

# Projekt EKG

Projektrapport – E3PRJ3 3. semester

Gruppe 3 – Vejleder: Tore Skogberg

Udarbejdet af:

Kristian

Joakim

Martin

Mathias Casper Lynge Holding

Karsten

Rune

Jacob

Anton

## 1. Resume

Emnet for dette projekt er at udvikle et system til måling af EKG. Målingen af EKG-signalet skal vises grafisk på en pc.

En af de største udfordringer for EKG-måling er, at der skal måles små signaler i et meget støjfyldt miljø, og det har også været den vigtigste overvejelse omkring det overordnede systemdesign. For at imødegå denne udfordring, og undgå basale fejl, er der valgt en iterativ udviklingsmodel, med henblik på at afdække fundamentale problemstillinger tidligt i projektets forløb, hvilket skulle give gode muligheder for at ende med et implementeret velfungerende system.

Det implementerede system består grundlæggende af:

- Analog forstærkning og filtrering
- AD-konvertering
- Digital signalbehandling
- Grafisk visning af EKG-signal
- Visning af BPM

Systemet er blevet implementeret og testet og har levet op til de i kravspecifikationen angivne krav, hvilket vil sige, at grafen bliver tegnet pænt uden nævneværdig støj, og at BPM bliver vist korrekt.

## Abstract

The subject of this project is to develop a system for measuring ECG. The measurement of the ECG signal is to be displayed graphically on a PC.

One of the biggest challenges for ECG measurement is to measure small signals in a very noisy environment, and it has also been the most important consideration about the overall system design. To face this challenge and avoid basic errors, we have chosen an iterative development model, in order to uncover fundamental issues early in the progress of the project, which should make it possible to end up with a working implemented system.

The implemented system basically consists of:

- Analogue gain and filtering
- AD-converter
- Digital signal processing
- Graphic display of ECG signal
- Display of BPM

The system has been implemented and tested and has met the requirements of the specification requirements without significant deficiencies.

## Indholdsfortegnelse

<b>1. Resume</b>	<b>2</b>
<b>Indholdsfortegnelse</b>	<b>3</b>
<b>2. Indledning</b>	<b>4</b>
<b>3. Opgaveformulering</b>	<b>5</b>
<b>4. Systembeskrivelse</b>	<b>6</b>
<b>5. Krav</b>	<b>6</b>
<b>6. Projektafgrænsning</b>	<b>7</b>
<b>7. Projektbeskrivelse</b>	<b>8</b>
7.1. Projektgennemførelse	8
7.1.1. Ansvarsfordeling	10
7.2. Metoder	12
7.3. HW/SW Analyse	12
7.4. Systemarkitektur	14
7.5. HW/SW Design	15
7.5.1. Software Design	15
7.5.2. Hardware Design	19
7.6. HW/SW Implementering	21
7.6.1. Software implementering	21
7.6.2. Hardware implementering	30
7.7. Resultater	39
7.7.1. Accepttest	39
7.7.2. Sammenligning med professionelt optaget EKG	42
7.8. Opnåede erfaringer	44
7.8.1. Erfaring med udviklingsproces	44
7.8.2. Iterativ proces	44
7.8.3. Fumlebræt	44
7.8.4. Sampling af kontinuerte signaler	44
7.8.5. MCU-programmering	45
7.8.6. Flertrådet programmering og events	45
7.9. Udviklingsværktøjer	45
<b>8. Konklusion</b>	<b>46</b>
<b>9. Referencer</b>	<b>46</b>

## 2. Indledning

I dette projekt er formålet at konstruere et EKG-målingssystem (Elektrokardiogram). Systemet er beskrevet i større detalje i de følgende afsnit.

I arbejdet med dette system er der tilegnet og gjort brug af teknisk viden inden for C-programmering, objektorienteret C#-programmering, seriel kommunikation, filtrering og forstærkning af elektriske signaler, konvertering fra analog til digital signal, digital signalbehandling, galvanisk adskillelse, samt komponent-søgning og analyse. Viden om disse emner er primært tilegnet gennem semestrets kurser, men også igennem vejledere, samt egen opsøgning af viden.

I projektet er der lagt vægt på projektforsløbet, og dets faser. Der er i det tidlige forløb udarbejdet projektformulering og kravspecifikation/accepttest for at give et overblik over projektets udformning og funktioner. Derefter er systemarkitekturen udformet, hvori systemet er blevet nedbrudt i mindre blokke. Dette har givet mulighed for at arbejde isoleret med mindre dele. Systemet er blevet analyseret, designet og implementeret, hvorefter det er blevet samlet til det endelige produkt og er blevet testet. Der er sideløbende med selve projektdokumentationen produceret arbejdsdokumenter og mødereferater for at give overblik over forløbet i projektet, samt de aftaler der er indgået.

Der er i samarbejde med andre grupper blevet afholdt review-møder, i de indledende faser, for at give deltagerne en udefrakommende vinkel på projektudformningen.

Projektet er et tværfagligt projekt med forskellige hardware og software dele. Dette stiller krav til den indledende strukturering af projektet, da der skal arbejdes separat på forskellige elektronik- og software-dele. Det stiller også krav til samarbejde og kommunikation, for at systemerne i sidste ende skal kunne fungere samlet.

Projektets emne, "EKG-måler", er valgt, da det giver mulighed for arbejde med mange relevante emner (som beskrevet ovenfor). Projektet fokuserer hovedsageligt på elektronik- og softwareudvikling.

### Læsevejledning

I resten af rapporten benyttes følgende forkortelser:

SW: Software

HW: Hardware

GUI (Graphical User Interface): Brugergrenseflade

MCU: MikroController Unit. Den benyttede mikrocontroller er af mærket ATMEL mega32.

DSP: Digital Signal Processing

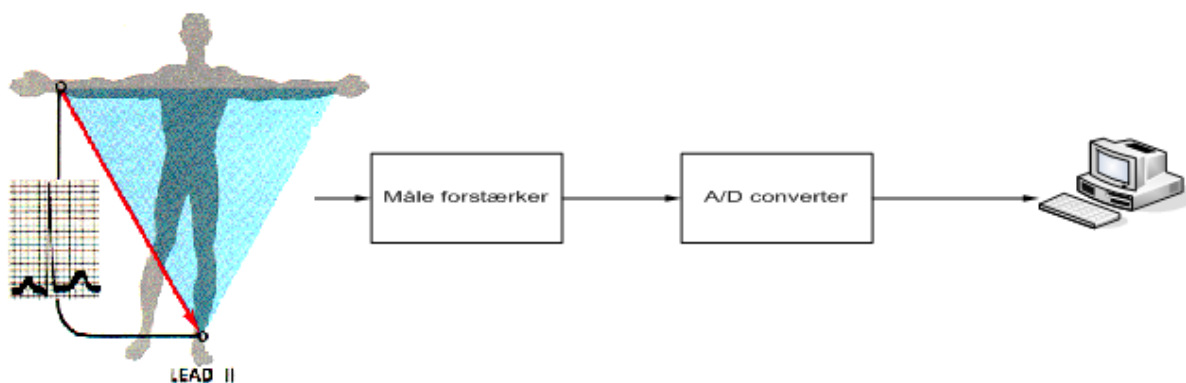
RTOS: Realtime Operating System

BPM: Beats Per Minute

ADC: Analog to Digital Converter

EKG: Elektrokardiogram

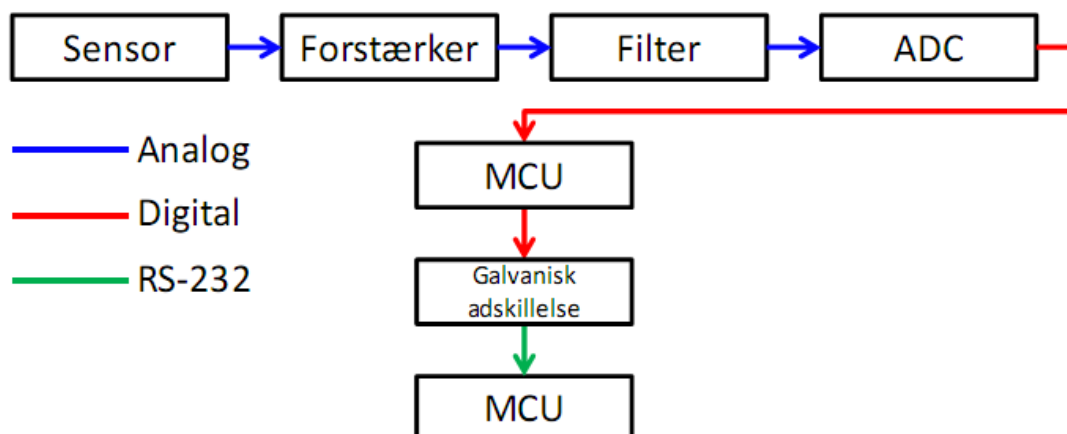
## 3. Opgaveformulering



Figur 1 - Blokdiagram over produkt

- Der skal konstrueres et udstyr, der kan måle elektrokardiogram som vist på ovenstående overordnede blokdiagram.
- Sensoren er standard elektroder (fx Ambu Blue Sensor Q), der giver et analogt signal. Måleforstærkeren skal kunne danne tilpasning mellem sensorens udgangssignal og A/D konverterens indgang(e). Der må ikke anvendes en måleforstærker integreret i en enkelt kreds.
- Valget af A/D konverter er en del af projektopgaven.
- De analoge signaler fra sensoren skal A/D konverteres og præsenteres digitalt på en monitor, enten ved hjælp af en mikroprocessor eller en PC.
- Ifm. databehandlingen skal det være muligt at gemme måledata. Måledata skal efterfølgende kunne præsenteres på brugerfladen og eventuelt sammenlignes med nye måledata.
- Software skal kunne betjenes af en bruger uden store IT-kompetencer (typisk skal udstyret anvendes af personer med en sundhedsmedicinsk uddannelse).
- Galvanisk adskillelse mellem "patient" og computer.
- Hvis kalibrering er nødvendig, skal denne kunne gemmes.

## 4. Systembeskrivelse



Figur 2 - Systemoversigt

Systemet kommer til at bestå af delene set på ovenstående diagram. Sensoren er en Ambu Blue Sensor R, som er placeret på patienten. Herefter kommer en forstærkerdel, der forstærker signalet et ønsket antal gange. Der følger så en filtreringsdel, der filtrerer uønskede frekvenser fra. Herefter konverteres det analoge signal til et digitalt, vha. en AD-konverter. Det digitale signal downsamples herefter i MCU-delen og sendes videre til pc'en gennem en RS232 serielport-forbindelse. På pc'en udføres der signalbehandling, som frasorterer støj og udregner patientens puls. Her findes den grafiske brugerflade der er brugerens adgang til systemet. Der skal på denne brugerflade vises et graf-vindue hvor EKG-signalet tegnes, et vindue hvor den målte puls vises (BPM), og følgende knapper: "Start", "Stop", "Vis gemt data".

## 5. Krav

Systemet skal designes til at måle et EKG-signal. Systemet skal betjenes gennem en grafisk brugerflade. Ved tryk på "Start"-knap skal brugeren kunne starte programmet, og ved tryk på "Stop"-knappen skal det stoppes. Brugeren skal også kunne aktivere "Vis gemt data"-knappen og derved se gemt data fra tidligere målinger. Målte data gemmes hvert 4. sekund. Den gemte data overskrives hver gang der gemmes data, så det er altid den nyeste gemte data der vises. Den grafiske repræsentation skal foregå i et vindue, som har 4 sekunder på x-aksen, og -5 til 5 mV på y-aksen. Vinduet skal også være inddelt i tern, der hver har 0,2 sekunder på x-aksen og 0,5 mV på y-aksen. Der stilles krav til målepræcision, som betyder at den målte værdi skal indtegnes med en maksimal afvigelse på 20 % eller  $\pm 100 \mu\text{V}$ . Signaler med en frekvens mellem 0,5 og 150 Hz skal kunne tegnes. Der skal også vises målt puls (BPM) på brugergrænsefladen. I intervallet 20-200 BPM skal målepræcisionen for denne være  $\pm 5 \%$ . Det er også et krav, at der af sikkerhedsgrunde skal være galvanisk adskillelse mellem patient og lysnettet.

## 6. Projektafgrænsning

I dette projekt er det valgt at fokusere mest på konceptet i at udvikle et apparat til EKG-måling, og det har ikke været et mål at frembringe et færdigt produkt. Dette skyldes, at det her er læringsprocessen, omkring tekniske løsninger, der har været i centrum. Samtidig har forventningen til EKG-målingen været, at det vil være svært at opnå en EKG-måling af god kvalitet. Der er derfor truffet følgende beslutninger:

- Det er valgt at lave et apparat der kun benytter 3 målesensorer, i stedet for 10 som er standard for professionelle EKG-målinger. Antallet 3 er valgt, da dette er et minimumskrav, for at kunne måle et acceptabelt EKG-signal.
- Der er ikke sat krav om, at det skal være muligt at identificere standardopdelingerne af et EKG-signal, P, Q, R, S og T.<sup>1</sup>

På grund af tidspres senere i projektet er følgende afgrænsninger også vedtaget:

- Det er valgt at anvende laboratoriets spændingsforsyning, som er CE-godkendt, fremfor et batteri.
- Det er ikke prioriteret at lave print til alle HW-dele.

