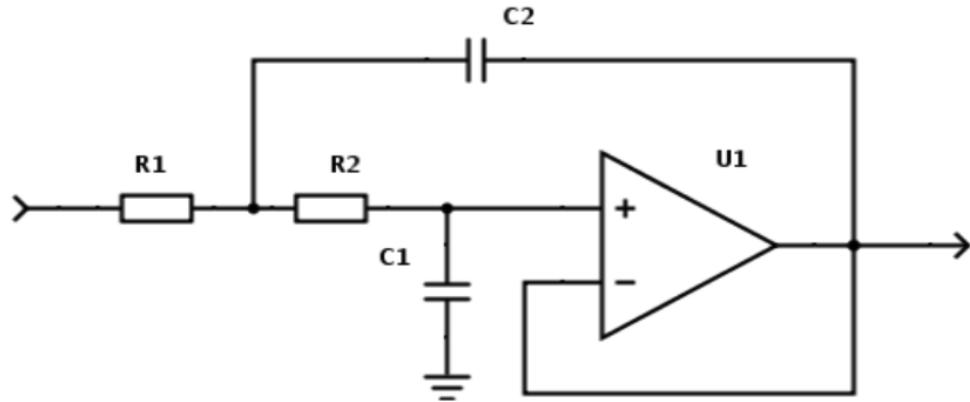
Gainmodstanden, R_G er en variabel modstand, der bestemmer forstærkningen i INA114. INA114 skal have en gain på 666,667. Ud fra ligning (3.3) kan R_G isoleres og beregnes.

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

²Se datablad for INA114 s. 2 i bilag

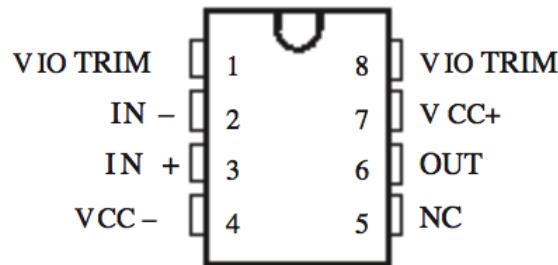
3.1.2 Filterblok

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 færdigtbehandlede signal kan måles. Ben fire forbindes med -9 V. Ben syv forbindes med +9 V.

Beregning af komponentværdier

Ud fra overføringsfunktion for filteret kan de forskellige komponentværdier beregnes.

Overføringsfunktion

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

³Se datablad for OP27G i bilag

Standardform

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

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

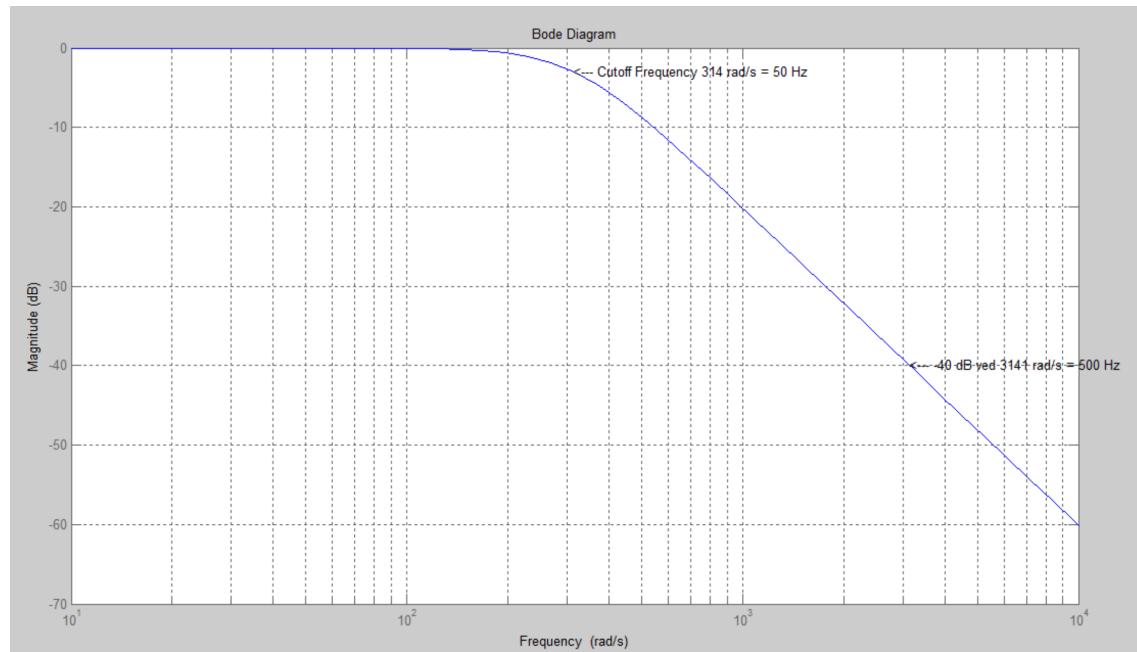
Kondensatoren, C2, skal være 680 nF og kondensatoren, C1, bestemmes til 340 nF, da denne kondensator kunne realiseres ud fra det, der var til rådighed. Der benyttes en kondensator på 330 nF og 10 nF til denne realisering. Modstandene R1 og R2 skal være samme værdi, $R = R_1 = R_2$

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

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

Magnitude Bodeplot

Bodeplottet i Figur 3.3 viser, at ved -3 dB, hvor cutoff frekvensen befinner 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.



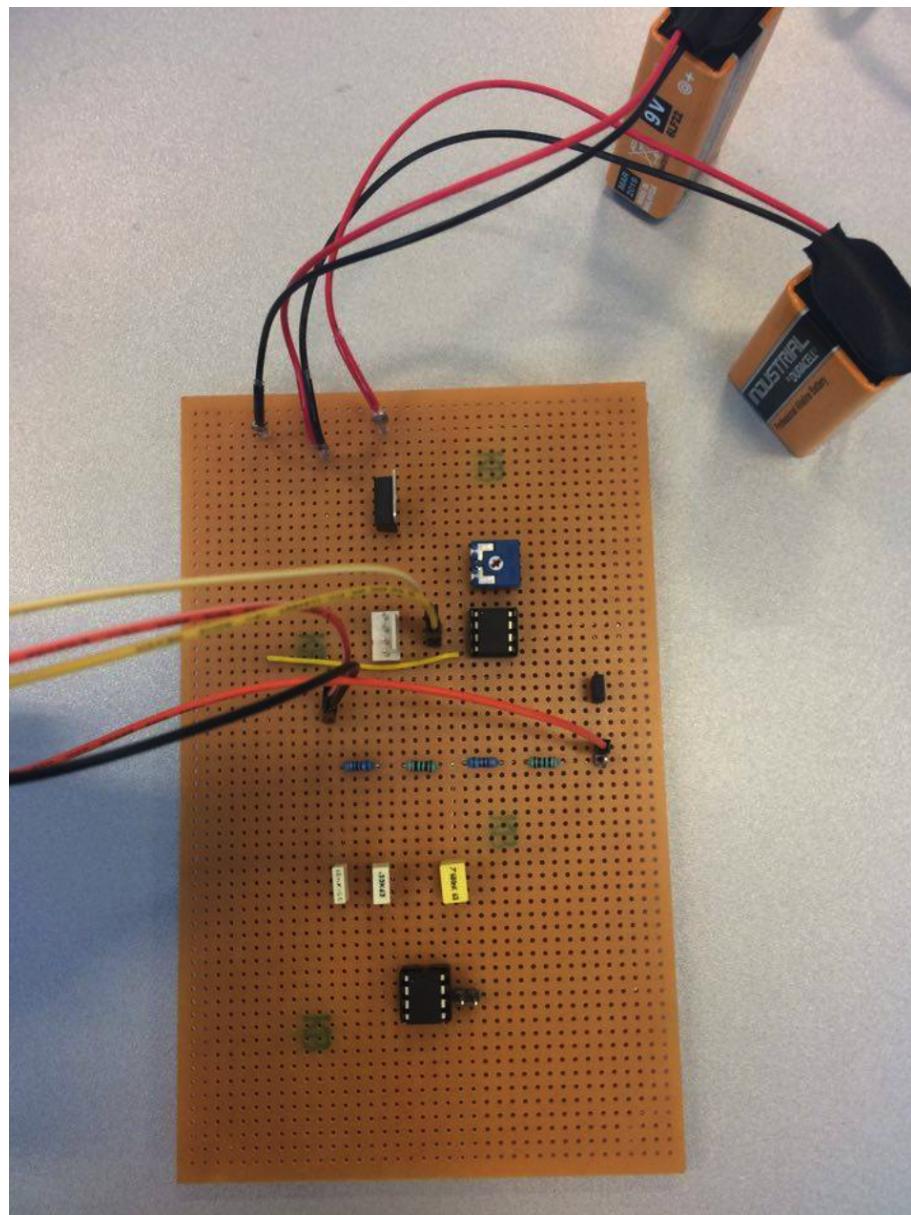
Figur 3.5: Magnitude Bodeplot Lavpasfilter

3.2 Test

Forstærkeren og filteret testes via Analog Discovery og Waveforms. Der foretages først unit test 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 3.4.



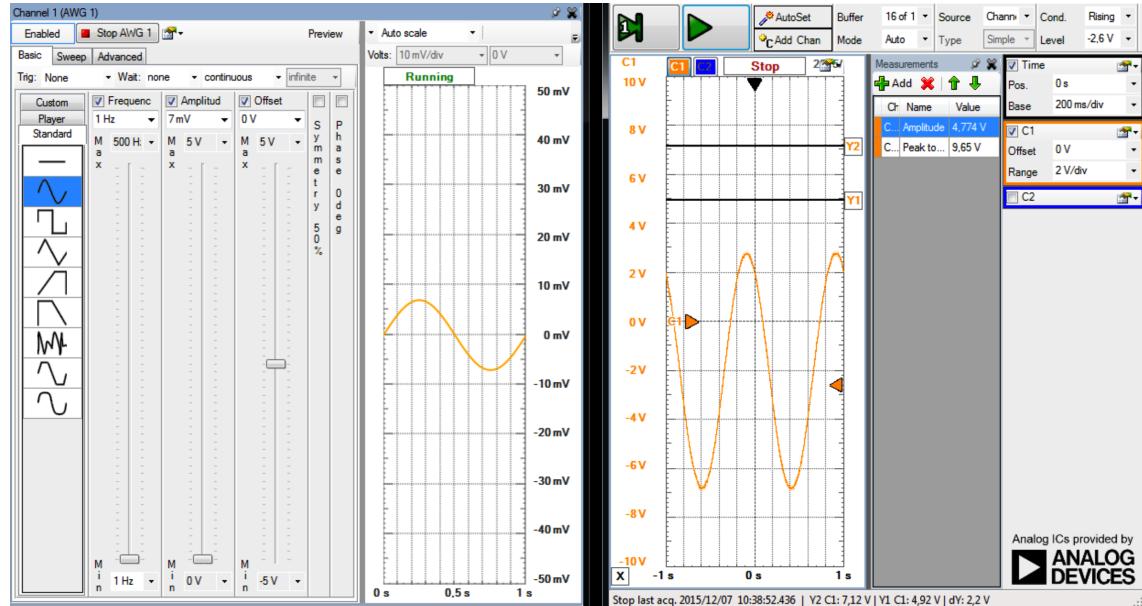
Figur 3.6: Testopstilling for Forstærkeren

Der ønskes, at ved en inputspænding på 7,5 mV vil outputspændingen være forstærket op til 5 V. Der kunne ikke påtrykkes en spænding på 7,5 mV, men istedet med 7 mV.

Outputspændingen ved 7 mV er

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

Dette ser at være det samme, når man tester Forstærkeren, se Figur 3.5.

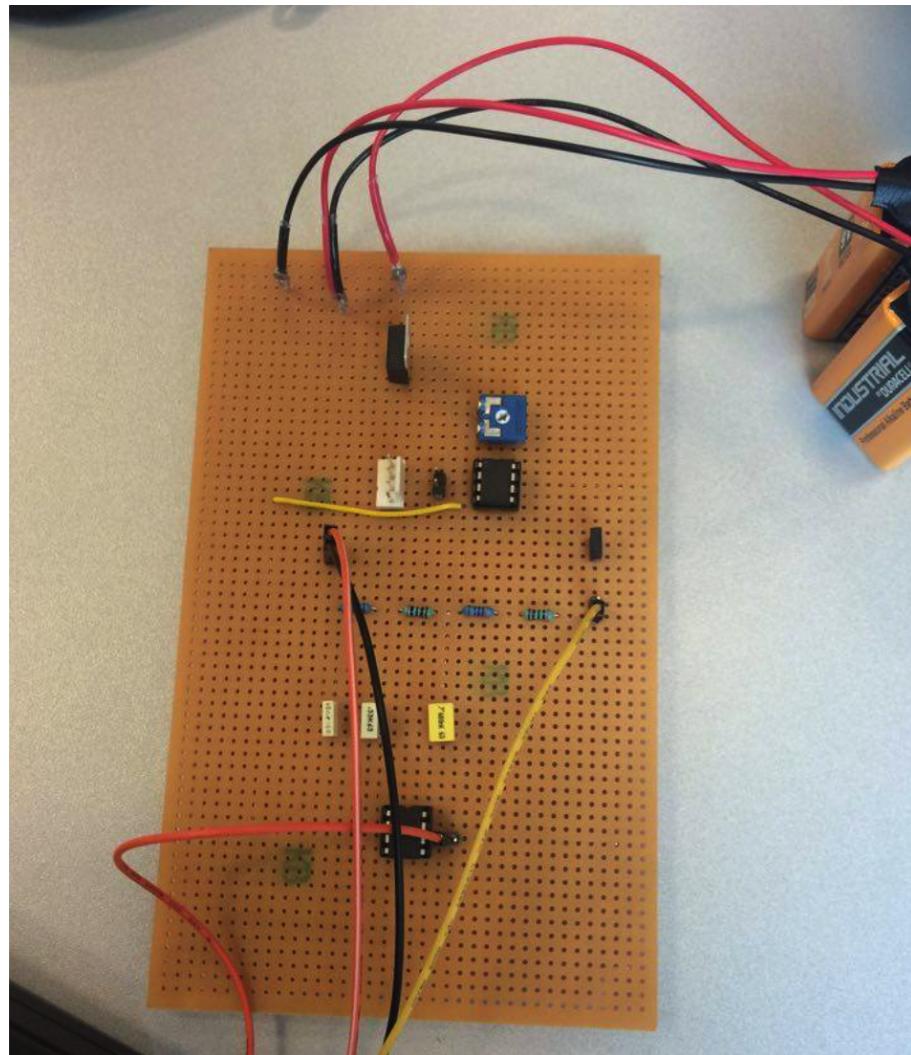


Figur 3.7: Forstærker respons ved 7 mV

3.2.2 Test af Filterblok

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

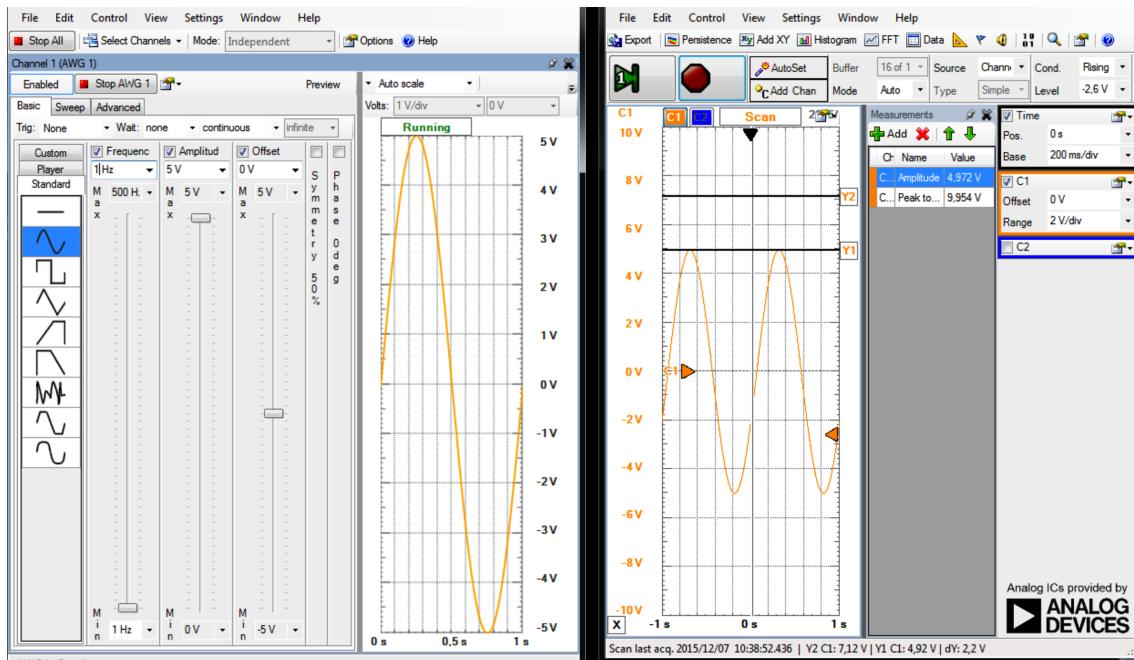
Testopstillingen ses på Figur 3.6.



