



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

BACHELOROPPGAVE

Studieprogram/spesialisering:	Vårsemesteret 2023
Bachelor i ingeniørfag / Automatisering og elektronikkdesign	Åpen eller Konfidensiell Åpen
Forfatter(e): Hassen Bekit Mohamedali	
Fagansvarlig: Morteza Esmaeili	
Veiledere: Morteza Esmaeili og Ståle Freyar	
Tittel på bacheloroppgaven: Kretsutvikling for elektrokardiografi og signalbehandling	
Engelsk tittel: Circuit development for electrocardiography and signal processing	
Studiepoeng: 20	
Emneord: Kretsutvikling. AD-omformer. Forsterker. Filter. C-programmering. TFT-skjerm	Sidetail: 59 oppgaver + 10 vedlegg/annet: Kode, Komponentlist, kretsskjema, Laboratoreit. Stavanger 15. mai 2021

Sammendrag

Denne bacheloroppgaven omhandler kretsutvikling og signalbehandling for elektrokardiografi (EKG) med fokus på måling av pulsraten ved bruk av et realiserende EKG-signal. EKG er et signal som beskriver hjertets elektriske aktivitet. Det registrerte EKG-signalet består av distinkte morfologiske trekk, inkludert en P-bølge som reflekterer atrie depolarisering, et QRS-kompleks som indikerer atrial repolarisering og ventrikulær depolarisering, og en T-bølge som skyldes ventrikulær repolarisering. Oppgaven begynner med en studie av hjertets elektriske systemer, etterfulgt av forsterkning og filtrering av EKG-signalet. Deretter blir signalet konvertert til digital form og utført digital signalbehandling. Til slutt presenteres den resulterende hjerteraten på en skjerm.

Generelt sett er oppgaven grundig gjennomført, og resultatene demonstrerer et vel fungerende system med potensial for videre forbedringer. Selv om systemet for måling av hjerterate ved hjelp av det realiserende EKG-signalet fungerer som forventet, er det fremdeles behov for ytterligere undersøkelser og analyser for å tilpasse det til praktiske formål. Gjennomføringen av oppgaven har vært utfordrende, men har også gitt en betydelig følelse av mestring innen flere områder. Denne oppgaven legger grunnlaget for videre utvikling innen elektrokardiografi og signalbehandling, og kan være et verdifullt bidrag til det medisinske feltet.

Forord

Denne oppgaven markerer avslutningen på et treårig studieløp i Automatisering og elektronikkdesign ved Universitetet i Stavanger. Oppgaven har vært utfordrende å gjennomføre som en enkelt person innenfor en begrenset tidsramme, men samtidig har den vært utrolig lærerik.

Jeg ønsker med dette å takke veileder Ståle Freyer for god veiledning og grundige tilbakemeldinger gjennom hele oppgaven. Din ekspertise og engasjement har vært en uvurderlig ressurs for meg, og jeg setter stor pris på den tiden og energien du har investert i å veilede meg.

Jeg vil også rette en takk til fagansvarlig Morteza Esmaeili for hans verdifulle råd og fleksibilitet i forbindelse med valget av oppgaven. Din innsikt og støtte har vært avgjørende for at jeg kunne fokusere på et tema som både interesserte meg og var relevant for studiet mitt.

Innhold

1	Introduksjon	1
2	Teori	3
2.1	Innledning	3
2.2	Hjertets elektriske system	3
2.3	Start av elektrisk impuls	4
2.4	EKG-registrering	4
2.5	EKG-signalkarakteristikk	5
2.6	Støy i EKG-signalet	6
2.6.1	Baselinjevandring	7
2.6.2	Interferens fra strømnettet	7
2.6.3	EMG støy	8
2.6.4	Elektrodebevegelsesartefaktstøy	8
2.7	Høyrebeinsdrift krets	9
3	Implementasjon av kretsen	10
3.1	Innledning	10
3.2	Metode	10
3.3	Maskinvareblokkdiagram av et EKG-system	11
3.4	Instrumenteringsforsterker	11
3.5	Båndpassfilter	13
3.5.1	Matematisk modellering av båndpassfilter	15
3.5.2	Praktisk utføring av båndpassfilteret ved et sinus-signal	17
3.5.3	Praktisk utføring av båndpassfilteret ved et EKG-signal	21
3.6	Kretsskjema av EKG-systemet	23
4	Analog til digital	24
4.1	Innledning	24
4.2	Analog til digital	24
4.3	Plattformkort	25
4.3.1	Mikrokontroller	26
4.3.2	UART	28
4.4	AD-omformer	30
4.4.1	AD-konfigurasjon	31
4.4.2	Timer konfigurasjon	34
4.4.3	Klokke konfigurasjon	36
4.4.4	Start og kalibrering av AD-omformeren.	36
4.4.5	Samplefrekvens	38

4.4.6	AD-Oppløsning	39
4.4.7	Presentasjon av digital EKG-signal	40
4.4.8	Resultat av AD-omformereren	41
5	Digital signalbehandlin	43
5.1	Innledning	43
5.2	R-bølge toppdeteksjon	43
5.3	R-bølge interval	48
5.4	Fesultat av pulsrate	49
6	Skjerm	50
6.1	Innledning	50
6.2	Skjerm	50
6.3	Maskinvareoppsett	51
6.4	Systemgrensesnitt	53
6.5	SPI kommunikasjon	54
6.6	Resultat av pulsrate på skjermen	54
7	Diskusjon	56
7.1	Forslag til forbedring	56
7.2	Konklusjon	57
A	Kode	58
B	Komponentliste	60
C	Oversikt av fysisk kretsskjema	61
D	Laboratoriet	62

Kapittel 1

Introduksjon

Denne bacheloroppgaven omhandler kretsutvikling og signalbehandling for elektrokardiografi (EKG). Elektrokardiografi er en medisinsk prosedyre som anvendes for å registrere den elektriske aktiviteten i hjertet. Metoden gir informasjon om hjertets tilstand og er nyttig ved ulike hjertesykdommer, inkludert hjerterytmeforstyrrelser, ledningsforstyrrelser i hjertet, endringer i hjertemuskulaturen og oksygenmangel. Resultatet av undersøkelsen er en grafisk representasjon kjent som et elektrokardiogram. Denne fremstillingen kan vises og leses av på en dataskjerm eller gjennom utskrift, og en kardiolog kan deretter vurdere hjertets tilstand basert på resultatene.

EKG består av tre hovedtyper: Holter EKG, hendelseovervåking EKG og diagnostisk EKG. Holter EKG brukes til 24-timers overvåking og har en signalbåndbredde på 40 Hz. Denne typen EKG brukes primært for å oppdage spesifikke sykdommer som ikke nødvendigvis krever høyt detaljnivå. Hendelseovervåking EKG brukes vanligvis på sykehus for kontinuerlig overvåking av kritiske pasienter, og det utløses alarmer ved eventuelle problemer. Denne typen EKG har en signalbåndbredde på 100 Hz. Diagnostisk EKG brukes for å oppdage et bredere spekter av kardiovaskulære sykdommer, og det krever større detaljnivå, det vil si større båndbredde på 150 Hz.

Etttersom EKG har flere ulike bruksområder, så skal denne oppgaven vil begrense seg til å måle hjerteraten ved hjelp av EKG. En normal hvilepuls for en gjennomsnittlig person ligger vanligvis mellom 50 og 80 slag per minutt [10]. Imidlertid kan hjerterpuls variere betydelig, og den kan ligge i et område fra 30 til 200 slag per minutt. I et slikt tilfelle vil grunnfrekvensen for EKG-målingen være mellom 0,67 Hz opp til 3,33 Hz. Av praktiske årsaker vil denne oppgaven imidlertid forholde seg til en bredere signalbåndbredde på 0,5 Hz til 10 Hz. Dette valget tas for å sikre tilstrekkelig informasjon om hjerterpuls og muligheten for å registrere eventuelle avvikende mønstre eller uregelmessigheter i pulssignalet.

Oppgaven er strukturert i fem kapitler. Kapittel 2 gir en generell kunnskap om hjertets elektriske system og dets funksjonalitet. Her blir beskrevet hvordan den elektriske impulsen oppstår, og hvordan registrering av EKG-signalet utføres. Kapittel 3 beskriver på implementeringen av kretsen og analyserer komponenter som instrumenteringsforsterker og båndpassfilter. I dette kapitlet undersøkes også frekvensresponsen til båndpassfilter. Kapittel 4 beskriver om digital signalbehandling, der en mikrokontroller brukes til å konvertere EKG-signalet fra analogt til digitalt. Kapittel 5 beskrives på digital signalbehandling, spesielt deteksjon av R-bølgen og beregning av intervallene mellom R-bølgene for å finne pulsraten. Til slutt, i kapittel 6, presenteres den resulterende pulsraten som ble funnet i kapittel 5, og hvordan den vises på skjermen.

I oppgaven har det oppstått flere utfordringer under gjennomføringen av undersøkelsen. Disse utfordringene har blitt nøye undersøkt og forsøkt løst. Enkelte av utfordringene krever betydelig tidsinnsats, og jeg har inkludert dem som forslag til forbedringer. Det er viktig at leseren tar med seg informasjonen om utfordringene og deres løsninger for å kunne følge rådetråden gjennom oppgaven.

Kapittel 2

Teori

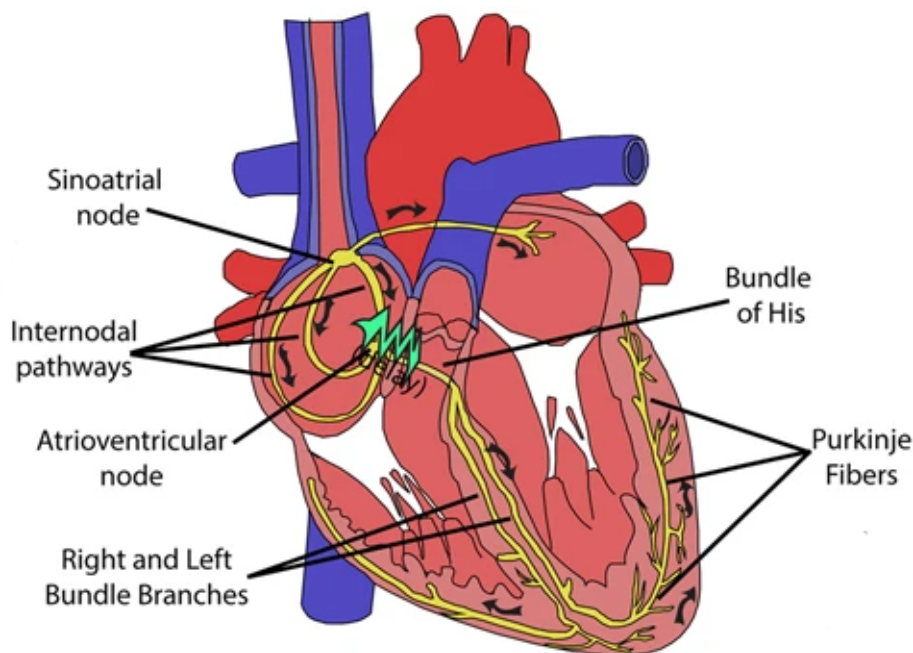
2.1 Innledning

Teori delen av denne oppgaven vil omhandle en generell kunnskap om hjertes elektriske system og dens funksjonalitet, hvordan den elektriske impulsen starter, og hvordan EKG-signalet blir registrert.

2.2 Hjertets elektriske system

Hjertet er en muskel som kontinuerlig pumper blod gjennom kroppen. Et hjerteslag er en rytmisk sammentrekning av de fire kamrene i hjerte. Hvert hjerteslag blir utløst av elektriske signaler som sprer seg gjennom hjertets eget nervesystem (**Figur 2.1**). Disse signalene kan oppdages og registreres ved hjelp av elektrokardiografi (EKG). Et EKG er en medisinsk teknologi som har utviklet for å detektere og beskrive den samlede elektrisk aktiviteten fra alle muskelcellene, målt på huden på brystkassen. Hjertet er delt inn i fire kammer, to kammer på høyre siden og to kammer på venstre siden. Disse kamrene er kjent som atrier (forkamre) og ventrikler (hertekamre), og deres hovedoppgave er å pumpe blodet ut i kretsløpet. Ved hvert hjerteslag fungerer kamrene som to pumper på samme side av hjertet. For å sikre optimal blodsirkulasjon og hjertefunksjon, er det nødvendig med et velkoordinert rytmesystem mellom atrier og ventrikler, som sikrer en effektiv pumpefunksjon. I tillegg er det viktig å ha kontroll over pulsfrekvensen, som refererer til hastigheten på hjerteslagene. Mekanismen som regulerer hjerteslagene, er avhengig av hjertets elektriske systemer [13].

2.3 Start av elektrisk impuls



Figur 2.1: Ledningsnettets og utbredelsen av den elektriske impulsen. Bildet er tilpasset fra [3] med tillatelse.

2.3 Start av elektrisk impuls

Det elektriske signalet i hjertet begynner i sino-atrial-knuten (SA-knuten). Dette er hjertes naturlige pacemaker¹. SA-knuten starter alle hjerteslag og bestemmer hjertefrekvensen. De elektriske impulser av SA-knuten sprer seg deretter gjennom hjertet, og når en impuls passerer trekker hjertemuskelfibrene seg sammen. Videre forsetter impulsen gjennom atrioventrikulærknuten (AV-Knuten). AV-knuten er plassert på den andre siden av høyre atrium, og fungerer som elektrisk port til ventriklene. Det forsinket strømming av elektriske impulser til ventriklene, denne forsinkelsen skyldes for å sikre at atriene har kastet ut alt blodet inn i ventriklene før ventriklene trekker seg sammen. AV-knuten mottar signalet og sender dem videre til His-buntens. Denne buntens deles inn i høyre og venstre buntgrener som leder impulsen i høyre og venstre ventrikkel [14].

2.4 EKG-registrering

Når en muskelcelle er i hvile, oppstår en elektrisk negativ ladning inne i cellemembranen i forhold til det ytre miljøet. Dette hvilepotensialet kan variere med opptil 90 millivolt [3] og blir referert til som cellens hvilepotensial. Denne polariseringen skyldes på grunn av fordeling av ioner, elektrisk ladde partikler, på innsiden og utsiden av membranen.

¹Impulsgenerator som stimulerer hjertet

2.5 EKG-signalkarakteristikk

Før hjertemuskelcellen trekker seg sammen, blir den omkringliggende membranen en kort stund elektrisk ladet. Når hjertemuskelcellen trekker seg sammen, fører det til en depolarisering av membranen, som resulterer i en endring i cellens membranpotensial. Depolarisering er en elektrofysiologisk prosess som refererer til en midlertidig endring i membranpotensialet til en celle. Denne prosessen tillater overskytende ioner å strømme gjennom membranen. Den resulterende oppbyggingen av ioner skaper et aksjonspotensial². Innsiden av cellemembranen blir da positivt elektrifisert i forhold til utsiden. Denne situasjonen varer noen få millisekunder.

Membranen repolariseres³ og hvilepotensialet gjenopprettes, noe som gjør cellen klar for en ny sammentrekning. De elektriske strømmene generert av milliarder av aktive hjerteceller kan registreres på overflaten av kroppen. Opptaket gjøres ved hjelp av elektroder festet til huden. Elektrodene leder de svake strømmene til en elektrokardiograf, som i utgangspunktet er en forsterker som overfører resultatene til en skriver. Den registrerte bølgeformen kalles et elektrokardiogram [3].

2.5 EKG-signalkarakteristikk

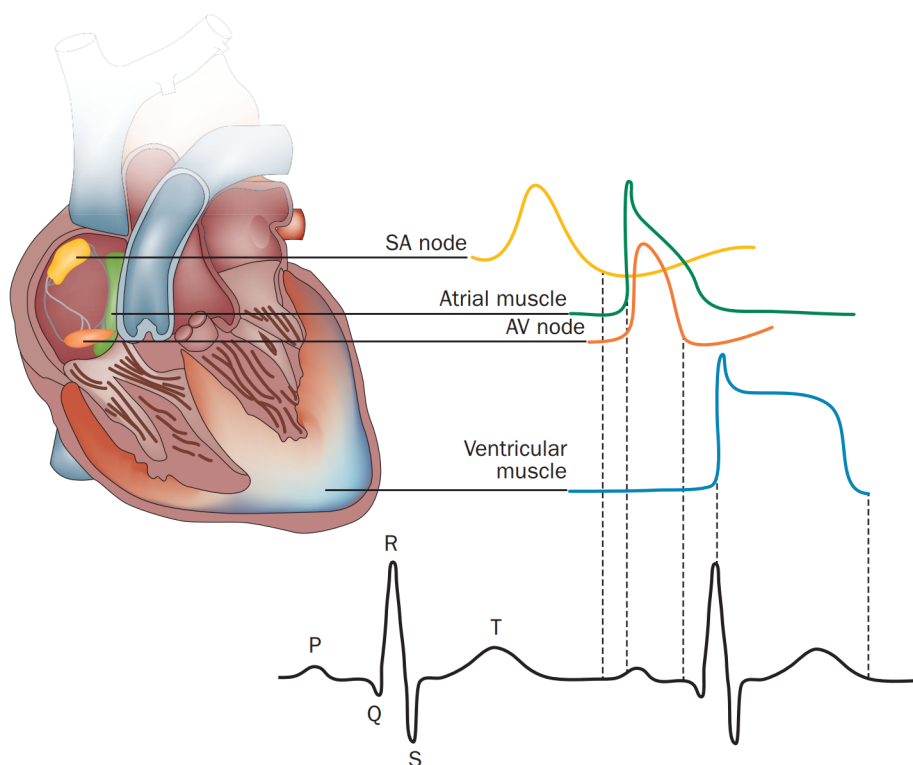
EKG-signalet har et typisk utseende der utgangspunktet er en flat elektriske linje med karakteristiske utslag. Disse utslagene har fått bokstavbenevnelser (**Figur 2.2**). En elektrisk impuls gjennom hjerte gir følgende utslag:

- P-bølgen er et utslag som beskriver hvordan impulsen brer seg i atriene.
- QRS-bølgen er et større utslag som beskriver hvordan impulsen brer seg i ventriklene.
- T-bølgen er et mindre utslag som beskriver at hjertet gjenvinner normal elektriske stabilitet.

²Aksjonspotensial er elektriske potensialvariasjoner av noen millisekunders varighet som registreres fra nerve- og muskelceller under aktivite

³Repolarisering er en elektrofysiologisk prosess som refererer til tilbakegangen av membranpotensialet til en celle til sitt hvilemembranpotensial etter en depolarisering.

2.6 Støy i EKG-signalet



Figur 2.2: PQRST bølger. Bildet er tilpasset fra [5] med tillatelse.

Tabellen 2.1 viser amplituden og varigheten av EKG bølgeformintervall. Alle viktige intervaller i denne tabellen er innenfor normale områder.

komplekse	Intervall [s]	Amplitude [mV]	Frekvens [Hz]
<i>P</i>	0.08	0.25	5 - 30
<i>PR</i>	0.12 - 0.20		
<i>QRS</i>	0.06 - 0.12		8 - 50
<i>ST</i>	0.05 - 0.15		
<i>T</i>		0.1 - 0.5	
<i>R</i>		0.5 - 1.0	0 - 10

Tabell 2.1: Normale EKG-verdier for bølger og intervaller[17, 6]

2.6 Støy i EKG-signalet

EKG-signalet er et støyutsatt signal, og det er noe man bør ta hensyn til. Identifikasjon av et representativt EKG-signalet kan bli påvirket av støy og artefakter⁴. Disse er beskrevet som forstyrrende signaler som er et resultat av alt som ikke tilhører den elektriske aktiviteten som genereres av hjertet. Kildene til disse artefaktene kan være fysiologiske, som muskelaktivitet eller hudbevegelser, eller ikke fysiologiske som følge av elektriske enheter i områder eller kan være feil bruk av utstyret. Her skal man se på de 4 mest vanlige støy signaler som forstyrrer EKG-signalet. Disse forstyrrende

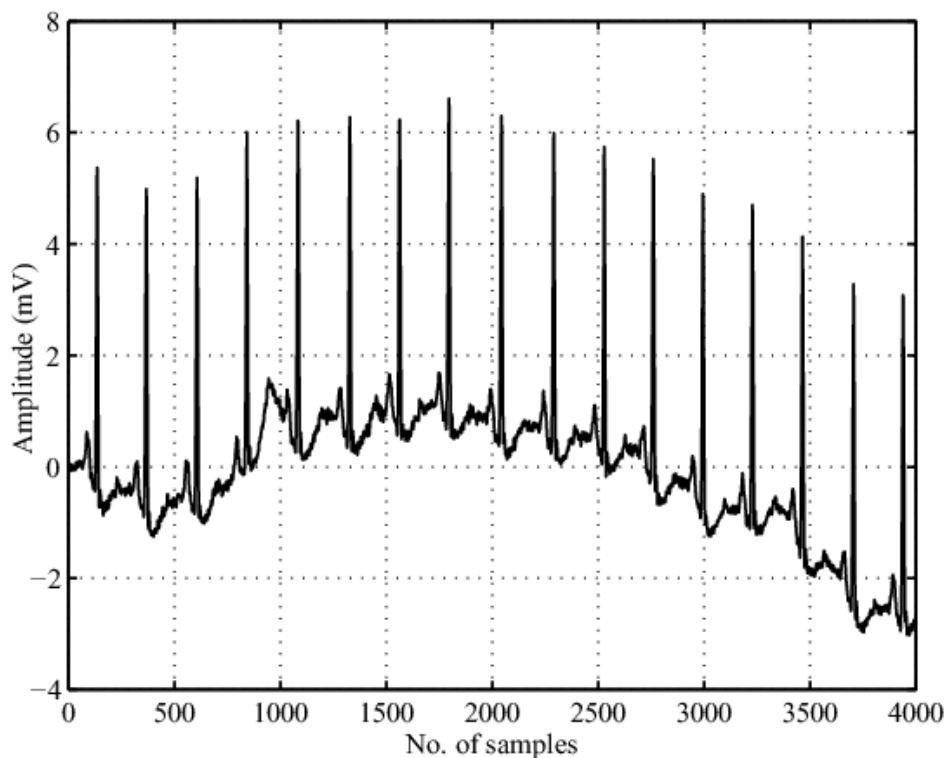
⁴Ordet brukes for eksempel om måleresultater som skyldes feil ved apparatet, metodefeil eller lignende. For eksempel hvis et EKG signalet ikke representerer en reell signal, kalles en artefakt.