Som det beskrives senere<sup>2</sup>, er det valgt at anvende en udviklingsmodel bestående af 2 iterationer. Det var oprindeligt planlagt, at der i 2. iteration skulle implementeres ekstra funktionaliteter som log, kalibrering og flere måleafledninger ud fra ekstra målesensorer. Det viste sig dog, at arbejdet med at opfylde kravene, stillet i forbindelse med 1. iteration, var omfattende nok til at udfylde hele projektforsløbet.

---

<sup>1</sup> Se Figur 19 på side 27

<sup>2</sup> 7.2 Metoder på side 11

## 7. Projektbeskrivelse

### 7.1. Projektgennemførelse

Oprindeligt var det tænkt at anvende en iterativ udviklingsmodel for projektet bestående af 2 dele. Første iteration skulle munde ud i et simpelt system, hvor ideen var at komme hurtigt i gang med det praktiske arbejde, og dermed afdække fundamentale udfordringer i forbindelse med at udvikle et EKG-måleapparat. Derfor var de funktionelle krav holdt til et minimum. I 2. iteration skulle systemet udbygges med flere funktioner, og mere raffinerede hardwareløsninger. Det viste sig dog, at omfanget af 1. iteration indeholdte nok arbejde til at udfylde næsten hele projektforsløbet. Derfor blev der midtvejs i projekter truffet en beslutning om ikke at påbegynde en ny iteration, men derimod at optimere de valgte løsninger fra første iteration. Indenfor hver iteration var det tænkt at bruge vandfaldsmodellen, og når der ses tilbage på projektets forløb, er det denne udviklingsmetode, der er mest beskrivende for projektet.

Projektet er blevet gennemført gennem 5 faser:

#### **Fase 1. Projektformulering**

I denne fase blev projektets overordnede mål og ønsker udformet.

#### **Fase 2. Kravspecifikation**

I denne fase blev der udformet specifikke krav til projektet. Det blev også formuleret hvordan disse krav skulle testes. Fasen er dokumenteret i kravspecifikationen og accepttesten.

#### **Fase 3. Systemarkitektur**

I denne fase blev de overordnede HW- og SW-komponenter, samt grænsefladerne mellem disse komponenter, fastlagt på baggrund af kravene i kravspecifikationen. Der blev herefter defineret ansvar til de enkelte systemblokke og hvordan grænsefladerne mellem dem skulle udformes. Denne fase blev dokumenteret i systemarkitekturen.

#### **Fase 4. Design/strukturering, implementering og test af HW og SW**

I denne fase blev den endelige strukturering/design udformet for henholdsvis HW og SW. Herefter blev den endelige struktur forsøgt implementeret og testet.

#### **Fase 5. Accepttest**

I den sidste fase blev det samlede implementerede system testet i henhold til den udformede accepttest.



Den endelig tidsplan:

Tidsplan i Uger	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Kravspec	X	X	X	X											
Accepttest	X	X	X	X											
Systemarkitektur				X	X										
HW/SW Strukturing					x	X	X	X	X		X	X	X		
Design						x	X	X	X		X	X	X		
Implementering						x	X	X	X		X	X	X		
<b>STATUS</b>								14-apr							
Test														X	
Dokumentering	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Rapport														X	X
Aflevering															X
Påske ferie															
Afleverings dato 27 maj															
Status 05-apr Hul igennem systemtet															

Figur 3 - Tidsplan for gruppe 3

### 7.1.1. Ansvarsfordeling

#### Projekt Dokumentation

- Kravspecifikation
  - Alle
- Systemarkitektur
  - Alle
- Design/Implementering/Test
  - SW pc-del
    - Joakim
    - Kristian
  - SW pc-del signalbehandling/BPM
    - Karsten \*
  - SW MCU-del
    - Rune
- HW Analog Sensor kredsløb
  - Jacob
  - Mathias
  - Anton
- HW ADC/Galvanisk adskillelse
  - Martin
  - Karsten
- Accepttest
  - Alle

\*Karsten K B har i softwaredelen stået for arbejdet med signalbehandling/BPM og har derved ingen ansvar for resten af software delen i projekt EKG.

### Projekt Rapport

Alle er ansvarlig for de overordnet punkter i Projektrapporten, dog undtagen disse punkter:

- Design/Implementering
  - SW pc-del
    - Joakim
    - Kristian
  - SW pc-del signalbehandling/BPM
    - Karsten K      B      \*
  - SW MCU-del
    - Rune Haahr Hansen
  - HW Analog Sensor kredsløb
    - Jacob
    - Mathias
    - Anton
  - HW ADC/Galvanisk Adskillelse
    - Martin
    - Karsten

\* Karsten K      B      har i softwaredelen stået for arbejdet med signalbehandling/BPM og har derved ingen ansvar for resten af software delen i projekt EKG.

## 7.2. Metoder

Den samlede proces består af projektformulering, kravspecifikation, fastlæggelse af systemarkitektur, samt analyse, design, implementering, test af hhv. software og hardware og afsluttende accepttest.

Der er anvendt specifikke metoder til udarbejdelse af de forskellige dele af projektet, hvoraf de vigtigste beskrives her:

- UML:
  - Til udarbejdelse af kravspecifikation, er der anvendt use cases, som definerer de funktionelle krav til produktet.
  - I forbindelse med softwaredesign er der anvendt systemsekvensdiagrammer, klassediagrammer og aktivitetsdiagrammer, som bruges til at planlægge softwarens struktur og funktionalitet.
- Iterativ udvikling:
  - Da der var brug for at få praktiske erfaringer med hardware- og softwareløsninger tidligt i projektet, blev der valgt en iterativ udviklingsmodel for projektet, som skulle give tidlige erfaringer omkring de udfordringer der måtte være med hensyn til hardware- og softwareimplementering.
- Review:
  - I løbet af projektet er der blevet holdt review af kravspecifikation og accepttest. Review er gennemført på tværs af projektgrupperne, med henblik på at få input fra udenforstående, og dermed klarlægge eventuelle fejl.

## 7.3. HW/SW Analyse

For at kunne opfylde kravene, stillet i kravspecifikationen, skal det samlede system kunne opfylde følgende primære funktionaliteter:

- Måling af EKG-signal.
- Visning af EKG-signal på en pc.
- Visning af udregnet BPM på en pc.
- Visning af et gemt EKG-signal på en pc.

Samtidig er der stillet specifikke tekniske krav, hvoraf de vigtigste, i forbindelse med det overordnede design, er:

- Inputsignaler med værdier mellem  $\pm 5$  mV skal reproducere i den grafiske grænseflade med en maksimal afvigelse på 20 % eller  $\pm 100$   $\mu$ V. Den af disse afvigelser, der bliver den største, bliver den bestemmende.
- Inputsignaler skal kunne måles i frekvensområdet fra 0,5 til 150 Hz.
- Der skal være galvanisk adskillelse mellem patient og computer/lysnettet.
- Der skal fjernes 50 Hz støj.

I den forbindelse er der nogle overvejelser til det overordnede system design, som kan deles op i følgende hovedpunkter:

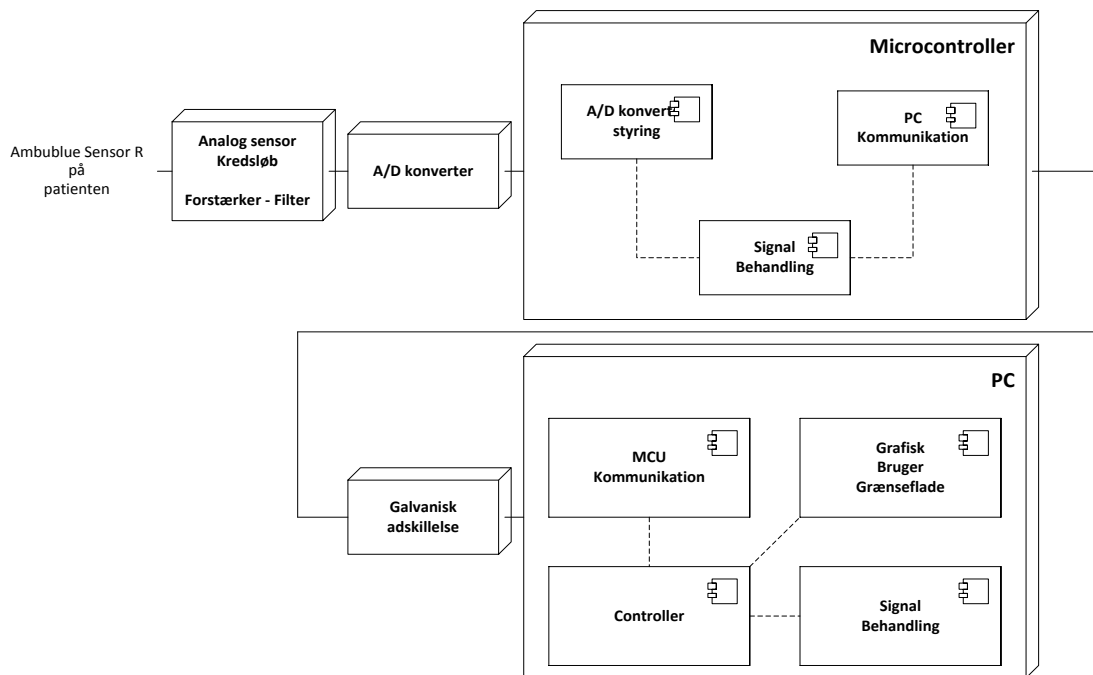
- Bestemmelse af samplerate for EKG-signalet der sendes til pc'en:  
For at opfylde Shannon kriteriet skal samplefrekvensen mindste være 2 gange den højeste frekvens for frekvensmåleområdet. Det giver en minimums samplefrekvens på 300 Hz. Der vælges en lidt højere samplefrekvens på 500 Hz, for at give bedre mulighed for signalbehandling.
- Beslutning om opløsningen for AD konverteringen:  
Der vælges en opløsning på 8 bit, 0-255 i værdi og der vælges en ADC med et dynamisk spændingsområde på 0 til 5 V.
- Bestemmelse af nødvendig lavpas-filtrering for at undgå aliaseringer:  
Med den valgte samplingfrekvens på 500 Hz vil der opstå aliaseringer for frekvenser over 250 Hz. For at undgå dette skal signalet dæmpes med 54 dB for frekvenser over 250 Hz, sådan at amplituden for disse frekvenser kommer ned under værdien for  $\frac{1}{2}$  bit.  
Det er her valgt at opdele filteret i en analog og digital del:
  - Analog filter:  
Det vælges, at AD konverteren skal sample med 1500 Hz. Kravet til analog filtrering bliver dermed, at signalet skal være dæmpet mindst 54 dB ved 750 Hz. (Nyquist/Shannon kriteriet)
  - Digital filter:  
For at det oprindelige krav, om 54 dB dæmpning ved 250 Hz, bliver opfyldt skal det digitale filter dæmpe signalet med 54 dB for frekvenser over 250 Hz. (Nyquist/Shannon kriteriet)
- Bestemmelse af den nødvendige forstærkning af det analoge signal før AD konverteringen: Området  $\pm 5$  mV skal forstærkes op til 0-5 V før sampling.

De satte tekniske og funktionelle krav deles ud i nødvendige software- og hardware-dele:

- Måling af EKG-signal:  
HW: Analog filtrering, forstærkning, AD-konvertering og galvanisk adskillelse.  
SW: AD-konverter styring, digital filtrering og downsampling.
- Visning af EKG-signalet på en pc:  
SW: Fjernelse af 50 Hz støj ud fra modtaget data og grafisk visning af de filtrerede data.
- Visning af udregnet BPM på en pc.  
SW: Udregning af BPM, ud fra modtaget data, og grafisk visning af dette.
- Visning af et gemt EKG-signal på en pc.  
SW: Grafisk visning af et gemt EKG-signal.

Gruppen har valgt at opfylde disse strukturelle krav og kravene til produktet ved at opdele projektet i en pc-del, en MCU-del og en hardware-del. Sammenlagt munder hele analysen ud i en overordnet løsningsmodel illustreret ved deployment diagrammet i systemarkitekturen. Se Figur 4 nedenfor.

#### 7.4. Systemarkitektur



Figur 4 - Deployment Diagram

EKG-systemet er opbygget i 5 blokke.

Et analogt sensorkredsløb registrerer data fra patienten. Dette kredsløb filtrerer uønskede frekvenser fra og forstærker de ønskede.

Signalet bliver sendt videre til AD-konvertering. I ADC'en foregår der en højpasfiltrering for at DC-afkoble signalet fra det analog sensor kredsløb og endvidere gives signalet et DC-offset på det halve af det dynamiske spændingsområde for indgangssignalet til ADC'en.