Figur 3.8: Testopstilling af filteret

Channel 1 er inputspændingen og viser den varierende frekvens. Ændringen ses i venstre side af efterfølgende skærmbilleder. Oscilloskopet mäter outputspændingen på filteret og resultatet vises i højre side af efterfølgende skærmbilleder.

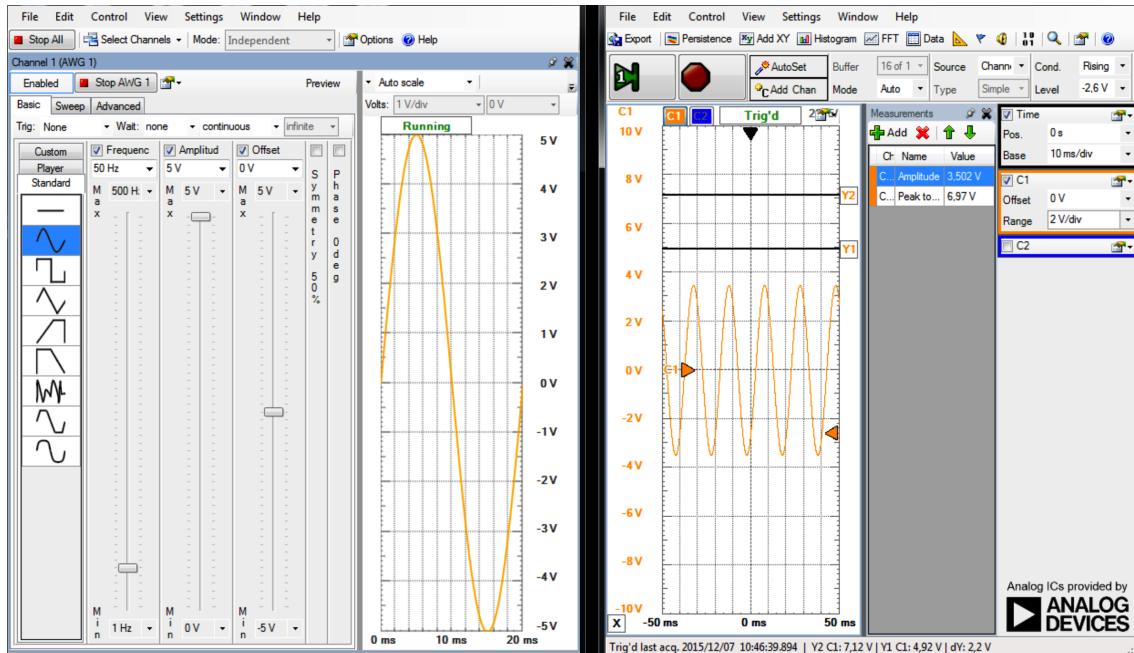
Filtering ved 1 Hz



Figur 3.9: Lavpasfilter respons 1 Hz

På Figur 3.7 ses det, at inputspændingen og outputspændingen er den samme - 5 V ca. Dette er også det ønskede resultat, da filteret er et lavpasfilter, der er designet til at lukke de lave frekvenser igennem indtil 50 Hz.

Filtering ved 50 Hz



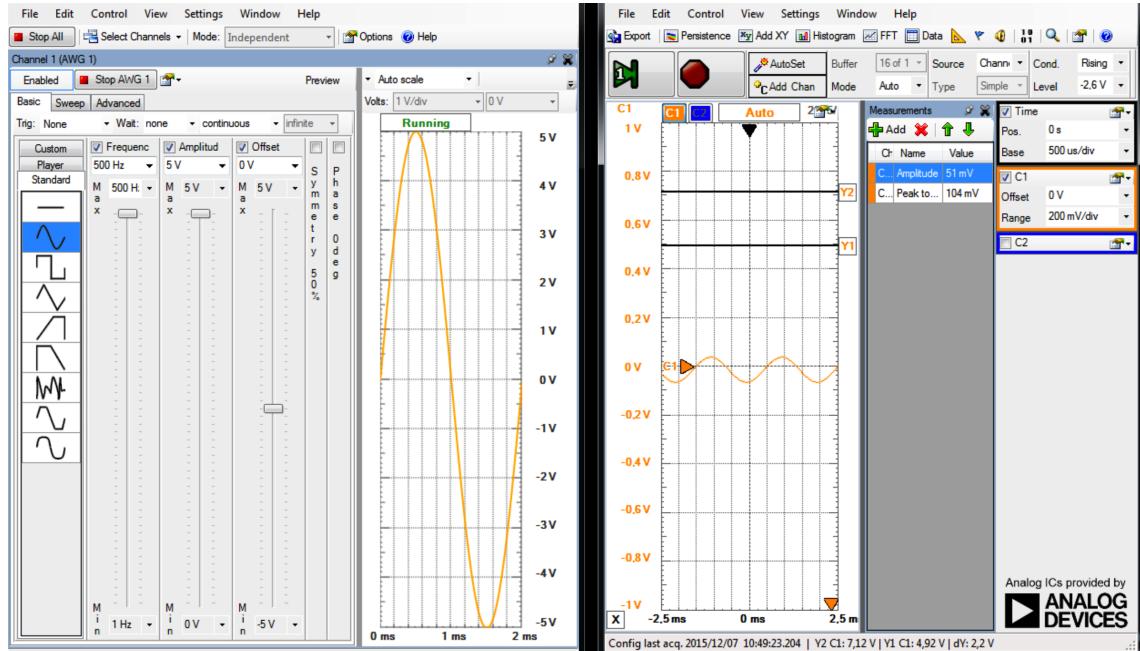
Figur 3.10: Lavpasfilter respons 50 Hz

Ved 50 Hz, som er filteret cutoff frekvens, skal inputspændingen være dæmpet med 3 dB efter filteringen. I ligningen (3.9) beregnes, hvad outputspændingen er efter en dæmpning med 3 dB.

$$3dB = 20 \cdot \log(x) \Rightarrow x = 0,707 \Rightarrow \text{outputs}pnding = 5V \cdot 0,707 \Rightarrow \text{outputs}pnding = 3,535V \quad (3.9)$$

På Figur 3.8 ses, at det i praksis er 3,502 V, hvilket der er acceptabelt.

Filtering ved 500 Hz



Figur 3.11: Lavpasfilter respons 500 Hz

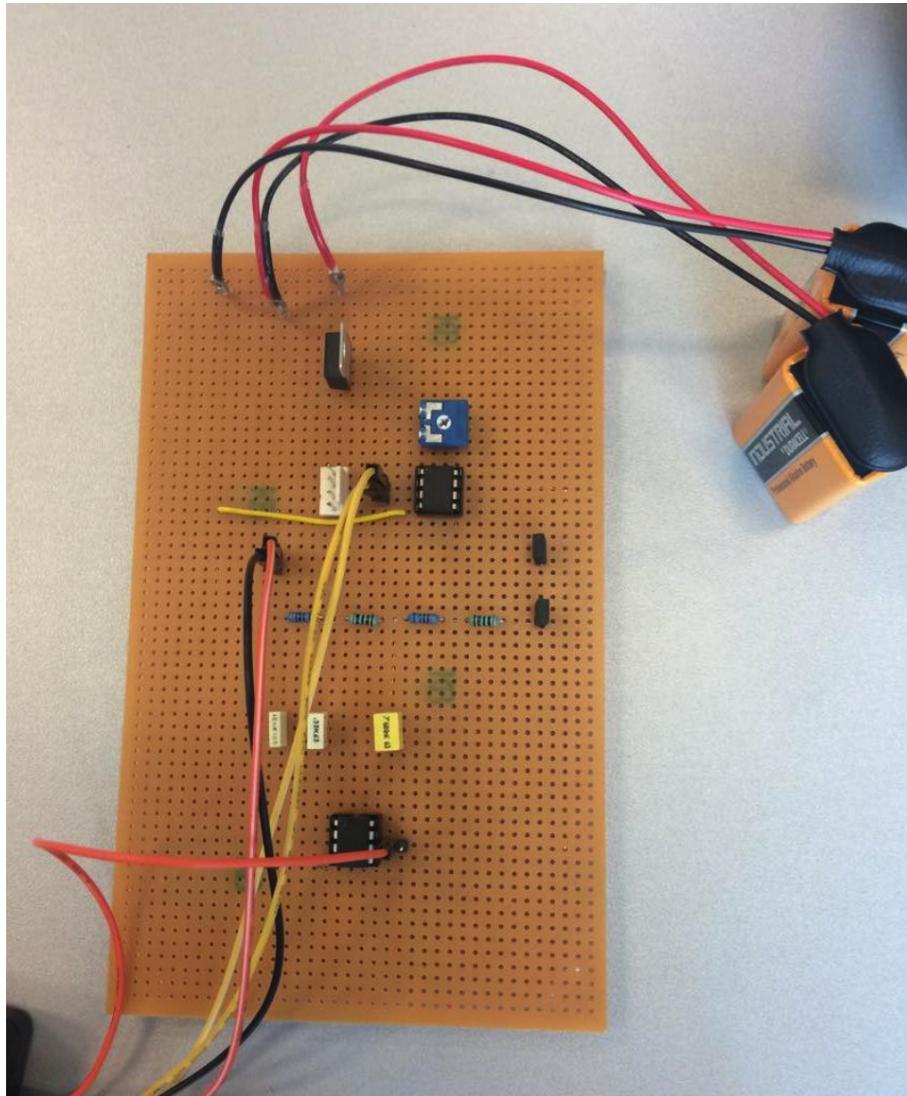
Her forventes der en dæmpning med 40 dB, da det er et anden ordens lavpasfilter, og 500 Hz er en dekade efter cutoff frekvensen. I ligning (3.10) ses beregningen, hvad outputspændingen er efter en dæmpning på 40 dB.

$$40dB = 20 \cdot \log(x) \Rightarrow x = 0,01 \Rightarrow \text{outputs}pnding = 5V \cdot 0,01 \Rightarrow \text{outputs}pnding = 0,05V \quad (3.10)$$

På Figur 3.9 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 3.10.



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.

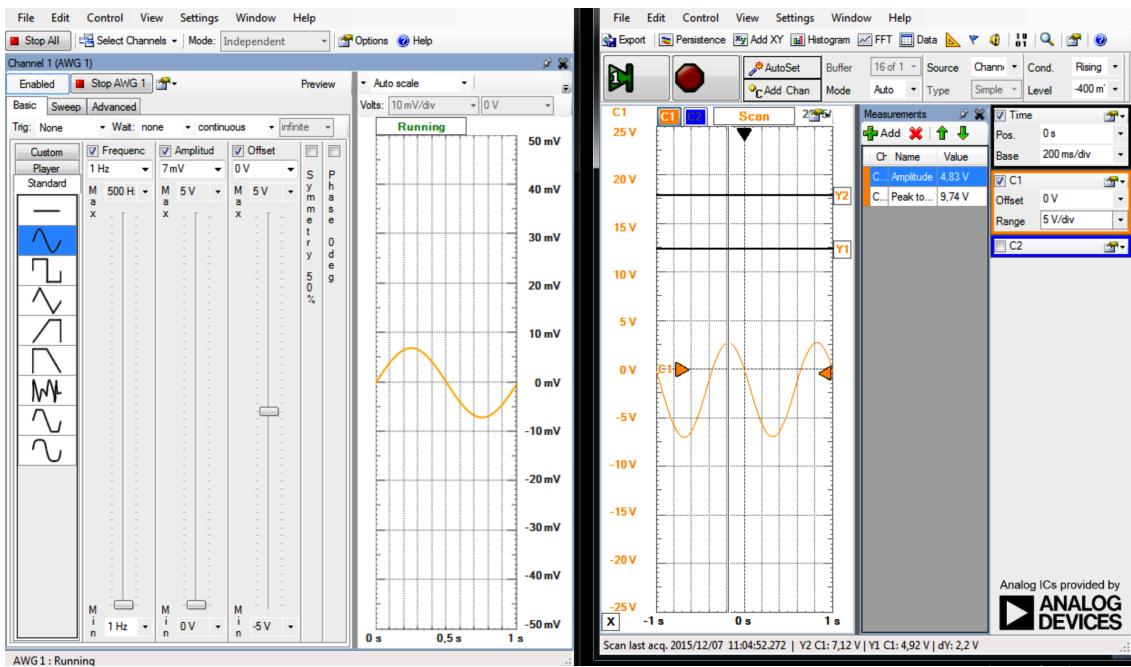
Test ved 1 Hz

På Figur 3.11 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 filterets cutoff frekvens.

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

3.2. Test

ASE

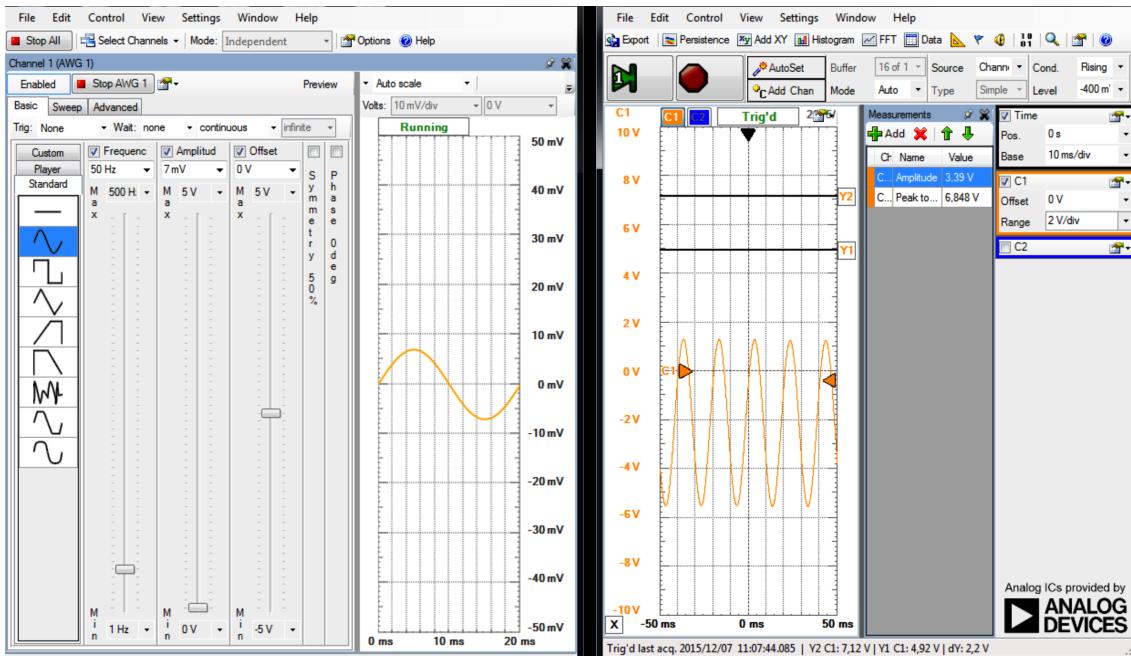


Figur 3.13: Signalbehanlding respons 1 Hz

Test ved 50 Hz

På Figur 3.12 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 3,535 V.