2.6 Støy i EKG-signalet

signaler er baselinjevandring, interferens fra strømnettet, elektromyografisk (EMG) og elektrodebevegelsesartefakt.

2.6.1 Baselinjevandring

Baselinjevandring eller baselinjedrift er den effekten når EKG-signalet i x-akse av signalet begynner å vandre eller beveger seg opp eller ned enn å forholde EKG-signalet på rettlinje. Som det er vist tidligere i **Figur 2.3** så har EKG-signalet en karakterstisk baselinje. Ved en baselinjevandring i en EKG signalet kan føre til endring fra det normale karakterstisk baselinjen. Vandringen forårsaket på grunn av feil på elektrodene, pasientensbevegelse og pust. Baselinjevandring frekvensinnholdet er i området på 0.5 Hz. Frekvensinnholdet av baselinjevandringen øker imidlertid, når kroppen beveger seg under trening eller stresstest. Dermed er det nødvendig å filtrere vekk baselinjevandringen i EKG-signalet for å opprettholde baselinjen. Da baselinjevandring er lav frekvenssignal, derfor er det hensiktsmessig å bruke en høypass filter med en knekkfrekvens på 0.5 Hz for å filtrere vekk støyet [2].



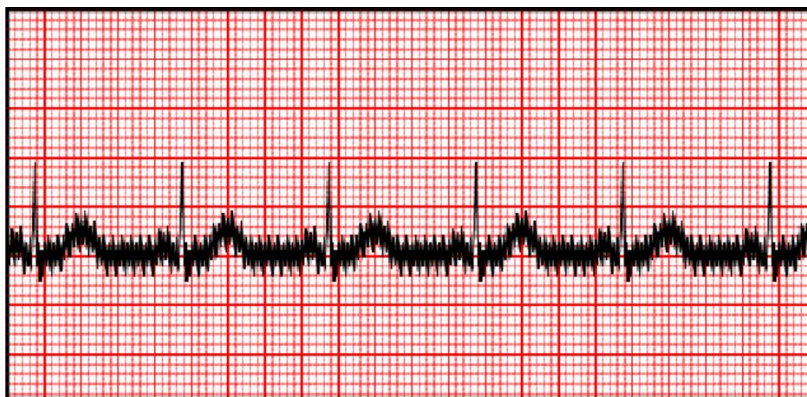
Figur 2.3: Eksempel av en EKG-signal påvirket av baselinjevandring [2] (fra en artikkel med Open Access Licence - Attribution CC BY).

2.6.2 Interferens fra strømnettet

De elektromagnetiske feltene som genereres av strømnettet er en vanlig kilde til støy i EKG-signalet, og andre bioelektriske signaler registrert fra kroppsoverflaten. Denne type støy er kjennetegnet som sinusformet interferens med en frekvens på 50 Hz (i Europa) eller 60 Hz (i USA). Denne forstyrrelsen av støyet gjør det vanskelig å

2.6 Støy i EKG-signalet

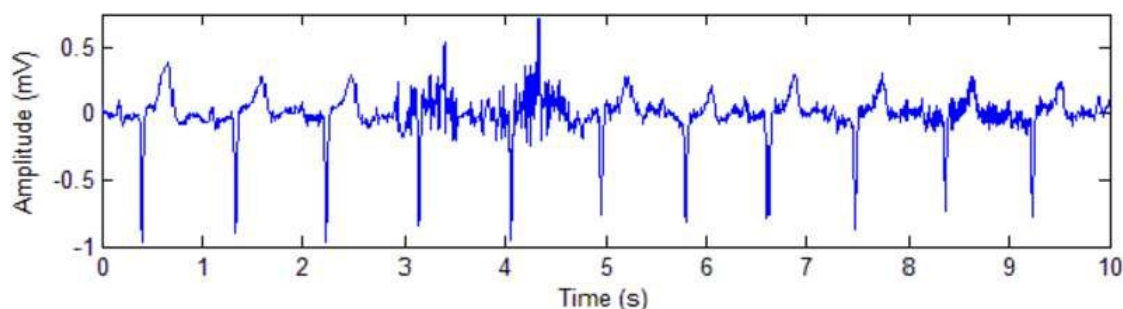
analysere og tolke EKG-signalet. Det er nødvendig å eliminere støyet som oppstår i EKG-signalet fra strømmettet, ettersom denne forstyrrelsen legger seg over de lavfrekvente EKG-bølgene, som for eksempel P-bølgen og T-bølgen (**Figur 2.4**). Dette netstøyet filtreres vekk med å bruke et notch filter [2].



Figur 2.4: Eksempel av en EKG-signal påvirket av strømmettet [2] (fra en artikkel med Open Access Licence - Attribution CC BY).

2.6.3 EMG støy

Elektromyografi (EMG) er en metode som registrerer musklens bioelektriske aktivitet. Den gir informasjon om de indre motoriske nevronene, den nevro-muskulære overgangen og muskelen [11]. Tilstedeværelsen av muskelstøyet representerer et stort problem i mange EKG-applikasjoner, foreksempel når pasienten beveger seg eller er under aktivitet. EMG frekvensrekken varierer mellom 50 til 150 Hz [11]. EMG-støyet forstyrrer innholdet i EKG-signalet, siden PQRST-komplekset blir overlappet av EMG-støyet, på annet ord av muskelaktivitet (**Figur 2.5**). Siden det er nødvendig å ha et støyfritt EKG-signal, så er det et krav å redusere støyet. Dette kan oppnås med å bruke et notch filter [2]. Figuren nedenfor viser et eksempel av EKG-signal som er påvirket av EMG-støy.



Figur 2.5: Eksempel av en EKG-signal påvirket av EMG-støy [2] (fra en artikkel med Open Access Licence - Attribution CC BY).

2.6.4 Elektrodebevegelsesartefaktstøy

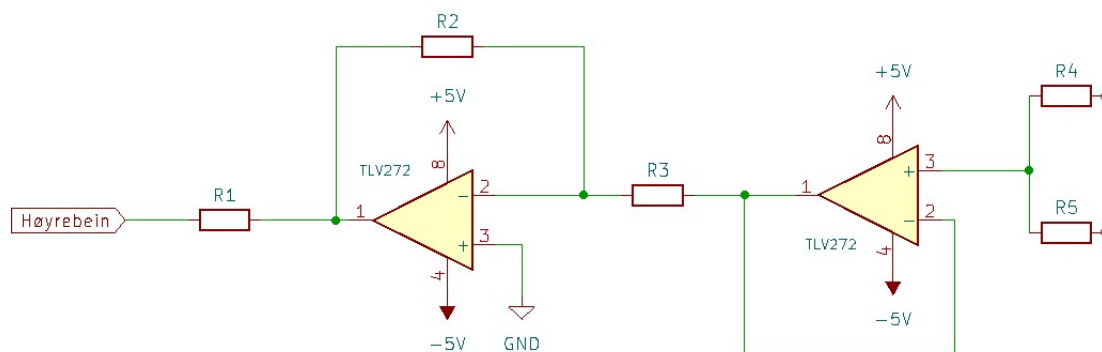
Elektrodebevegelsesartefaktstøy oppstår når det er uønskede forstyrrelser eller artefakter i EKG-signalet som skyldes bevegelse av elektrodene på pasientens hud. Dette

2.7 Høyrebeinsdrift krets

kan være forårsaket av flere faktorer, inkludert pasientens muskelaktivitet, feil plassering av elektrodene eller dårlig festing av elektrodene på kroppen. Når elektrodene beveger seg, for eksempel på grunn av pasientens bevegelser eller dårlig festing, kan det føre til forstyrrelser i det elektriske signalet som registreres under EKG-målingen. Dette kan resultere i støy i EKG-signalet, og det kan bli vanskelig å tolke og analysere EKG-signalet nøyaktig.

2.7 Høyrebeinsdrift krets

For å sikre pasienten fra strømnettet og samtidig redusere interferans fra forsterkeren kan en designe en høyrebeinsdrift krets. Høyrebeinsdrift kretsen inverterer og forsterker gjennomsnittlig *common mode* signal tilbake til pasientens høyre bein. Dette tiltaket reduserer effekten av nettstøy (50 Hz) og baselinjedrift. Jo mer forsterkning som kan brukes i tilbakekoblingsløkken, forbedrer også *common mode rejection ratio* CMRR. Å kansellere støy på denne måten reduserer behovet for demping fra CMRR i instrumentforsterkeren. Mer om CMRR kommer vi tilbake i neste kapittel. En høyrebeinsdrift krets fungerer også som en beskyttelseskrets mot overstrøm til kroppen. Den består av et buffertrinn for å unngå belastning på instrumentforsterkerens interne kretser, samt et inverterende forsterkertrinn for negativ forsterkning av signalet. Ved utgangen er høyrebeinsdriftkretsen koblet til en høy motstand, som primært har til hensikt å beskytte pasienten mot overstrøm fra strømnettet og under transiente forhold. **Figur 2.6** viser et eksempel på en høyrebeinsdriftkrets [18].



Figur 2.6: Eksempel av en høyrebeinsdrift krets. Kretsen er designet i KiCad.

Kapittel 3

Implementasjon av kretsen

3.1 Innledning

Dette kapittelet omhandler implementasjonen av EKG-kretsen med en kombinasjon av teoretiske og praktiske bruk. Fokuset ligger på å presentere en grundig og omfattende beskrivelse av metoder og prosedyrer som brukes for å forsterke det svake EKG-signalet til en brukbar spenningsignal ved hjelp av en instrumenteringsforsterker. Videre blir signalet filtrert gjennom et båndpassfilter for å oppnå en ønsket frekvensrespons.

3.2 Metode

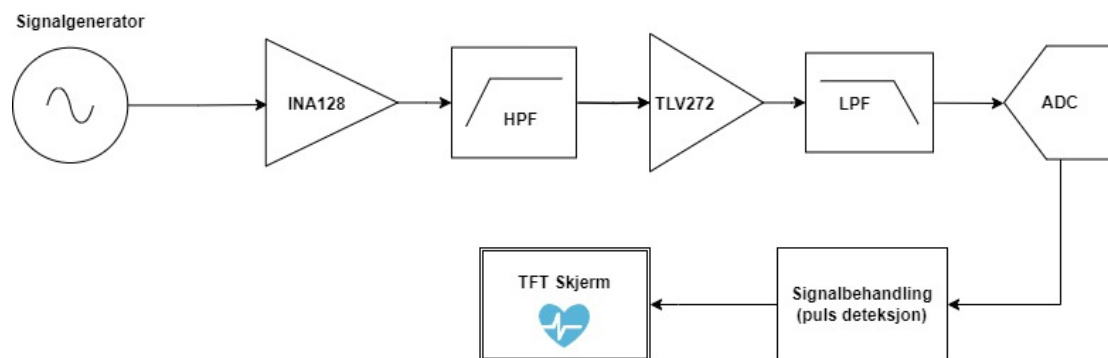
Forskrifter om medisinsk utstyr har som formål å sikre at medisinsk utstyr ikke utgjør noen fare for pasienter, brukere og eventuelt andre personers sikkerhet i forbindelse med produksjon, konstruksjon, omsetning og bruk av slikt utstyr [9]. På grunn av disse forskriftene er implementasjonen av denne kretsen ikke egnet for medisinsk bruk, da den ikke oppfyller de nødvendige kravene. Derfor skal kretsen unngå å bli brukt til fysiske målinger på mennesker, for å sikre at sikkerhetsstandardene i medisinsk utstyr blir ivaretatt. Som et alternativ oppstod ideen om å benytte et oscilloskop for å realisere EKG-signalet, ettersom det kan demonstrere og generere en rekke forskjellige signaler, inkludert et EKG-signal. Det betyr inngangsspenning på instrumenteringsforsterker er 20 mV i dette tilfellet.

Oscilloskopet som ble brukt er [MSO-X 3012A](#) produsert av *KEYSIGHT*. Dette oscilloskopet har en utgangsspenningsgrense på 20 mV. Siden EKG-spenningen normalt ligger mellom 0,5 mV og 1 mV, ble det forsøkt å bruke en spenningsdeler for å redusere utgangssignalet fra oscilloskopet. På grunn av oscilloskopets spenningsgrense var det imidlertid ikke mulig å måle det reduserte utgangssignalet fra spenningsdeleren. Derfor ble det bestemt å beholde en utgangsspenning på 20 mV, da dette er den laveste spenningen oscilloskopet kan levere.

3.3 Maskinvareblokkdiagram av et EKG-system

Oscilloskopet genererer et rensert EKG-signal med en spenningsreferanse på 20 mV. Dette signalet forsterkes deretter av en instrumenteringsforsterker (*INA128*), til en spenning på 2,5 volt. Deretter filtreres signalet ved hjelp av et båndpassfilter for å oppnå ønsket båndbredde på 0,5-10 Hz. Dette båndpassfilteret er konstruert ved å kombinere et høypassfilter og en forsterker (*TLV272*), samt et lavpassfilter.

Etter filtreringen blir det analoge EKG-signalet konvertert til digital form ved hjelp av en STM32-mikrokontroller for videre signalbehandling. Signalbehandlingen har som formål å detektere R-bølgen. Når R-bølgen er identifisert, kan en beregne puls-raten. Pulsraten vil bli vist på en *TFT-skjerm* for visuell presentasjon. **Figur 3.1** viser blokkdiagram for hele EKG-systemet.



Figur 3.1: Blokkdiagram for hele EKG-systemet.

3.4 Instrumenteringsforsterker

En Instrumenteringsforsterker, er en elektronisk enhet som brukes til å forsterke differansen mellom to inngangssignaler, samtidig som den undertrykker signaler som er felles for begge innganger. Dette oppnås ved å eliminere signaler med samme potensial på begge inngangene. Instrumentasjonsforsterkere er spesielt nyttige når det gjelder å forsterke svært små signaler fra transdusere og andre kilder. De brukes mye i medisinsk utstyr, som for eksempel i EKG. De bidrar til å forsterke og isolere det svake EKG-signalet fra omgivelses støy og andre elektriske forstyrrelser.

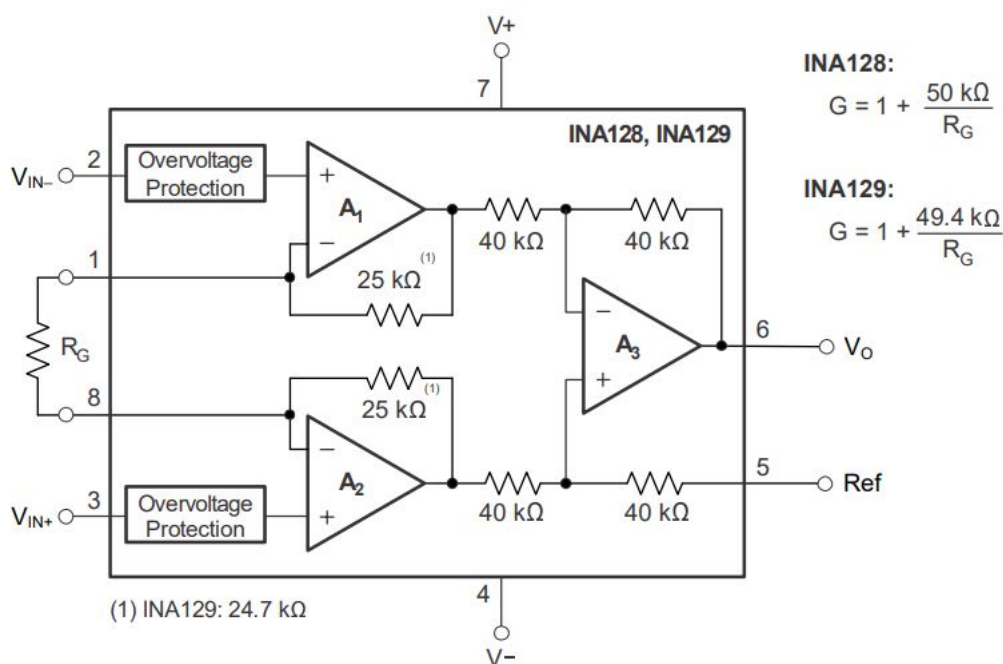
Når en ønsker å forsterke det svake EKG-signalet fra hjertet, er det avgjørende å bruke en instrumenteringsforsterker med spesifikke egenskaper. En eksempel på en slik forsterker er *INA128*, som kan finnes i dokumentet [INA128](#) (**Figur 3.2**). Det som gjør denne type forsterker spesiell, er dens høye forsterkning (gain), som kan variere fra 1 til 10000 V/V, og dens høye *common-mode rejection ratio* (CMRR) på 120 dB.

CMRR er evnen til å undertrykke signaler som har samme potensial på begge innganger samtidig som den forsterker differensielle signaler. For at instrumenteringsforsterker skal fungere effektivt, må den være i stand til å forsterke mikrovolt-nivå

3.4 Instrumenteringsforsterker

signaler samtidig som den undertrykker CMRR-spenningen på inngangen. Det er også viktig at den er i stand til å opprettholde høy CMRR over det relevante båndbreddområdet. Dette krever en instrumenteringsforsterker med svært høy CMRR over hele frekvensområdet av interesse.

Siden forsterkeren skal håndtere svært lave inngangsspenninger, er det viktig at den ikke inkluderer sitt eget støy til det signalet som skal forsterkes. Dette betyr at forsterkeren må ha lav støyegenskaper for å sikre at den ikke forvrenger det svake EKG-signalet. Forsterkeren **INA128** har veldig lav støynivå i de fleste bruksområder. Det lavfrekvente støyet ligger vanligvis i området 0,1 til 10 Hz når forsterkningen (gain) er større enn 100 [1]. Dette betyr at forsterkeren gir minimal støyforvrengning i det lave frekvensområdet, noe som er viktig for å opprettholde signalkvaliteten og nøyaktigheten i det forsterket EKG-signalet [16].



Figur 3.2: Forenklet skjema av INA128 [16].

forkortelsen G_I står for gain for IN-amp.

$$G_I = \frac{2.5V}{20mA} = 125 \quad (3.1)$$

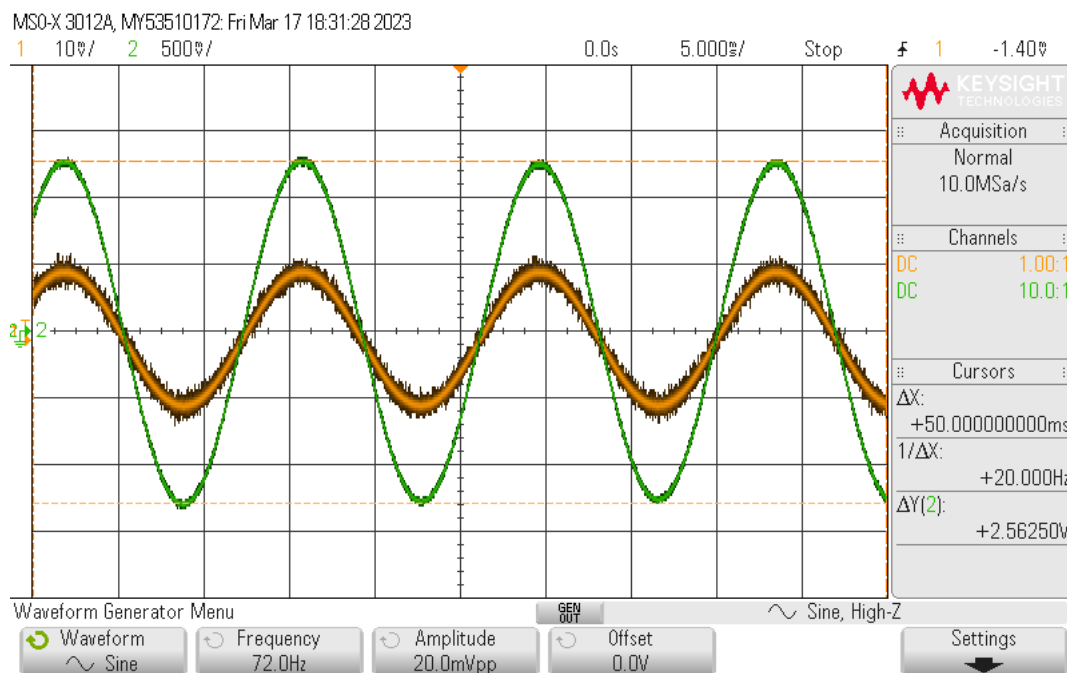
forkortelsen R_G står for gain motstand.

$$R_G = 1 + \frac{50K\Omega}{125} = 400\Omega \quad (3.2)$$

Forsterkeren ble testet ved å påføre et sinus-signal med en amplitud på 20 mV og frekvens på 72 Hz, og ved å variere verdien av motstanden R_G ble forsterkningsfaktoren G_I bestemt. Det er ønskelig at forsterkeren forsterke signalet fra 20 mV til 2.5 volt, som er ønskelig spenningsområdet for videre behandling.