MCU'en modtager det digitaliserede signal. MCU'en står for at styre AD-konverteringen, samplingsfrekvensen på 1500 Hz og kommunikationen mellem MCU'en og pc'en ved hjælp af en RS-232 forbindelse.

Før signalet kommer til pc-delen, skal det igennem en galvanisk adskillelse. Denne adskillelse er sat ind for at beskytte patienten fra pc-delens 230 V forsyning ved evt. fejl.

Pc-delen modtager det digitale signal med en frekvens på 500 Hz.

På pc'en kan brugeren observere EKG-signalet og BPM på den grafiske brugergrænseflade. Herudover får brugeren mulighed for at hente gemte EKG-målinger. Pc'en står også for 50 Hz filtrering.

## 7.5. HW/SW Design

### 7.5.1. Software Design

Med henblik på at imødekomme den objektorienterede programmeringsfilosofi med indkapsling af funktioner i klasser, er der ved design af et projekt behov for overblik over de fornødne klasser samt hvilke funktioner de skal indeholde.

En designproces, hvor use cases bearbejdes til systemsekvens- og sekvensdiagrammer, sikrer at de ønskede funktionaliteter i kravspecifikationen opfyldes.

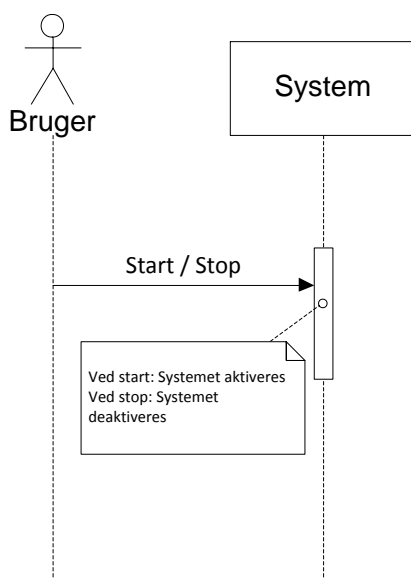
I implementeringsdelen kan det ses, at der ikke er den store forskel mellem design og implementeringen. Dette skyldes at det i designfasen af projektet har været nødvendigt, at tilegne sig viden gennem praktisk arbejde med koden. Der er altså foregået indledende test og implementering, inden design-fasen var afsluttet, og derfor passer de to dele overens.

Designet er således blevet fastlagt på baggrund af systemsekvens- og sekvensdiagrammer udarbejdet ud fra use case'ene, samt erfaringerne opnået gennem arbejdsprocessen. Diagrammerne opfylder de ønskede funktionaliteter, der fremgår af kravspecifikationen.

Under software-designforløbet er der udarbejdet systemsekvensdiagrammer samt detaljerede sekvensdiagrammer for begge use cases. Samtlige systemsekvensdiagrammer og detaljerede sekvensdiagrammer kan ses i projektdokumentationens dokument om detaljeret SW-design.

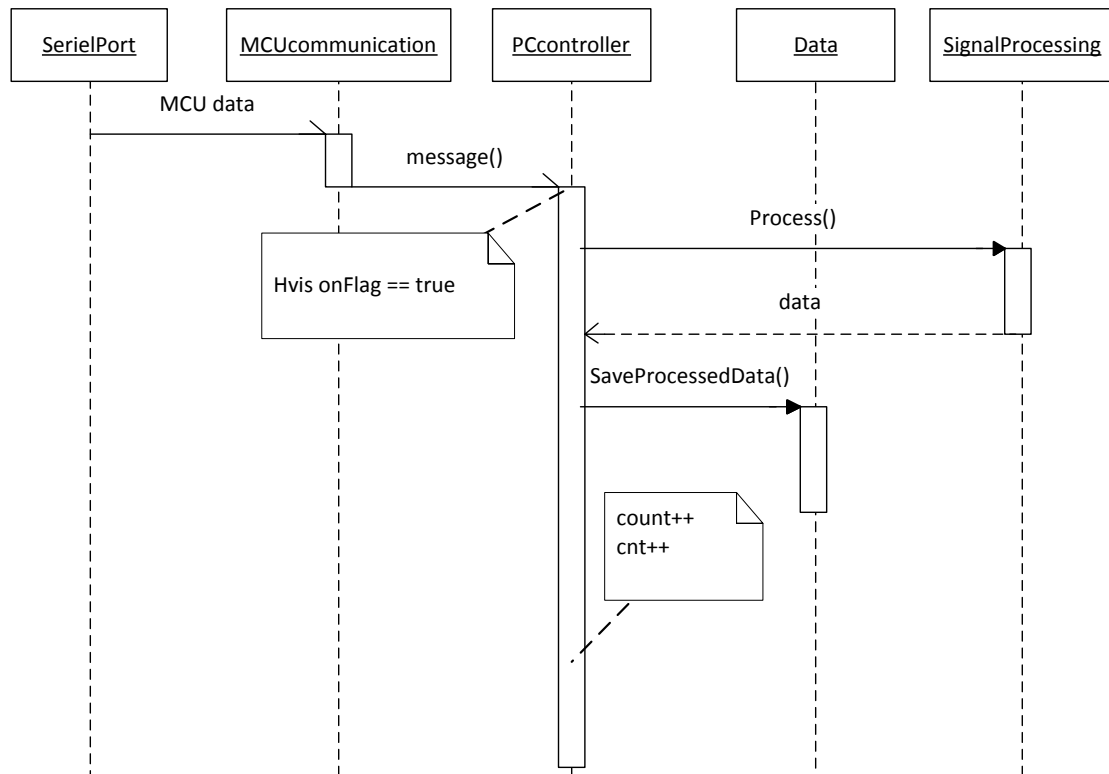
Eksempel på systemsekvensdiagram:

Use case 1 – Start / Stop program



Figur 5 - Systemsekvensdiagram for Use Case 1 "Start/stop program"

Her ses et eksempel på et detaljeret sekvensdiagram. I diagrammet ses hovedforløbet for programmet projekt EKG:



Figur 6 - Detaljeret sekvensdiagram for hovedprogrammet i Projekt EKG

Ud fra samtlige systemsekvens- og sekvensdiagrammer, er de nødvendige SW-klasser blevet identificeret som foreningsmængden af de fremkomne klasser på de detaljerede systemsekvensdiagrammer. De fremkomne klasser er vist på klassediagrammet på Figur 8 på side 19.

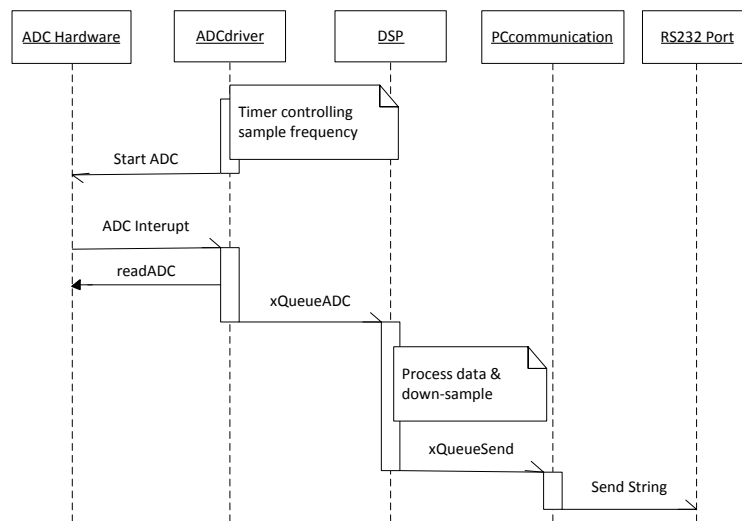


MCU softwaren indeholder 3 dele:

- AD-konverter styring
- Signalbehandling
- Pc kommunikation

Dette kan ses som 3 klasser, med AD-konverter styring(ADCdriver), pc kommunikation(PCcommunication) som boundary klasser og signalbehandling(DSP) som en controller klasse.

Sekvensdiagram til overblik over signalgennemgangen på MCU'en.



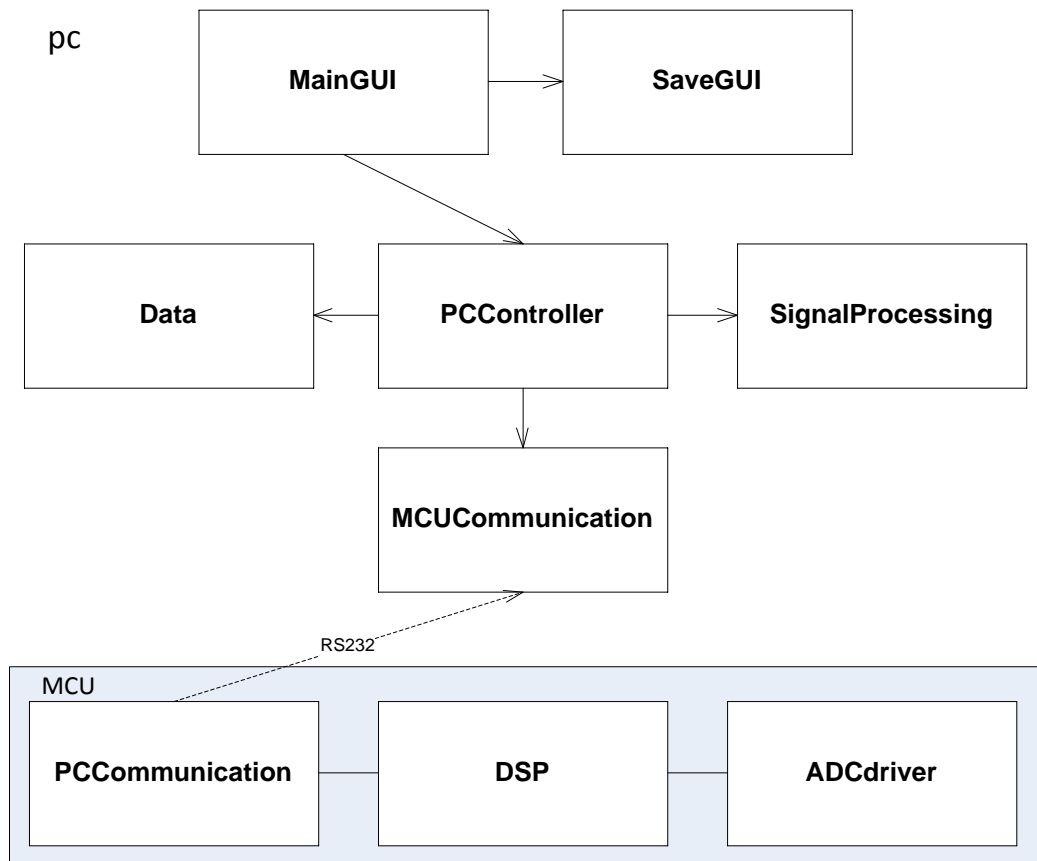
Figur 7 - Detaljeret sekvensdiagram for signalgennemgangen i MCU'ens software

#### Overvejelser angående brug af realtids-operativ-system på MCU'en.

- Et realtidssystem kan lægge beslag på nogle af MCU'ens funktioner og kræver en smule af processor tiden til at håndtere bagvedliggende operativsystemskode.
- Et realtidssystem giver bedre/flere muligheder for strukturering af MCU programmeringen.

Der er valgt at køre med et RTOS system kaldet FreeRTOS grundet muligheden for en bedre strukturering af MCU programmeringen.

### 7.5.1.1. Overordnet klassediagram



Figur 8 - Overordnet klassediagram for software delen af projektet

Samtlige klasser er beskrevet nærmere i klassebeskrivelserne nedenfor.

### 7.5.1.2. Klassebeskrivelser

**MainGUI:** Klassens ansvar består i at oprette brugergrænsefladen. Her findes relevante knapper, der kan aktiveres af brugeren. Det er også her EKG-grafen tegnes.

**SaveGUI:** Ansvar for denne klasse er at oprette et nyt vindue når use case 2 aktiveres. I dette vindue skal grafen for gemt data tegnes.

**PCcontroller:** PCcontroller-klassens ansvar er at sørge for afviklingen af programmet.

**Data:** Data-klassens ansvar er at lagre relevant data.

**SignalProcessing:** Ansvar for denne klasse er at behandle de modtagne data for at frasortere støj. Her skal også udregnes puls (BPM).

**MCUcommunication:** Denne klasse står for at modtage data fra serielporten.

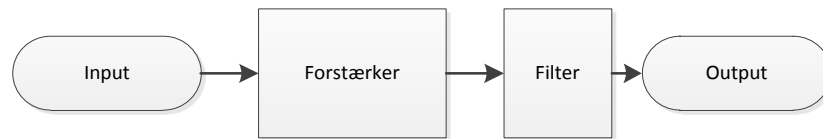
**PCcommunication:** Klassens ansvar er at sende data via serielporten til pc'en.

**DSP:** Klassen står for at udføre digital signalbehandling i form af et lavpas-filter. Den står også for at downsample signalet fra 1500 til 500 Hz.