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

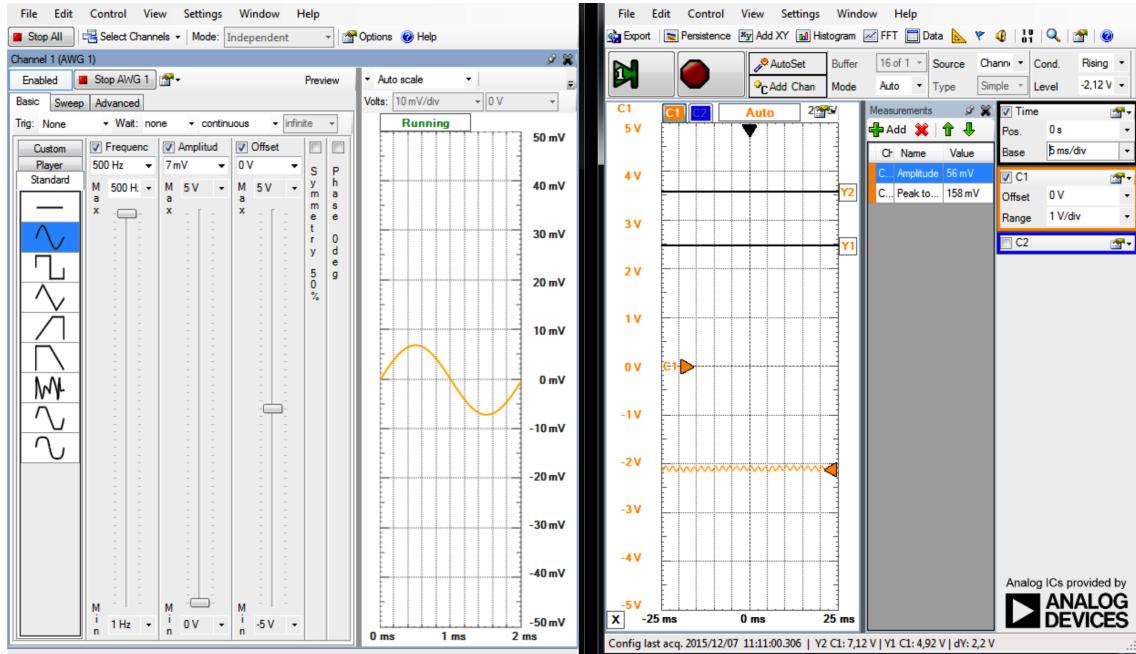


Figur 3.14: Signalbehanlding respons 50 Hz

Test ved 500 Hz

På Figur 3.13 ses resultatet ved 500 Hz. Der forventes, at Forstærkeren har forstørret 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.

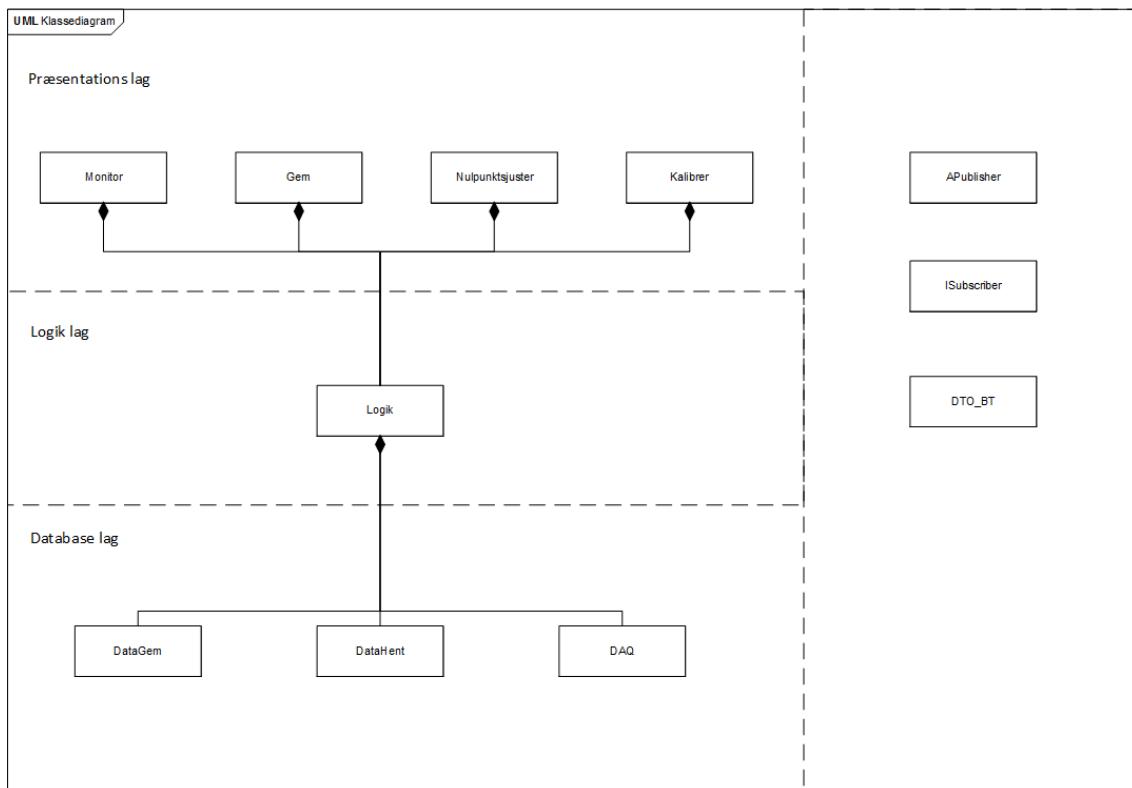


Figur 3.15: Signalbehandling respons 500 Hz

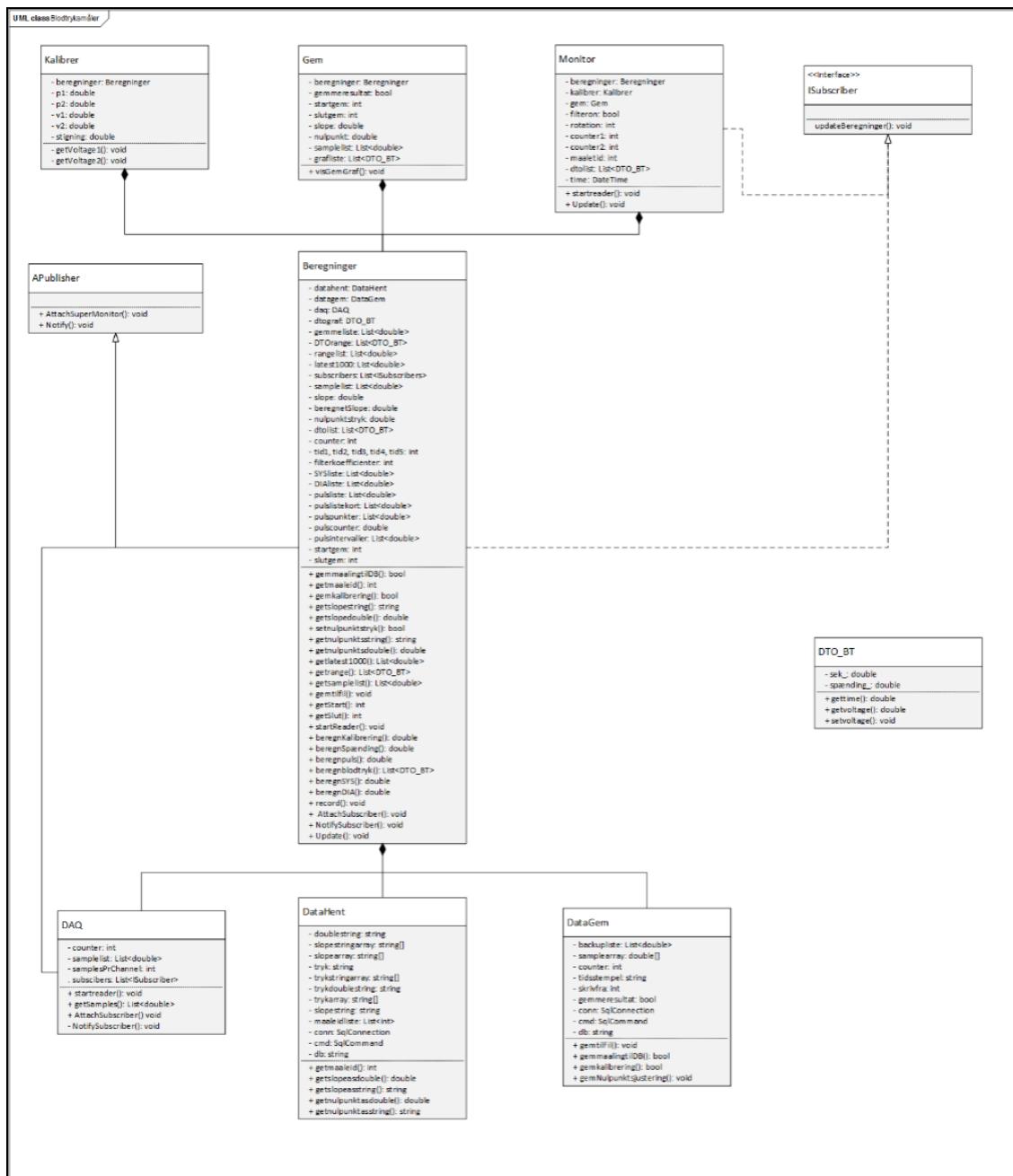
SW implementering og test 4

4.1 Implementering

Softwareen i blodtryksmålesystemet er opbygget omkring trelagsmodellen.



Figur 4.1: UML kompositions klassediagram



Figur 4.2: Klassediagram

4.1.1 Datalag

Datalaget kommunikerer med DAQ, Database og computerens harddisk.

DAQ

Herunder er en klasse, DAQ, hovedsageligt med software fra National Instruments, der kan udveksle data med den fysiske DAQ 6009. Denne kan betragtes som af typen "black box". Dette betyder, at undertegnede ikke har et begreb om den interne mekanisme i klassen. Som udgangspunkt er funktionen af klassen at levere 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 Database og lokale tekstmapper. Fra Databasen indhentes data om tidligere målingers nummerering således, at de målinger, der skal gemmes i fremtiden kan blive nummereret korrekt.

De lokale tekstmapper indeholder data omkring tidligere kalibreringer og nulpunktsjusteringer.

DataGem

Klassen gemmer data på Database og lokale tekstmapper.

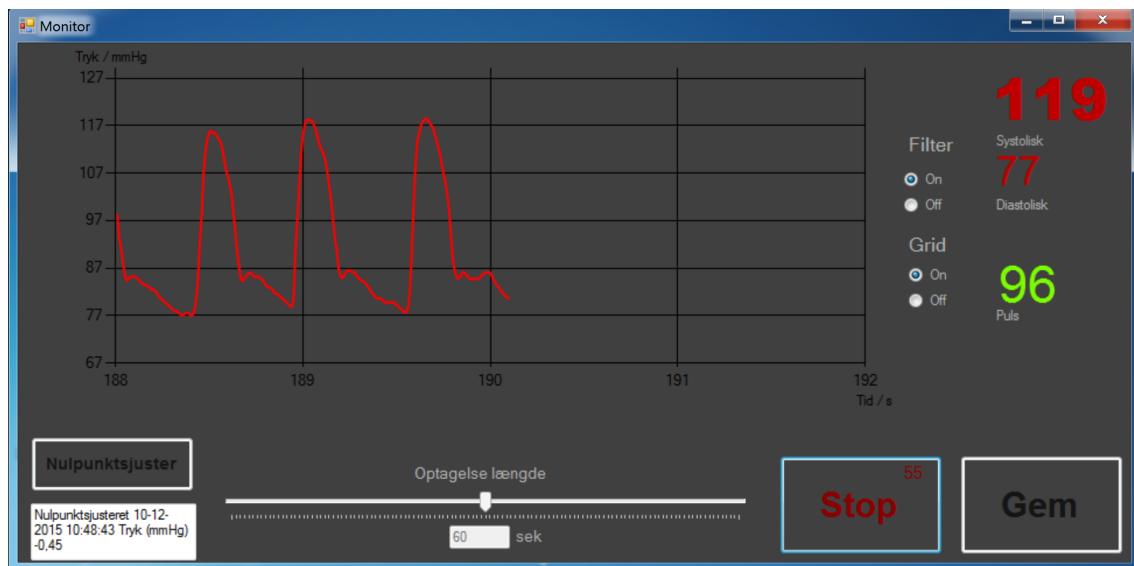
4.1.2 Logiklag

Beregninger

Beregninger er hovedsageligt et lag, hvor logiske beregninger foretages.

Herunder beregnes ting som 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



Figur 4.3: Monitor med igangsat måling

Monitor

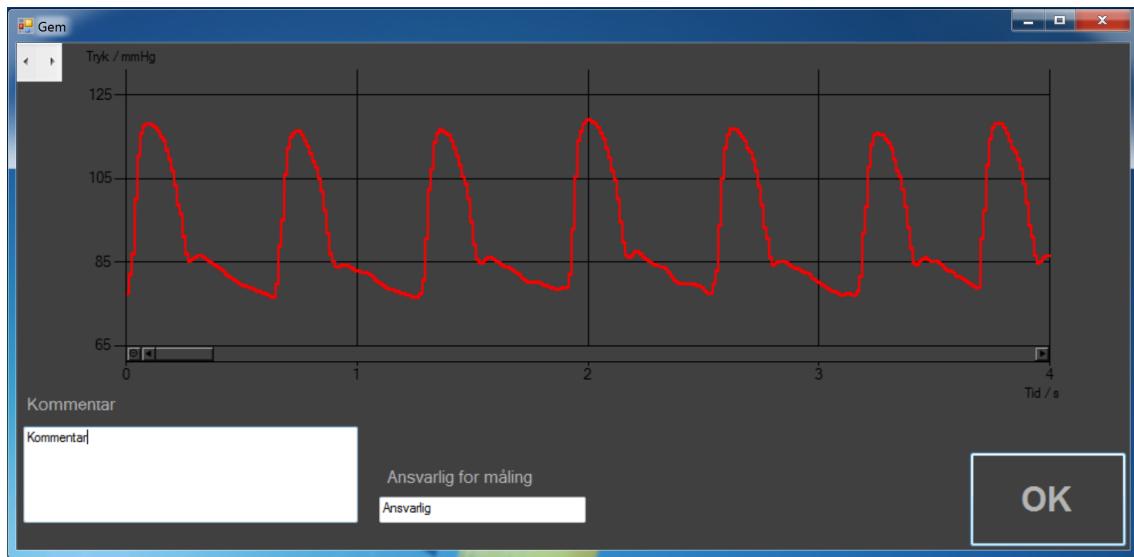
Monitorlaget er Forskerens generelle visuelle interaktionslag med systemet. Heri observeres udskrevne blodtrykssignaler over 4-sekunders intervaller. Værdierne puls, systolisk og diastolisk blodtryk udskrives ligeledes her. Designet er inspireret fra i forvejen eksisterende blodtryksmonitoreringssystemer.

Der er mulighed for at vælge en optagelseslængde for målingen, og start/stop med 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 radio buttons. Y- og X-akse kan påsættes med to radio buttons.