3.5 Båndpassfilter

Resultatet av instrumenteringsforsterker viser inngangsspenningen som en oransje kurve med en spenningsamplitude på 20 mV generert av oscilloskopet (**Figur 3.3**). Etter forsterkningen blir utgangsspenningen en grønn sinusformet kurve med en spenningsamplitude på 2,56 V. Dette samsvarer med forventningene, da instrumenteringsforsterkeren er designet for å forsterke det svake inngangssignalet til et mer målbart og anvendbart nivå. Den observerte spenningsamplituden på 2,56 V bekrefter at forsterkeren fungerer som forventet og gir den ønskede forsterkningen av EKG-signalet.



Figur 3.3: Resultat av INA128-forsterkeren.

3.5 Båndpassfilter

Et aktiv ikke-inverterende båndpassfilter har blitt valgt for å fjerne baselinjedrift, elektrodepotensiale, og redusere folding. Et aktivt båndpassfilter er et båndpassfilter som gir forsterkning til inngangssignalet. Grunnet det ble valgt ikke-inverterende båndpassfilter er at inngangssignalet og utgangssignalet kan være i fase med hverandre. Dette filteret vil slippe gjennom en spesifisert frekvensbåndbredde, mens det filtrerer ut frekvenser under og over dette båndet. Båndpassfilteret er konstruert ved å kaskadekoble et høypassfilter, lavpassfilter, og en operasjonsforsterker. Som tidligere beskrevet i introduksjonen, er oppgaven begrenset til en båndbredde på 0.5 - 10 Hz. Dette betyr båndpassfilteret er konstruert for å fjerne uønskede frekvenser som er under 0.5 Hz og over 10 Hz. Filteret har en bandstopp på +/- 20 dB/Decade. Årsaken til valget av et aktivt båndpassfilter skyldes dets fordeler.

Fordeler med aktive filtre:

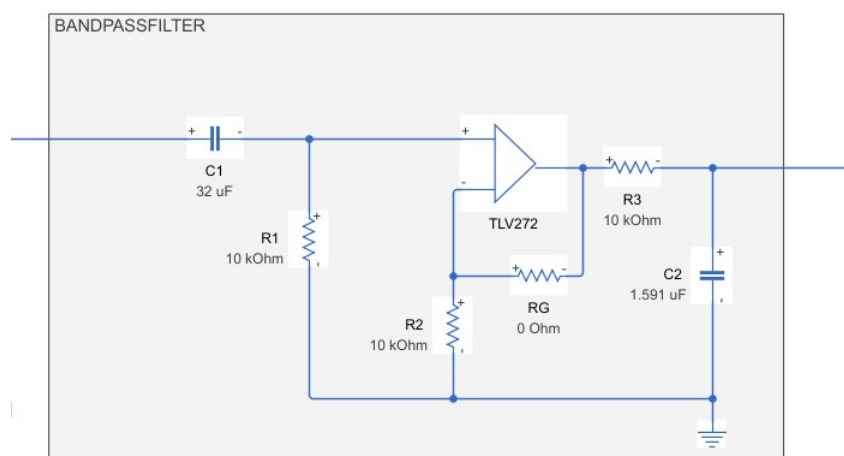
- Ingen resonansproblemer.
- Ingen lasteproblemer.

3.5 Båndpassfilter

- Kan eliminere alle harmoniske.
- Mer ansvarlig enn passive filtre
- Null innsettingstap

Ulemper med aktive filtre:

- De kan være dyre.
- De har et komplekst kontrollsystem..
- Begrenset frekvensområde.
- Krever DC strømforsyning.



Figur 3.4: Krettskjema av båndpassfilter. Kondensator C2 er regnet ut til å være 1.59 uF, men forholdet seg til en standard verdi på 1.8 uF. Kretsen er laget i Simulink

Båndpassfilteret i **Figur 3.4** bearbeider EKG-signalet på følgende måte:

1. først signalet passerer gjennom høypassfilteret, som tillater signaler med frekvenser over 0,5 Hz å passere, samtidig som det demper eller stopper signaler med lavere frekvenser.
2. Deretter går signalet videre til operasjonsforsterker, som forsterker signalet i henhold til formel 3.3. I denne deloppgaven ønsker vi imidlertid ikke å forsterke signalet, for å kunne analysere forholdet mellom inngang og utgangssignalet. Dermed motstanden RG er satt til null.
3. Til slutt går signalet gjennom lavpassfilteret, som tillater signaler med frekvenser under 10 Hz å passere, samtidig som det stopper signaler med høyere frekvenser.

Denne båndpassfilter kretsen er designet for å filtrere ut uønsket støy som ikke er relevante for EKG-signalet. Ved å bruke høypass- og lavpassfiltre begrenser vi båndbredden til det ønskede området (0,5-10 Hz).

3.5.1 Matematisk modellering av båndpassfilter

For å kunne analysere filteret matematisk og utlede overføringsfunksjonen, tar vi utgangspunkt i ønsket om at forsterkeren TLV272 har en forsterkning (gain) på $G_o = 1$. Dette valget er gjort for å kunne observere hvordan spenningsamplituden på filteret varierer ved ulike frekvenser. Derfor er verdien av motstanden R3 bestemt til å være null, slik at den ikke påvirker forsterkningen. Detaljert utledning av overføringsfunksjonen av filteret krever en grundig analyse av kretsens komponenter, som resistanser, kondensatorer og operasjonsforsterkere, og deres tilkoblingskjema. Videre vil det være nødvendig å anvende relevante kretsanalysemetoder, som Kirchhoffs lover og impedansanalyse, for å bestemme overføringsfunksjonen.

Forsterking(gain) for G_o (operasjonsforsterker TLV272):

$$G_o = 1 + \frac{RG}{R2} \quad (3.3)$$

Bestemme verdien for kondensator C1 med å velge motstanden R1 til å være $10K\Omega$

$$F_L = \frac{1}{2\pi R1C1} \Rightarrow C1 = \frac{1}{2\pi R1F_L} = \frac{1}{2\pi 10 \cdot 10^3 \cdot 0.5} = 32\mu F \quad (3.4)$$

Bestemme verdien for kondensator C2 med å velge motstanden R3 til å være $10K\Omega$

$$F_H = \frac{1}{2\pi R3C2} \Rightarrow C2 = \frac{1}{2\pi R3F_H} = \frac{1}{2\pi 10 \cdot 10^3 \cdot 10} = 1.59\mu F \quad (3.5)$$

F_L representerer for minste knekkfrekvens, og F_H representerer for høyeste knekkfrekvens.

Likning av overføringsfunksjon :

Med å analysere forholdet mellom inngang og utgangssignalet i kretsen i **Figur 3.4** kan man finne overføringsfunksjon av båndpassfilteret.

Utlede overføringsfunksjon av båndpassfilteret for høypassfilter delen:

$$H(f)_H = \frac{j\omega R1C1}{1 + j\omega R1C1} \quad (3.6)$$

Utlede overføringsfunksjon av båndpassfilteret for lavpassfilter delen:

$$H(f)_L = \frac{1}{1 + j\omega R3C2} \quad (3.7)$$

Utlede knekkfrekvensen i rad/sec

$$w_1 = \frac{1}{R_1 C_1} = 3, 14 \text{ rad/sec} \quad (3.8)$$

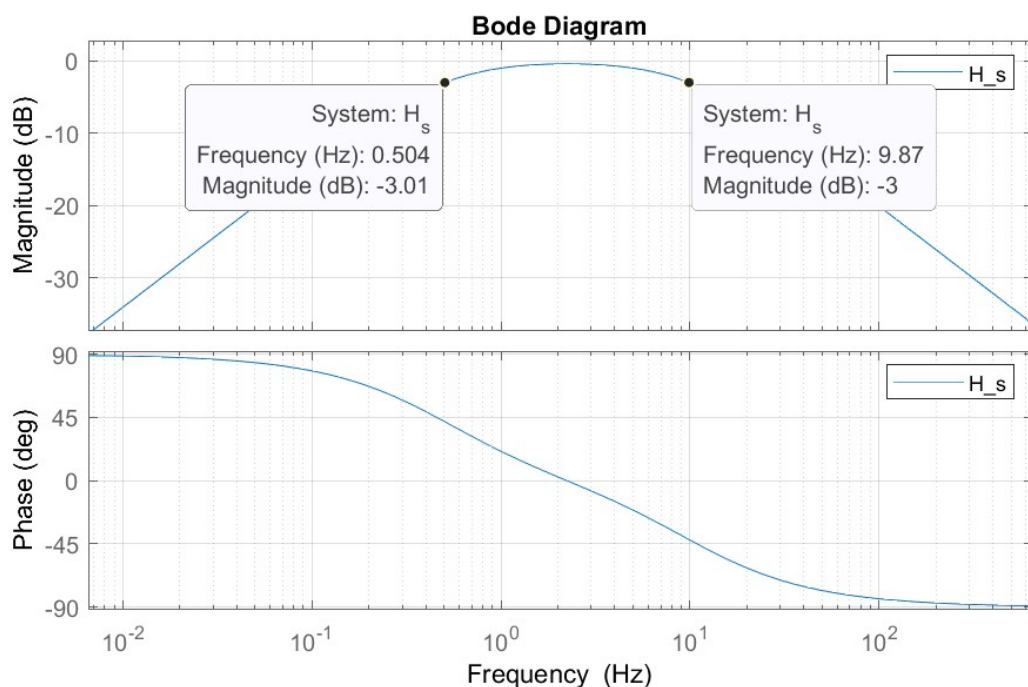
$$w_2 = \frac{1}{R_2 c_2} = 62, 5 \text{ rad/sec} \quad (3.9)$$

3.5 Båndpassfilter

Utleder overføringsfunksjon for båndpassfilteret:

$$\begin{aligned}
 H(s) &= G \cdot H(s)_H \cdot H(s)_L \\
 &= \left(1 + \frac{R_G}{R_2}\right) \cdot \left(\frac{sR_1C_1}{1 + sR_1C_1}\right) \cdot \left(\frac{1}{1 + sR_4C_2}\right) \\
 &= \frac{sR_4C_1}{(1 + sC_1R_1)(1 + sR_4C_2)} \\
 &= \frac{\frac{s}{\omega_1}}{\left(1 + \frac{s}{\omega_1}\right)\left(1 + \frac{s}{\omega_2}\right)} \\
 &= \frac{\frac{s}{3,14}}{\left(1 + \frac{s}{3,14}\right)\left(1 + \frac{s}{62,5}\right)} \\
 &= \frac{s62,5}{s^2 + s65,64 + 197}
 \end{aligned} \tag{3.10}$$

Resultatene for overføringsfunksjonen til båndpassfilteret er presentert i Bode-plottet, som vist i **Figur 3.5**. Ved å analysere figuren kan man observere at ved knekfrekvensen $F_L \approx 0.5$ Hz og $F_L \approx 10$ Hz, samsvarer avlesningene med de forventede frekvensverdiene i henhold til utledningene fra likningene 3.8 og 3.9. Dette indikerer at Bode-plottet bekrefter den forventede oppførselen til båndpassfilteret. Som en konklusjon kan det fastslås at båndpassfilteret er designet i tråd med de ønskede spesifikasjonene.

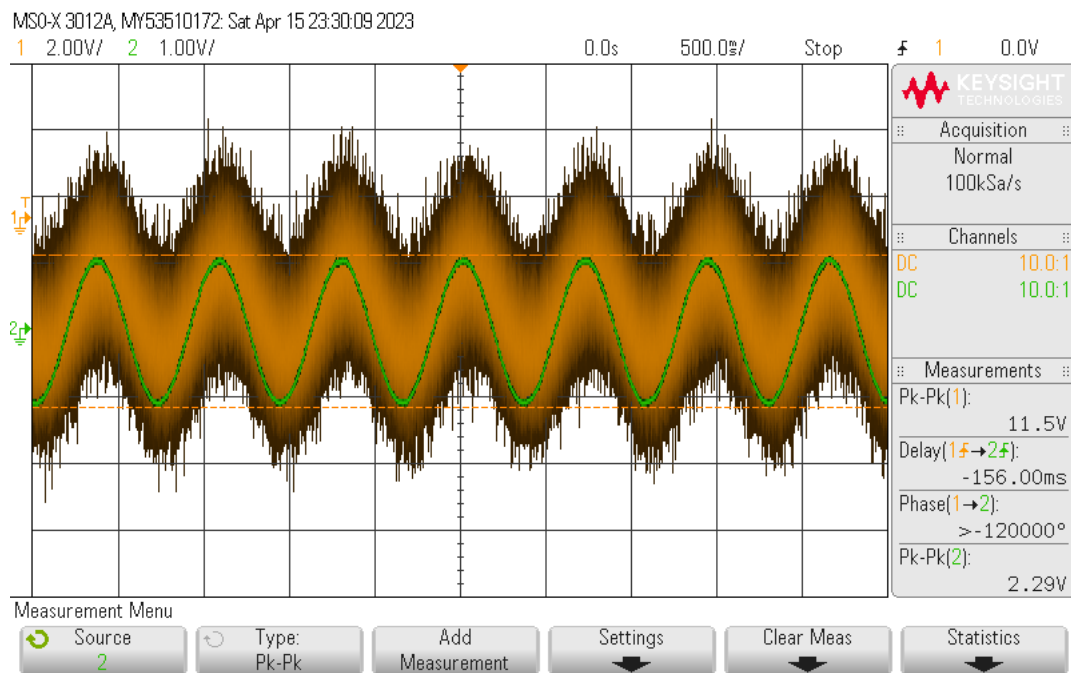


Figur 3.5: Ideelt sett Bode-plot av båndpassfilterets overføringsfunksjonen. Plottet er simulert i MATLAB (Version R2022a. Natick, Massachusetts: The MathWorks Inc.) ved å benytte formel 3.10.

3.5 Båndpassfilter

3.5.2 Praktisk utføring av båndpassfilteret ved et sinus-signal

Båndpassfilteret ble testet med et sinusformet inngangssignal ved å justere forskjellige frekvenser på oscilloskopet for å analysere responsen i virkeligheten. Valget av et sinusformet inngangssignal skyldes at det er enklere å analysere oppførselen til signalet. Under testingen ble det undersøkt nærmere faseforskyvninger, tidsforsinkelser og inngangs- og utgangsamplituder. Det ble også lagt til støy ved sinus-inngangssignalet for å absorbere båndpassfilter responsen. Støyet som er lagt til sinus-inngangssignalet er tilfeldig generert av oscilloskopet (**Figur 3.6**).



Figur 3.6: Resultat av båndpassfilter ved frekvens 10 Hz. Oransje sinus-signal er inngangssignalet ved støyet, grønn sinus-signal er filtrert utgangssignalet av båndpassfilter.

Tabellen 3.1 viser den praktiske analysen av båndpassfilteret. Analysen utført ved å endre forskjellige frekvensen på oscilloskopet. I tabellen er det vist verdier i grønn og rødt farge. Rødt farge indikere for laveste og høyeste knekkfrekvensen, mens grønn indikere for senterfrekvensen. I praktisk er det umulig å oppnå et ideelt båndpassfilter. Ved å se nærmere i **Tabel 3.1** ser man at filteret demper ikke alle frekvenser utenfor ønsket frekvensområdet fullstendig. Det er også en viss mengde uønsket signal som vil passere gjennom filteret, selv om det ikke ligger innfor det ønskede passbåndet. I tillegg vil filteret ha en viss faseendring eller forsinkelser i signalet innenfor passbåndet.

Det er forskjellige grunner til at båndpassfilteret ikke er ideell. En av grunnene er tap i filteret, som betyr at signalet som går gjennom vil bli redusert i intensitet. Dette skyldes blant annet resistive tap i kretsen. Annen grunn er at båndpassfilter vil ha en vis inngangsimpedans og utgangsimpedans, som kan påvirke signalet og

3.5 Båndpassfilter

belaste umiddelbart kretsen.

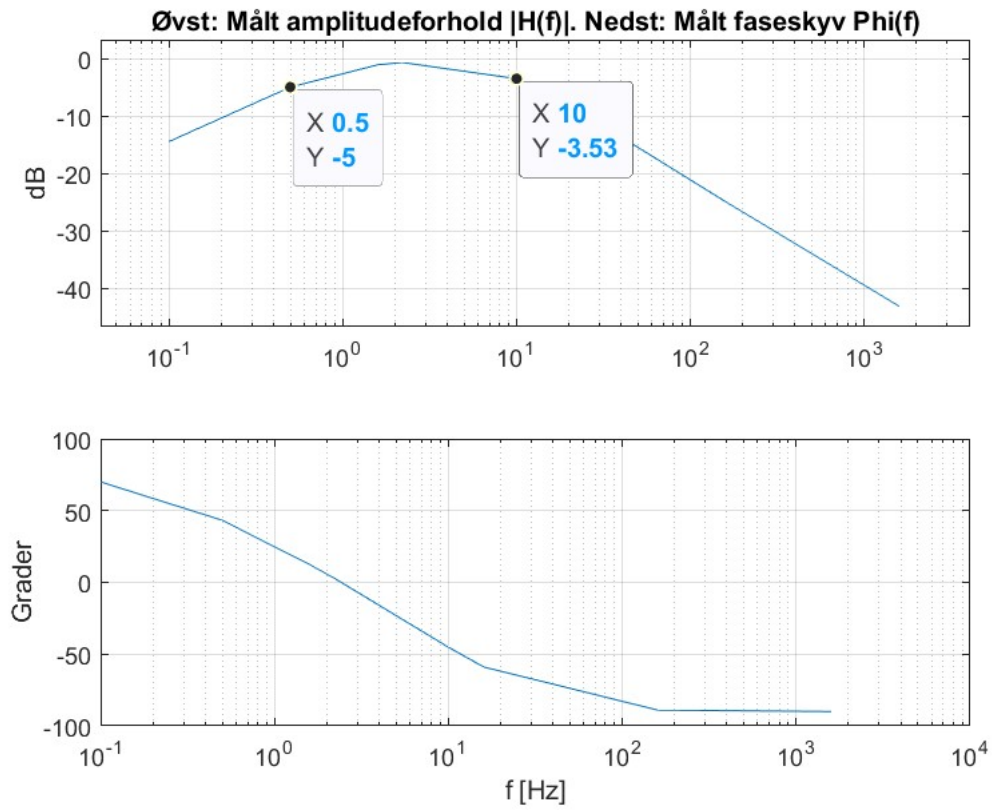
frekvens(Hz)	$\Delta\Phi$	Δt	V_{ut}/V_{in}	$(V_{ut}/V_{in})dB$	V_{ut}	$V_{in}[V]$
0.1	-70	-1.90s	0.18	-14.5	481mV	2.6
0.5	-43.3	-240ms	0.57	-5	1.48V	2.6
1.6	-12.3	-21.3ms	0.88	-1.1	2.29V	2.6
2.2	-3	-3.7ms	0.92	-0.7	2.40	2.6
10	45.2	12.54ms	0.66	-3.53	1.73v	2.6
16	59	10.24ms	0.488	-6.22	1.27v	2.6
160	89	1.54ms	0.057	-25	148mv	2.6
1591	90	140us	0.007	-43.2	18mv	2.6

Tabell 3.1: Viser frekvensrespons resultatet for båndpassfilteret, ved forskjellige frekvens verdier. Forkortelsene $\Delta\Phi$, Δt , V_{ut} og V_{in} står for faseforskyvning, Tidsforsinkelse, utgangsspenning og inngangsspenning.