**ADCdriver:** Klassens ansvar er at styre AD-konverteren.

## 7.5.2. Hardware Design

### 7.5.2.1. Analog sensor kredsløb



Figur 9 - Analog sensor blokdiagram

#### Forstærkning

Da indgangssignal fra elektroderne har en peek-to-peek spænding på op til 10 mV AC og kravet fra ADC'en til signalet er at det skal ligge mellem 0 og 5 V, skal signalet forstærkes.

Det ønskes at et udsving på 10 mV ved indgangen af kredsløbet skal svare til et udsving på 5 V ved udgangen, derfor skal forstærkningen være.

$$\text{Forstærkning} = \frac{\pm 2.5 \text{ V}}{\pm 5 \text{ mV}} = \frac{5 \text{ V}}{10 \text{ mV}} = 500$$

#### Filter

Frekvenser over den halve samplings frekvens ønskes dæmpet til mindre end det laveste trin (LSB) på ADC'en. Da ADC'en har en reference spænding på 5 V og en opløsning på 8 bit svarer LSB på ADC'en til:

$$LSB = \frac{ADC_{\text{referencen}}}{ADC_{\text{opløsning}}} = \text{Trin}_{\text{spændingen}} \Rightarrow LSB = \frac{5 \text{ V}}{2^8} = 19,5 \text{ mV}$$

Frekvenser over den halve samplings frekvens ønskes dæmpet til det halve af ADC'ens LSB.

$$20 \log \left( \frac{\frac{19.5 \text{ mV}}{2}}{5 \text{ V}} \right) \approx -54 \text{ dB}$$

$$\text{den halve samplings frekvens} \frac{1500 \text{ Hz}}{2} = 750 \text{ Hz}$$

Det betyder at frekvenser over 750 Hz skal dæmpes med 54 dB eller mere.

### 7.5.2.2. ADC

#### Overordnede krav til ADC-kredsløbet

Det overordnede planlagte design for EKG-systemet medfører følgende krav til valget af ADC og design af tilhørende elektronik:

- Samplingsfrekvens: 1500 Hz
- ADC'en skal kunne konvertere et analogt indgangssignal, med et dynamisk spændingsområde på 0-5 V, til en bit-værdi mellem 0 og 255.
- Signalet fra Analogt Sensorkredsløb skal DC-afkobles, og skal samtidig gives et DC-offset på det halve af det dynamiske spændingsområde for indgangssignalet til ADC'en.
- Bit-fejlraten skal være lav for at kunne overholde de overordnede krav til EKG-systemet omkring målepræcision.
- Forsyning til ADC og tilhørende elektronik skal forsynes med 5 V.
- Inputsignaler skal kunne måles indenfor et frekvensområde, 0,5-150 Hz.

Disse krav medfører begrænsninger i forhold til valg af ADC, som er beskrevet mere i detalje i følgende afsnit.

#### Samplingsfrekvens

For at opfylde kravet til samplingsfrekvensen, skal ADC'en kunne gennemføre en AD-konvertering på tiden:

$$t_{konvertering} = \frac{1}{f_s} = \frac{1}{1500 \text{ Hz}} = 67 \text{ ms}$$

#### DC-afkobling af indgangssignalet

Der skal implementeres et indgangstrin, der sørger for DC-afkobling af signalet fra Analog Sensorkredsløb, og samtidig giver signalet et DC-offset, svarende til midten af det dynamiske spændingsområde for indgangssignalet til ADC'en.

Indgangstrinnet skal give 1 gangs forstærkning af signalet.

#### Spændingsforsyning til ADC og MCU

Det er aftalt at al elektronik i EKG-systemet skal forsynes med et 9 V batteri. Da det samtidigt er aftalt at ADC, MCU og tilhørende elektronik skal køre på 5 V, så skal der implementeres spændingsregulering, der sørger for at omsætte de 9 V til 5 V.

### 7.5.2.3. Galvanisk adskillelse

En nødvendig funktionalitet for EKG-systemet er at patienten er afskærmet fra lysnettet. Målet er derfor at kunne lave en adskillelse, der både afskærmer 230 V spændingen fra lysnettet, og kan lade et registrerbart digitalt signal passer igennem fra ADC'en til MCU'en. Komponenten skal dog opfylde kravet til MCU'ens Baud rate på 57600.

## 7.6. HW/SW Implementering

### 7.6.1. Software implementering

#### 7.6.1.1. *pc-del*

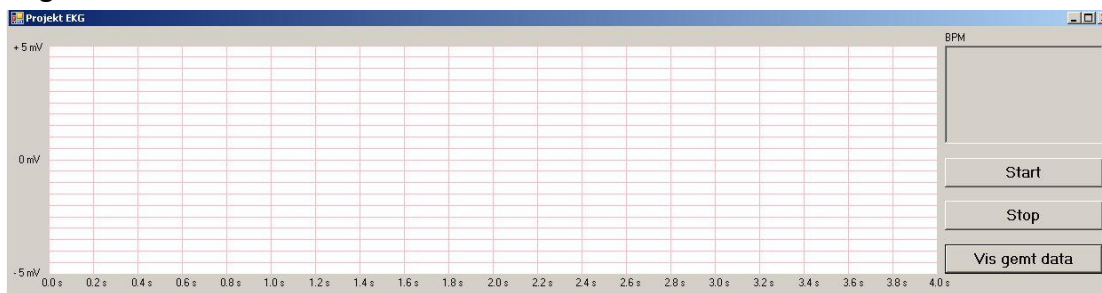
Implementeringen af systemet er i videst muligt omfang sket på baggrund af de i designfasen udarbejdede systemsekvens- og sekvensdiagrammer samt det heraf resulterende klassediagram.

Den fulde dokumentation for software-implementeringen fremgår af projektdokumentationen.

I følgende afsnit bruges begrebet data-pakker, og med dette menes at der er modtaget 3 chars og en terminering fra MCU'en, hvilket svarer til en måleværdi.

#### Implementering af MainGUI

Objektet af denne klasse oprettes når programmet startes. I constructoren til denne klasse oprettes knapper og vinduer til den grafiske brugergrænseflade. Det får følgende udseende:



Figur 10 – Programvindue af MainGUI

Metoden `UpdateGUIEvent()` bliver kaldt vha. eventet `UpdateGui()` fra `PCcontrolleren`. Dette sker hver gang der er modtaget 10 data-pakker fra MCU'en. Den kalder så metoderne `DrawGraph()` og `WriteBPM()`. Disse to metoder står for at tegne grafen og skrive den målte BPM på brugergrænsefladen.

DrawGraph() modtager de 10 data-pakker som parameter og tegner punkterne på grafen.

```
#region Graf
private void DrawGraph(double[] data) //Målinger indtegnes på graf
{
    Pen pen = new Pen(Color.Blue);
    pen.Width = 2;

    for (int i = 0; i < 5; i++)
    {
        x = i + 5*drawkaldt;
        y = 256 - (float)data[i*2];
        paper.DrawLine(pen, xOld, yOld, x, y);
        xOld = x;
        yOld = y;
        y = 256 - (float)data[i*2 + 1];
        paper.DrawLine(pen, xOld, yOld, x, y);
        yOld = y;
        if (x == 999)
        {
            paper.Clear(Color.White);
            x = 0;
            xOld = 0;
            table();
        }
    }
    drawkaldt++;
    if (drawkaldt == 200)
    {
        drawkaldt = 0;
    }
    pictureBox1.Image = myBitmap;
    pen.Dispose();
}
```

Figur 11 - Metoden DrawGraph() i C# kode

WriteBPM() opdaterer BPM feltet med den beregnede værdi.

Metoden table() tegner tern på graf-vinduet iht. kravene i kravspecifikationen.

```
private void table() //Der tegnes tern i graf-vinduet
{
    Pen pen = new Pen(Color.LightPink);
    for (int i = 0; i < pictureBox1.Height / 5; i++)
    {
        paper.DrawLine(pen, 0, i * pictureBox1.Height / 20, pictureBox1.Width, i * pictureBox1.Height / 20);
        paper.DrawLine(pen, i * pictureBox1.Width / 20, 0, i * pictureBox1.Width / 20, pictureBox1.Height);
    }
}
```

Figur 12 - Metoden table() i C# kode

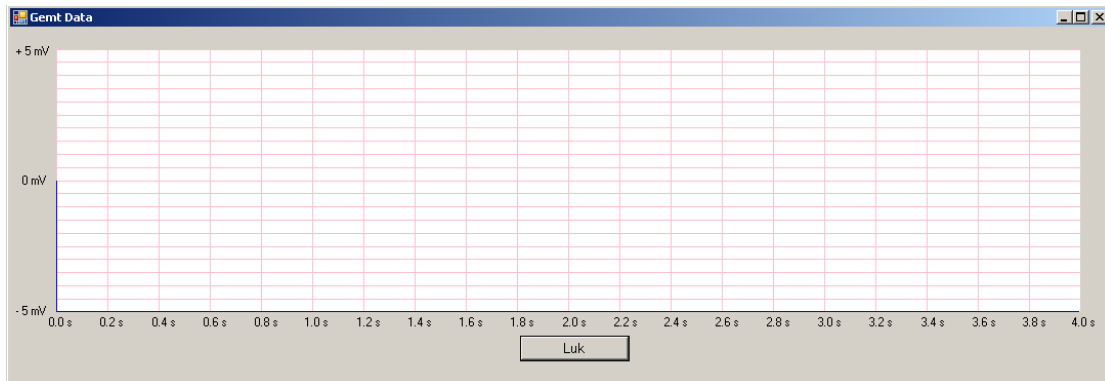
button1\_Click() metoden aktiveres, når brugeren trykker på "Vis gemt data"-knappen. Metoden opretter et objekt af SaveGUI-klassen, og kalder relevante metoder i den, for at tegne grafen for det gemte data i det nye vindue.

button2\_Click() kaldes når brugeren trykker på Start-knappen. Knappen sætter variabelen onFlag = true.

button3\_Click() kaldes når brugeren trykker på Stop-knappen. Knappen sætter variabelen onFlag = false.

### Implementering af SaveGUI

Objektet af SaveGUI bliver oprettet når brugeren trykker på "Vis gemt data"-knappen. I constructoren oprettes det nye vindue med følgende udseende:



Figur 13 - Programvindue af SaveGUI

GetOldDataForGraph() metoden henter den gemte data. DrawOldGraph() tegner så data i vinduet:

```
--
40 public void DrawOldGraph() //Målinger indtegnes på graf
41 {
42     Pen pen = new Pen(Color.Blue);
43     pen.Width = 2;
44
45     for (int i = 0; i < 1000; i++)
46     {
47         x = i;
48         y = 256 - (float)TempArray[i*2];
49         paper.DrawLine(pen, xOld, yOld, x, y);
50         xOld = x;
51         yOld = y;
52         y = 256 - (float)TempArray[i*2 + 1];
53         paper.DrawLine(pen, xOld, yOld, x, y);
54         yOld = y;
55     }
56
57     pictureBox2.Image = myBitmap2;
58
59     //Clear grafen når den når til enden
60
61     pen.Dispose();
62 }
63
```

Figur 14 - Metoden DrawOldGraph() i C# kode

button1\_Click() metoden kaldes, når brugeren trykker på "Luk"-knappen, og vinduet lukkes.

table() virker ligesom i MainGUI og bliver kaldt i constructeren til klassen.

### Implementering af PCcontroller klassen

Objektet af PCcontroller-klassen oprettes af MainGUI, når programmet startes. I constructoren til PCcontroller oprettes objekter af klasserne MCUcommunication, SignalProcessing og Data.

DataReceievedEvent() metoden kaldes af evente message() i MCUcontroller-klassen.

Dette sker når der er modtaget en data-pakke fra MCU'en. I metoden findes et if/else statement der holder øje med om variablen onFlag er true eller false, og kun i tilfælde af true vil den udføre noget. Metoden kalder Process() metoden i SignalBehandling og data bliver hermed behandlet. Herefter gemmes den behandlede data vha.