Gem

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



Figur 4.4: Gem-vindue med 10 sekunders måling

Kalibrér

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

4.1.4 Datamodel Laget

Datamodel 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 (blodtryk) korrigert for nulpunktstryk samt 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 udkrivning 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 Beregningeklassen, og fra Beregningeklassen til Monitorklassen ved hjælp af publisher/subscriber mønstret. Når data befinner sig i Monitorklassen kan der laves logiske beregninger og manipulation på data i Beregningeklassen. Forholdet mellem klasserne er således; DAQklassen og Beregningeklassen har et publisher/subscriber forhold, og Beregningeklasse og Monitorklassen har et publisher/subscriber forhold.

Hierakiet fremgår af figurerne nedenfor.

```

namespace DataLog
{
    class DAQ : APublisher
    {
        private AnalogMultiChannelReader analogInReader;
        private Task myTask;
        private Task runningTask;
        private AsyncCallback analogCallback;
    }
}

```

Figur 4.6: DAQ klassens arvehierarki

```

namespace Logik
{
    public class Beregninger : APublisher, ISubscriber
    {
        DataHent datahent = new DataHent();
        DataGem datagem = new DataGem();
        DAQ daq = new DAQ();
    }
}

```

Figur 4.7: Beregninger klassens arvehierarki

```

namespace GUI
{
    public partial class Monitor : Form, ISubscriber
    {
        // Metoders Attributter
        Beregninger beregninger;
        Kalibrer kalibrer;
    }
}

```

Figur 4.8: Monitor klassens arvehierarki

4.1.5 Metodebeskrivelser

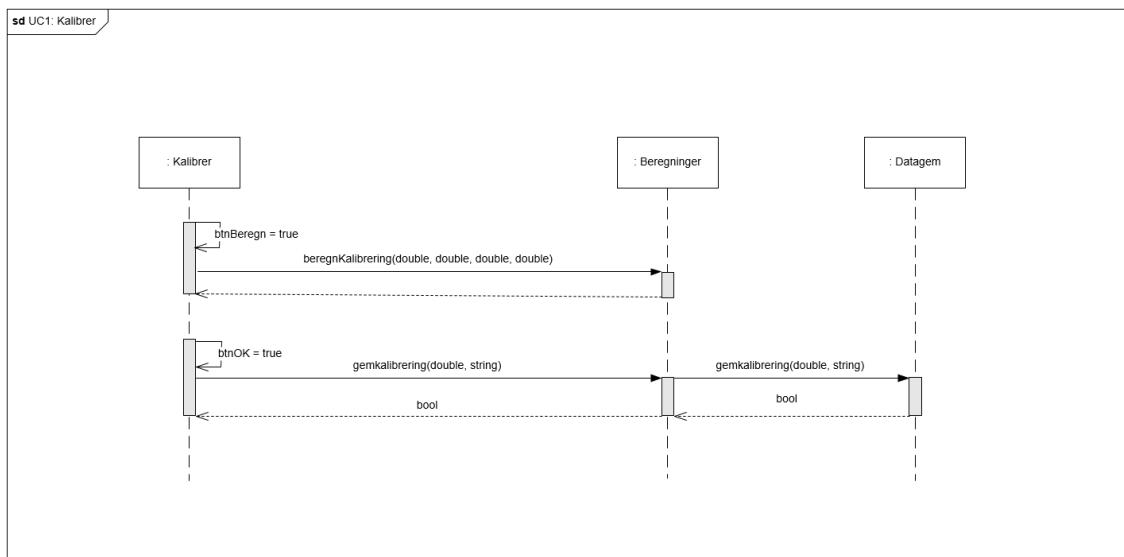
Use casene er beskrevet vha. sekvensdiagrammer, samt korte beskrivelser herunder. Dette gøres for at se hvilke klasser og metoder i softwaren der bruges i de forskellige use cases. Sekvensdiagrammerne overskueliggør klassernes involvering i de forskellige use cases, og beskriver kommunikationen herimellem.

Aktivitetsdiagrammer beskriver udførelsen, af enkelte metoder, step for step. Dette giver et fuldkomment 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.

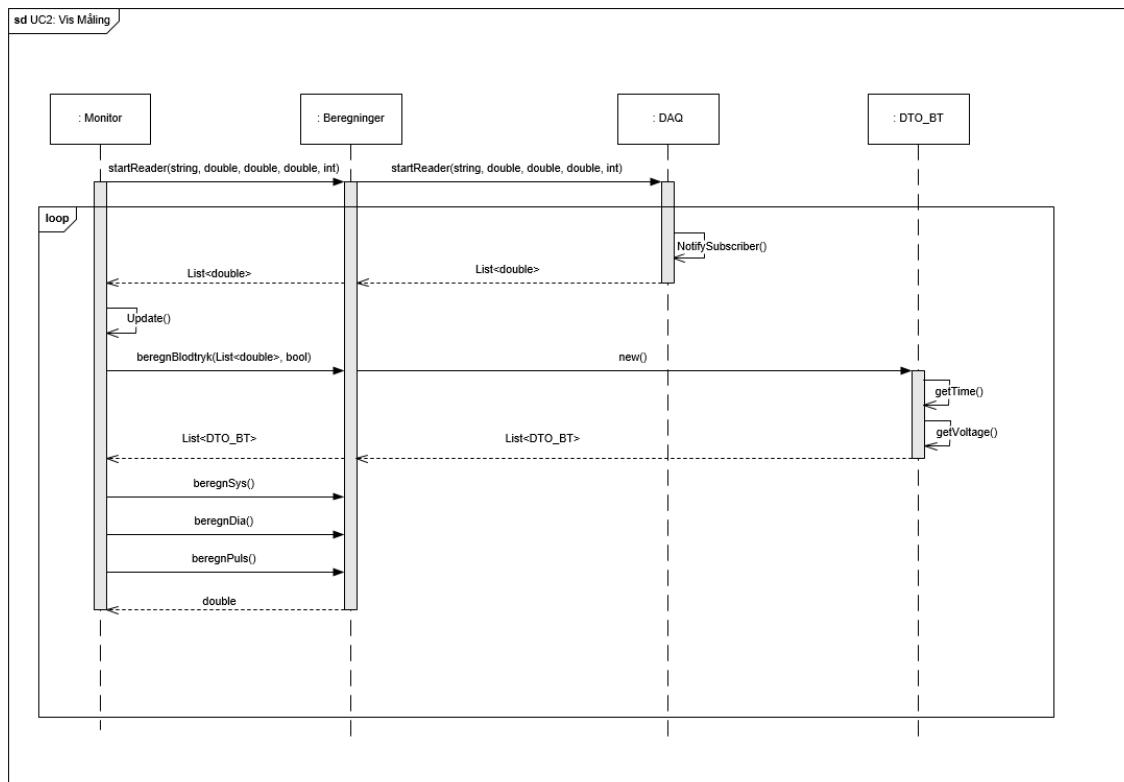


Figur 4.9: Sekvensdiagram for Use Case 1

Use Case 2: Vis måling

Når Monitor-vinduet åbnes ved opstart af program afvikles metoden 'startreader' i DAQ. 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 4.9 beregner pulsen, når der er mellem 3000 og 10000 samples til rådighed - altså når den fysiske DAQ har målt mellem 3 og 10 sekunder.

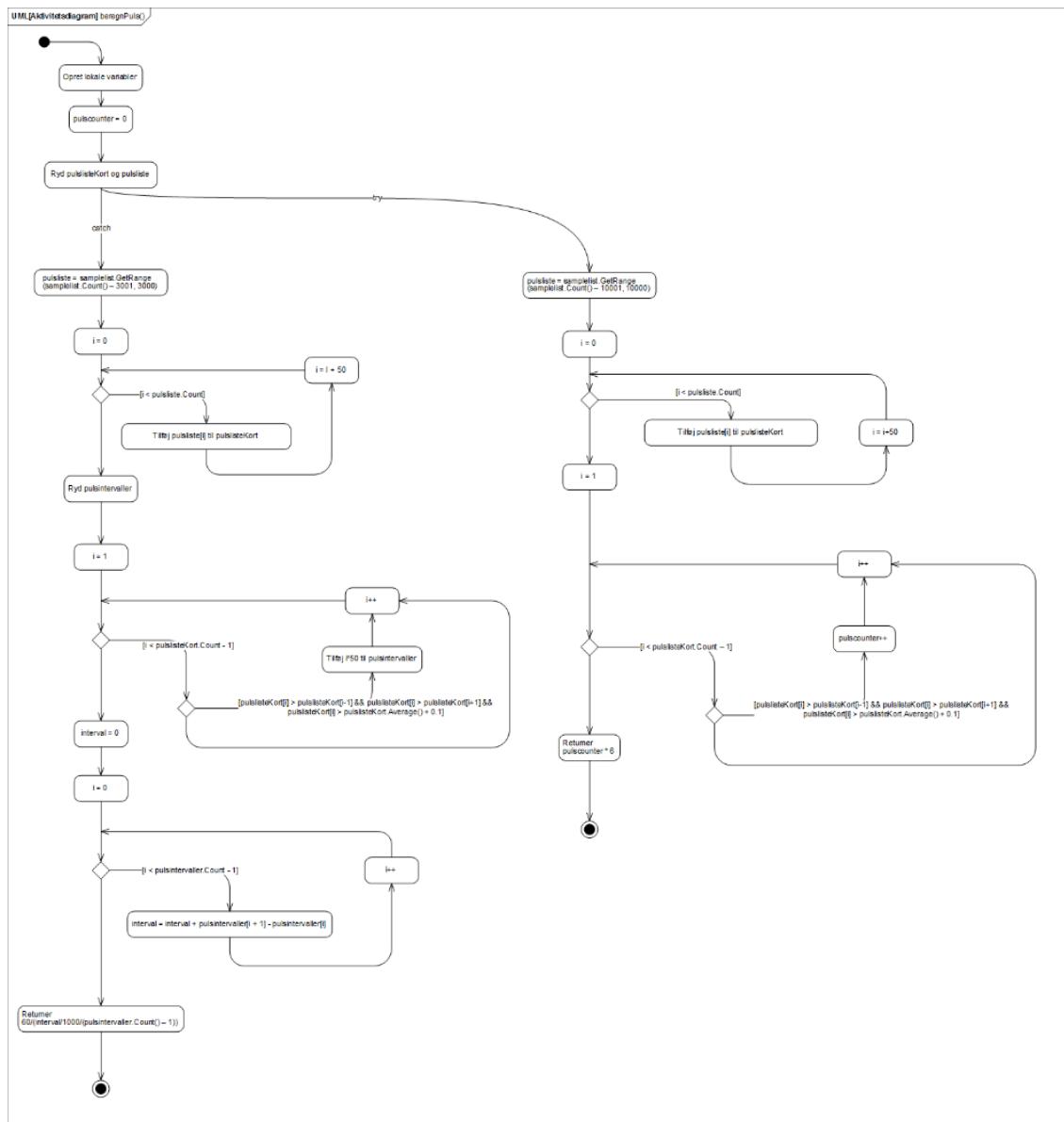
Metodikken består i at lokalisere toppunkter i signalet for at udregne det gennemsnitlige interval herimellem. Herudfra kan pulsen estimeres.

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.

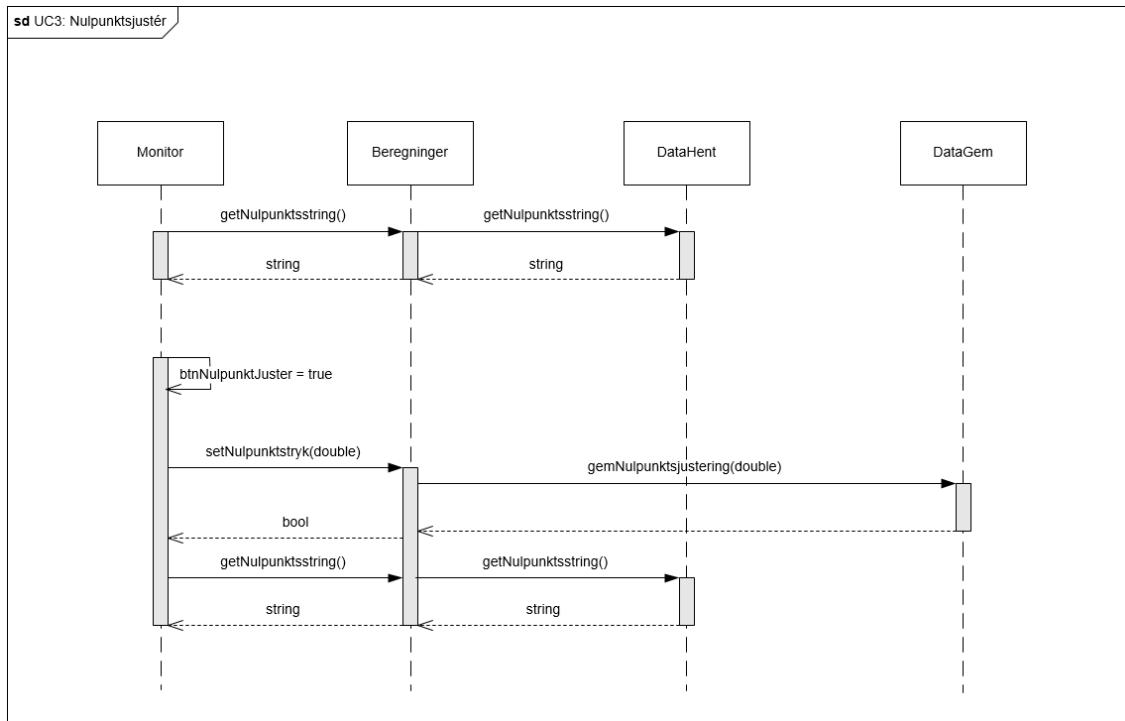


Figur 4.11: Aktivitetsdiagram over beregnpuls

Use Case 3: Nulpunktjuster

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

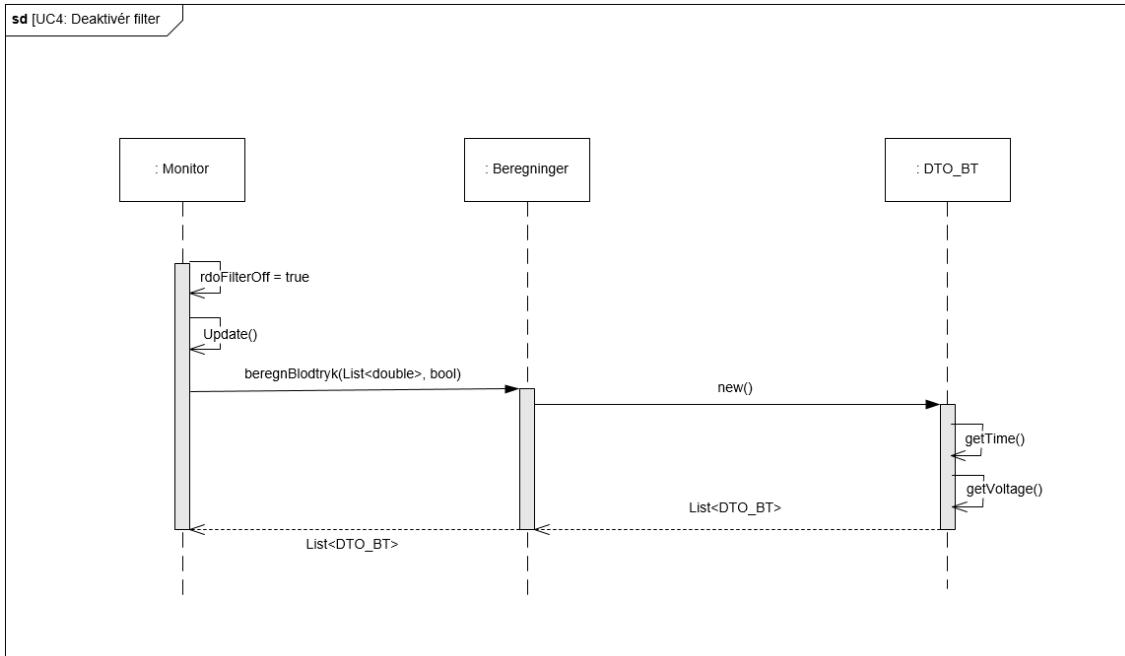
Forsker trykker på knappen 'Nulpunktsjuster', og Beregninger udfører en nulpunktsjustering, gemmer data i Datagem og returnerer en ny string der beskriver seneste (denne) nulpunktsjustering.