Når fasedifferansen er positiv, betyr det at utgangssignalet følger inngangssignalet med en viss forsinkelse. Dette kan virke ulogisk ved første øyekast, da alle fysiske systemer er kausale og virkningen av et inngangssignal ikke kan forekomme før selve inngangssignalet kommer inn til systemet. Imidlertid er dette tilsynelatende ulogiske fenomenet forårsaket av at signalene som betraktes, er periodiske og gjentar seg med en periode på enten 2π eller 360 grader. Derfor, hvis fasedifferansen er for eksempel +60 grader, så er dette det samme som at utgangssignalet ligger $+60 - 360 = 300$ grader etter inngangssignalet. Med andre ord, en positiv fasedifferanse $\Delta\Phi$ tilsvarer en negativ fasedifferanse lik $\Delta\Phi - 360^\circ$.

Båndpassfilteret har blitt undersøkt gjennom både teoretisk og praktisk analyse. Basert på disse analysene kan man trekke den konklusjonen at responsen til båndpassfilteret er i samsvar med de forventede resultatene.

3.5 Båndpassfilter



Figur 3.7: Bode-diagram av praktisk målteverdier i **Tabel 3.1**

3.5 Båndpassfilter

Matlab kode for Bode-diagramet, vist i **Figur 3.7**. Koden er brukt i faget ELE200, lagt av Morten Tengedal.

```
1
2
3 %-----
4 ns = 8; % Talet på ulike frekvensar som er brukt, lik 6 i eksemplet
5       % under.
6
7 %Deklarasjon av datavektorar
8 %-----
9
10 %Frekvensvektor
11     frekv_vektor = zeros(1,ns);
12
13 %Vektor for utrekna amplitudeforholdsverdiar i decibel
14     ampl_dB_vektor = zeros(1,ns);
15
16 %Vektor for utrekna faseskyv i grader.
17     fase_vektor = zeros(1,ns);
18
19 %Frekvensverdiar:
20     frekv_vektor = [0.1 ,0.5 ,1.6,2.2, 10, 16,160,1591];
21
22 %|H(f)|db-verdiar:
23     ampl_dB_vektor = [ -14.5 , -5 , -1.1, -0.7, -3.53, -6.22, -25, -43.2];
24
25 %Phi(f)-verdiar i grader.
26     fase_vektor = [70,43.3 ,12.3, 3, -45.2, -59,-89,-90];
27
28 echo on
29
30 clf
31 subplot(211),semilogx(frekv_vektor,ampl_dB_vektor)
32 grid
33 title('Øvst: Målt amplitudeforhold |H(f)|.
34 Nedst: Målt faseskyv Phi(f)')
35 ylabel('dB')
36 subplot(212),semilogx(frekv_vektor,fase_vektor)
37 grid
38 xlabel('f [Hz]')
39 ylabel('Grader')
40
```

3.5.3 Praktisk utføring av båndpassfilteret ved et EKG-signal

I denne seksjon vil undersøkelsen av EKG-inngangssignalets responsen gjennom et båndpassfilter bli analysert. Signalet ble generert av et oscilloskop og testet på samme måte som sinus-inngangssignalet. Etter flere forsøk ble det observert at EKG-signalet var mest følsomt i båndpassområdet mellom 0,5 og 10 Hz. QRS-komplekse viste seg til å være følsom ved smal båndbredde, spesielt S-bølgen som representerer depolariseringen av ventriklene i hjertet. S-bølgen ved 0.5 - 10 Hz flatet seg ut til null, dette indikerer til tap av informasjon i ventriklene. Oppgaven er begrenset til å fokusere på måling av hjerteraten. Selv om S-komplekset i elektrokardiogrammet kan være forstyrret og flatet ut, har dette ingen betydning for oppgavens mål. Siden oppgaven primært handler om å avlese hjerteraten, da er R-komplekset den mest avgjørende faktoren å ta hensyn til.

Fokuset vil derfor være på dette området for å analysere signalets egenskaper og eventuelle avvik. Analysering av signalets egenskaper utført ved å stille først oscilloskopet på 1Hz, da normal hvilepuls til en person er på 60 slag per minutt **Figur 3.8**

Ved å analysere **Figur 3.8** ser vi at EKG-inngangssignalet, i oransje farge, som er forsterket av instrumenteringsforsterkeren fra 20 mV til 2.57 V (Pk-Pk). Deretter passerer inngangssignalet (oransje) gjennom båndpassfilteret, hvor signalet filtreres til ønsket båndbredde. Utgangssignalet av båndpassfilteret vises i grønn farge.

Observasjonen av det grønne utgangssignalet fra båndpassfilteret i forhold til inngangssignalet (oransje) viser en tydelig forskjell. Dette er tidlig en virkningen av høypassfilteret, som er en del av båndpassfilteret. Utgangssignalet er redusert i amplitude. S-bølgen, som er en del av det opprinnelige signalet, er ikke lenger synlig i det grønne signalet. Dette skyldes at S-bølgen har en høyere signalfrekvens, og dermed vil høypassfilteret filtrere ut dette signalet. Noe som resulterer i tap av S-bølgen i utgangssignalet. Dette fenomenet kan observeres i **Figur 3.8**.

Den smale båndbredden (0.5-10 Hz) som er brukt i båndpassfilteret, har en innvirkning på QRS-komplekset i det elektriske hjertesignalet. Dette fører til forstyrrelser som påvirker S-bølgens amplitude, som blir flatet ut til null. En slik forstyrrelse kan potensielt føre til feilaktig tolkning av utslaget som representerer spredningen av elektrisk aktivitet i ventriklene. Da formålet er å finne hjerterepulsen, er R-bølgen av størst interesse i dette tilfellet. Men det er viktig å merke seg at fraværet av S-bølgen som et resultat av forstyrrelser i båndpassfilteret ikke vil påvirke direkte målingen av hjerteraten, da R-bølgen fortsatt vil være til stede. Derfor vil det ikke være en direkte feilaktig tolkning av hjerterepulsen.

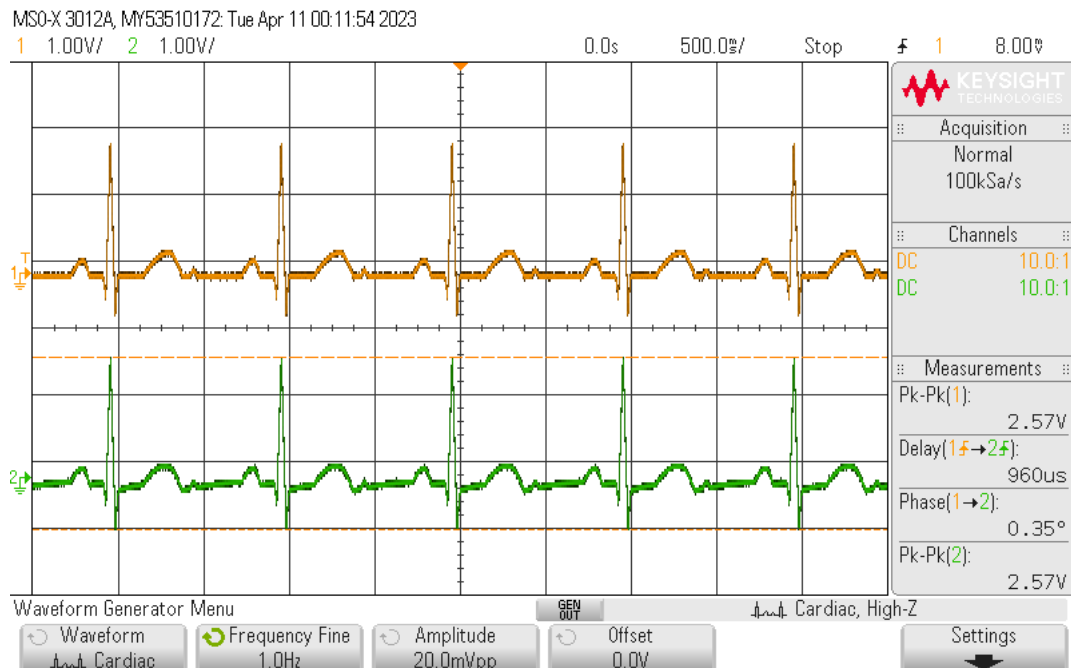
Imidlertid kan mangelen på S-bølgen ha andre konsekvenser når det gjelder analyse og tolkning av det elektriske hjertesignalet. S-bølgen gir informasjon om spredningen av elektrisk aktivitet i ventriklene, og dens fravær kan begrense den fulle forståelsen av hjertets elektriske mønster. Det er derfor viktig å være klar over at bruk av et smalbåndet båndpassfilter kan føre til tap av visse komponenter i det elektriske hjertesignalet, som for eksempel S-bølgen.

3.5 Båndpassfilter



Figur 3.8: Resultat av QRS-komplekset ved å bruke en smal båndbredde på mellom 0,5 og 10 Hz. En kan se S-bølgen i blå utgangssignalet eksisterer ikke.

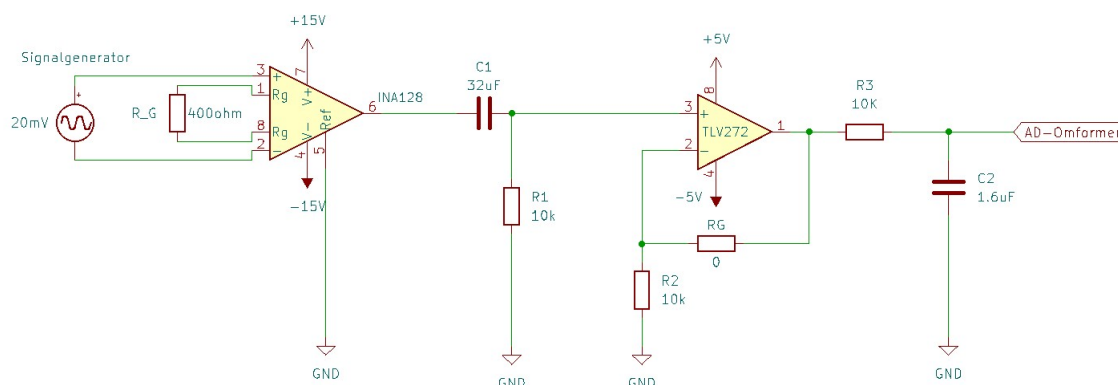
Det ble gjort et forsøk på å observere eventuelle forstyrrelser i QRS-komplekset ved å bruke en bred båndbredde på mellom 0,5 og 130 Hz. Resultatene viste at inngangssignalet (oransje) var identisk med utgangssignalet (grønn) **Figur 4.9**.



Figur 3.9: Resultat av QRS-komplekset ved å bruke en bred båndbredde på mellom 0,5 og 130 Hz.

3.6 Kretsskjema av EKG-systemet

Kretsskjema består av fire komponenter: en signalgenerator, en instrumenteringsforsterker, et aktivt båndpassfilter og en AD-omformer. I tillegg kunne det være en fordel å inkludere en notch-filter i kretsen for å eliminere netstøy på 50 Hz. Etersom oppgavens båndbredde er begrenset, er det ikke behov for en slik type filter. Signalgeneratoren gir et utgangssignal på 20 mV, som deretter forsterkes til 2,5 V av instrumenteringsforsterkeren. Deretter filtreres signalet til ønsket båndbredde og sendes videre til en analog-til-digital-konverter for etterfølgende digital signalbehandling (**Figur 3.10**).



Figur 3.10: kretsskjema av EKG-systemet Kondensator. Kondensator C2 er regnet ut til å være 1.59 uF, men forholdet seg til en standar kondensator verdi på 1.8 uF. Kretsen er designet i KiCad.

Kapittel 4

Analog til digital

4.1 Innledning

Dette kapitlet omhandler digital signalbehandling av EKG-signalet, som innebærer å konvertere det analoge EKG-signalet til digital form for videre behandling. Konverteringen vil bli utført ved hjelp av en mikrokontroller av typen STM32F103RB. Hovedfokuset i kapitlet vil være på konfigureringen av AD-omformerer og bruken av en Timer i forbindelse med AD-omformerer. Vi vil også se nærmere på ulike utfordringer som oppstår underveis, samt hvordan disse utfordringene ble håndtert og løst.

I introduksjon 1 ble det nevnt at oppgaven skulle arbeide med en smal signalbåndbredde på 0.5 Hz til 10 Hz. Denne begrensede båndbredden har ført til betydelige utfordringer til nå. I delseksjon 3.5.3 ga en forklaring på hvilke problemer den smale båndbredden førte med seg. Selv om jeg har løst utfordringen ved å konkludere at S-komplekset har ikke noen betydning for oppgavens formål, har dette likevel skapt flere utfordringer i dette kapitlet.

Derfor vil denne oppgaven i dette kapitlet jobbe med en bredere båndbredde på 0.5 Hz til 130 Hz. Dette oppnås ved å endre verdien av kondensatoren C2 i **Figur 3.4** til 120 nF. Jeg vil forklare hvilke utfordringer som oppstår og hvordan de ble løst etter hvert som kapitlet utvikler seg. Det er imidlertid viktig å merke seg at oppgaven fra dette kapitlet og videre vil forholde seg til en bredere båndbredde på 0.5 Hz til 130 Hz.

4.2 Analog til digital

Det analoge EKG-signalet, som opprinnelig er generert av oscilloskopet, skal gjennomgå en serie behandlinger før det blir konvertert til et digitalt format ved hjelp av STM32F103RB-mikrokontrolleren. Under konverteringsprosessen fra analog til digital form, er det flere sentrale faktorer som må tas i betraktning. Disse faktorene inkluderer oppløsning, samplingsfrekvens, signalstøy, kalibrering og foldingsfeil.

4.3 Plattformkort

Oppløsningen til en analog-til-digital omformer bestemmer hvor nøyaktig signalet blir representert i digitale verdier. Høyere oppløsning gir mer nøyaktige verdier, men krever også mer prosesseringskraft og større lagringskapasitet. Samplingsfrekvensen bestemmer hvor ofte signalet blir målt og konvertert til digitale verdier. En høy samplingsfrekvens kan bidra til å bevare informasjonen i det opprinnelige signalet, mens en for lav samplingsfrekvens kan føre til tap av viktig data. Analogt signal kan være utsatt for støy og forvrengning, som kan påvirke nøyaktigheten av den digitale konverteringen. Det er viktig å vurdere og minimere støyen før digital konvertering. Det er også viktig å kalibrere analog-til-digital omformeren for å sikre nøyaktighet og pålitelighet av de digitale verdiene. Analogt signal kan ha frekvenskomponenter som overstiger samplingsfrekvensen, som kan føre til aliasing og feilaktige digitale verdier. Det er derfor viktig å inkludere antialiasing-filter for å hindre aliasing.

4.3 Plattformkort

I faget *ELE210* ble plattformkortet Nucleo-F103RB distribuert til alle studenter som deltok i faget. Formålet var å gi studentene muligheten til å bli kjent med STM32 mikrokontrolleren. Dette gjorde det unødvendig å kjøpe et nytt plattformkort. Tilgjengeligheten av plattformkortet forenklet bruken i oppgaven.

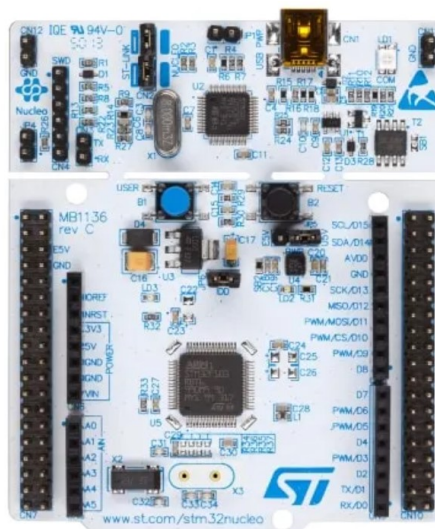
STM32F103RB-mikrokontrolleren er en 32-bits mikrokontroller som er produsert av STMicroelectronics (**Figur 4.1**). Den er utstyrt med en ARM Cortex-M3 prosessor og er vanligvis anvendt i innebygde systemer. Programmeringen av mikrokontrolleren baserer seg hovedsakelig på C-språket. For å muliggjøre en effektiv og lesbar programmering og kommunikasjon med mikrokontrolleren, har ARM-prosessorens utviklere etablert en standard kjent som CMSIS¹. Denne standarden gir et grensesnitt som tillater en standardisert tilnærming for programmering og overføring av kode til mikrokontrolleren. Ved å følge CMSIS-standarden, blir utviklere i stand til å dra nytte av en enhetlig og strukturert metode for å utvikle programvare for STM32F103RB-mikrokontrolleren. Dette bidrar til å forbedre effektiviteten og lesbarheten til koden som er utviklet for mikrokontrolleren [12].

¹Cortex Microcontroller Software Interface Standard

4.3 Platformkort

MCU	STM32F103RB
Family	ARM Cortex-M3
Vendor	ST Microelectronics
RAM	20KiB
Flash	128KiB
Frequency	up to 72MHz
FPU	no
Timers	7 (2x watchdog, 1 SysTick, 4x 16-bit)
ADCs	1x 12-bit (16 channels)
UARTs	3
SPIs	2
I2Cs	2
RTC	1
USB	1
CAN	1
Vcc	2.0V - 3.6V

(a) Detaljer om mikrokontrolleren



(b) Nucleo-F103RB [7]

Figur 4.1: Mikrokontroller Nucleo-F103RB

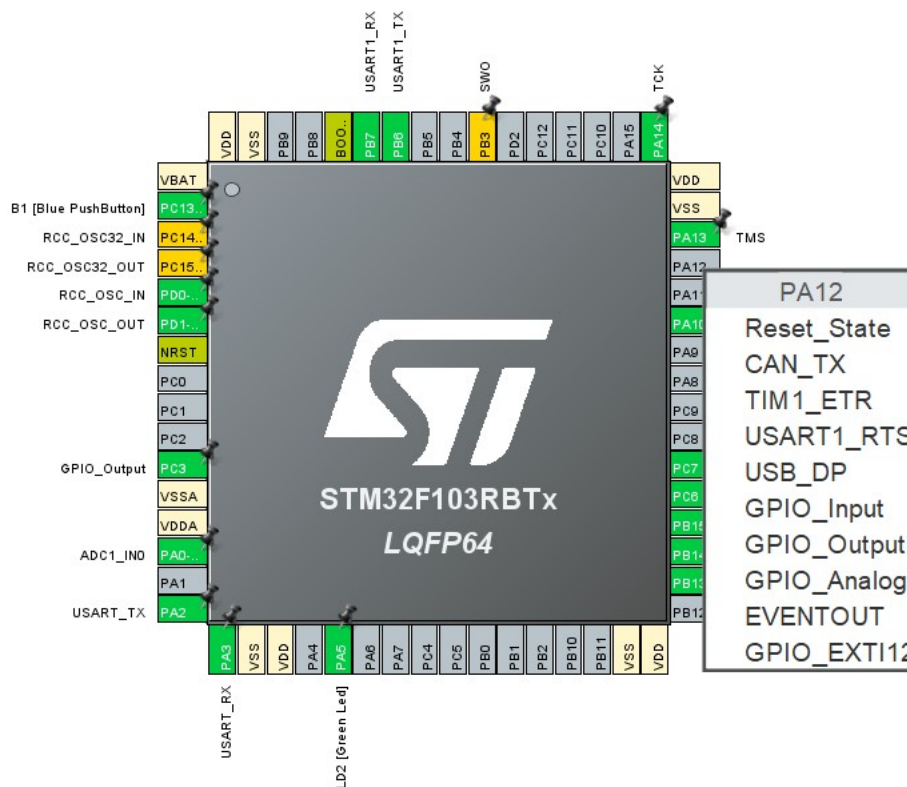
4.3.1 Mikrokontroller

For analog-til-digital konvertering av mikrokontrolleren vil det benyttes STM32CubeIDE ("Cube"), et integrert utviklingsmiljø (IDE) utviklet av STMicroelectronics (**Figur 4.2**). STM32CubeIDE er spesifikt tilpasset for programmering av STMicroelectronics' samling av mikrokontrollere. Ved å benytte dette standardiserte IDE-et kan man oppnå kode som er godt lesbar, verifiserbar og testbar.

Når man utvikler et prosjekt ved hjelp av STM32CubeIDE, går man gjennom to hovedfaser: maskinvareoppsett og programvareutvikling. I maskinvareoppsettsfasen setter man opp konfigurasjonen for mikrokontrolleren og andre tilkoblede komponenter. Dette inkluderer å definere pinnefunksjoner, aktivere og konfigurere moduler som er nødvendige for analog-til-digital konvertering, samt å tilpasse kretsen til de spesifikke kravene i prosjektet. Etter å ha fullført maskinvareoppsettet, kan man begynne med programvareutviklingen. Dette innebærer å skrive koden som styrer analog-til-digital konvertering, samt eventuelle andre funksjoner som er nødvendige for prosjektet. STM32CubeIDE gir et brukervennlig grensesnitt og verktøysett for å skrive og teste koden. Koden kan utvikles ved hjelp av C eller C++ programmeringsspråk, og man kan dra nytte av standardbiblioteker og funksjoner som er tilgjengelige i STM32CubeIDE for å forenkle utviklingsprosessen.