SaveProcessedData() metoden i Data-klassen. Variablerne cnt og count inkrementeres herefter. Når variabelen count == 10 aktiveres eventet UpdateGui(), og når variabelen cnt == 2000 aktiveres metoden SaveAsOldData() i Data-klassen.

```

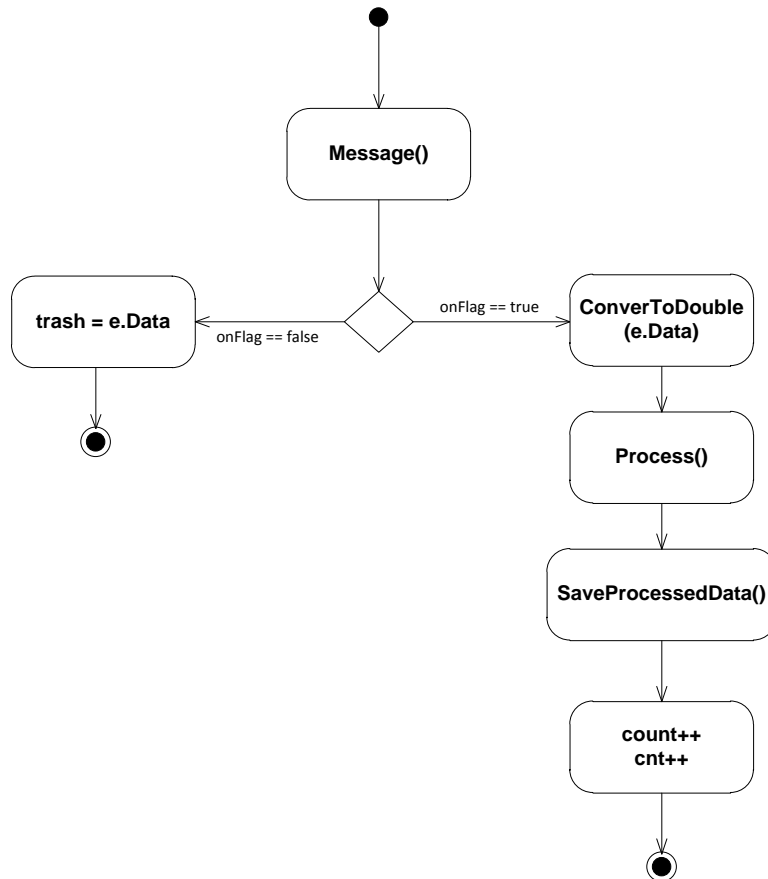
63 private void DataReceievedEvent(object sender, SerialPortEventArgs e)
64 {
65
66     if (onFlag == true)
67     {
68         try
69         {
70             data.SaveProcessedData(signalprocessing.Process(Convert.ToDouble(e.Data)));
71         }
72         catch
73         {
74             data.SaveProcessedData(signalprocessing.Process(128));
75         }
76         count++;
77         cnt++;
78     }
79     else
80     {
81         trash = e.Data;
82     }
83
84     if (count == 10) // sender måling til GUI med 50 Hz.
85     {
86         UpdateGui(this, new UPdateGUIEventArgs(data.GetProcessedData(), signalprocessing.BPM));
87         count = 0;
88     }
89
90     if (cnt == 2000)
91     {
92         data.SaveAsOldData();
93         cnt = 0;
94     }
95
96 }

```

Figur 15 - Metode DataReceievedEvent() i C# kode

På Figur 17 Side 27 ses også aktivitetsdiagrammet for metoden:





Figur 16 - Aktivitetsdiagram for metoden `DataReceivedEvent()`

### Implementering af Data-klassen

Metoden `SaveProcessedData()` gemmer den behandlede data i et array på 2000 pladser kaldet `ProcessedData`.

Metoden `GetProcessedData()` henter de 10 nyeste elementer fra arrayet `ProcessedData`.

`SaveAsOldData()` metoden kaldes når der skal gemmes data (`cnt == 2000`). Metoden gemmer indholdet i `ProcessedData` i et nyt array kaldet `OldProcessedData`.

`GetOldData()` metoden kaldes når den gamle data skal bruges (når brugeren trykker på "Vis gemt data"-knappen). Den returnerer data fra `OldProcessedData` arrayet.

### MCUcommunication

MCUcommunication-klassens ansvar er at opsætte og modtage kommunikation fra MCU'en. Den opsætter seriel-porten til den valgte kommunikation-protokol, og når en char er modtaget, bliver den vha. et event sat i en buffer i metoden serialPort\_DataRecieved(). Når metoden så registrerer en terminering i form af "/r", så sendes bufferen afsted via et event message() over til PCcontroller-klassen, og herefter tømmes bufferen.

```
private void serialPort_DataReceived(object source, SerialDataReceivedEventArgs e)
{
    char[] inputData = new char[1];

    while (serialPort.BytesToRead > 0)
    {
        // Get one char at the time
        inputData[0] = (char)serialPort.ReadChar();

        // if not terminators then put in buffer
        if ((inputData[0] != '\r') && (inputData[0] != '\n'))
        {
            buffer += inputData[0];
        }

        // if terminator...
        if ((inputData[0] == '\r') && (buffer != ""))
        {
            // ShowSerialData(s); //do not work - invoke is necessary!
            message(this, new SerialPortEventArgs(buffer));
            buffer = string.Empty;
        }
    }
}
```

Figur 17 - Metoden serialPort\_DataReceived() i C# kode

## Signalprocessing

I følgende afsnit bliver klassens vigtigste metoder beskrevet. Hjælpefunktioner er udeladt, men beskrivelser heraf kan findes i projektdokumentation.

Process():

Metoden modtager data, sørger for at der bliver udført signalbehandling og beregning af BPM, og returnerer den seneste behandlede data.

FilterReceivedData():

Metoden udfører digital signalbehandling der skal fjerne 50 Hz støj. Dette er implementeret med et 2. ordens IIR-filter efter følgende differensligning:

$$y[n] = \frac{b_0x[n] + b_1x[n-1] + b_2x[n-2] - a_1y[n-1] - a_2y[n-2]}{a_0}$$

hvor feedforward-koefficienterne er:

$$b_0 = 10,472135954999583$$

$$b_1 = -16,944271909999166$$

$$b_2 = 10,472135954999583$$

og feedback-koefficienterne er:

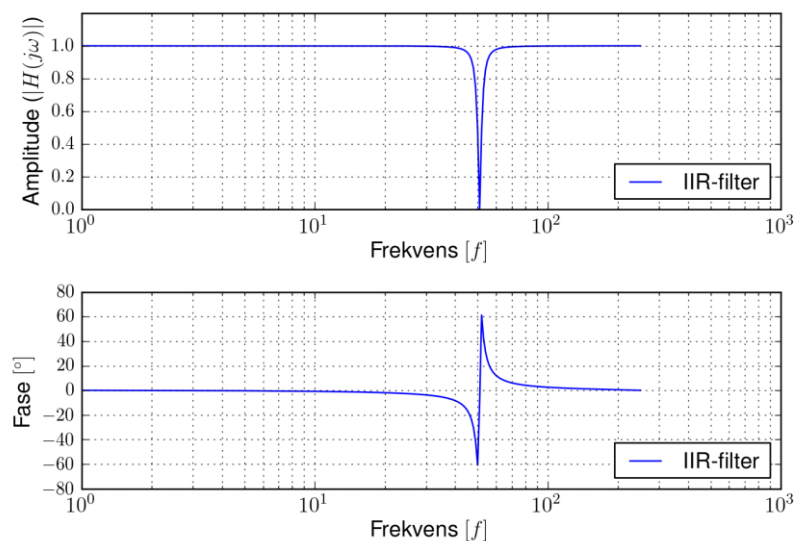
$$a_0 = 10,710196696839827$$

$$a_1 = -16,944271909999166$$

$$a_2 = 10,23407521315934$$

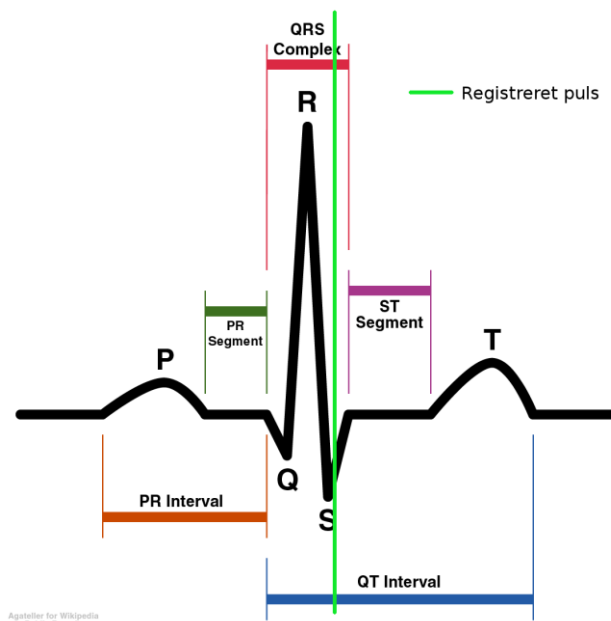
Filteret er designet med metoden "Biliniær transformation", hvor et filter designet i s-domænet transformeres til z-domænet, og derefter transformeres til en differensligning i samplingdomænet.

Frekvenskarakteristik for det implementerede filter



Figur 18 - Frekvenskarakteristik genereret i SciPy

CalculateBPM():



Figur 19 - Registrering af pulsslag: <http://en.wikipedia.org/wiki/Electrocardiography>

Metoden detekterer pulsslag og beregner BPM ud fra tiden mellem pulsslagene. Til detektering af pulsslag er der implementeret en algoritme, der grundlæggende gennemløber følgende punkter:

1. Er hældningen positiv, negativ eller 0?
2. Hvis hældningen er negativ (gældende for områderne P, Q, R-S, T):
  - a. Mål amplituden siden hældningen sidst var positiv.
3. Hvis hældningen er positiv (gældende for områderne P, Q-R, S-, T):
  - a. Registrer puls hvis amplituden for seneste område med negativ hældning langt overstiger den typiske maksimale amplitude for T (Dette er kun gældende for området lige efter S, og dermed identificeres område R-S).

Det betyder at registreringen af et pulsslag sker som vist på følgende Figur 19

Når der er registreret pulsslag gemmes tiden, der er passeret siden sidste registrerede pulsslag.

For at undgå alt for stor følsomhed overfor ændringer af pulsens hastighed, bliver værdien for BPM beregnet som gennemsnittet af de sidste tre BPM-værdier.

### 7.6.1.2. MCU-del

#### Brug af FreeRTOS

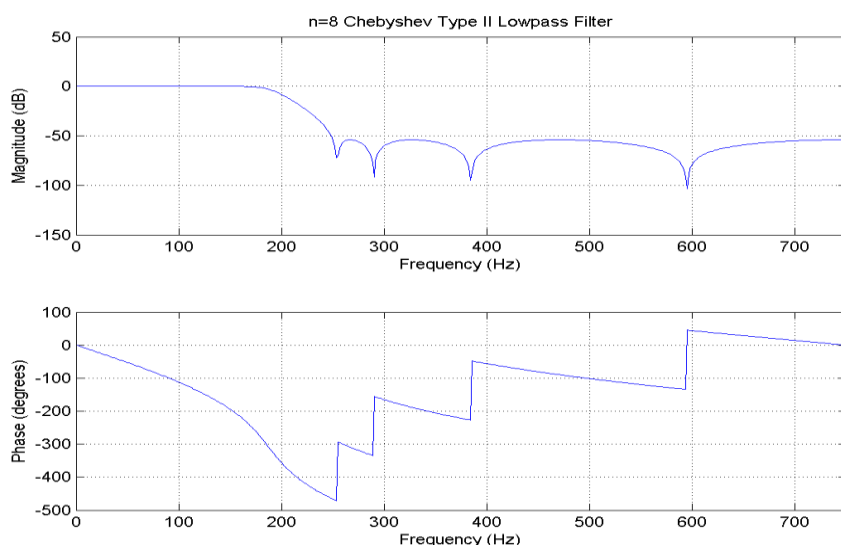
Der gøres brug af realtids-operativ-systemet FreeRTOS. Det er sat op til 1500 Hz ticks og der bliver brugt indbyggede metoder fra queues og tasks.

#### Implementering af ADCdriver

ADCdriveren består af 2 dele. Første del er timings delen der står for at starte AD-konverteren. Den består af en task der aktivere AD-konverteren for derefter at vente 1 tick. Denne task sættes til højeste prioritet, således at aktiveringen af AD-konverteren altid startes hver gang FreeRTOS tæller et tick. Anden del består af et interrupt, der læser data fra AD-konverteren, og sender det videre via en queue til DSP. Interruptet aktiveres af AD-konverteren via et ekstern interrupt ben.

#### Implementering af DSP

DSP står for at signalbehandle data, downsample fra 1500 Hz ned til 500 Hz og sende data videre til PCcommunication. Downsamplingen tager hver tredje måling, og sender den videre. Signalbehandlingen omregner signalet, så der er et pasbånd op til 150 Hz og et stopbånd fra 250 Hz og op. Der er en maksimal dæmpning i pas båndet på 0.03 dB og en minimumsdæmpning i stop båndet på 54 dB.



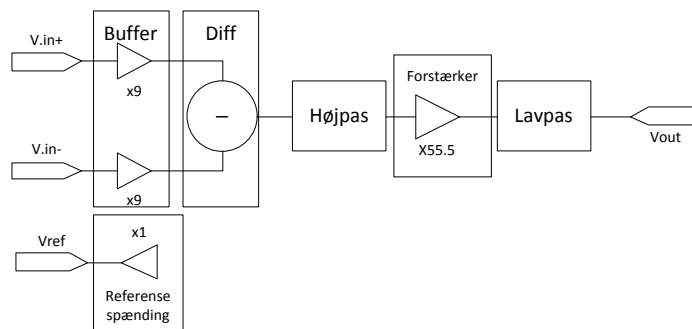
Figur 20 - Frekvenskarakteristik af et 8. ordens Chebyshev filter

#### Implementering af PCcommunication

PCcommunication består af 2 dele. Første del består af en task, der modtager data fra DSP, omregner tallet til en streng af karakterer, sender den første af sted og lægger resten i en queue. Denne task sættes til næst højeste prioritet. Anden del består af et interrupt der sender den forreste karakter i queue'en. Interruptet aktiveres når UART'en er færdig med at sende en karakter.