Figur 4.12: Sekvensdiagram for Use Case 3

Use Case 4: Deaktivér filter

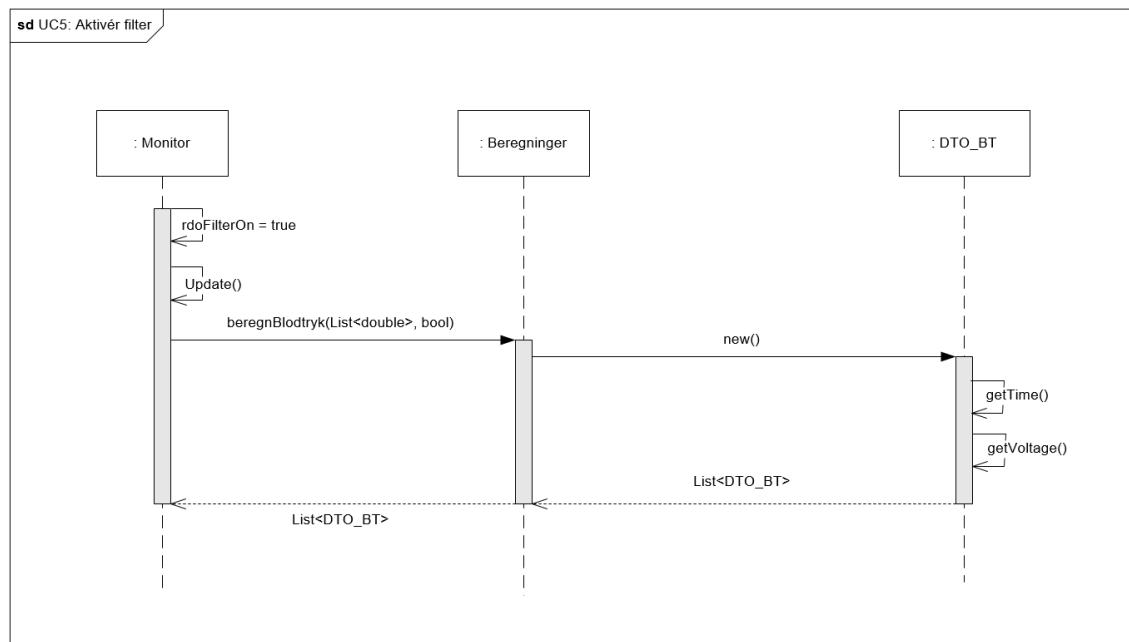
Forsker sætter radobutton for filter til "off". Bool "filteron" sættes til false, og sendes med metoden, der udregner blodtryk 'beregnBlodtryk', i beregninger. Den returnerede liste af DTO'er, som udskrives i Monitor, er ufiltrerede.



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 et gennemsnitsværdi af de foregående 10 værdier. Forsker sætter radiobutton for filter til "on". Bool "filterOn" sættes til true, og sendes med metoden, der udregner blodtryk 'beregnBlodtryk', i beregninger. Den returnerede liste af DTO'er, som udskrives i Monitor, er filtrerede.



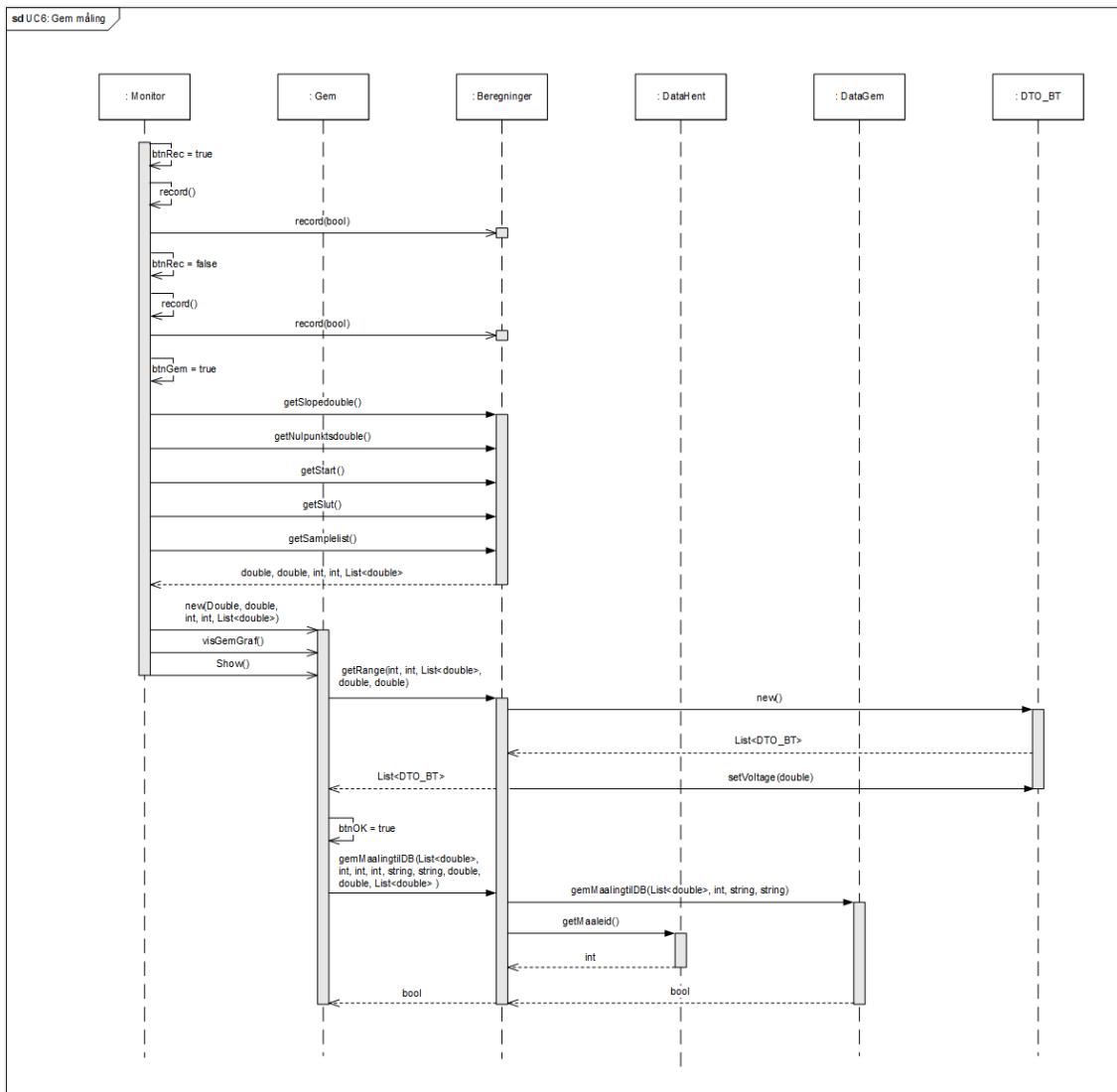
Figur 4.14: Sekvensdiagram for Use Case 5

Use Case 6: Gem måling

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

Det optagne 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, samt metadata gemmes.



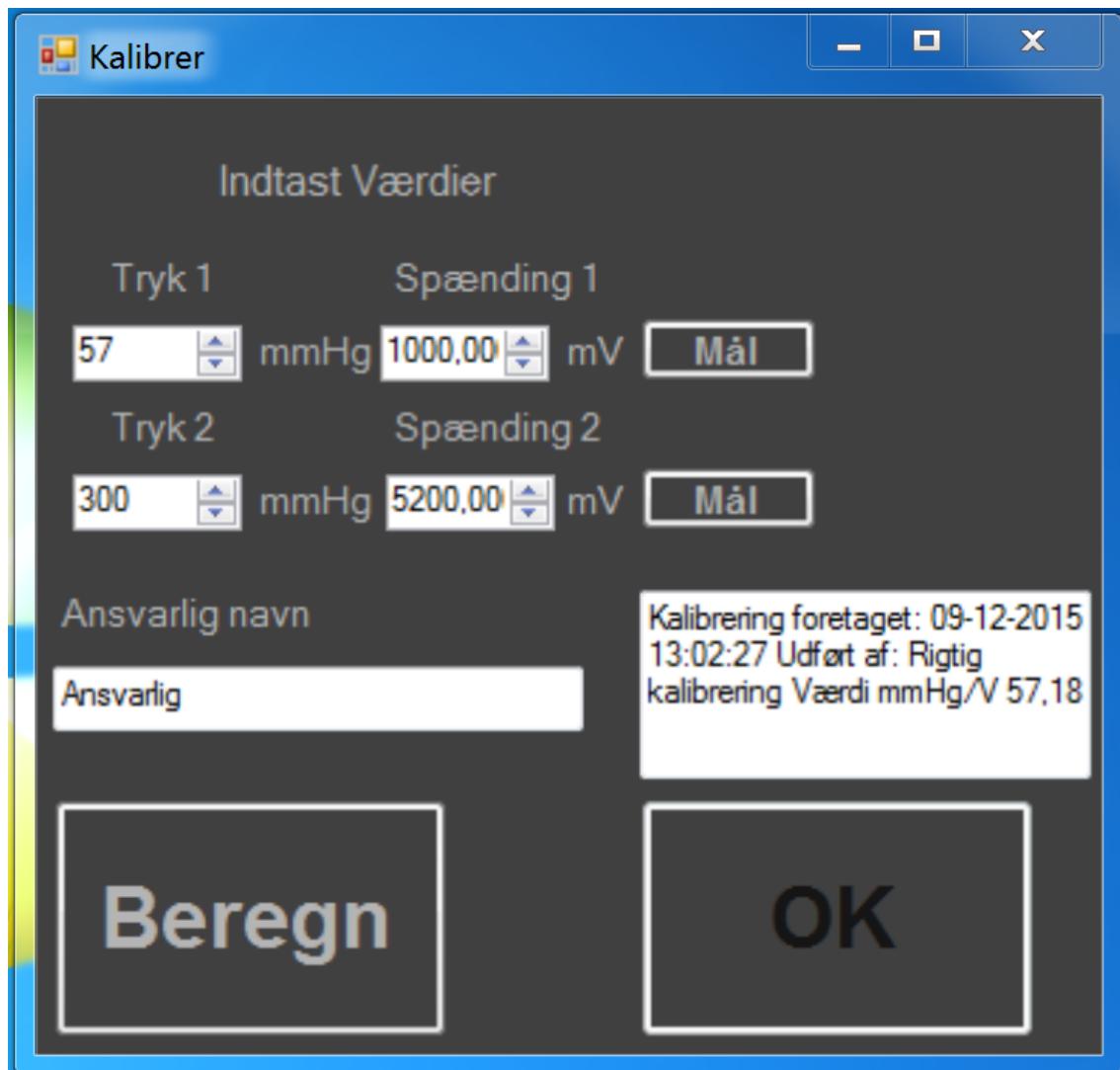
Figur 4.15: Sekvensdiagram for Use Case 6

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.

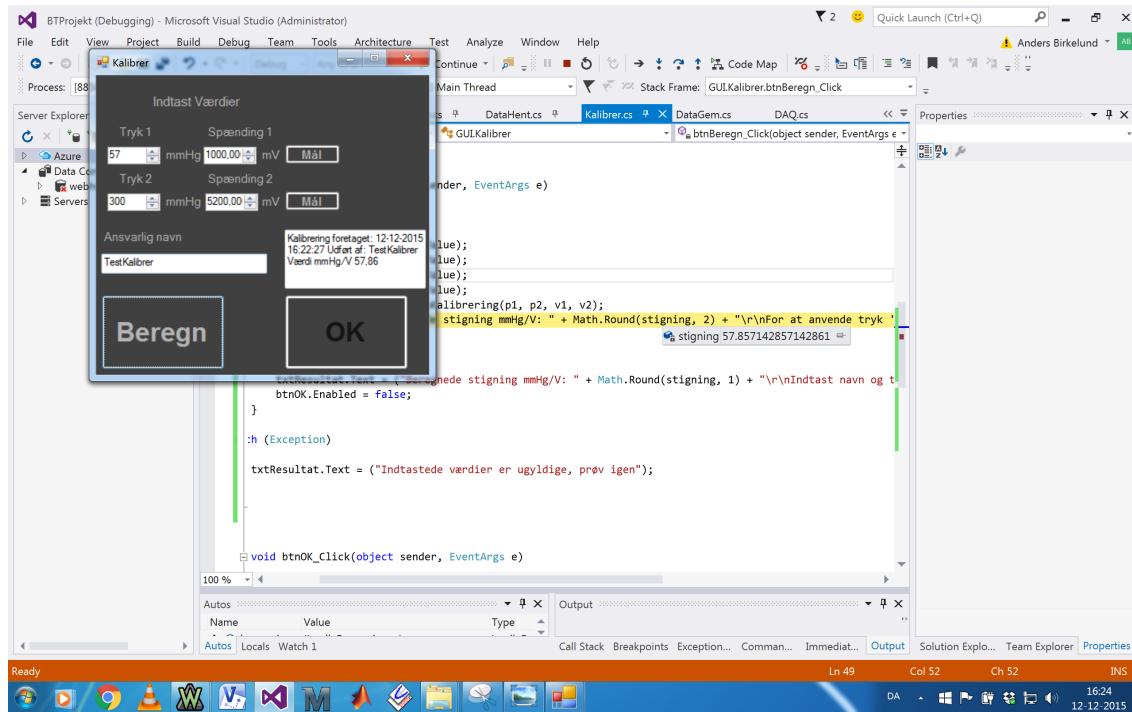
4.2.1 Use Case 1: Kalibrer

Forsker indtaster målte kalibreringsdata. Kalibreringsdata for seneste kalibrering vises.



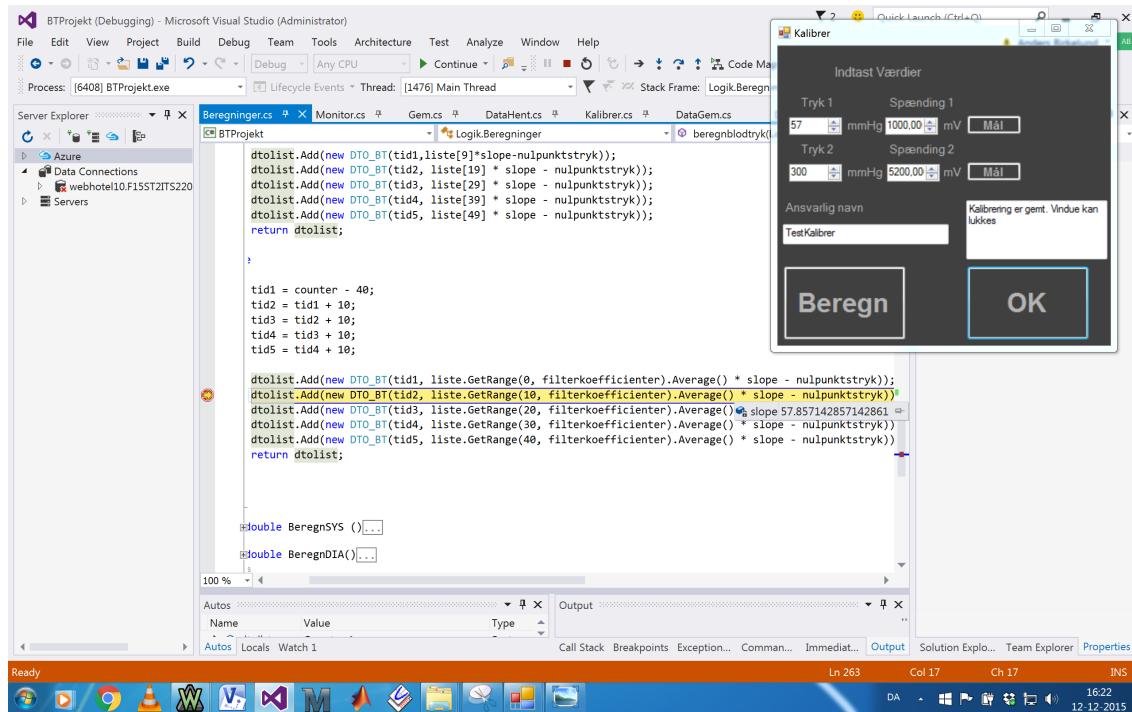
Figur 4.16: Kalibreringsdata indtastet

De målte værdier beregnes.