Ved å følge denne prosessen kan man utvikle et prosjekt for analog-til-digital konvertering ved hjelp av STM32CubeIDE, og dra nytte av standardiseringen som tilbys for å oppnå godt lesbar, verifiserbar og testbar kode [12].

4.3 Platformkort



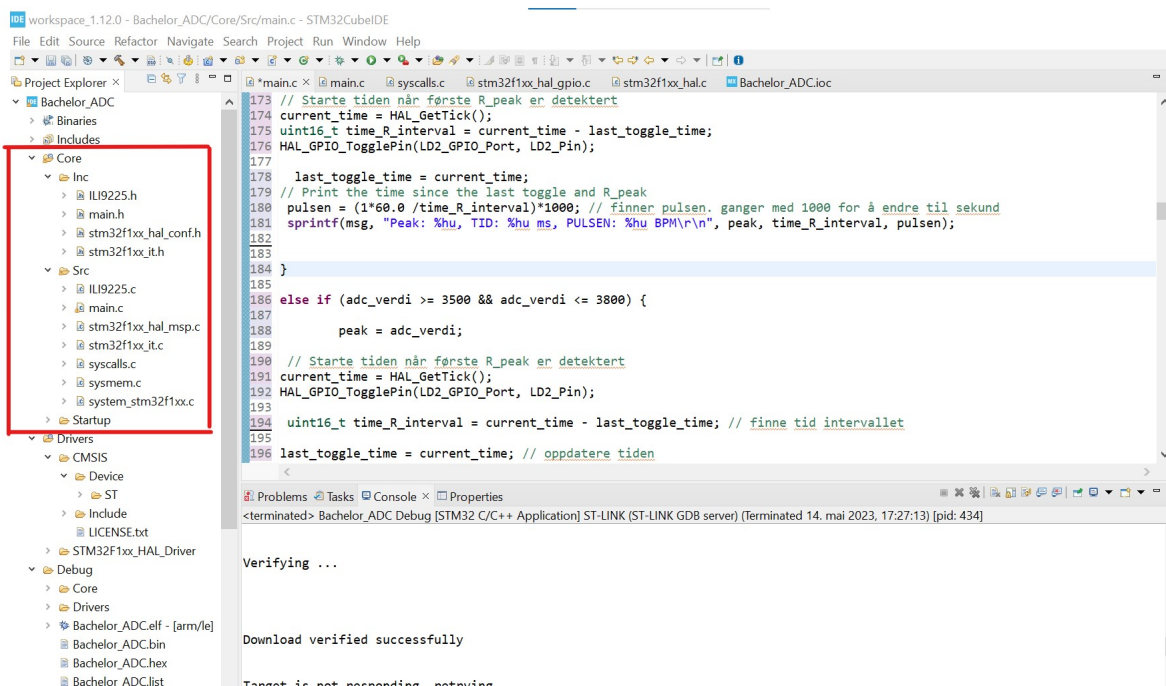
Figur 4.2: Standard maskinvareoppsett i *STM32CubeIDE*.

I dette tilfellet har pinnen PA12 blitt venstreklikket i *STM32CubeIDE*, og det åpnes en nedtrekksmeny som viser tilgjengelige funksjoner som PA12-pinnen kan kobles til. Ved å velge en av disse funksjonene, aktiveres pinnen og tildeles den valgte funksjonen. Etter å ha spesifisert maskinvareoppsettet og lagret det, genereres automatisk konfigurasjonskode for maskinvaren.

Som et resultat av dette vil prosjektoppsettet inneholde hovedfiler med følgende egenskaper: **.h** og **.c**. Disse filene er vanligvis kjent som henholdsvis hjelpefiler (header) og kildefiler (source). hjelpefiler (**.h**) inneholder deklarasjoner av funksjoner, definisjoner av konstanter og strukturer som er nødvendige for prosjektet. Disse filene gir en oversikt over grensesnittet og funksjonaliteten til koden. Mens kildefilene (**.c**) inneholder den faktiske implementasjonen av funksjonene og logikken i prosjektet. Her skrives koden som utfører spesifikke oppgaver, inkludert styring av PA12-pinnen og håndtering av den valgte funksjonen (**Figur 4.3**).

Ved å organisere koden i separate hjelpefiler- og kildefiler oppnår man en strukturert og modulær tilnærming, noe som gjør det enklere å vedlikeholde og utvide prosjektet over tid. Hjelpefiler gir en oversikt over tilgjengelige funksjoner og grensesnitt, mens kildefilene inneholder den faktiske implementasjonen av koden.

4.3 Platformkort



Figur 4.3: Prosjekthierarkiet i STM32CubeIDE.

4.3.2 UART

For å muliggjøre kommunikasjon mellom en mikrokontroller og en datamaskin, er det nødvendig at begge enhetene kan utveksle informasjon ved hjelp av samme kommunikasjonsprotokoll. Dette er spesielt viktig i sammenheng med analog-til-digital konvertering av EKG-signaler, da det er nødvendig å kunne effektivt overføre de konverterte digitale verdiene til datamaskinen for visualisering. Det finnes flere kommunikasjonsprotokoller som kan benyttes for dette formålet, for eksempel UART¹, SPI² eller I2C³. Valg av kommunikasjonsprotokoll avhenger av faktorer som hastighetskrav, avstand mellom enhetene, antall tilkoblede enheter og tilgjengelige grensesnitt på både mikrokontrolleren og datamaskinen.

En effektiv kommunikasjon mellom mikrokontrolleren og datamaskinen sikrer at de konverterte digitale verdiene fra EKG-signalet kan overføres pålitelig og i sanntid. Dette gjør det mulig for datamaskinen å visualisere EKG-dataene og utføre eventuell videre analyse eller behandling. Ved å skape en stabil kommunikasjonsprotokoll mellom mikrokontrolleren og datamaskinen kan man sikre at de to enhetene kan utveksle informasjon effektivt og uten tap av data. Dette er avgjørende for å oppnå nøyaktig visualisering av de konverterte EKG-dataene på datamaskinen.

For å oppnå kommunikasjonen mellom mikrokontrolleren og datamaskinen ble programvaren Putty benyttet (**Figur 4.4**). Putty er en gratis og åpen kildekode-

¹Universal Asynchronous Receiver/Transmitter

²Serial Peripheral Interface

³Inter-Integrated Circuit

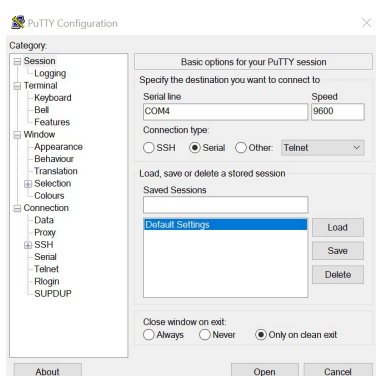
4.3 Platformkort

programvare som gjør det mulig å opprette en seriell tilkobling via UART-kommunikasjon mellom datamaskinen og en ekstern modul, som for eksempel mikrokontrolleren. Ved å bruke Putty kan man opprette en tilkobling mellom datamaskinen og mikrokontrolleren via en seriell port. Dette muliggjør sending og mottak av data mellom enhetene. I dette tilfellet gjør det at man kan vise og analysere de digitale verdiene som er konvertert fra EKG-signalet.

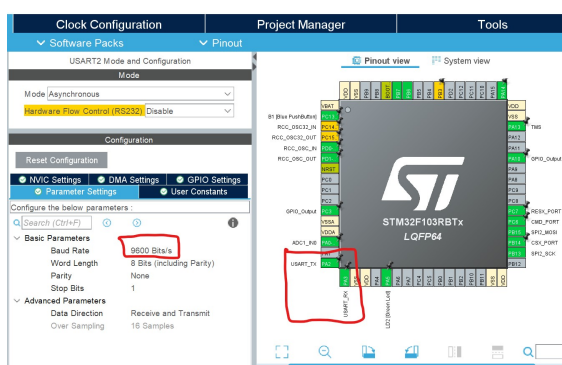
Putty gir en enkel og brukervennlig måte å skape og administrere en seriell forbindelse. Programvaren støtter forskjellige innstillinger for seriell kommunikasjon, for eksempel overføringshastighet og protokollparametere. Den gir også funksjoner for å vise og logge data som sendes og mottas mellom mikrokontrolleren og datamaskinen. Ved å benytte Putty-programvaren har en effektiv kommunikasjonskanal mellom mikrokontrolleren og datamaskinen. Dette gir oss muligheten til å overføre data og informasjon mellom enhetene, og dermed legge grunnlaget for videre analyse og bearbeiding av de digitale EKG-verdiene i kommende trinn.

For å bruke *Putty* for *UART* kommunikasjon, følges en disse trinnene:

1. konfigurere *UART* i *STM32CubeIDE*. I *parameter settings* kan en sette verdien i *Baud Rate* til 9600 Bits/s.
2. Start Putty-programmet på datamaskinen.
3. I *Putty-vinduet* velger en *Serial* som tilkoblingstype. Deretter angir en riktig COM-port og innstillingene for overføringshastighet *baud rate* som er konfigurert for *UART*-kommunikasjonen. Husk *baud rate* i *Putty-vinduet* og *UART* må være samme.
4. Deretter klikke en på *open* for å opprette tilkoblingen mellom datamaskinen og enheten via *UART*



(a) Putty konfigurasjon



(b) UART konfigurasjon

Figur 4.4: Konfigurasjon av kommunikasjon mellom datamaskin og mikrokontrolleren gjennom UART.

Etter tilkoblingen er opprettet, kan en sende og motta data via UART-porten. Alt som sendes fra mikrokontrollen via UART vises i Putty-vinduet.

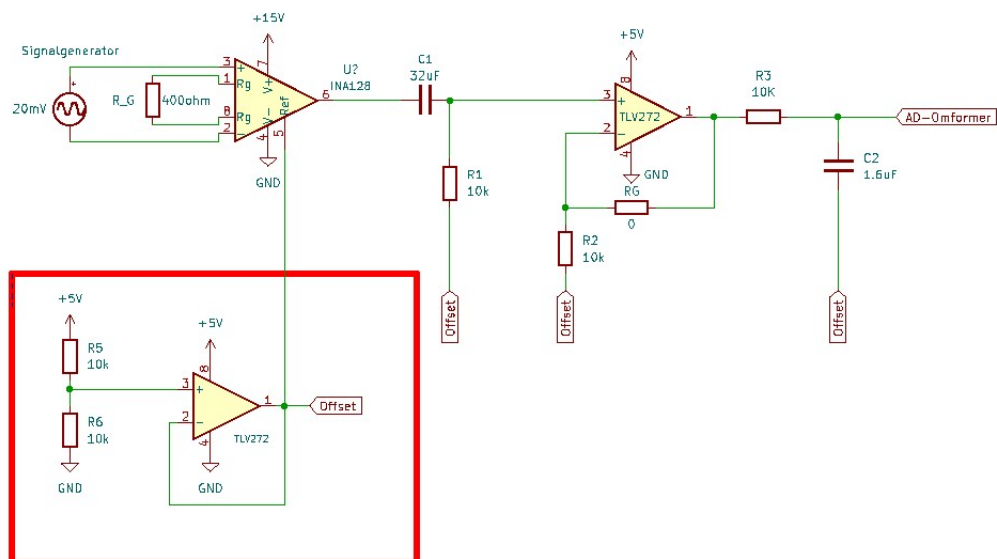
4.4 AD-omformer

Mikrokontrolleren benytter en spenningsreferanse fra 0 til 3,3 V for sin analog-til-digital konvertering. Som en konsekvens av dette, vil AD-omformerene kun kunne behandle positive spenningsverdier. Det vil si AD-omformerene er unipolar. Da EKG-signalet er vekselstrøm (AC) vil veksle mellom positiv og negativ spenning. Som et resultat vil negativ spenning av EKG-signalet tolkes som null av AD-omformerene.

For å løse utfordringen med at AD-omformerene kun kunne behandle positive spenningsverdier, ble offseten på oscilloskopet justert til 900 μV . Slik at signalet ble skyve opp fra jorden. Og dermed gjort hele EKG-signalet positivt. Imidlertid førte denne løsningen til at høypassfilteret i kretsen måtte fjernes, siden høypassfilter har som hensikt å fjerne DC-signaler fra et signal, og offseten på 900 μV representerer et DC-signal. Dette er en kritisk løsning som fører til uønskede konsekvenser. Tidligere i oppgaven ble det påpekt hvor støyende et EKG-signal kan være, og hvor viktig rollen til et høypassfilter er for å rense opp i signalet. Å fjerne høypassfilter fører nå til en upraktisk løsning med potensielle negative konsekvenser for datakvaliteten.

I løpet av arbeidet med oppgaven utviklet jeg etter hvert en forbedret og mer effektiv løsning på utfordringen, men jeg hadde ikke tilstrekkelig tid til å implementere denne løsningen. Den foreslåtte løsningen involverer design av en spenningsreferansegenerator, som er tydelig markert i en rødt firkantet boks i **Figur 4.5**. Spenningsreferansen genererer en offsetspenning for instrumentforsterkerens utgang, og dette sikrer at utgangsspenningen til instrumentforsterkeren alltid er positiv og ikke negativ. Ved å benytte denne løsningen kan behovet for å justere offset på oscilloskopet elimineres, samtidig som behovet for å fjerne høypassfilteret ikke lenger er nødvendig. Dette resulterer en effektiv og praktisk løsning.

4.4 AD-omformer



Figur 4.5: I dette tilfellet er spenningsgeneratoren påtrykt en spenning på 5 V, og den genererer en referansespenning på 2,5 V som brukes som referanse for hele kretsen.

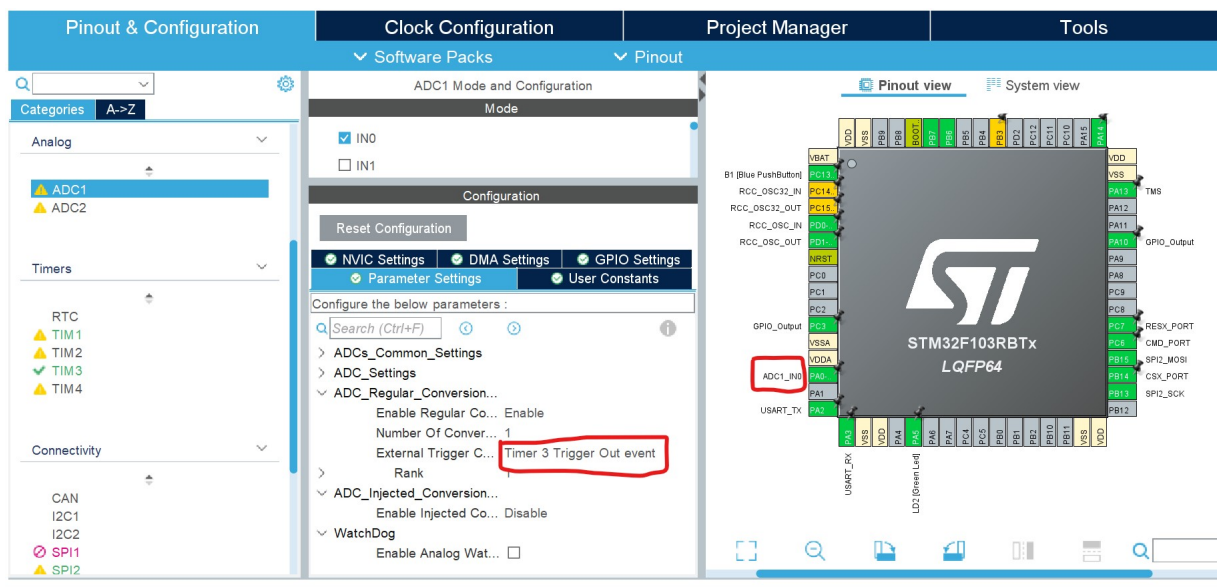
Det finnes forskjellige typer AD-omformerer, direkte konvertering, SAR (Successive Approximation Register), og pipeline AD-omformer. Generell sett STM32F103RB mikrokontrolleren har en SAR AD-omformer modul med følgende karakterstikk. Mikrokontrolleren har en oppløsning på 12 bit (løst), kan ikke endres. Samt med en maksimal klokkefrekvens på 12 MHz. Den har opptil 18 multiplexede kanaler, som gjør at den kan måle signaler fra 16 eksterne og to interne kilder. AD-omforming av de ulike kanalene kan utføres i enkelt-, kontinuerlig-, skanne- eller diskontinuerlig modus. Resultatet av AD-omformaren blir lagret i et 16-bitt dataregister som kan justeres til venstre eller høyre. Konverteringstiden kan programmeres individuelt for hver kanal [8].

STM32F103RB mikrokontrolleren har flere innebygde timermoduler som kan konfigureres for å starte AD-konverteringen. I dette kapitlet er det et ønske å benytte en *Timer* til å generere periodiske hendelser som triggesignaler for AD-konverteringen. En kan konfigurere *Timeren* i *STM32CubeIDE*. En kan sette timeren til å generere en avbruddshendelse med ønsket frekvens. Dette er spesielt nyttig når en ønsker å sample inn data med en fast samplingshastighet. Ved bruken av *Timer* gjør det enklere å oppnå nøyaktig tidsstyring for konvertering.

4.4.1 AD-konfigurasjon

For å konvertere det analog EKG-signalt til digital er det nødvendig å konfigurere omformaren. Dette gjør man i programvare STM32CubeIDE. Det skal benyttes som sagt en *Timer* for å utløse AD-konverteringer ved en konfigurert samplingfrekvens. Her skal vises enhetskonfigurasjon steg for steg.

4.4 AD-omformer



Figur 4.6: Enhetskonfigurasjon (Device Configuration Tool) i STM32CubeIDE programvare for ADC1.

Det ble brukt inngang *AD1* inngang kanal 0 for AD-omformereren. En kan også legge merke til at *hadc1.Init.ContinuousConvMode* er konfigurert til *emphDISABLE*, dette er fordi, det skal bli brukt en Timer til å trigge AD-konverteringen. Koden en ser i *kode1* er oppsette som bli gjort i Devise Configuration Tool i programvaren, som blir automatisk generert til *main.c* filen (Figur 4.6).

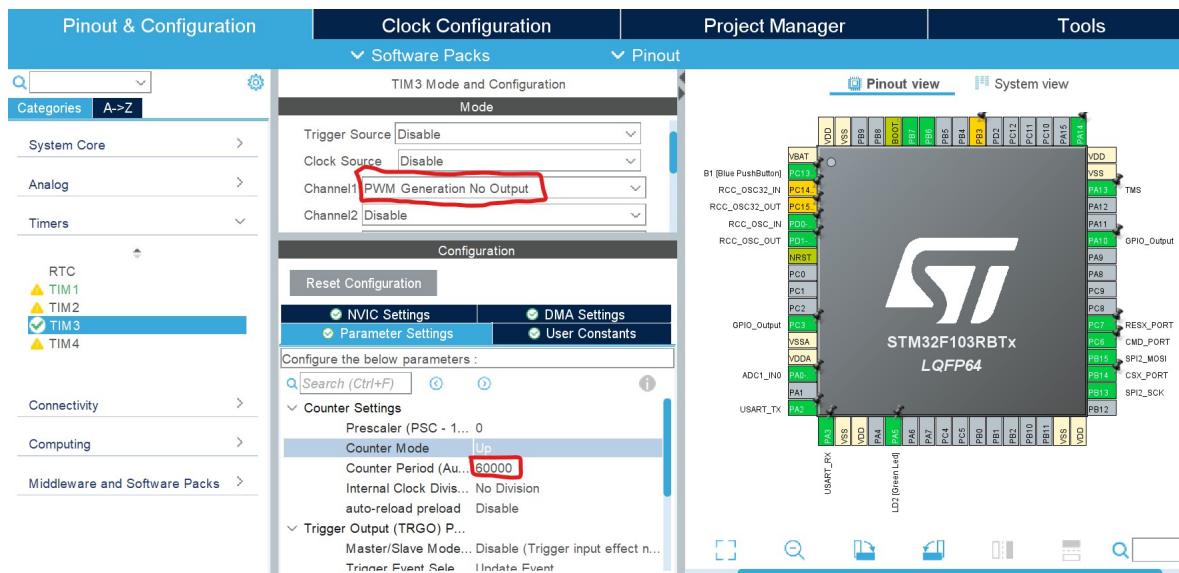
Kode1: Generert kode av STM32CubeIDE for ADC1

```
1 static void MX_ADC1_Init(void)
2 {
3     ADC_ChannelConfTypeDef sConfig = {0};
4
5     /** Common config
6     */
7     hadc1.Instance = ADC1;
8     hadc1.Init.ScanConvMode = ADC_SCAN_DISABLE;
9     hadc1.Init.ContinuousConvMode = DISABLE;
10    hadc1.Init.DiscontinuousConvMode = DISABLE;
11    hadc1.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T3_TRGO;
12    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
13    hadc1.Init.NbrOfConversion = 1;
14    if (HAL_ADC_Init(&hadc1) != HAL_OK)
15    {
16        Error_Handler();
17    }
18
19    /** Configure Regular Channel
20    */
21    sConfig.Channel = ADC_CHANNEL_0;
22    sConfig.Rank = ADC_REGULAR_RANK_1;
23    sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
24    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
25    {
26        Error_Handler();
27    }
28    /* USER CODE BEGIN ADC1_Init 2 */
29
30    /* USER CODE END ADC1_Init 2 */
31
32 }
```

4.4 AD-omformer

4.4.2 Timer konfigurasjon

For å utløse AD-konverteringen er det valgt å bruke timeren *TIM3*. Konfigurasjonen av *TIM3* ble initiert ved å konfigurere utløserkanalen (Channel1) og velge PWM-generering uten noen utgang (PWM Generation No Output). Dette ble gjort fordi timeren er satt opp til å trigge AD-omformerer internt. Deretter ble tellerperioden (Counter Period) satt til 60000. Valget av denne spesifikke verdien vil bli nærmere diskutert i neste seksjon.