## 7.6.2. Hardware implementering

### 7.6.2.1. Analog sensor kredsløb



Figur 21 – Blokdiagram: Analog sensor kredsløb

#### Placering af modul

Signalet modtaget af bufferen bliver forstærket 9 gange, da det ønskes at støj fra modstande og operationsforstærkere, der er placeret senere i kredsløbet, skal bidrage med så lidt støj som muligt.

Differensforstærkeren sørger for at forskellen mellem de to input signaler sendes videre til højpas-filteret.

Signalet højpas-filtreres, for at fjerne evt. DC-niveau inden det forstærkes igen.

Signalet forstærkes igen med 55,5 gange for at opnå de 500 ganges forstærkning, som er kravet for grænsefladen til ADC'en.

$$55,5 \cdot 9 = 499,5 \approx 500$$

Lavpas-filteret dæmper frekvenser over 750 Hz med 54 dB eller mere, for at undgå aliasering når signalet konverteres til et digitalt signal. Samtidigt må der ikke være betydende dæmpning ved frekvenser under 150 Hz.

#### Afkobling af Operationsforstærkerne

Der er anvendt afkobling på forsyningerne til samtlige operationsforstærkere, for at undgå at støj fra strømforsyningen påvirker operationsforstærkerne.

#### Referencespændingen Vref

For at få bufferne til at arbejde i operationsforstærkernes lineære område, løftes reference spændingen fra 0 V til det halve af forsyningsspændingen. Ved at betragte denne spænding som stel, kan forsyningen betragtes som  $\pm 4,5$  V.

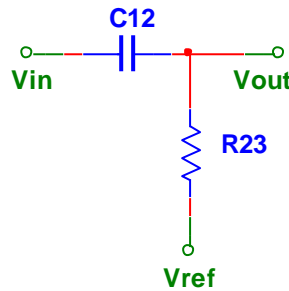
## Filter

### Højpas-filter

Højpas-filterets funktion er at fjerne evt. DC spænding. I kravspecifikationen er det opgivet, at frekvenser over 0,5 Hz skal kunne måles, og derfor sættes knækfrekvensen til ca. 0,2 Hz.

### Løsning

Det er valgt at anvende et 1. ordens passivt filter:



Figur 22 - Kredsløbsoversigt for et 1.ordens passivt filter

Der vælges en standard komponentværdi for kondensatoren og en passende modstandsværdi beregnes:

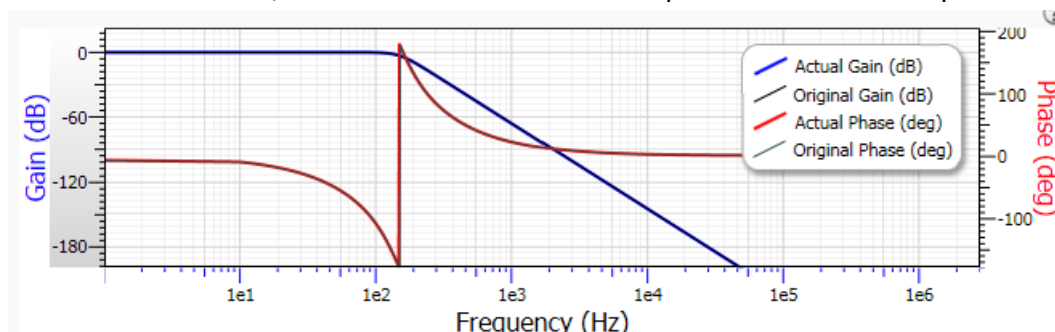
$$C_{12} = 1 \mu F \Rightarrow R_{23} = \frac{1}{2\pi \cdot 1 \mu F \cdot f_c} = 795,8 k\Omega \approx 800 k\Omega$$

### Lavpas-filter

Lavpas- eller antialiaserings-filteret skal dæmpe frekvenser over den halve samplings frekvens (750 Hz) med minimum 54 dB. Det forventes, at EKG signalet ligger i frekvens-området under 150 Hz. Derfor bør signalet ikke dæmpes under denne frekvens.

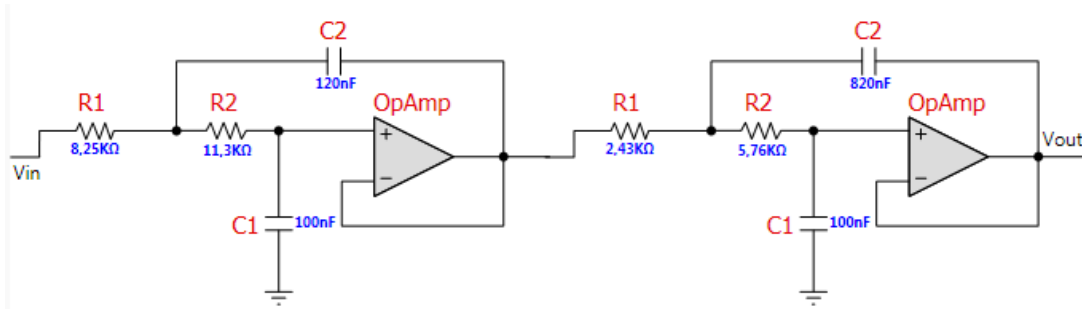
### Løsning

Da signalet skal påvirkes så lidt som muligt under knækfrekvensen, faldt valget af filter på et Butterworth-filter, da dette har den fladeste amplitudekarakteristik i pasbåndet.



Figur 23 - Diagram over amplitude- og fase-karakteristik over lavpasfilter

Til beregning af komponentværdier er der benyttet et program til beregning af filtre, Filter Pro.



Figur 24 - Kredsløb diagram udregnet af Filter pro

Dæmpningen ved 150 Hz er ca. 3 dB som ønsket.

Frequency (Hz)	Frequency (rad/sec)	Gain (V/V)	Gain (dB)	Phase (Degrees)	Phase (Radians)	Group Delay (uSec)
151,356	950,999	0,69	-3,226	176,844	3,087	3882,68
Frequency (Hz)	Frequency (rad/sec)	Gain (V/V)	Gain (dB)	Phase (Degrees)	Phase (Radians)	Group Delay (uSec)
749,894	4711,724	0,002	-56,032	30,087	0,525	112,662

Figur 25 - Tabel over beregnet værdier for amplitude og fasekarakteristik

Dæmpningen ved 750 Hz er ca. 56 dB, som opfylder kravet om 54 dB dæmpning.

## Forstærkning

### Buffer

Da impedansen mellem signal og kredsløb er en ukendt størrelse, der kan ækvivaleres med en modstand på 10 kΩ - 1 MΩ, er der valgt at sætte buffere imellem elektroderne og differensforstærkeren. Ved at forstærke signalet så meget som muligt i dette led, kan støjbidraget fra senere led gøres mindre betydende for det endelige signal støj forhold.

### Løsning

Det er oplyst, at elektroderne kan have et offset op til 0,3 V, at operationsforstærkeren har et offset på 5 mV og at signalet ligger i spændingsområdet  $\pm 5$  mV, hvilket giver en maksimal spænding på:

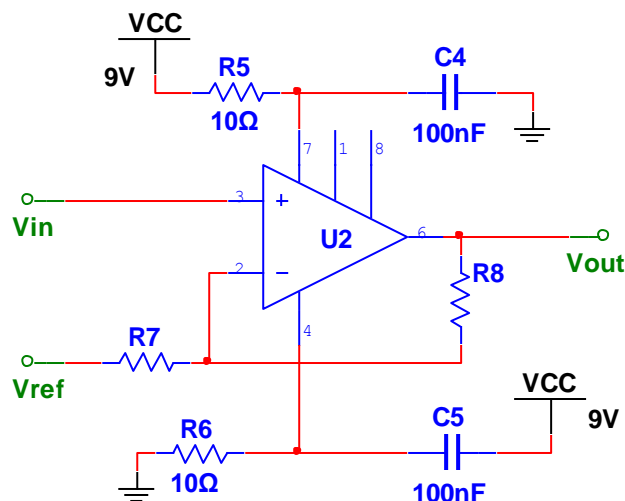
$$V_{in} = 300 \text{ mV} + 5 \text{ mV} + 5 \text{ mV} = 310 \text{ mV}$$

Det oplyses også, at operationsforstærkeren har et spændingsfald på 1,5 V fra forsyningsspændingen (9 V) til maksimal output og en referencespænding på 4,5 V. Ud fra dette beregnes den maksimale forstærkning, der bør anvendes i dette led:

$$Max_{Gain} = \frac{9 \text{ V} - 1,5 \text{ V} - 4,5 \text{ V}}{310 \text{ mV}} = 9,67 \approx 9$$

Ud fra dette vælges der modstande, som begrænser strømmen nok til ikke at overbelaste operationsforstærkerens udgange, samt med så små værdier, at de ikke bidrager med nogen nævneværdig termisk spændingsstøj.





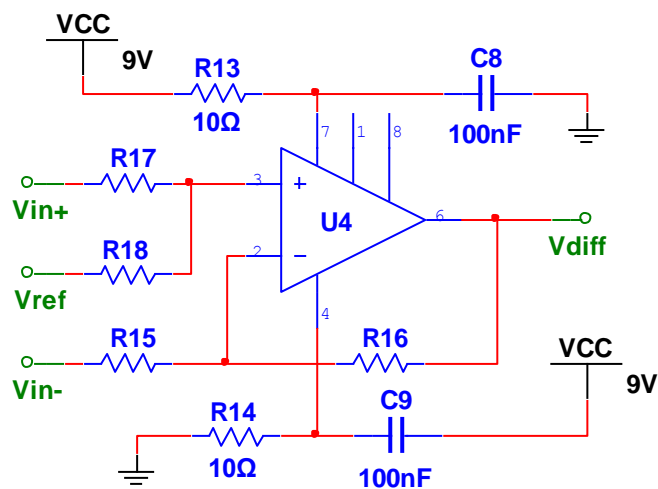
Figur 26 - Kredsløb diagram over buffer

$$R_7 = 1\text{k}\Omega \text{ og } R_8 = 8\text{k}\Omega$$

### Differens

Da det er forskellen mellem indgangene der ønskes forstærket, så forbindes udgangene fra bufferne til en differensforstærker. I dette led ønskes signalet ikke forstærket pga. DC forskelle mellem indgangene.

### Løsning



Figur 27 - Kredsløb diagram over differensforstærker

Da signalet ikke skal forstærkes, er den eneste bekymring, mht. modstandsværdierne, strømbegrænsningen af signalet fra bufferne, og den termiske støj de bidrager med. Komponentværdierne sættes til:

$$R_{17} = R_{18} = R_{15} = R_{16} = 1\text{ k}\Omega$$

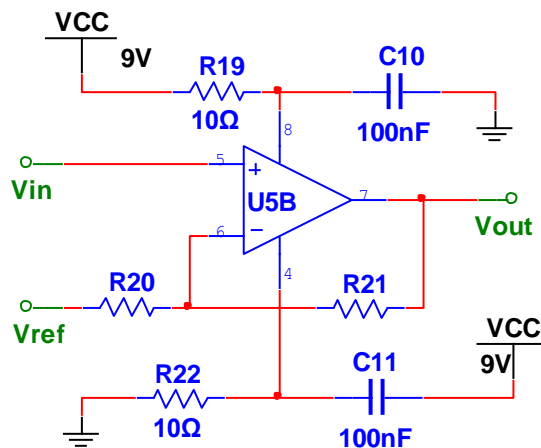
### Forstærker Led

Da det endelige signal skal forstærkes 500 gange, og det indtil videre kun er blevet forstærket 9 gange, skal signalet forstærkes yderligere.