Figur 4.17: Kalibreringskonstant udregnes

Kalibreringskonstanten implementeres ved udskrivning på graf.



Figur 4.18: Kalibreringsdata anvendes

Det kontrolleres, at kalibreringsfilen er opdateret

	Name	Date modified	Type	Size
	Kalibreringsdata	12-12-2015 16:22	Text Document	8 KB

Content of the file "Kalibreringsdata":

```

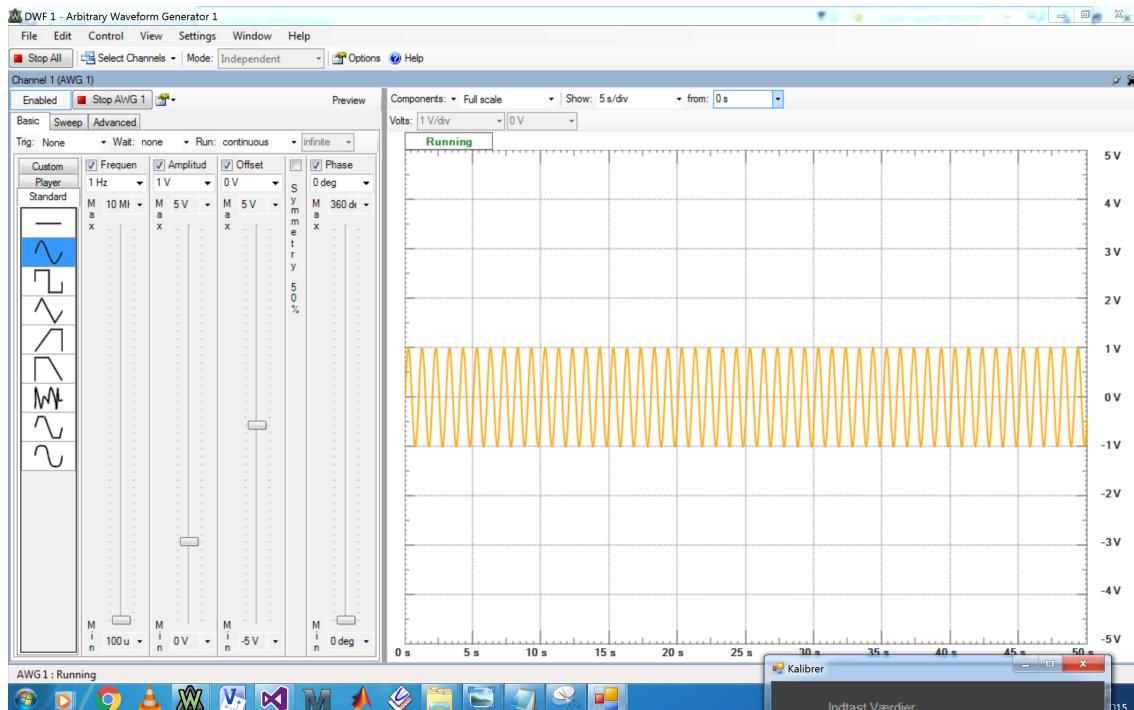
Kalibrering foretaget: 08-12-2015 12:42:53 Udført af: sdffffsd Værdi mmHg/V; 60
Kalibrering foretaget: 08-12-2015 12:43:27 Udført af: sdffffsd Værdi mmHg/V; 57,6923076923077
Kalibrering foretaget: 08-12-2015 12:43:36 Udført af: sdffffsd Værdi mmHg/V; -78,9473684210526
Kalibrering foretaget: 08-12-2015 12:43:55 Udført af: sdffffsd Værdi mmHg/V; 60
Kalibrering foretaget: 09-12-2015 08:34:55 Udført af: sdffffds Værdi mmHg/V; 60
Kalibrering foretaget: 09-12-2015 09:19:08 Udført af: Gruppen Værdi mmHg/V; 57,1823415788522
Kalibrering foretaget: 09-12-2015 11:20:35 Udført af: Jim Bob Værdi mmHg/V; 60
Kalibrering foretaget: 09-12-2015 11:31:03 Udført af: fht Værdi mmHg/V; 56,6037735849057
Kalibrering foretaget: 09-12-2015 11:59:58 Udført af: jgjgh Værdi mmHg/V; 100
Kalibrering foretaget: 09-12-2015 12:00:53 Udført af: kjhgkjhkjh Værdi mmHg/V; 57,6923076923077
Kalibrering foretaget: 09-12-2015 12:18:43 Udført af: Dr Anders Værdi mmHg/V; 57,1823415788522
Kalibrering foretaget: 09-12-2015 13:02:27 Udført af: Rigtig kalibrering Værdi mmHg/V; 57,1823415788522
Kalibrering foretaget: 12-12-2015 16:22:27 Udført af: TestKalibrer Værdi mmHg/V; 57,8571428571429

```

Figur 4.19: Kalibreringsdata er gemt til fil

4.2.2 Use Case 2 Vis måling

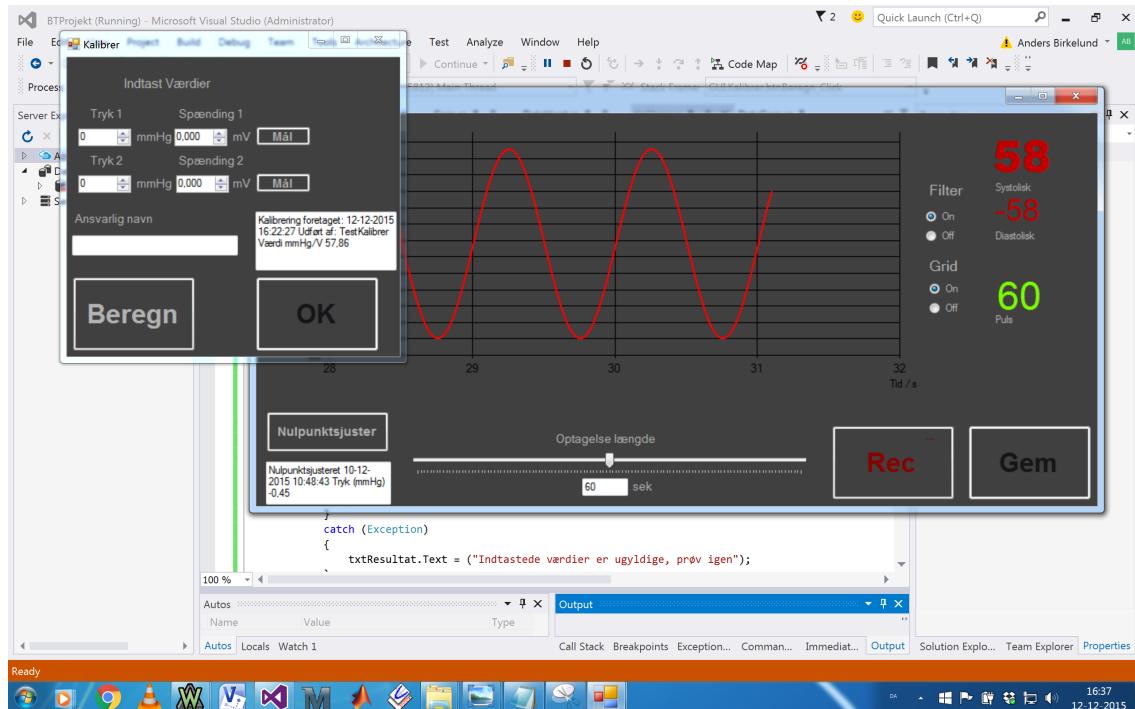
Et inputsignal på 1 Hz, 1 V amplitude og 0 V offset bruges til input på den fysiske DAQ.



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).

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.

```
public Monitor()
{
    InitializeComponent();
    filteron = new bool();
    beregninger = new Beregninger();
    kalibrer = new Kalibrer(beregninger);
    kalibrer.Show();
    kalibrer.TopMost = true;
    beregning.AttachSubscriber(this);
    startreader("Dev1/ai0", -5, 5, 1000, 50);

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

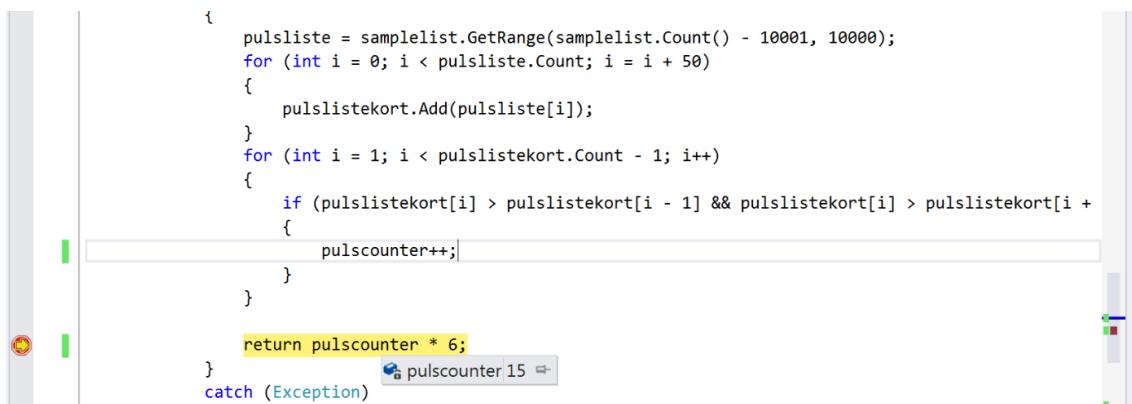
    //GRAPH VISUALS

    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.



```

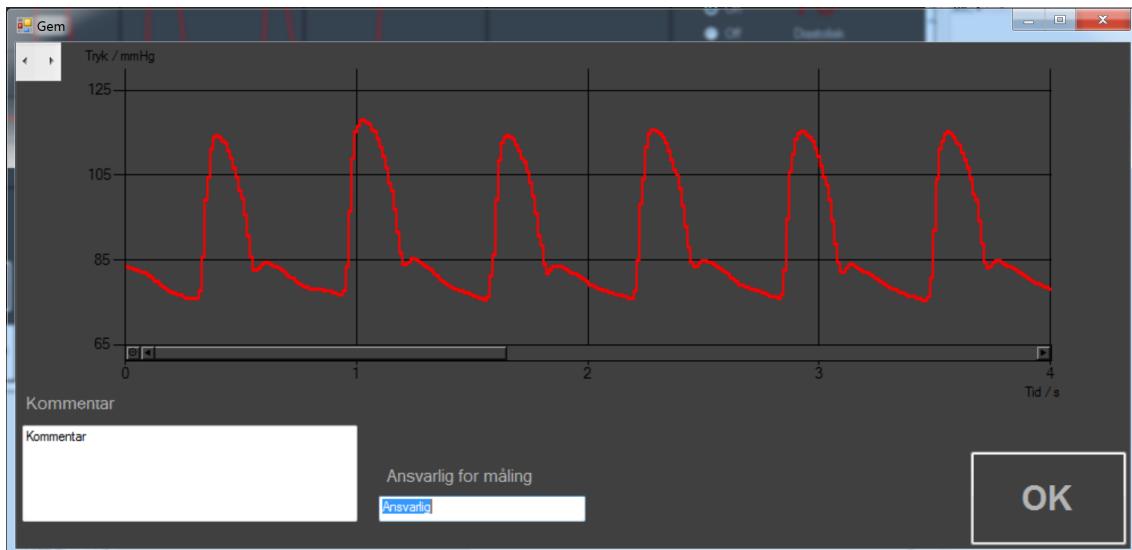
    pulsliste = samplelist.GetRange(samplelist.Count() - 10001, 10000);
    for (int i = 0; i < pulsliste.Count; i = i + 50)
    {
        pulslistekort.Add(pulsliste[i]);
    }
    for (int i = 1; i < pulslistekort.Count - 1; i++)
    {
        if (pulslistekort[i] > pulslistekort[i - 1] && pulslistekort[i]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