Figur 4.7: Enhetskonfigurasjon(Device Configuration Tool) i STM32CubeIDE for timeren *TIM3*.

Generert kode av STM32CubeIDE for TIM3

```
1 static void MX_TIM3_Init(void)
2 {
3
4
5     TIM_MasterConfigTypeDef sMasterConfig = {0};
6     TIM_OC_InitTypeDef sConfigOC = {0};
7
8     /* USER CODE BEGIN TIM3_Init 1 */
9
10    /* USER CODE END TIM3_Init 1 */
11    htim3.Instance = TIM3;
12    htim3.Init.Prescaler = 0;
13    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
14    htim3.Init.Period = 60000;
15    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
16    htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
17    if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
18    {
19        Error_Handler();
20    }
```

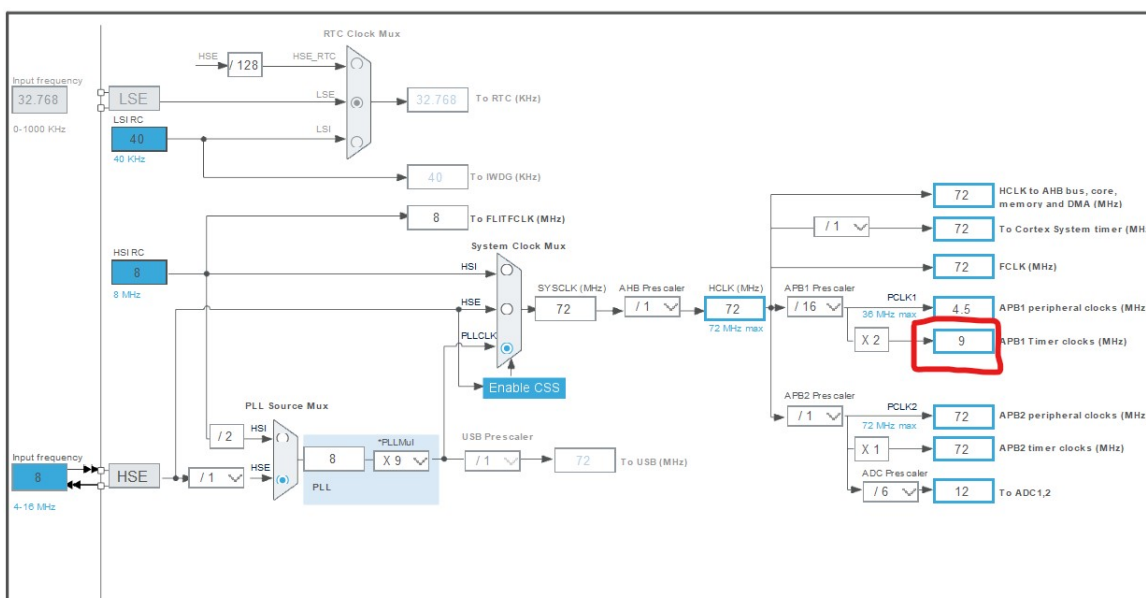
4.4 AD-omformer

```
20 }
21 sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
22 sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
23 if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig)
24     != HAL_OK)
25 {
26     Error_Handler();
27 }
28 sConfigOC.OCMode = TIM_OCMODE_PWM1;
29 sConfigOC.Pulse = 0;
30 sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
31 sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
32 if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1)
33     != HAL_OK)
34 {
35     Error_Handler();
36 }
37 /* USER CODE BEGIN TIM3_Init 2 */
38
39 /* USER CODE END TIM3_Init 2 */
40
41 }
42
43
44
```


4.4.3 Klokke konfigurasjon

Hvor fort mikrokontrolleren skal utføre en handling eller en oppgave er avhengig av klokkefrekvensen. Klokkefrekvensen er definert som antall klokkesykluser per sekund, og det er denne klokkehastigheten som bestemmer mikrokontrollerens hastighet og evne til å behandle instruksjoner og utføre oppgaver raskt. Jo høyere klokkefrekvensen er, desto raskere kan mikrokontrolleren utføre beregninger og håndtere data. I figur 4.8 kan man observere at frekvensen til mikroprosessoren er satt til maksimalt 72 MHz. I tillegg er klokkefrekvensen til timeren, som er markert med en rød firkant, satt til 9 MHz for å oppnå ønsket samplingsfrekvens.

Ved å begrense klokkefrekvensen til timeren til 9 MHz, oppnår man en passende hastighet for å oppfylle den ønskede samplingsfrekvensen. Dette gjøres for å sikre at AD-konverteringen utføres med riktig timing og gir nøyaktige digitale verdier fra de analoge signalene. Ved å tilpasse klokkefrekvensen til timeren til 9 MHz i forhold til mikroprosessorens maksimale frekvens på 72 MHz, oppnår man en balanse mellom ytelse og presisjon for å oppfylle kravene til samplingsfrekvensen i systemet.



Figur 4.8: Klokke konfigurasjon for Timeren er satt på 9MHz. Denne klokkefrekvensen er avgjørende for samplefrekvensen for AD-omformer

4.4.4 Start og kalibrering av AD-omformer

`HAL_ADCEx_Calibration_Start(&hadc1)`, dette kaller en funksjon fra HAL-biblioteket som starter en kalibrering på en AD-enhet. `&hadc1` er en peker til AD-enheten som skal kalibreres. `HAL_ADC_Start_IT(&hadc1)`, dette kaller en funksjon fra HAL-biblioteket som starter en AD-konvertering med interrupt-støtte på en AD-enhet. `&hadc1` er en peker til AD-enheten som skal brukes til konvertering. `HAL_ADC_Start_IT(&hadc1)` starter timeren.

4.4 AD-omformer

```
1 // calibrate ADC for better accuracy and start it w/ interrupt
2 if(HAL_ADCEx_Calibration_Start(&hadc1) != HAL_OK)
3     Error_Handler();
4
5 if(HAL_ADC_Start_IT(&hadc1) != HAL_OK)
6     Error_Handler();
7
8 // start pwm generation
9 if(HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3) != HAL_OK)
10     Error_Handler();
11
```

Funksjonen tar en peker til AD-enheten som en parameter. Funksjonen vil bli kalt når en AD-konvertering er fullført. Derav blinker den grønne LED-lampen ved å endre tilstanden til LD2_Pin ved hjelp av `HAL_GPIO_TogglePin()`. Ved å koble LD2_Pin på oscilloskopet kan en se AD-omformerens samplefrekvens. Derav setter `adcFlag` til SET. Dette indikerer at en AD-konvertering er fullført og at AD-verdien er klar til å bli behandlet utenfor callback-funksjonen.

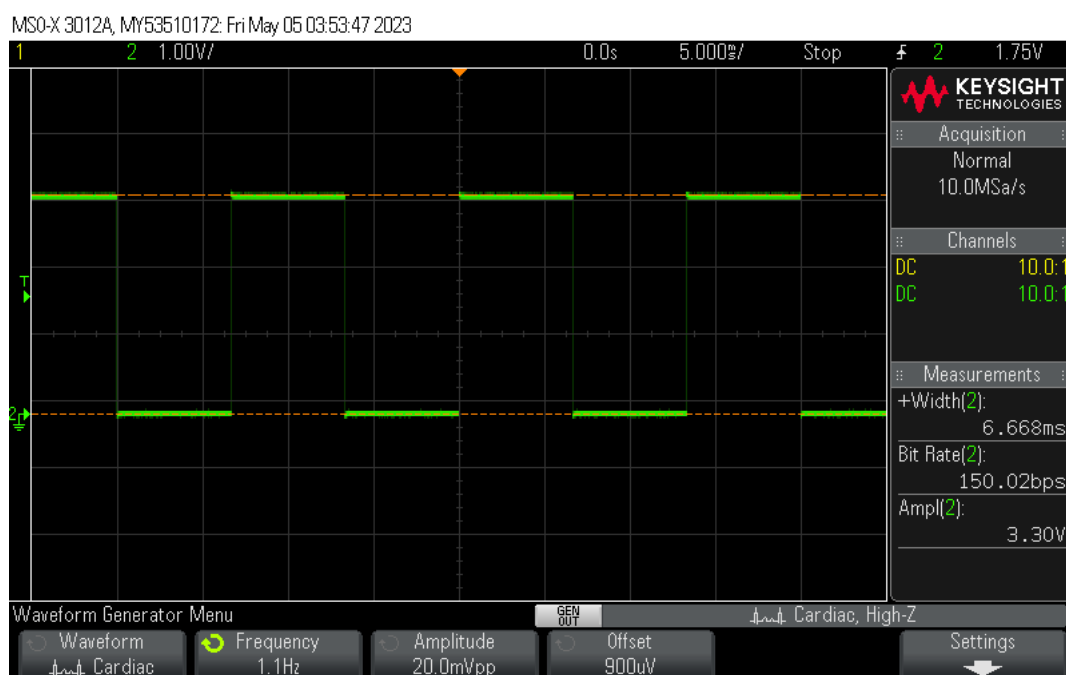
```
1 /* USER CODE BEGIN 4 */
2     void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
3     {
4         // Toggle the Green LED
5         HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
6         adcFlag = SET;
7     }
8 /* USER CODE END 4 */
```

I while-løkken sjekker om variabelen `adcFlag` er satt. Hvis `adcFlag` henter AD-verdien ved å kalle funksjonen `HAL_ADC_GetValue(&hadc1)`, og lagre verdien i variabelen `adc_verdi`. Deretter konverterer AD-verdien til en tegnstring ved hjelp av `sprintf()`-funksjonen og lagrer den i variabelen `msg`. Derav Setter `adcFlag` til RESET for å indikere at AD-verdien er behandlet. Tilslutt Sender tegnstringen over UART-grensesnittet ved å kalle `HAL_UART_Transmit()` funksjonen, som overfører tegnstringen `msg` som en bytestring over UART-grensesnittet. Deretter fortsetter løkken og gjentar disse trinnene kontinuerlig så lenge while (1) betingelsen er sann.

```
1
2 while (1)
3     {
4     if (adcFlag) {
5         adc_verdi = HAL_ADC_GetValue(&hadc1);
6         sprintf(msg, "%hu\r\n", adc_verdi);
7         adcFlag = RESET;
8         HAL_UART_Transmit(&huart2, (uint8_t*)msg, strlen(msg), HAL_MAX_DELAY);
9     }
```

4.4.5 Samplefrekvens

Når man utfører analog-til-digital konvertering, er det viktig å ta hensyn til Nyquist-teoremet. Ifølge teoremet må samplingsfrekvensen være minst dobbelt så høy som den høyeste frekvensen i signalet for å unngå aliasing, som er forvrengning eller tap av informasjon [15]. I vår tilfelle varierer frekvensen til EKG-signalet fra 1 til 2.5 Hz. For å kunne reprodusere signalet og samtidig ikke overskride Nyquist-frekvensen, er samplefrekvensen til AD-omformerens beregnet til å være på 150 Hz. Noe som er tilstrekkelig for å representere EKG-signalet på en passende måte. Samplefrekvensen bestemmes av *timer klokkefrekvensen* delt på *Counter Period*. Hvor *Counter Period* er satt til å være 60000 perioder, i seksjon 4.4.2. Samt *timer klokkefrekvensen* er satt til å være 9 MHz, i seksjon 4.4.3. Verdiene i konfigurasjonene er bestemt på grunnlag av ønsket samplefrekvensen på 150Hz. Dette betyr at hvis EKG-signalets periode er 1 sekund, vil AD-omformerens sample signalet 150 ganger.



Figur 4.9: Bildet viser LED-LD2 blinker hver 6.6ms, som indikerer at AD-omformerens sampler ved 150Hz som ønsket. En legges merke til at i nedste ramme i figuren står det *frequency* 1.1Hz. Denne frekvensen er EKG-inngangssignalet ikke samplefrekvensen

$$SF = \frac{TK}{CP} = \frac{9 \cdot 10^6 Hz}{6 \cdot 10^4 perioder} = 150 Hz \quad (4.1)$$

TK = Timer klokke. Verdien er konfigurert i figur 4.8.

CP = Counter Period. Verdien er konfigurert i figur 4.7.

4.4.6 AD-Oppløsning

Mikrokontrolleren som benyttet for å konvertere analog-til-digitalt er en 12 bit omformer ved 3.3 V spenning referanse. Dette vil si en kan representere EKG-signalet til 2^{12} (4096) diskrete nivåer. Jo høyere oppløsning, jo mer nøyaktig vil AD-omformeren kunne representere det analoge inngangssignalet. Hvor nøyaktig en ønsker å representere et signal variere avhengig av applikasjonen. Under digital konvertering av EKG-signalet ble det observert at de diskrete nivåene ikke nådde den maksimale verdien på 4095 digitale verdier. I stedet varierte de rundt 3500 til 3800 verdier. Denne avvikende oppførselen skyldtes en feil i konverteringsprosessen, og det var nødvendig med en grundig undersøkelse for å identifisere årsaken. Problemet kunne potensielt oppstå på grunn av faktorer som elektrisk støy, kalibreringsfeil eller feil implementering av AD-omformeren. Før undersøkelsen startet, ble det gjennomført en dobbeltsjekk for å bekrefte om inngangssignalet til AD-omformeren var 3.3 V. Selv om denne verifiseringen viste seg å være korrekt, viste det seg dessverre at dette tiltaket alene ikke var tilstrekkelig for å løse problemet. Derfor måtte alternative metoder prøves. Gjennom en rekke eksperimenter og feilforsøk ble det til slutt konkludert med at problemet skyldtes spenningdelen i kretsen.

Problemet med avvikende verdier skyldes spenningdelen mellom motstanden R9 i **Figur 3.10** og en intern motstand i mikrokontrolleren kalt R_{AIN} , som vist i figur 4.10. I seksjon 3.6 er kretsskjemaet designet for å tilpasse AD-omformeren til et spenningsområde på 3.3V. Dette betyr at utgangsspenningen fra båndpassfilteret, med hjelp av motstanden RG, er beregnet til å forsterke signalet til 3.3 V. Under feilundersøkelsen ble det oppdaget at selv om inngangsspenningen til AD-omformeren var ment å være 3.3 V, var den faktiske spenningen 2.75 V, som illustrert av formel 5.1. Derfor, årsaken til at AD-omformeren ikke nådde 4095 digitale verdier, ligger i spenningdelen mellom motstandene i kretsen.

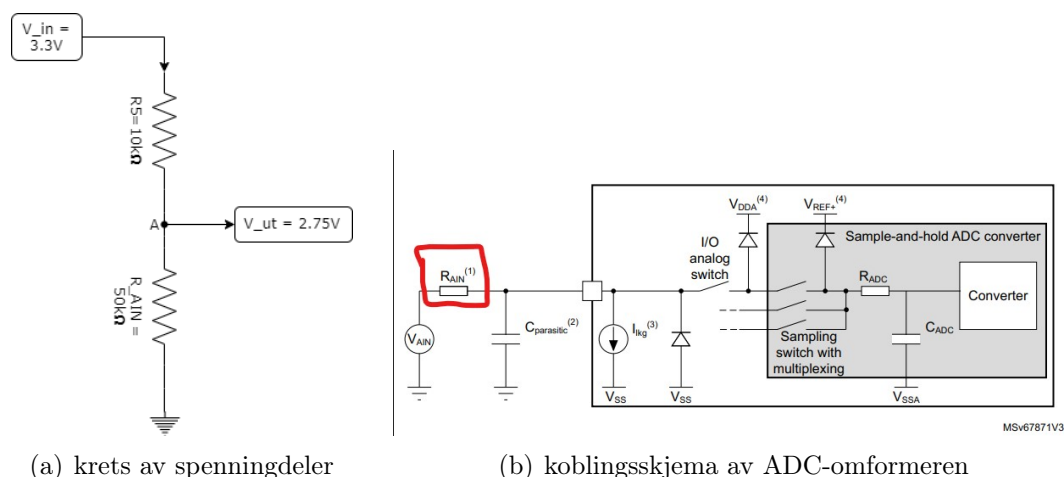
$$V_{ut} = \left(\frac{R_{AIN}}{R9 + R_{AIN}} \right) \cdot V_{in} \Rightarrow \left(\frac{50K\Omega}{10K\Omega + 50K\Omega} \right) \cdot 3.3V = 2.75V \quad (4.2)$$

V_{in} er spenningen som kommer ut av båndpassfilter inn på ADC1 kanal 0.

V_{ut} er konverteringsspenningsområde som AD-omformeren bruke for å konvertere.

På grunn av spenningssenknningen var det nødvendig å justere forsterkningen i både operasjonsforsterkeren og instrumentasjonsforsterkeren for å tilpasse inngangsspenningen til AD-omformeren i seksjon 3.6. For å oppnå det ønskede spenningsområdet for AD-omformeren måtte samlesettforsterkningen være på 4 V som inngangsspenning (V_{in}) på ADC1 kanal 0. Ved hjelp av formel 5.1 ble V_{ut} i dette tilfellet justert til 3.3 V. På denne måten ble problemet løst, og resultatet viste seg å være den maksimal digital verdi på 4095 som ønsket.

4.4 AD-omformer

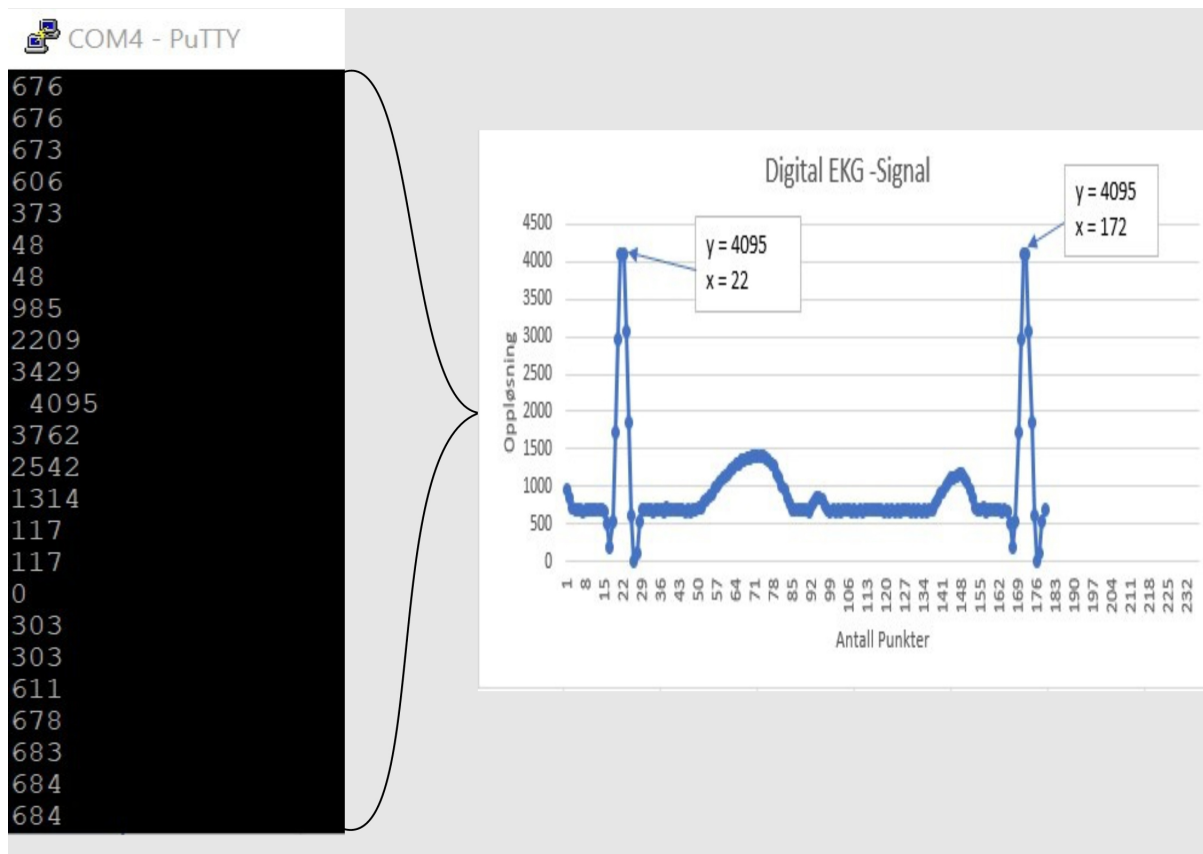


Figur 4.10: Koblingsskjema(b) og verdien på R_{AIN} er hentet av STM32F103RB databladet.