### Løsning

For at finde de modstandsværdier, der skal bruges i forstærkerleddet, beregnes der hvor meget yderligere forstærkning, der er nødvendig.

$$\text{Forstærker}_{\text{Gain}} = \frac{500}{9} \approx 55.5$$

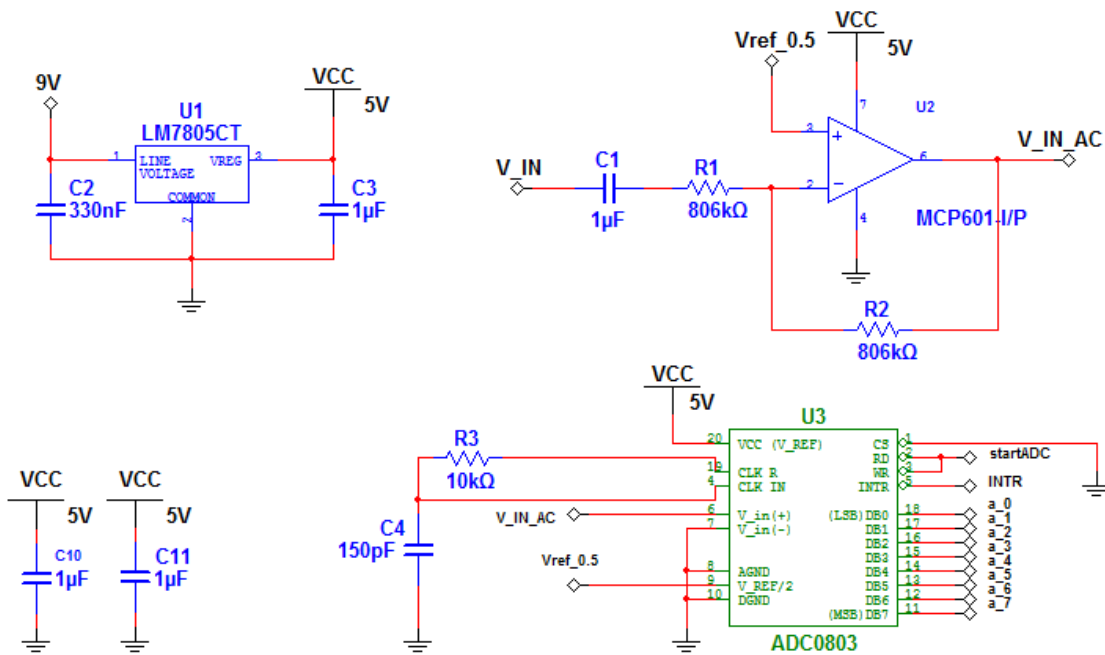


Figur 28 - Kredsløb diagram over forstærker led

Modstanden  $R_{20}$  sættes til 1 kΩ, da det antages at denne værdi begrænser strømmen nok til ikke at overbelaste operationsforstærkeren.  $R_{21}$  beregnes nu:

$$\text{Gain} = \left(1 + \frac{R_{21}}{R_{20}}\right) = 55,5 \Rightarrow R_{21} = (55,5 - 1)R_{20} = 54,5 \text{ k}\Omega$$

### 7.6.2.2. ADC



Figur 29 - Kredsløb diagram over ADC

#### Valg af ADC

De tekniske krav til ADC'en, beskrevet under HW-Design, bliver opfyldt med komponenten ADC0803, hvilket vises i de følgende afsnit.

#### Konverteringstid, bit-fejlrate og opløsning

Med den valgte samplingsfrekvens på 1500 Hz, må en AD-konvertering maksimalt tage 67 ms. I databladet for ADC0803 er det opgivet, at tiden for en konvertering maksimalt er:

$$T_C = 73 \cdot \frac{1}{f_{CLK}}$$

Den ønskede konverteringstid på 67 ms medfører følgende minimumskrav til  $f_{CLK}$ :

$$T_C = 67 \text{ ms} = \frac{73}{f_{CLK}} \Leftrightarrow f_{CLK} = \frac{73}{67 \text{ ms}} = 1,09 \text{ kHz}$$

Samtidigt skal konverteringstiden holdes under de 67 ms, for at sikre at Linearity Error for LSB bliver under 0,5.

I implementeringen er der valgt en typisk opsætning af ADC'en, med følgende valgte værdier for modstand og kondensator i oscillator-leddet:

$$R = 10 \text{ k}\Omega \quad \text{og} \quad C = 150 \text{ pF}$$

Dette giver en clock-frekvens på:

$$f_{CLK} \cong \frac{1}{1,1 \cdot R \cdot C} \cong 606,1 \text{ kHz}$$

Og en konverteringstid på:

$$T_C = 73 \cdot \frac{1}{606,1 \text{ kHz}} \cong 12 \text{ ms}$$

Med den valgte  $f_{CLK}$ , er kravene til konverteringstiden overholdt, og bit-fejlraten holdes samtidig under  $\pm 0.5$  LSB.

ADC0803 har en opløsning på 8 bit, og kan konvertere spændinger mellem 0 og 5 V.

## Indgangstrin

### Løsning

Ud fra kravet til indgangstrinnet, skal der sørges for, at signalet får et DC-offset, som svarer til det halve af forsyningsspændingen til ADC'en. Derudover skal indgangstrinnet bruges som DC-afkobling for signalet modtaget fra analog sensor kredsløb.

For at opfylde disse krav implementeres der et højpas-filter. Højpas-filteret skal opfylde følgende krav for at passe ind i systemet. Forstærkningen skal være 1 og dæmpningen ved 0,2 Hz må maksimalt være 3dB.

Komponentværdierne til indgangstrinnet findes ved hjælp af Laplace-transformation og knudepunktsgligninger. Overføringsfunktionen  $T(s)$  for kredsløbet findes til

$$T(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{-R_2}{\frac{1}{C_1 \cdot s} + R_1}$$

Da der stilles krav om at filteret skal have en forstærkning på 1 sættes  $R_2=R_1=R$ . Dette betyder at overføringsfunktionen kan omskrives til et produkt af standardled:

$$T(s) = -\frac{\frac{1}{RC_1}}{\frac{1}{RC_1} + s} \cdot \frac{s}{\frac{1}{RC_1}}$$

Da 3 dB frekvensen skulle være 0,2 Hz kan der ud fra det fundne standard led, findes en modstandsværdi for R, som findes i E96-rækken.

$C_1$  sættes til 1  $\mu F$ .

$$R_1 = \frac{1}{2\pi \cdot 10^{-6} F \cdot 0,2 Hz} \approx 806 k\Omega$$

Den implementerede 3 dB frekvens beregnes med modstandsværdi fra E96-rækken:

$$f_{3dB} = 0,2 Hz = \frac{1}{2\pi \cdot 806 k\Omega \cdot 10^{-6} F_1} = 0,197 Hz$$

Et andet krav til op-amp'en er at den skal kunne give et output rail-to-rail. Derfor vælges der en MCP601.

### **Spændingsforsyning**

ADC'en er galvanisk adskilt fra lysnettet, og det er derfor valgt at forsyne denne del med et 9 V batteri. Da der kun skal bruges en spænding på 5 V, på henholdsvis ADC'en, indgangstrinnet og MCU'en, skal der anvendes et regulerings kredsløb.

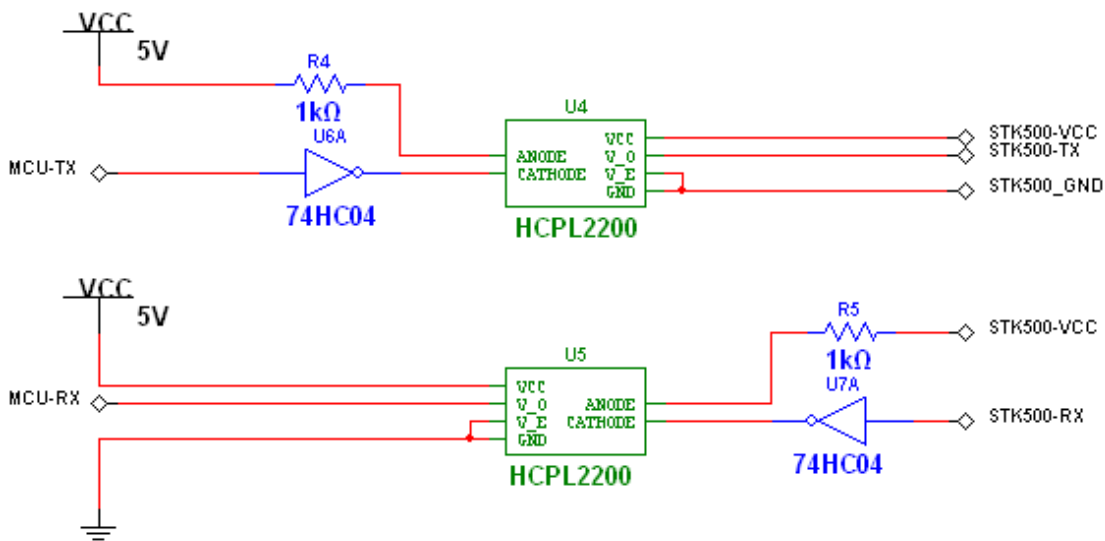
### **Løsning**

Ved hjælp af en MC7805 positive voltage regulator, implementeres der et passende kredsløb. Komponentværdierne findes ved at slå op i databladet for MC7805:

$C_1 = 330 \text{ nF}$

$C_2 = 1 \text{ }\mu\text{F}$

### 7.6.2.3. Galvanisk adskillelse



Figur 30 - Kredsløb diagram over galvanisk adskillelse

#### Beregninger/Beskrivelse

Til den galvaniske adskillelse bliver der brugt en HCPL-2200 optokobler sammen med en M74HC04B hex inverter.

For at dette kredsløb skal kunne virke, skal optokobleren og Hex inverteren kunne operere hurtigt nok for at kunne opfylde kravene i kravspecifikationen.

Ved at se i databladene for disse 2 komponenter, kan det nu beregnes om det samlede rise og fall time vil kunne holde sig inde for kravene.

Ved en Baud rate på 57600, må det samlede rise eller fall time max være:

$$T = \frac{1}{f} = \frac{1}{57600\text{Hz}} = 1,7 \cdot 10^{-5} \text{ sek}$$

Den samlede rise- eller fall-time i kredsløbet findes. På optokobleren er det 70 ns og for Hex inverteren er det 8 ns. Dette giver en samlet rise-time på ca. 78 ns, og fall-time er endnu kortere, og dermed er der ikke problemer med at overholde kravet om en Baud rate på 57600.

$$f = \frac{1}{78 \cdot 10^{-9} \text{ sek}} = 12,8 \text{ MHz}$$

Ifølge databladet for Mega32 MCU må komponenterne max have en rise time på 300 ns og en fall-time på 300 ns. Dette bliver også overholdt.

De resterende komponentværdier vælges på følgende måde.

R1 findes ved hjælp af diagrammet i databladet for typisk af HCPL-2200 optokobleren. Derudover er der anvendt afkobling.

Sikkerhedsmæssigt opfylder HCPL-2200 optokobleren også kravet. Da den ifølge databladet har en maksimum virkende isolering over for spænding, hvor følgende værdier er opgivet:  $V_{IORM}$  på 630 V<sub>peak</sub> og en højeste tilladelige overspænding (i 10 sek)  $V_{IOTM}$  på 6000 V. Herudover har den en isolerende modstand ved  $T_s$ ,  $V_{IO}=500$  V på  $\geq 10^9 \Omega$ .

Da der blev lavet print skete der dog en fejl, og ved et uheld er der blevet trukket baner under optokoblerne. Dette bevirker, at sikkerhedskravene reelt ikke er overholdt.

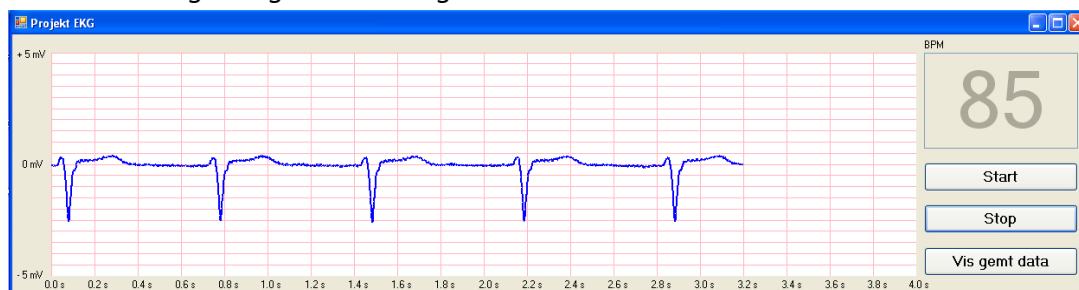
## 7.7. Resultater

Der er, i forbindelse med udviklingen af systemet, udført en række enhedstests af de specifikke software- og hardwareblokkes funktionaliteter. Resultater og detaljer om disse kan læses i projektdokumentationen. Derudover er der gennemført en afsluttende accepttest, der afgør om systemets funktionalitet og præcision lever op til kravene stillet i Kravspecifikationen. I de følgende afsnit vil de vigtigste resultater af accepttesten blive beskrevet, og en optagelse herfra vil derefter blive sammenlignet med en professionelt udført EKG-måling.

### 7.7.1. Accepttest

Det er lykkedes at konstruere et EKG-målingssystem med et tilfredsstillende resultat. Det målte EKG-signal tegnes på grafen som forventet, og alle use cases udføres som beskrevet i kravspecifikationen. Herunder ses et screenshot fra accepttesten, der viser udseendet af brugergrænsefladen og EKG-signalet.