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

```

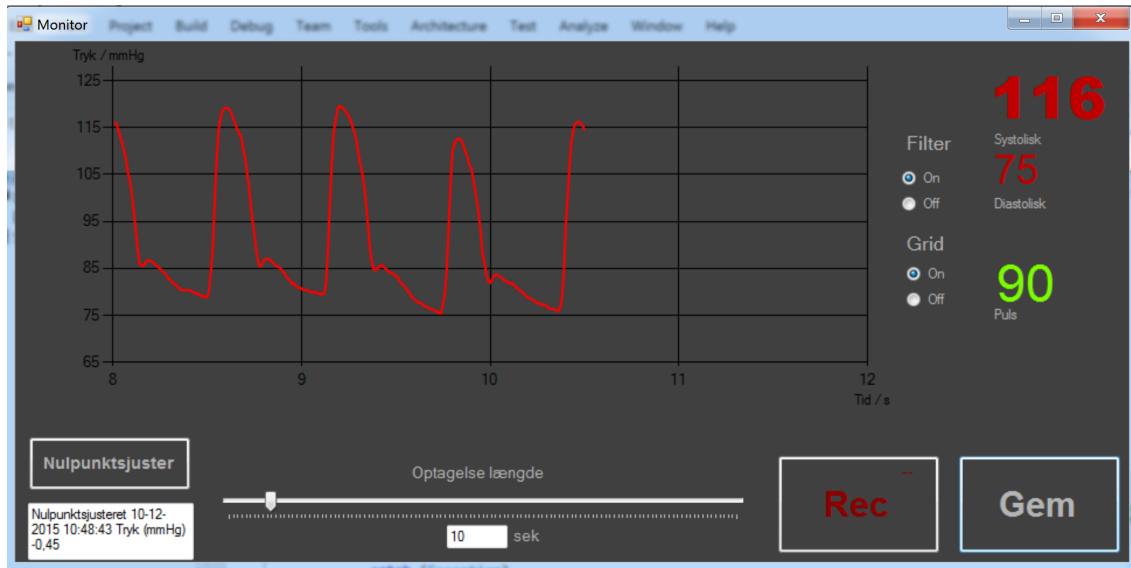
Figur 4.23: Puls efter 10 sekunders måling

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



Figur 4.24: Puls efter 10 sekunders måling

Det talte antal pulsslag holdes op mod dem, der er talt af metoden beregnpuls.

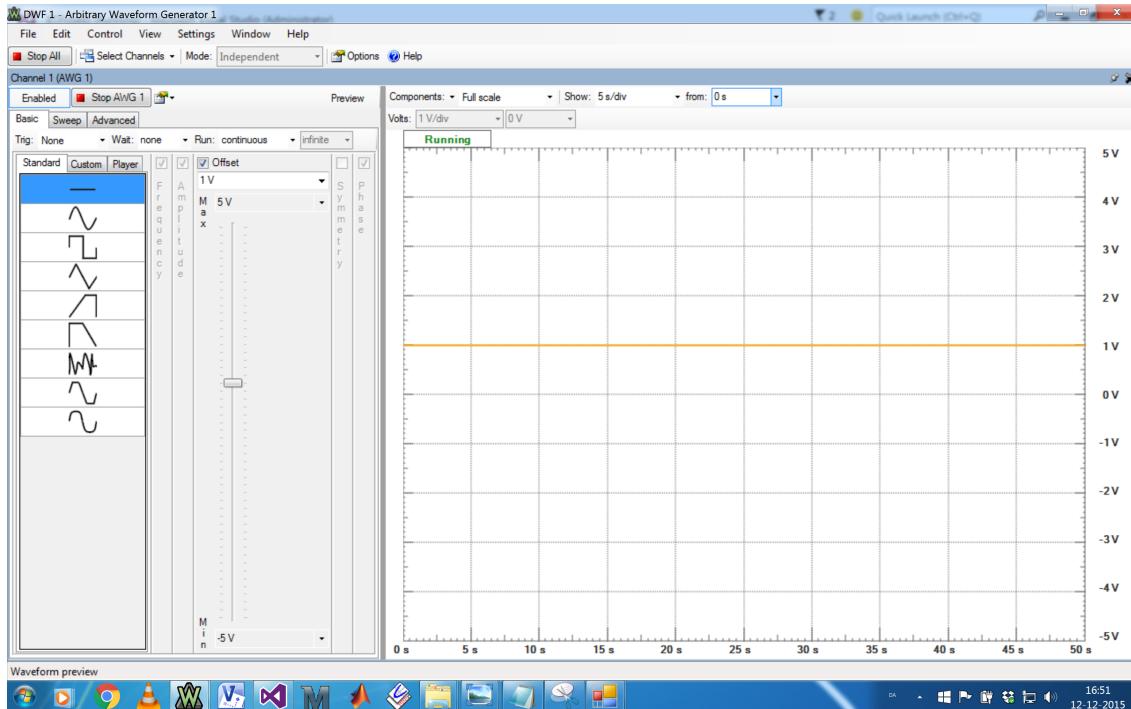


Figur 4.25: Puls efter 10 sekunders måling

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

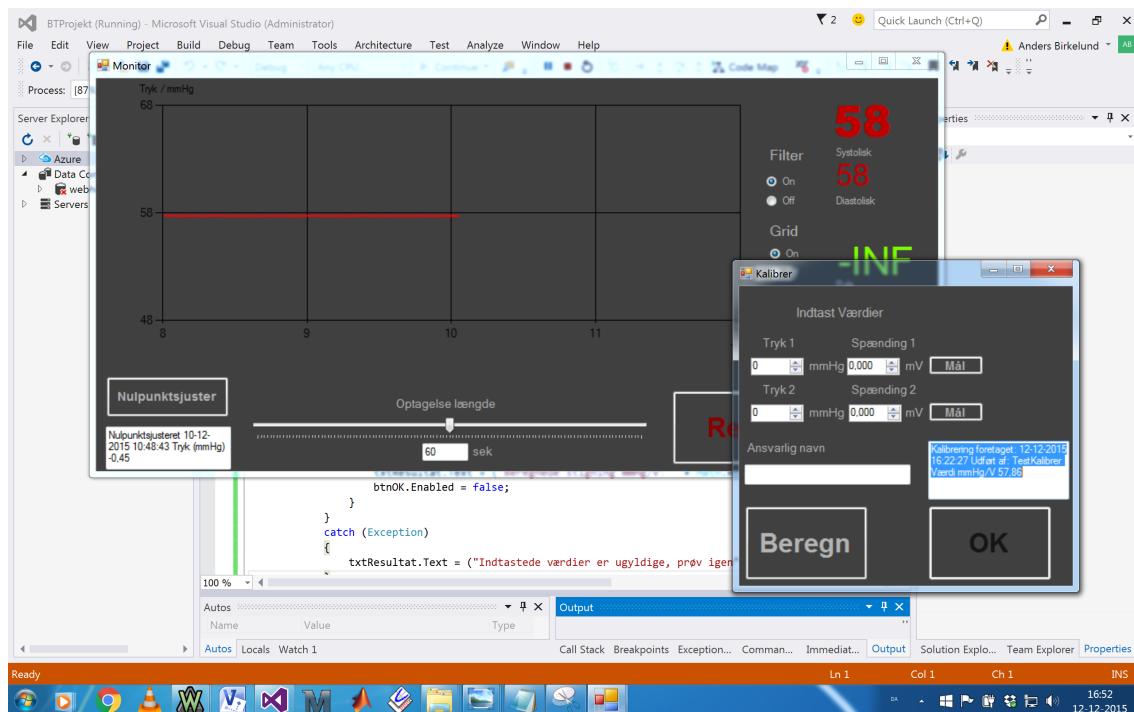
4.2.3 Use Case 3: Nulpunktsjuster

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



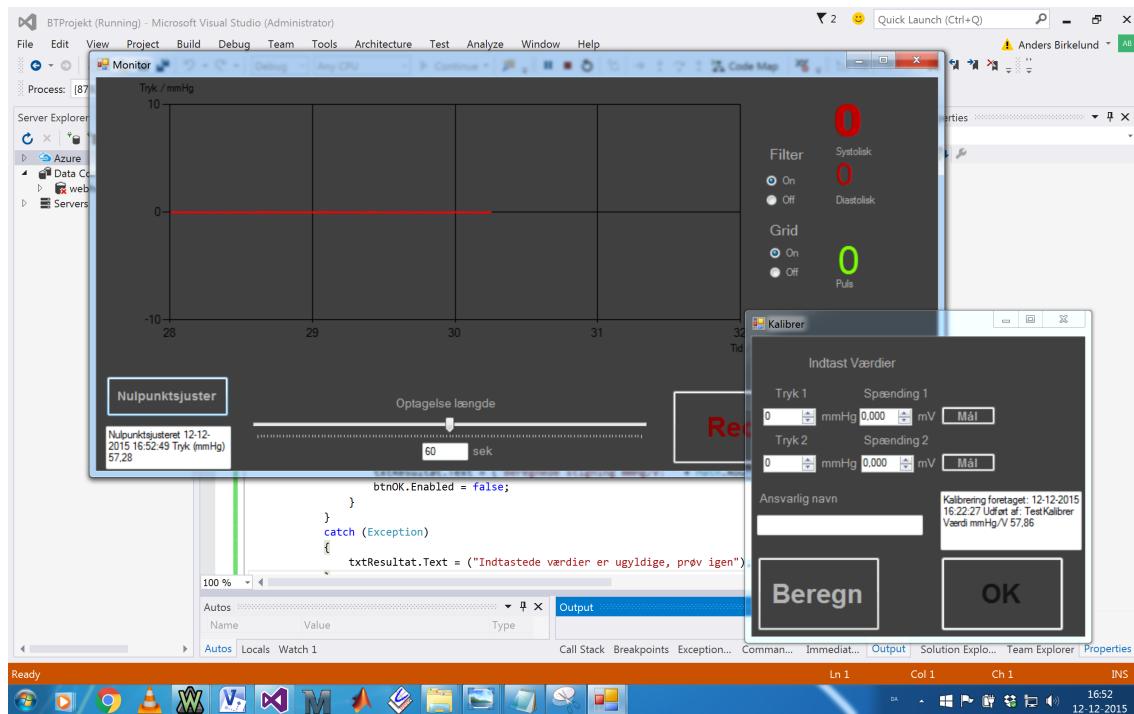
Figur 4.26: Inputsignal til DAQ

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



Figur 4.27: Inputsignal til DAQ

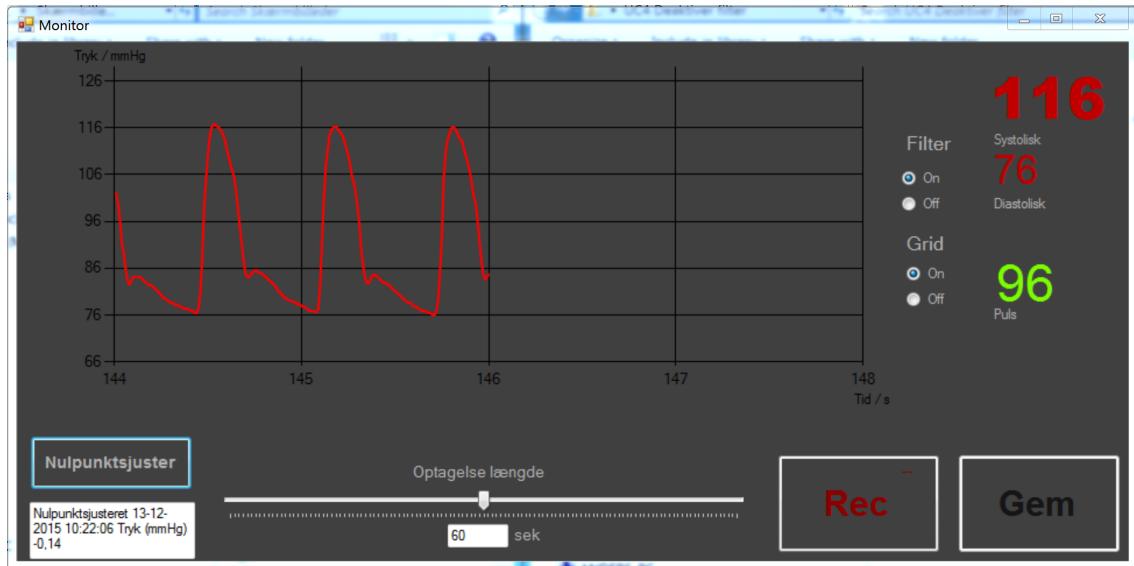
Der trykkes på knappen "Nulpunktsjuster", og grafens værdi falder til 0.



Figur 4.28: Inputsignal til DAQ

4.2.4 Use Case 4: Deaktiver filter

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



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.

```

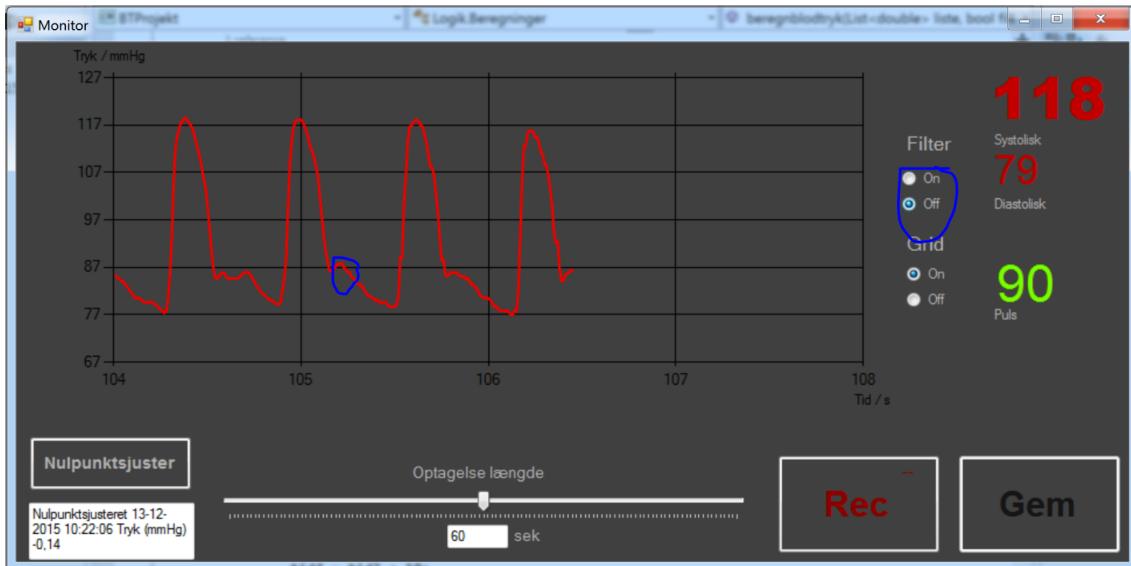
public List<DTO_BT> beregnBlodtryk(List<double> liste, bool filterOn) // returnerer EN DTO tra
{
    counter = counter + 50;
    dtolist.Clear();
    if (filterOn == false)
    {
        filterOn true
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 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: Beregningeklassen opretter filtrerede DTO'er

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



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

Beregninger klassen beregner med metoden uden filter, da "filteron= true".

```

public List<DTO_BT> beregnblodtryk(List<double> liste, bool filterOn) // returnerer EEN DTO fra
{
    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
    {
        filterOn true
        tid1 = counter - 40;
        tid2 = tid1 + 10;
        tid3 = tid2 + 10;
        tid4 = tid3 + 10;
        tid5 = tid4 + 10;

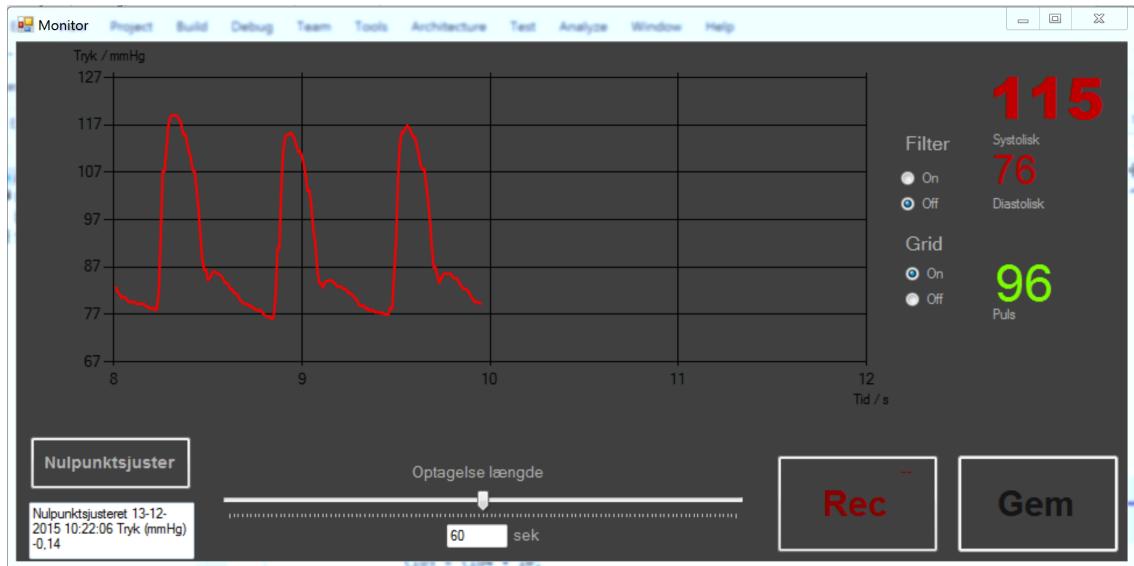
        dtolist.Add(new DTO_BT(tid1, liste.GetRange(0, filterkoefficienter).Average() * slope -
        dtolist.Add(new DTO_BT(tid2, liste.GetRange(10, filterkoefficienter).Average() * slope -
    }
}

```

Figur 4.32: Beregningeklassen opretter ufiltrerede DTO'er

4.2.5 Use Case 5: Aktiver filter

Filteret er ikke aktiveret, og det ufiltrerede signal udskrives.