4.4.7 Presentasjon av digital EKG-signal

I denne delen vil jeg presentere de digitale verdiene som har blitt digitalt omformet. **Figur 4.11** viser bildet som representerer resultatet av analog-til-digital omformingen. På den venstre siden av **Figur 4.11** er de konverterte digitale verdiene vist. Disse digitale verdiene ble deretter plottet i Excel, og plottet demonstrerer en betydelig likhet med det opprinnelige EKG-signalet (**Figur 4.12**).

Det analoge EKG-signalet generert av oscilloskopet ble konfigurert med en frekvens på 1 Hz under plottingen, hvilket betyr at det analoge EKG-signalet hadde en periode på 1 sekund. Det er essensielt å merke seg at samplingsfrekvensen til AD-omformer er beregnet til å være 150 Hz, i henhold til formel 4.1. Dette indikerer at AD-omformer bruker 6,66 ms for å sample en digital verdi. I **Figur 4.11** kan man observere at amplitudeforskjellen mellom punkt $x_1 = 172$ og $x_2 = 22$ tilsvarer 150 punkter, som forventet. På bakgrunn av dette kan jeg konkludere med at samplingsfrekvensen stemmer overens med plottet.

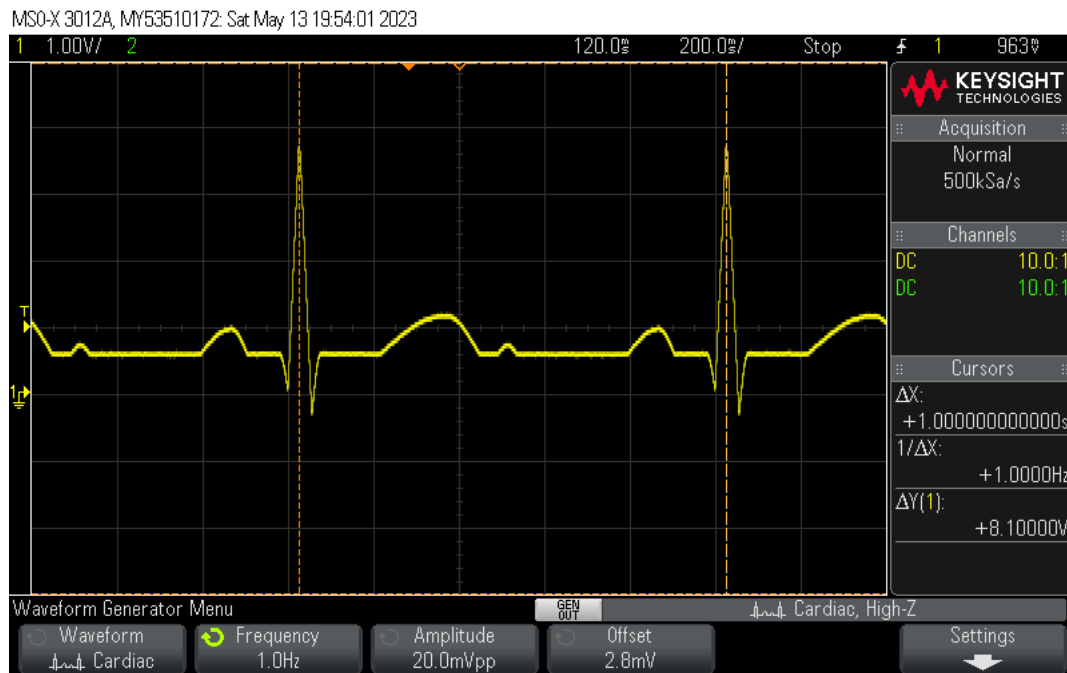


Figur 4.11: På venstre side av bildet fremvises den diskrete digitale verdien av EKG-signalet, mens på høyre side fremstilles en plott av hver enkelt digital verdi.

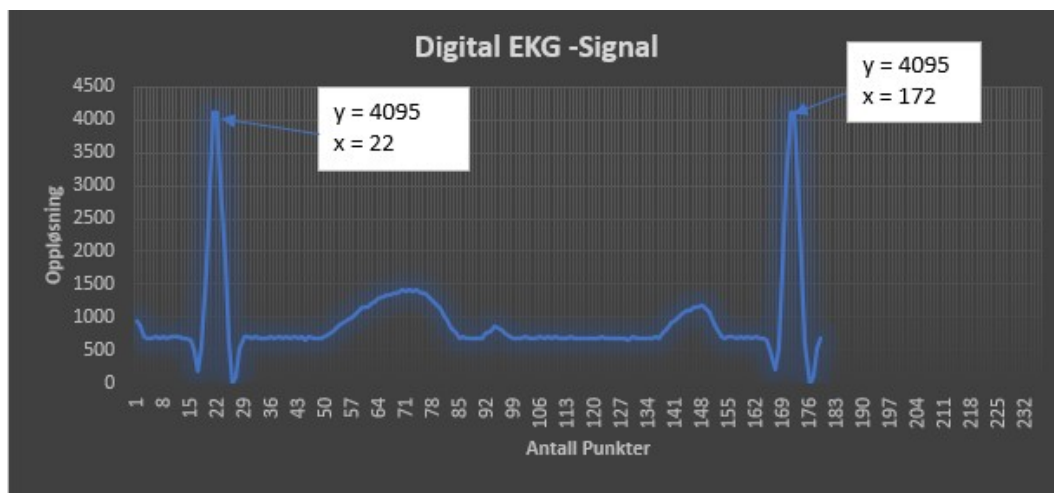
4.4.8 Resultat av AD-omformeren

Ved å sammenligne det analoge EKG-signalet med det digitale plottet, ser man en tydelig likhet mellom dem. Basert på denne observasjonen konkluderer jeg med at EKG-signalet har blitt vellykket digitalisert.

4.4 AD-omformer



Figur 4.12: Original analog EKG-signalet generert av oscilloskopet.



Figur 4.13: Plott av det vellykkede digitaliserte EKG-signalet.

Kapittel 5

Digital signalbehandling

5.1 Innledning

Dette kapitlet skal omhandle digital signalbehandling. Dette betyr behandling av det digitale EKG-signalet. Her skal kapitlet fokusere på programmeringsalgoritme for å finne pulsraten.

5.2 R-bølge toppdeteksjon

For å kunne finne pulsraten er det nødvendig å identifisere R-bølgetoppen i EKG-signalet. Det finnes flere avanserte metoder for R-bølgedeteksjon, men i dette tilfellet vil den enkleste og mest effektive toppdeteksjonsalgoritmen bli brukt. Programmeringen av R-bølgedeteksjonen vil bli utført i STM32CubeIDE ved hjelp av C-programmering.

Figur 5.1 illustrerer flytskjemaet som representerer programstrukturen for R-bølge-toppdeteksjon. Når mikrokontrolleren kobles til strømforsyningen eller når resett-knappen trykkes, initierer prosessoren en oppstartsrutine. Denne rutinen omfatter initialisering av stakkpekeren, etterfulgt av kallingen av hovedprogrammet, kjent som *main()*. Deretter vil alle instruksjoner innenfor *while-løkken* bli utført kontinuerlig så lenge betingelsen for *while-løkken* er sann (det vil si så lenge *while-løkkevariabelen* er lik 1).

Variabelen *adc_verdi* inneholder de digitale EKG-signalene i området 0 til 4095. Før jeg startet med programmeringen, gjennomgikk jeg en grundig studie av variabelen *adc_verdi*. De digitale verdiene avhenger av stabiliteten til spenningsamplituden i EKG-signalet. For å gjøre det klart, er det ønskelig at frekvensen på inngangssignalet varierer mellom 1 og 2.5 Hz for å kunne registrere pulsen ved forskjellige hjertefrekvenser. Denne frekvensvariasjonen førte imidlertid til en reduksjon i inngangsspenningen, noe som igjen førte til ustabil deteksjon av R-bølgetoppen. Dette var hovedårsaken til at jeg valgte å begrense oppgaven til en bredere båndbredde i kapittel 4.

Da oppgaven var pregnet til smal båndbredde på 0.5 til 10 Hz, ble det observert

5.2 R-bølge toppdeteksjon

at spenningsamplituden var svært følsom. Kretsen ble designet for å gi en spenning på 4 V til AD-omformerens ved en frekvens på 1 Hz. Imidlertid, når frekvensen på EKG-signalet ble økt til 1,5 Hz, ble spenningen redusert til 3 V fra de opprinnelige 4 V. Dette betyr at på grunn av spenningsdeleren, som ble forklart i delseksjon 4.4.6, tilsvarer denne spenningsreduksjonen en verdi på 2,5 V på AD-omformerens. Dette betyr at jeg nå representerer EKG-signalet med halvparten av de digitale verdiene, som er 2558. Dessverre er dette ikke tilstrekkelig og ikke praktisk innenfor digital signalbehandling. Den begrensede spenningsamplituden ved høyere frekvenser fører til redusert nøyaktighet og detaljer i de digitale verdiene som representerer EKG-signalet.

Derfor ble det nødvendig å utvide oppgaven for å ta hensyn til en bredere båndbredde på 0,5 til 130 Hz. Ved bruk av en bredere båndbredde viste det seg at spenningsamplituden ikke var like følsom som ved smalere båndbredde. Ved en frekvensområde fra 1 til 1,5 Hz var spenningsamplituden stabil på 4 V. Ved en frekvensområde fra 1,5 til 2 Hz var spenningsamplituden på 3,8 V. Den utvidede båndbredden bidro til bedre stabilitet i spenningsamplituden, noe som resulterte i mer stabile digitale verdier. Dette igjen førte til enklere og mer effektive programalgoritmer. En stabil digital verdi gjør det enklere å implementere og analysere algoritmer for R-bølge toppdeteksjon. Dermed bidro valget av en bredere båndbredde til å forbedre nøyaktigheten og effektiviteten til programmet for R-bølge toppdeteksjon.

I flytskjemaet i **Figur 5.1** kan vi observere to testblokker som har to mulige utfall: "ja eller nei". Den første testblokken sjekker om verdien til `adc_verdi` er større enn 3800 og mindre enn 4095. Hvis utfallet er "ja", betyr det at en R-bølge er detektert. Denne testblokken brukes når EKG-signalet har en frekvens på 1 til 1,5 Hz, og i dette tilfellet vil verdiene for R-bølgene variere mellom 3800 og 4095.

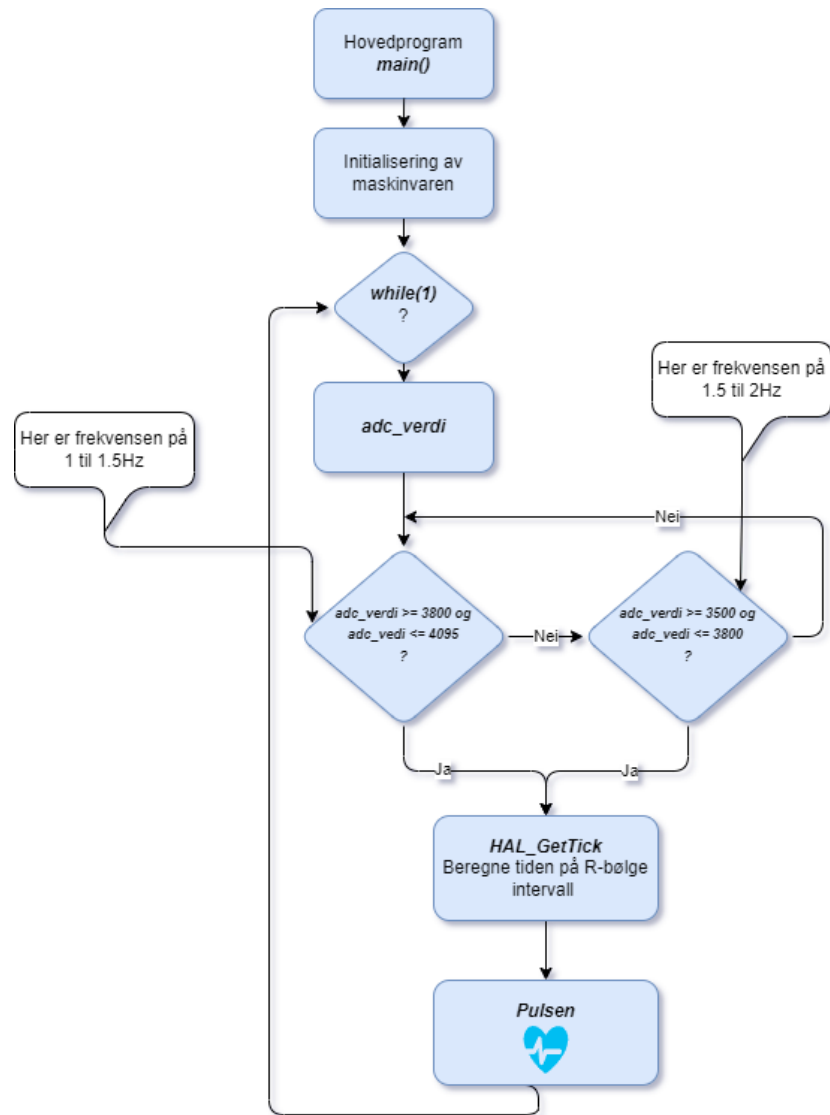
Hvis utfallet i den første testblokken er "nei", vil programmet gå videre til testblokk nummer to. Her sjekkes det om den digitale verdien ligger mellom 3500 og 3800. Hvis utfallet er "ja", betyr det at en R-bølge er detektert. Denne testblokken brukes når EKG-signalet har en frekvens på 1,5 til 2 Hz. Hvis utfallet i den andre testblokken også er "nei", vil programmet gå tilbake til den første testblokken og gjenta instruksjonene.

Ved å ha en bredere båndbredde for EKG-signalet, som tillater frekvenser mellom 0,5 og 130 Hz, kan vi begrense antall testblokker som trengs. Hvis vi hadde valgt en smalere båndbredde, som førte til ustabil spenningsamplitude, ville vi måttet bruke flere testblokker for hver frekvens (for eksempel 1,1 Hz, 1,2 Hz osv.) Dette ville ha ført til en mer kompleks og tidkrevende programalgoritme.

5.2 R-bølge toppdeteksjon

Ved å forholde oss til en bredere båndbredde kunne vi begrense antall testblokker og opprettholde stabiliteten i spenningsamplituden. Dette gjorde programmet mer effektivt og mindre sårbart for ustabile spenningsverdier. Derfor var det hensiktsmessig å velge en bredere båndbredde for å redusere ustabiliteten i spenningsamplituden og unngå behovet for avanserte og tidkrevende programalgoritmer.

5.2 R-bølge toppdeteksjon



Figur 5.1: Programstruktur for R-bølge toppdeteksjon, samt puls beregning.

5.2 R-bølge toppdeteksjon

```
1 while (1)
2 {
3     // Convert to string and print
4     sprintf(msg, "%hu\r\n", adc_verdi);
5
6     if (adc_verdi >= 3800) {
7         peak = adc_verdi; // R_peak er detektert
8
9         // Starte tiden når første R_peak er detektert
10        current_time = HAL_GetTick();
11        uint16_t time_R_interval = current_time - last_toggle_time;
12        HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
13        last_toggle_time = current_time;
14
15        // Print the time since the last toggle and R_peak
16        // finner pulsen. ganger med 1000 for å endre til sekund
17        pulsen = (60.0 / time_R_interval) * 1000;
18        sprintf(msg, "Peak: %hu, TID: %hu ms, PULSEN: %hu BPM\r\n",
19                peak, time_R_interval, pulsen);
20    }
21    else if (adc_verdi >= 3500 && adc_verdi <= 3800) {
22        peak = adc_verdi;
23
24        // Starte tiden når første R_peak er detektert
25        current_time = HAL_GetTick();
26        HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
27
28        // finne tid intervallet
29        uint16_t time_R_interval = current_time - last_toggle_time;
30        last_toggle_time = current_time;
31
32        // finner pulsen. Ganger med 1000 for å endre til sekund
33        pulsen = (60.0 / time_R_interval) * 1000;
34        sprintf(msg, "Peak: %hu, TID: %hu ms, PULSEN: %hu BPM\r\n",
35                peak, time_R_interval, pulsen);
36    }
37
38    // Sender verdien til consollet ved å bruke UART
39    HAL_UART_Transmit(&huart2, (uint8_t*)msg, strlen(msg), HAL_MAX_DELAY);
40 }
41 /* USER CODE END WHILE */
```

5.3 R-bølge interval

Etter R-bølge oppdeteksjonen i testblokkene går programmet videre til neste sekvensstrinn, altså firkantblokken `HAL_GetTick()`. See figur 5.1. Hvergang det er en R-bølge toppdeteksjon vil funksjon `HAL_GetTick()` vil bli kalt. `HAL_GetTick()` måler tidsintervallet mellom R-bølgene. La oss gå gjennom koden linje for linje:

```
1 current_time = HAL_GetTick();
```

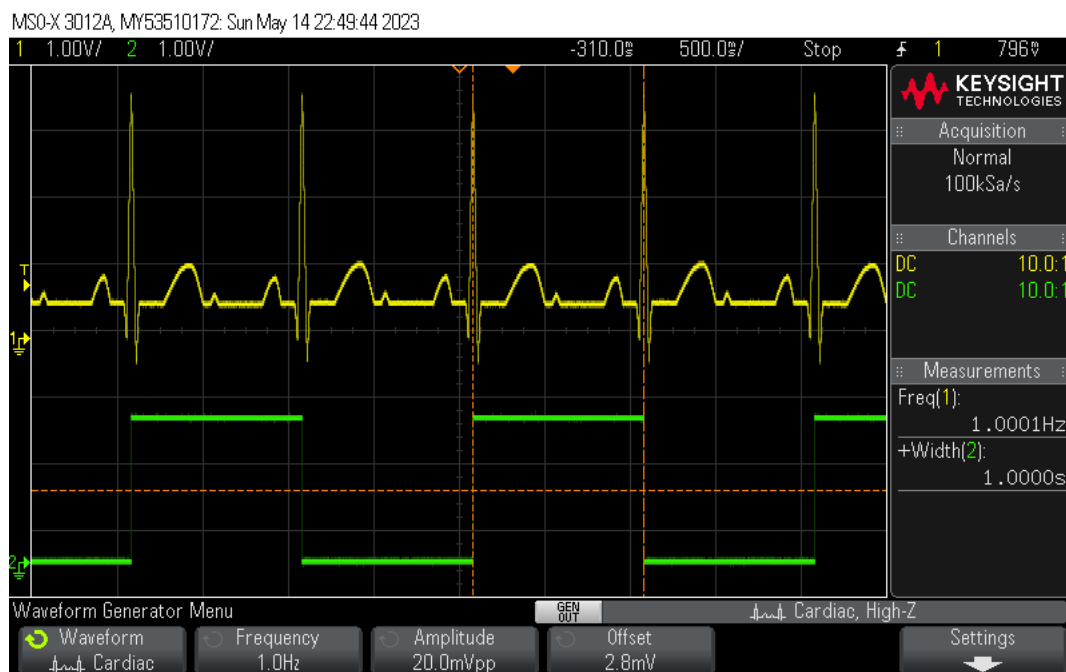
Her vil `HAL_GetTick()` hente den nåværende systemtick-tiden og lagrer den i variabelen `current_time`

```
1 uint16_t time_R_interval = current_time - last_toggle_time;
```

Her beregner tidsintervallet (`time_R_interval`) ved å trekke den forrige lagrede tiden (`last_toggle_time`) fra den nåværende tiden (`current_time`). Dette vil gi tidsforskjellen mellom to R-bølge topper.

```
1 HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
```

Denne linjen slår av/på på LD2 så lenge `time_R_interval` varer. Deretter koblet probe på `LD2_Pin` utgangen inn på oscilloskopet for å verifisere tidsintervallet på R-bølgen (**Figur 5.2**). Signalbredden på LD2 viser tidsintervallet mellom R-bølge topper på 1 sekund ved 1 Hz. Dette ser ut til å stemme.



Figur 5.2: R-bølge interval.

5.4 Resultat av pulsrate

Pulsen er definert som antall hjerteslag per minutt (bpm). Når R-bølge toppene er identifisert og tidsintervallet mellom dem er beregnet, kan pulsen beregnes ved å inverttere tidsintervallet og multiplisere med en faktor på 1000 for å konvertere til minutter.