*GUI med EKG-signal og BPM-måling:*



Figur 31 – Billede af EKG-måling

Et udsnit af testen af de funktionelle krav ses herunder:

*Use case 1, "Start/Stop program" hovedscenarie:*

Step	Handling	Forventet observation/resultat	Resultat/Kommentar
1	Brugeren trykker på "start" knap	Der tegnes en graf med EKG'et på skærmen og der vises BPM	OK
2	Brugeren trykker på "stop" knap	Systemet holder op med at tegne grafen og BPM opdatere ikke længere	OK

*Use case 2, "Vis gemt data" hovedscenarie:*

Step	Handling	Forventet observation/resultat	Resultat/Kommentar
1	Brugeren trykker på "Vis gemt data" knap	Systemet viser de sidste gemte data i et nyt vindue	OK
2	Brugeren trykker på "luk" knap	Vinduet lukkes og use case 2 afsluttes	OK



Der er også udført test af de ikke-funktionelle krav til målepræcision. Præcisionen af den viste spænding testes ved brug af en signalgenerator, der sender et givent signal ind i vores system. Et udsnit af de målte resultater ses herunder:

*Målepræcision:*

Input Signal - 2Hz sinus	Forventet observation/resultat	Observation/ Måle resultat	Resultat/Kommentar
1mV	1mV	1,1mV	OK
2mV	2mV	2,0mV	OK
3mV	3mV	3,2mV	OK
4mV	4mV	4,0mV	OK
5mV	5mV	5,1mV	OK
10mV	10mV	10mV	OK

Det ses, at alle målinger er indenfor de stillede krav til præcision. Ved yderligere test af andre signal-frekvenser, ses det at kravene er opfyldt i det ønskede frekvensområde på 0,5-150 Hz.

Præcisionen af BPM beregningerne testes ved at generere et sinus-signal, med en given frekvens, der kan omsættes til en bestemt værdi for BPM. Resultaterne ses herunder:

*BPM målepræcision:*

Frekvens[Hz]	Forventet observation/resultat	Observation/ Måle resultat	Resultat/Kommentar
0,1	0	0	OK
0,2	<15	<15	OK
0,3	18	<15	(OK)
0,4	24	25	OK
0,5	30	29-30	OK
1	60	58-60	OK
2	120	119-120	OK
3	180	176-182	OK
3,3	198	194->200	OK
3,4	>200	>200	OK

Det ses at BPM målingerne stemmer overens med der forventede i alle tilfælde med undtagelse af ved 0,3 Hz. Dette svarer dog til en puls på 18 og, ifølge vores kravspecifikation, garanteres der ikke for resultaterne uden for puls-området 20-200 BPM. BPM målingerne overholder kravet om, at det skal kunne måles med en præcision indenfor  $\pm 5\%$ .

De samlede resultater af accepttesten kan ses i accepttestdokumentet i projektdokumentationen.

Alt i alt opfylder systemet næsten alle krav stillet i kravspecifikationen, og dermed også kravene fra opgaveformuleringen. Den eneste undtagelse er, at den galvaniske adskillelse ikke er implementeret helt korrekt.

Et særligt tilfredsstillende resultat er omkring målepræcisionen, for det viste EKG-signal, da det i forvejen var forventet, at der ville blive målt meget støj.

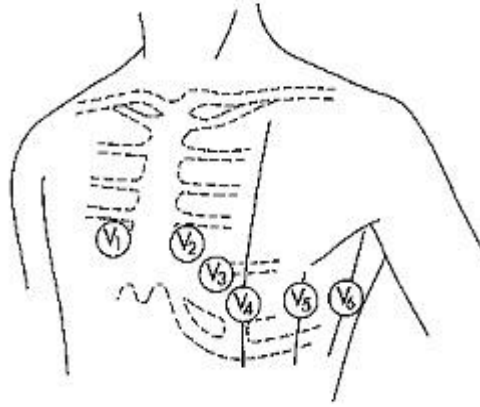
Resultatet indikerer, at de implementerede analoge og digitale filtre effektivt fjerner støj, samt at AD-konverteringen foregår korrekt både med hensyn til timing og skalering.

Målepræcisionen af BPM passer også godt, hvilket viser, at algoritmen til beregning af BPM virker helt efter hensigten og er meget præcis.

### 7.7.2. Sammenligning med professionelt optaget EKG

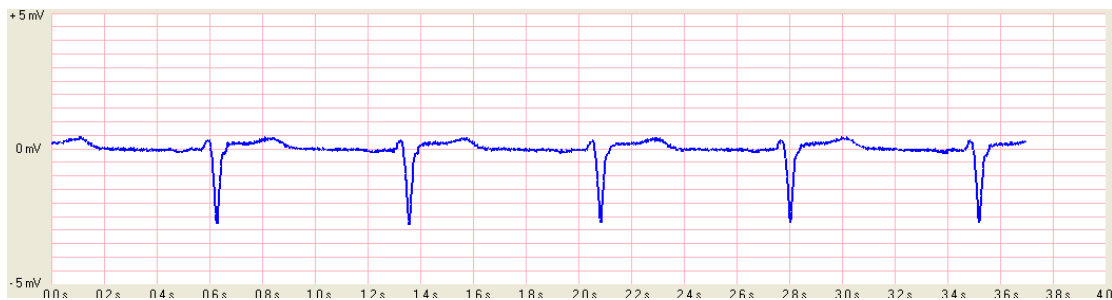
I det følgende afsnit er EKG-signalet, optaget fra accepttesten, sammenlignet med en EKG-måling foretaget af praktiserende læge, Bjarke Skov. Begge målinger er foretaget på Martin Nors Skov.

Der blev til accepttesten foretaget en EKG-måling med sensorerne placeret på punkterne V2 og V6 og med et referencepunkt på venstre arm.

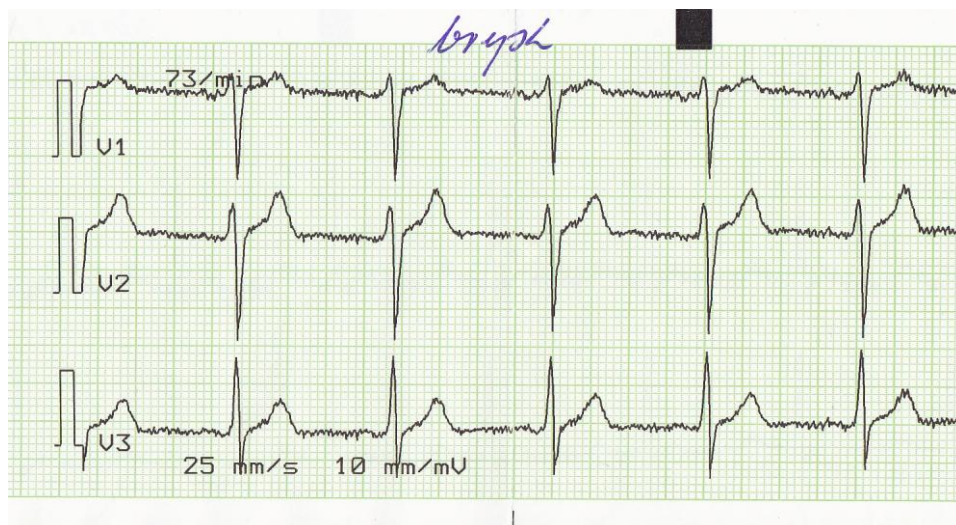


Figur 32 – Illustration af elektrodeplacering

Dette skulle give en måleafledning, svarende til det der ses som en V1 eller V2 måling fra den praktiserende læges EKG-måling. Da der, i forbindelse med accepttesten, kun er anvendt 3 elektroder, og disse er blevet placeret af en person uden lægefaglig baggrund, kan denne sammenligning kun bruges til at indikere om EKG-målingen er korrekt.



Figur 33 – EKG-signal målt i accepttesten



Figur 34 – Professionelt udført EKG-måling

Hvis man sammenligner længden af den længste tak på acceptestens afledninger og V1 afledningen, kan man se, at der ca. er 1mV til forskel. Dette kan bl.a. skyldes elektrodernes placering. Bjarke Skov har dog set et billede af EKG-målingen foretaget i forbindelse med acceptttesten, og han har vurderet, at det ville kunne bruges til diagnosticering.

EKG-målingen foretaget ved den praktiserende læge er taget med et 20 år gammelt analogt apparatur. Derfor observeres der en smule mere støj på denne måling end på accepttestens EKG-måling. Støjen der observeres er 50 Hz brum fra lysnettet.

## **7.8. Opnåede erfaringer**

### **7.8.1. Erfaring med udviklingsproces**

Gennem arbejdet er der opnået kendskab til, og tilegnet erfaring med, strukturering og arbejdsmetoder inden for tværfagligt projektarbejde. Mere specifikt er projektet blevet struktureret, så der er opnået erfaring med at arbejde efter en problemformulering og ud fra den danne en kravspecifikation. Derefter er der udarbejdet en systemarkitektur, der har muliggjort, at projektet kunne deles op i mindre ansvarsområder. Denne fordeling har gjort det muligt for alle gruppemedlemmer at kunne arbejde fokuseret inden for et ansvarsområde, hvilket sammenlagt har givet en effektiv udviklingsproces.

### **7.8.2. Iterativ proces**

Der er opnået erfaring med at lave et projekt efter en iterativ udviklingsmetode, omend der har været både fordele og ulemper ved dette. Den iterative metode gjorde, at der hurtigt kunne arbejdes frem, og at der kunne gøres brug af tilegnede erfaringer løbende. Et eksempel:

I første iteration var det planlagt at al filtrering skulle ske analogt, men med de tidlige erfaringer og tilegnet viden fra semestrets fag, kunne denne filtrering med fordel deles op i et analogt filter samt 2 digitale filtre.

### **7.8.3. Fumlebræt**

Der er opnået erfaring med, at der på en forsøgsopstilling på fumlebrædt over tid kan opstå løse forbindelser. Dette kan fremover løses ved tidligere at flytte designet til print eller veroboard, sådan at der opnås stabile kredsløb, der giver mere pålidelige målinger fra gang til gang.

### **7.8.4. Sampling af kontinuerte signaler**

Når der skal udføres større digital signalbehandling, skal der bruges en digital signalprocessor, som kan køre med en højere clockfrekvens og samtidig har et minimeret instruktionssæt, der er beregnet til at udføre sådanne beregninger effektivt. Den anvendte MCU i dette projekt, atmega32, er ikke specielt egnet til dette, hvilket har begrænset antallet af beregninger, der kunne lægges i denne del.

#### 7.8.5. MCU-programmering

På MCU'en var der udfordringer mht. timingen mellem de forskellige klasser og brugen af processoren. Tidlige udgaver havde nogle problemer med at frekvensen for kommunikationen med pc'en faldt betydeligt under 500 værdier per sekund og at kommunikationsklassen sløvede systemet ned. Problemet var, at når der skulle sendes til pc'en, så ventede MCU'en på at alle karakterer i beskeden var sendt, før den var klar til at lave noget nyt.

#### 7.8.6. Flertrådet programmering og events

På pc'en var der udfordringer mht. multi-threading, hvor det viste sig, at det var nødvendigt at bruge invoke-kald for at opdatere GUI. Events har også været en udfordring, da de viste sig uundværlige, når der skal kommunikeres fra lavere-niveau klasser til højere.

#### 7.9. Udviklingsværktøjer

I projektet er der anvendt følgende værktøjer til udvikling af systemet:

- Microsoft Visual Studio 2010: Anvendt til skrivning, kompilering, test og debugging af C++-kode.
- Microsoft Visio 2010: Anvendt til tegning af UML-diagrammer.
- Microsoft Word 2007 og 2010: Anvendt til skrivning af rapport og dokumentation.
- Atmel AVR STK500: Development toolkit anvendt som platform for den anvendte MCU-chip, Atmel Mega32.
- MultiSim 10: Anvendt til simulering af filter.
- Matlab R2010: Til filter beregning.
- FilterPro 3.1.0: Anvendt til design af filter/forstærkning.
- Mathcad 14.0: Anvendt til beregninger i analyse.
- ScyPy: Til signalbehandling.
- Python: Til udvikling af algoritme til beregning af BPM.

## 8. Konklusion

Det er lykkedes at strukturere og følge et projektforsløb. Projektet har gennemløbet faserne med udarbejdelse af opgaveformulering, kravspecifikation, systemarkitektur, analyse, design, implementering og test. Den overordnede strukturering har været velfungerende, hvilket har givet stor frihed til arbejde på individuelle ansvarsområder, hvor det kun har været nødvendigt at lave mindre ændringer i forhold til det overordnede design.

Det er ikke lykkedes at gennemføre den planlagte 2. iteration, da det viste sig at være for omfattende inden for tidsrammen. Dog har den iterative udviklingsproces klarlagt problemstillinger, omkring implementering, tidligt i forløbet, hvilket har givet mulighed for at lave rettidige ændringer i tidsplan og design.

Generelt er projektet vellykket. Det har været muligt at opfylde samtlige krav stillet i kravspecifikationen, med undtagelse af den galvaniske adskillelse. Der er dermed implementeret et EKG-målesystem, som ville kunne videreudvikles til et færdigt produkt.

En særlig vellykket del af projektet, er EKG-systemets gennemgående filtrering af det målte EKG-signal. Resultatet er et vist signal, hvor det er muligt at identificere P, Q, R, S og T takke af et EKG-signal, hvilket ikke var et af de oprindelige krav. Kvaliteten af signalet er blevet vurderet af praktiserende læge Bjarke Skov, som har vurderet at EKG-målingen ville kunne bruges til diagnosticering.

## 9. Referencer

Der henvises til litteraturlisten i bilagssamlingen anført i projektdokumentationen.