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.

```

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

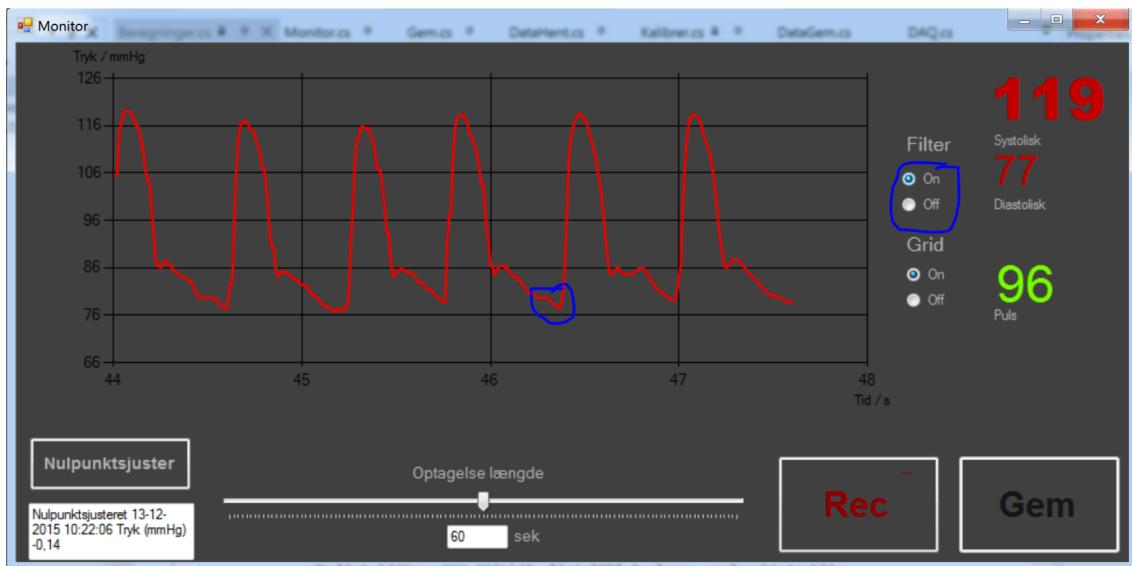
        dtolist.Add(new DTO_BT(tid1, liste[9]*slope-nulpunktstryk));
        dtolist.Add(new DTO_BT(tid2, liste[19] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid3, liste[29] * slope - nulpunktstryk));
        dtolist.Add(new DTO_BT(tid4, liste[39] * slope - nulpunktstryk));
        dtolist.Add(new 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 4.34: Beregningeklassen opretter ufiltrerede

Der trykkes på "Filter On" og det ses at det udskrevne signal, filtreres.



Figur 4.35: Ufiltreret signal udskrives i graf

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

```

Beregninger.cs X Monitor.cs Gem.cs DataHent.cs Kalibrer.cs DataGem.cs DAQ.cs ISubscriber.cs
BTProjekt Logik.Beregninger beregnBlodtryk(List<double> liste, bool filterkoefficienter)
{
    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;

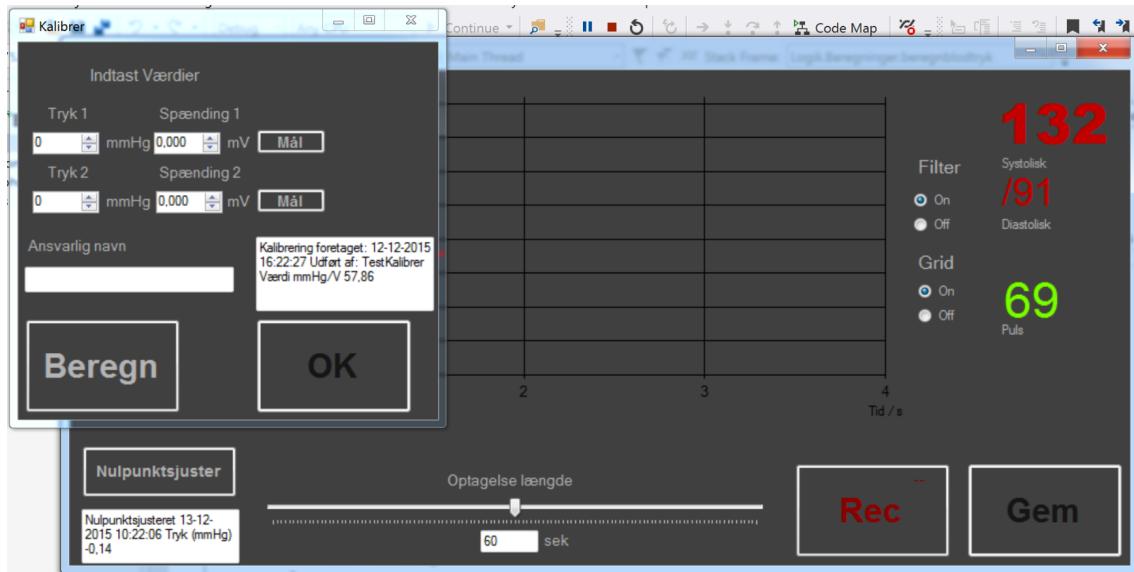
    dtolist.Add(new DTO_BT(tid1, liste.GetRange(0, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid2, liste.GetRange(10, filterkoefficienter).Average() * slope - nulpunktstryk));
    dtolist.Add(new DTO_BT(tid3, liste.GetRange(20, filterkoefficienter).Average() * slope - 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

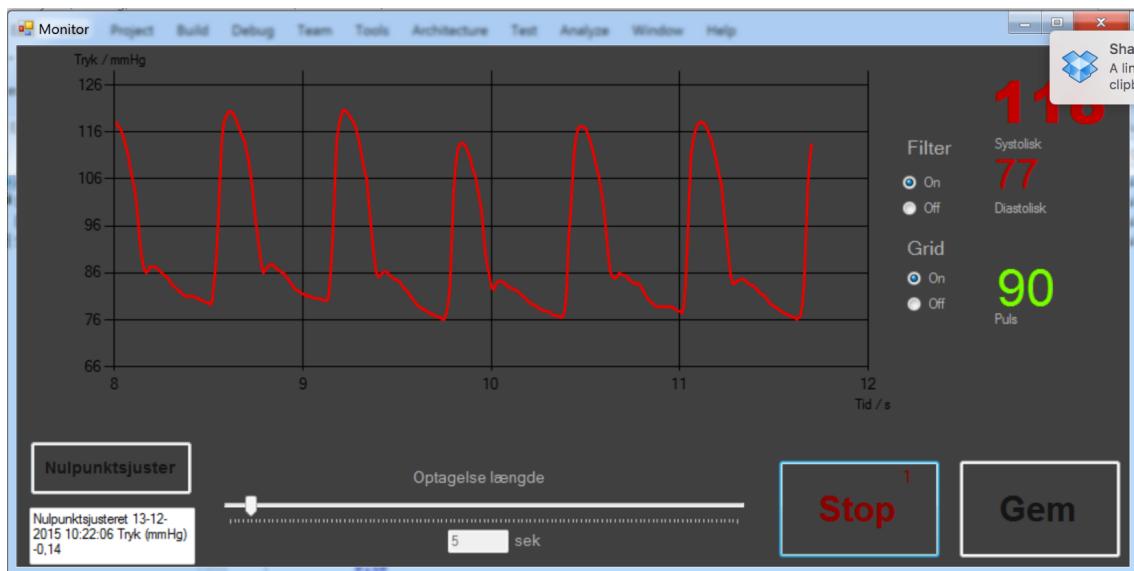
4.2.6 Use Case 6: Gem måling

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



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

Forsker vælger målelængde til fem sekunder, og trykker på "Rec".



Figur 4.38: Monitor-vindue efter tryk på "Rec"-knap

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

```

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 igen automatisk. Denne gang sættes slutværdien for målingen. Det observeres om differencen mellem startværdi og slutværdi er lig 5 sekunder dvs. 5000 samples.

```

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.

```

private void rdoFilterOff_CheckedChanged(object sender, EventArgs e)...
private void rdoFilterOn_CheckedChanged(object sender, EventArgs e)...
public void Update(List<double> liste)...
private void btnNulpunktuster_Click(object sender, EventArgs e)...
private void btnGem_Click(object sender, EventArgs e)
{
    äbnGem();
}

private void äbnGem()
{
    gem = new Gem(beregninger, beregning.getslopedouble(), beregning.getnulpunktdouble(), beregning.getStart(), beregning.getSlut(), beregning.getsamplelist());
    gem.visGemGraf();
    gem.Show();
}

```

Figur 4.41: Oprettelse af nyt Gem-vindue

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

The screenshot shows the Visual Studio debugger interface. The code editor displays the `Gem` class with its constructor and `Dispose` method. The `Locals` window below shows the current values of local variables:

Name	Type	Value
this	GUI.Gen	{GUI.Gem}
beregn	Logik.Beregninger	{Logik.Beregninger}
slope_	double	57.8571428571429
nulpunkt_	double	-0.136806404883507
startgem_	int	7000
slutgem_	int	12000
samplelist_	System.(List<double>)	Count = 12050

Figur 4.42: Gem-vinduets værdier, efter oprettelse

Det kontrolleres, om længden af "samplearray" i metoden `gemmaalingtilDB()` stemmer overens med de valgte fem sekunder dvs. 5000 samples.

The screenshot shows the `DataGem.cs` file with the `gemmaalingtilDB` method highlighted. The code uses Entity Framework to insert data into a database:

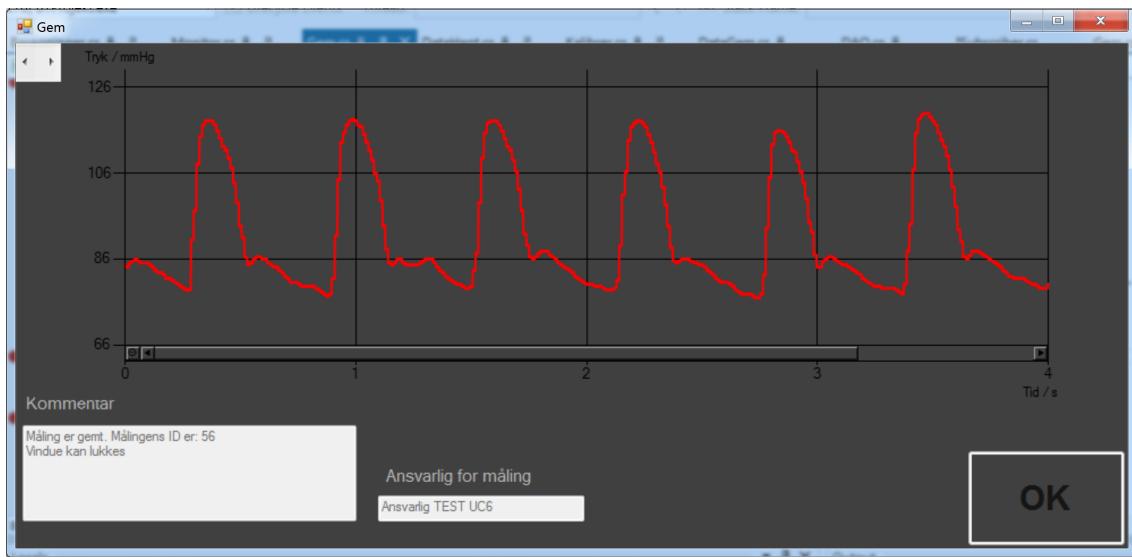
```

public void gemtilFil(List<double> backupliste) // Gemmer måling til .txt fil...
public bool gemmaalingtilDB(List<double> samplelist, int maaled, string ansvarlig, string kommentar)
{
    try
    {
        samplearray = samplelist.ToArray();
        conn = new SqlConnection("Data Source=DESKTOP-10.IHA.DK;Initial Catalog=" + db + ";Persist Security Info=True;User ID=" + db + ";Password=" + db + "");
        // Linq - henter alle verdier 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(); // Åbner forbindelse til SQL forbindelse som har navnet conn
    }
}

```

Figur 4.43: Array af blodtryksværdier, der gemmes til Database

Målingens ID noteres i Gem-vinduet.



Figur 4.44: Gem-vinduet efter gemt måling

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

The figure shows a screenshot of a Microsoft SQL Server Management Studio (SSMS) window. The title bar indicates the connection is to 'db_owner.BTmaaling [Data]'. The main area displays a table named 'maaleid' with columns 'maaleid', 'maaling', 'tid', 'ansvarlig', and 'kommentar'. The table contains five rows of data. At the bottom of the window, there is a status bar showing 'Connection Ready' and the session details 'webhotel10.ihha.dk | Anders-PC\Anders | F15ST2ITS2201404118'.

maaleid	maaling	tid	ansvarlig	kommentar
53	0x4EE14B993B905A40...	09-12-2015 16:13:57	Ansvarlig	Kommentar
54	0x41E68F3AA43635340...	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
3.0	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 stigningenstallet for, hvor mange mmHg/V udskrives i Kalibrerings-vinduet	Der er indtastet kalibreringsdata i de forskellige tekstdokumenter og stigningenstallet 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	✓
<i>Undtagelse</i>			

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	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 "Nulpunktsjustér"-knap	System nulpunktsjusterer	System nulpunktsjusterer	(✓)

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	✓

Undtagelser

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-vinduet	Det fremgår af Gem-vinduet, at målingen er gempt og der tjekkes i Databasen om målingen er gempt	Det fremgår af Gem-vinduet, at målingen er gempt og der tjekkes i Databasen om målingen er gempt	✓

Undtagelser

3.a	Stop optagelse af måling inden optagelses-længden 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 0-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 der indtastes gyldige værdier i Gem-vinduet og trykkes på "OK"-knap	System gemmer data i en Database og udskriver stempel for gemt data i Gem-vinduet	System gemmer data i en Database og udskriver stempel for gemt data i Gem-vinduet	✓
<i>Usability</i>				
Kalibreringsvinduet skal indeholde tekster til indtastning af kalibreringsdata	Der ses i Kalibreringsvinduet, om der er tekstbokse til 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

Der er en ”Beregn”-knap



Kalibrerings-vinduet skal indeholde en ”OK”-knap

Der ses i Kalibrerings-vinduet, om der er en ”OK”-knap

Der er en ”OK”-knap

Der er en ”OK”-knap



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

Der er et tidsstempel for seneste kalibrering



Monitor-vinduet skal indeholde en ”Gem”-knap

Der ses i Monitor-vinduet, om der er en ”Gem”-knap

Der er en ”Gem”-knap

Der er en ”Gem”-knap



Monitor-vinduet skal indeholde en ”Nulpunktsjustér”-knap

Der ses i Monitor-vinduet, om der er en ”Nulpunktsjustér”-knap

Der er en ”Nulpunktsjustér”-knap

Der er en ”Nulpunktsjustér”-knap



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

Der er et tidsstempel for seneste nulpunktsjustering



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 tekstdokse til indtastning af oplysninger om målingen	Der ses i Gem-vinduet, om der er tekstdokse til indtastning af oplysninger om målingen	Der er tekstdokse til indtastning af oplysninger om målingen i Gem-vinduet	Der er tekstdokse 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 pulsværdier på Monitor-vinduet fra 2 meters afstand med normalt syn

Der testes på 5 personer af forskellige aldre med en synspulsværdier 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



Reliability

Systemet skal have en effektiv MTBF på 99 timer og MTTR på 20 minutter

Performance

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 de 5 sekunder

Blodtrykssignalet vises indenfor de 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



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 kontinuerligt over et interval af 4 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	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	Blodtrykssignalets maksimum- og minimumsværdier vises	✓
<i>Supportability</i>				
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	✓
<i>Andet(+)</i>				
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 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 diastolisk blodtryk skal fremhæves øverst i højre hjørne ved størrelse end andre værdier i Monitor-vinduet (fx værdier på akserne)	Der ses i Monitor-vinduet, om det systoliske og det diastoliske blodtryk er fremhævet ved større skriftstørrelse end andre værdier i Monitor-vinduet	Det ses i Monitor-vinduet, at det systoliske og det diastoliske blodtryk er fremhævet ved større skriftstørrelse end andre værdier i Monitor-vinduet	Det ses i Monitor-vinduet, at det systoliske og det diastoliske blodtryk er fremhævet ved større skriftstørrelse end andre værdier i Monitor-vinduet	✓

Tabel 5.8: Accepttest af Ikke-funktionelle krav