Pulsen kan beregnes med følgende formel:

$$Puls(BPM) = \frac{60s}{Tidsintervallet\ mellom\ R\text{-}bølge\ toppene} \quad (5.1)$$

Koden for å beregne pulsen.

```
1 puls = (60.0 /time_R_interval_ms)*1000;
```

Resultatet av pulsrate basert på EKG-signalet er presentert i **Figur 5.3** for ulike signalfrekvenser. Ved å analysere disse resultatene kan det konkluderes med at pulsen har blitt vellykket identifisert. Pulsen er beregnet ved å anvende en metode som tar hensyn til tidsintervallet mellom R-bølger i EKG-signalet. Basert på de observerte verdiene kan det fastslås at pulsmålingen er pålitelig og gir en korrekt estimert pulsrate for de testede signalene. Disse funnene gir en bekreftelse på at det valgte algoritmet og metoden for pulsdeteksjon er effektiv og gir nøyaktige resultater.



Figur 5.3: Puls resultateter ved forskjellige frekvenser

Kapittel 6

Skjerm

6.1 Innledning

I foregående kapittel snakket vi og digital signalbehandl av EKG-signalet, og som resultat har vi funnet pulsen av EKG-signalet. I dette kapittelet skal det omhandler å vise den resulterende pulsen på en skjerm. Det skal benyttes TFT LCD skjerm ved driver ILI9225 for vise pulsraten.

6.2 Skjerm

Skjarmen er en enkeltbrikke SoC-driver med en oppløsning på 176RGBx220 punkter, for en TFT-flytende krystallskjerm. Den består av en 528-kanals kilde-driver, en 220-kanals gate-driver, 87120 byte RAM for grafisk data (**Figur 6.1**). Skjermen har fire typer systemgrensesnitt, inkludert i80/M68-systemet MPU-grensesnitt (8-/9-/16-/18-bit bussbredde), et serielt dataoverføringsgrensesnitt (SPI) og et RGB 6-/16-/18-bit grensesnitt. I RGB-grensesnittet muliggjør den kombinerte bruken av høyhastighets RAM-skrivefunksjon og vindusadressering visning av bevegelige bilder på en brukerspesifisert posisjon, samtidig som statiske bilder vises i andre områder på skjermen. Dette resulterer i en begrensning av overføringen av oppdateringsdata og reduserer strømforbruket [4].

På grunn av kompatibiliteten mellom TFT-skjermen og STM32-mikrokontrolleren, er det mulig å laste ned en Zip-mappe fra [GitHub.com](https://github.com), som forenkler konfigurasjonen og kommunikasjonen med mikrokontrolleren. Dette er hovedgrunnen for valget av akkurat denne type skjermen. Skjermen ble bestilt fra [Amazon.com](https://www.amazon.com) til en pris av 300 kr. Zip-mappen inneholder en hjelpefil kalt, *ILL9225.h* og en kildefil kalt *ILL9225.c*. Hjelpefilen inneholder funksjonsdeklarasjoner, mens kildefilen inneholder implementasjonen av disse funksjonene. Ved å utnytte disse filene oppnår man en enkel og effektiv kommunikasjon mellom TFT-skjermen og mikrokontrolleren.

6.3 Maskinvareoppsett

Name	Parameter
Display Color	16BIT RGB 65K color
SKU	MAR2201
Screen Size	2.2(inch)
Type	TFT
Driver IC	ILI9225
Resolution	176*220 (Pixel)
Module Interface	4-wire SPI interface
Backlight	3 White Led
Active Area	34.85x43.56 (mm)
Module PCB Size	42.46x67.20 (mm)
Operating Temperature	-20°C~60°C
Storage Temperature	-30°C~70°C
Operating Voltage	5V/3.3V
Power Consumption	TBD
Rough Weight(Package containing)	23 (g)



(a) Detaljer om TFT-skjerm.

(b) TFT-skjerm.

Figur 6.1: TFT-skjerm ILI9225 driver.

6.3 Maskinvareoppsett

TFT-skjermen bruker 8 pinner for å kontrollere skjermen, og disse pinnekonfigurasjonene er vist i **Figur 6.2**. Oppsettet av de 8 pinnene ble utført i STM32CubeIDE, der ønskede pinner ble konfigurert i systemet, og funksjonene til disse pinnene ble spesifisert. Kommunikasjonen med skjermen ble etablert ved hjelp av SPI2-grensesnittet. Det er verdt å merke seg at navnene som er definert som *RESX_PORT*, *CMD_PORT* og *CMD_PORT* i "Cube" (markert med rød firkant) ikke er identiske med pinnedefinisjonene i **Figur 6.2**. Dette skyldes den spesifikke navngivningen i ZIP-mappen. Hvis pinnene hadde blitt definert i samsvar med **Figur 6.2**, ville det vært nødvendig å endre alle pinnenavnene i ZIP-mappen, noe som ville vært tidskrevende. Derfor ble det bestemt å beholde pinneneavnene som er gitt i ZIP-mappen.

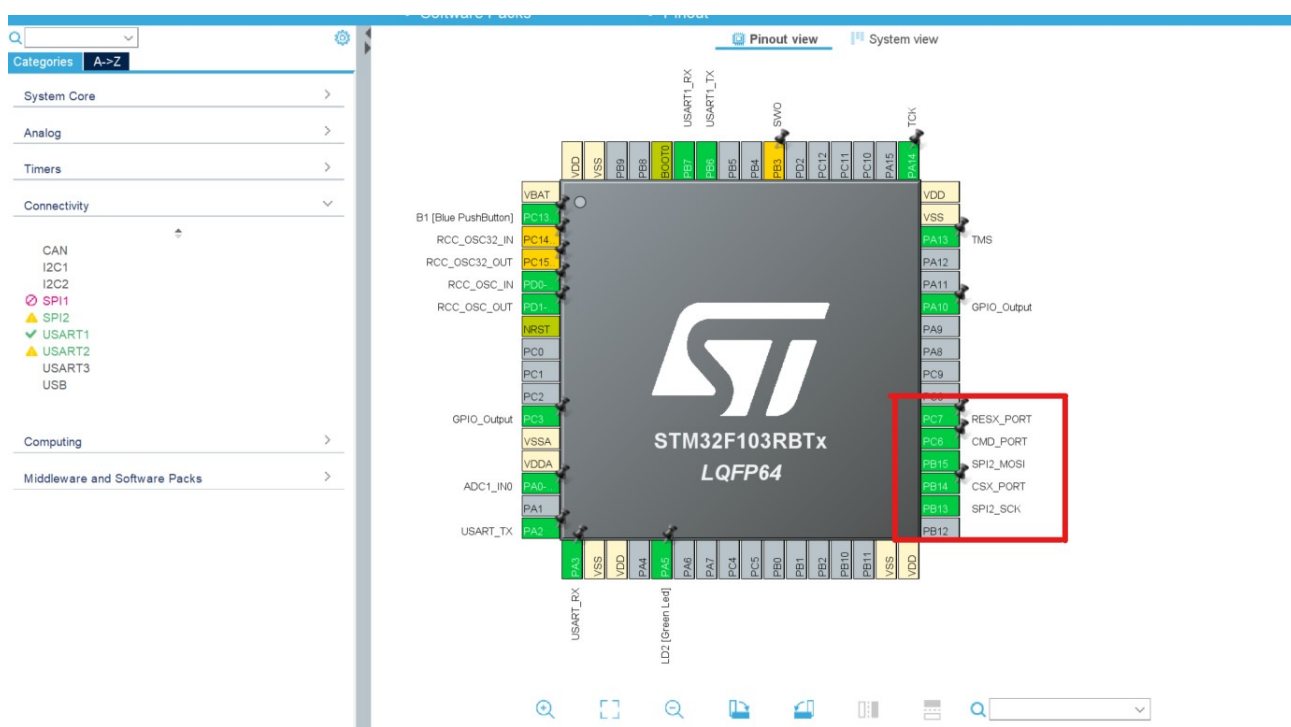
Number	Pin Label	Description
1	VCC	LCD Power positive (3.3V~5V)
2	GND	LCD Power ground
3	GND	LCD Power ground
4	NC	Not defined, reserved
5	NC	Not defined, reserved
6	LED	Backlight control, high level lighting, if not controlled, connect 3.3V always bright
7	CLK	LCD SPI bus clock signal
8	SDI	LCD SPI bus write data signal
9	RS	LCD register / data selection signal, high level: register, low level: data
10	RST	LCD reset signal, low level reset
11	CS	LCD chip select signal, low level enable

Figur 6.2: Pinne definisjoner av grensesnittet.

6.3 Maskinvareoppsett

Nr	Pinner Navn	Koblet Til
1	VCC	3.3 V
2	GND	GND
3	LED	3.3 V
4	CLK = SPI2_SCK	PB13
5	SDI = SPI2_MOSI	PB15
6	RS = CMD_PORT	PC6
7	RST = RESX_PORT	PC7
8	CS = CSX_PORT	PB14

Tabell 6.1: Tabellen viser navn på pinnene, og hvor hver av pinnene er koblet på mikrokontrolleren.



Figur 6.3: Pinner oppsett for TFT skjermen, i Cube.

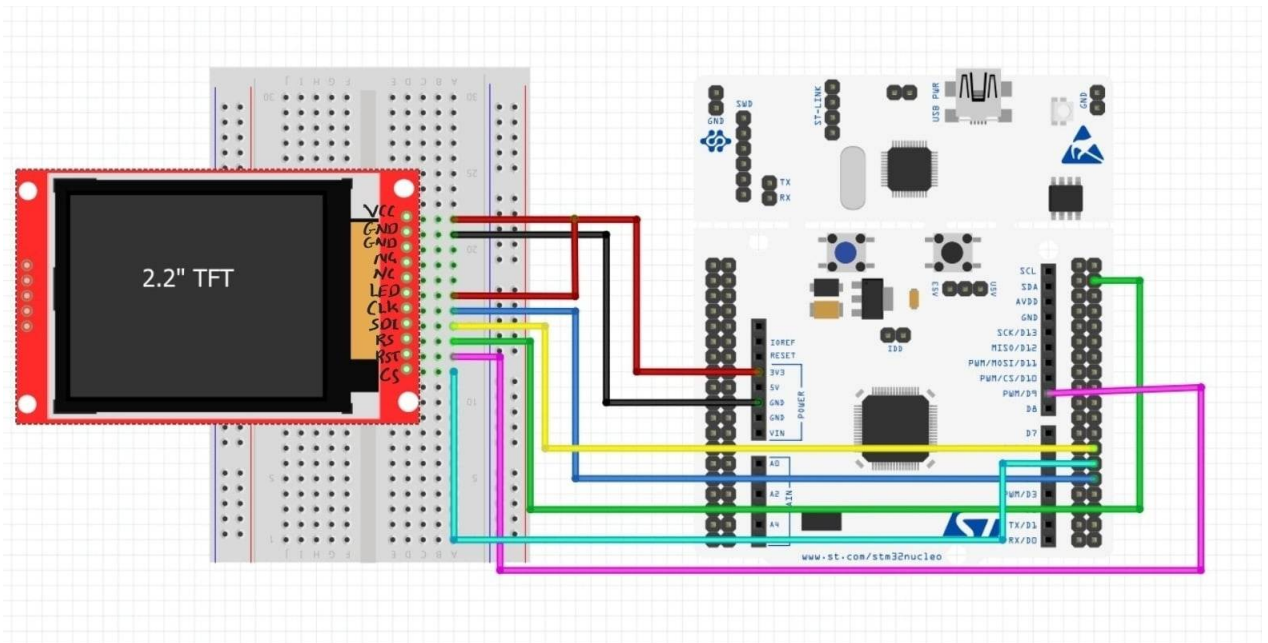
6.4 Systemgrensesnitt

Koden viser Kodeutdrag av hjelpefilen *ILL9225.h* som viser pinne definisjoner som vist i figuren.

```
1
2 #define CSX_PORT      GPIOB
3 #define CSX_PIN      GPIO_PIN_14 //Chip select
4 #define RESX_PORT    GPIOC
5 #define RESX_PIN     GPIO_PIN_7 //Reset pin
6 #define CMD_PORT     GPIOC
7 #define CMD_PIN      GPIO_PIN_6 //Command select
```

6.4 Systemgrensesnitt

Ved å følge oppsettet av **Tabell 6.1**, kan en koble opp de 8 TFT-pinnene på mikrokontrolleren, slik som på **Figur 6.4**.



Figur 6.4: Kretsskjema av TFT display koblet til STM32 mikrokontroller.

6.5 SPI kommunikasjon

Det skal benyttes en SPI-protokoll kommunikasjon mellom TFT-skjermen og mikrokontrollen. SPI¹ er en seriell kommunikasjonsprotokoll som brukes til å overføre data mellom integrerte kretser. SPI bruker en synkron metode kommunikasjon, for dataoverføringen ved hjelp av et klokkesignal. Dataene sendes i en sekvensiell strøm og blir mottatt på samme måte. I synkron kommunikasjon er timingen avgjørende for dataoverføringen. Senderen og mottakeren må være synkronisert slik at de vet når dataene skal overføres og mottas. Dette oppnås ved hjelp av en felles klokke som sørger for at både senderen og mottakeren opererer i takt. SPI-modulen i mikrokontrolleren er master, det vil si mikrokontrolleren styrer kommunikasjonen, mens skjermen er slave og skal kun ta imot data av mikrokontrolleren.

SPI bruker vanligvis fire signallinjer: *SCLK*, *MOSI*, *MISO*, og *SS*. *SCLK* (Serial Clock) er en klokkesignalpinne som dikterer timingen for dataoverføringen. Klokkesignalet er en osilerende signal som alltid generert ved master som forteller mottakeren når den skal sample bit på datalinjen. Mottakeren, altså skjermen, vil motta dataen ved å sample *MOSI* linjen ved hver stigende flanke av klokkesignalet, slik at masteren sender til slaven. Siden det er ønskelig å vise pulsen på skjermen, trenger ikke skjermen å sende data tilbake til mikrokontrolleren, og derfor er pinnen *MISO* ikke i bruk i dette tilfellet.

Pinnen som er kjent som *MOSI* (Master Out Slave In), er definert som utgangspinne for senderen (master), som i dette tilfellet er mikrokontrolleren. I henhold til **Figur 6.3**, er denne utgangspinnen spesifikt betegnet som *SPI2_MOSI*. Gjennom bruk av *SPI2_MOSI*-pinnen er det mulig å overføre data til TFT-skjermen.

Pinnen som er betegnet som *SS* (Slave Select) har som formål å velge en spesifikk slaveenhet når det er flere enheter tilkoblet samme SPI-buss. Når *SS*-signalet er aktivt, det vil si lavt, blir den tilknyttede slaveenheten valgt for kommunikasjon. I **Figur 6.3** er denne pinnen definert som *CSX_PORT*.

6.6 Resultat av pulsrate på skjermen

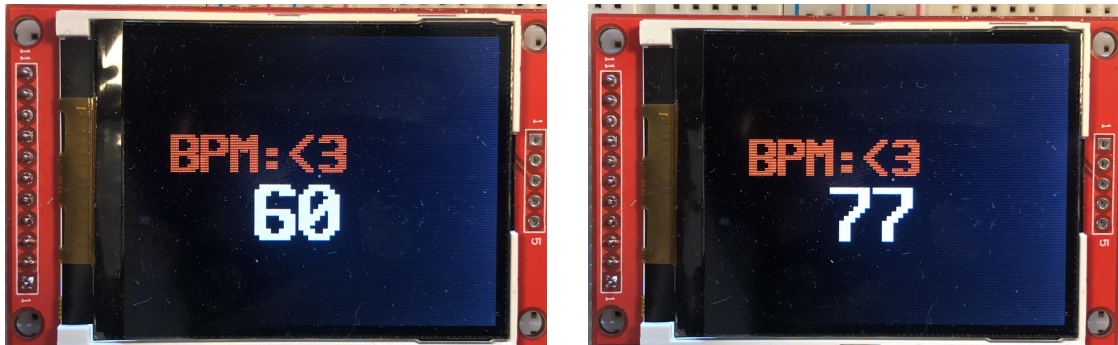
Det overordnede målet med oppgaven var å vise pulsrate som det endelige resultatet på skjermen. Størsteparten av tiden i denne delen gikk med til å analysere zib-filen som ble lastet ned fra github.com, spesielt hjelpefilen *ILI9225.h* og kildefilen *ILI9225.c*. Disse filene inneholdt flere funksjoner og implementasjonen av disse funksjonene. For å sende pulsrate til skjermen via SPI-kommunikasjon, ble funksjonen *draw_string()* brukt.

Funksjonen *draw_string()* skriver en streng til skjermen som en sekvens av tegn ved posisjon x, y med en angitt farge og størrelse. For hver streng som sendes til skjermen, skjer skrivingen én etter én og fortsetter kontinuerlig. Under sendingen av strengen ble det observert at AD-omformereren stoppet og ventet på at sendingen av strengen på skjermen skulle fullføres. Dette resulterte forstyrrelser i målingen av pulstiden. Dette kan antas å skyldes at mikrokontrolleren bruker tid på å utføre skrivingen

¹serial Peripheral Interface

6.6 Resultat av pulsrate på skjermen

av strengen til skjermen. Derfor anbefales det å utforske alternative metoder for å sende strenger til skjermen uten å forstyrre målingen av pulstiden. Resultatet av pulsrate i **Figur 6.5** er ikke egnet for praktiske formål.



(a) Pulsrate ved 1 Hz.

(b) Pulsrate ved 1.3 Hz.

Figur 6.5: Resultat av pulsrate. Tegning < 3 indikerer hjerte.

Kapittel 7

Diskusjon

7.1 Forslag til forbedring

Denne arbeidet har vurdert ulike forbedringsmuligheter knyttet til EKG-signalbehandling og måling. Gjennom analysen har det blitt identifisert flere utfordringer og begrensninger. Hvor begrensningene primært var pålagt av sikkerhetsmessige årsaker. Som et resultat av disse begrensningene ble EKG-signalet realisert ved hjelp av oscilloskopet. Denne realiseringen forenklet oppgaven betydelig, da oscilloskopet genererte et rent EKG-signal. Imidlertid førte dette også til at seksjonen [2.6] om støy i EKG-signalet i kapittel 2 ble neglisjert.

På grunn av denne begrensningen, kapittel 5 om digital signalbehandling er basert på det realiserte EKG-signalet. Dette betyr at algoritmen som ble utviklet for å detektere R-bølgetoppen, ikke er egnet for praktiske formål. Dette skyldes at en mer avansert algoritme, som også ville kreve mer tid, ville vært nødvendig. Derfor bestemte jeg meg for å forholde meg til algoritmen basert på det realiserte EKG-signalet. Dermed foreslår jeg at det utvikles en avansert algoritme for R-bølgetoppdeteksjon, slik at en praktisk anvendbar R-bølgetoppdeteksjon kan oppnås. En kan besøke nettsiden github.com, for å laste ned, utvikle eller dra nytte av koden for en bedre R-bølgetoppdeteksjon algoritme.

Det anbefales også å benytte den foreslått kretsen i **Figur 4.5**. Dette vil tillate at EKG-signalet kan registreres uten behov for å løfte det fra jordnivået ved hjelp av en offset på oscilloskopet. En slik valg vil muliggjøre bruk av et lavpassfilter samtidig som man har tilgang til hele EKG-signalet. Dette vil gi bedre muligheter for å oppnå høyere nøyaktighet og bedre datakvalitet. Samt anbefales det å utforske andre alternative metoder for å sende strenger til skjermen uten å forstyrre målingen av pulstiden.

7.2 Konklusjon

Denne bacheloroppgaven har omfattet en grundig undersøkelse av kretsutvikling for elektrokardiografi og signalbehandling med det overordnede målet om å finne pulsraten. Gjennom oppgaven har jeg studert hjertets elektriske system, forsterket og filtrert EKG-signalet, konvertert det til digital form, utført digital signalbehandling og presentert den resulterende hjertepulsen på en skjerm.

På bakgrunn av min funn og implementasjon av kretsen konkluderer jeg med at en bruke kretsskjema i **Figur 4.5**. Ved å benytte denne løsningen kan behovet for å justere offset på oscilloskopet elimineres, samtidig som behovet for å fjerne høypassfilteret ikke lenger er nødvendig. Dette resulterer en effektiv og praktisk løsning. Videre arbeidet bør rettes mot utviklingen av avanserte algoritmer for R-bølge toppdeteksjon, som vil bidra til å forbedre nøyaktigheten og påliteligheten til hjerte pulsanalyse.

Denne bacheloroppgaven har gitt meg en unik mulighet til å tilegne meg kunnskap og erfaring innen utviklingen av EKG-kretser samt de utfordringene som er knyttet til denne utviklingen. Videre har oppgaven også tillatt meg å fordype meg mer i emner som forsterking, filtrering, analog-til-digital konvertering og signalbehandling. Samtlige av disse tekniske utfordringene har bidratt til en betydelig økning i min dybdeforståelse av emnet. Jeg håper at denne oppgaven vil inspirere til videre studier og utvikling innen kardiovaskulær signalanalyse samt kretsdesign.

Tillegg A

Kode

```
1
2 if (adcFlag) {
3 //Henter AD-verdien og lagre digital verdiene i variabelen adc_verdi
4   adc_verdi = HAL_ADC_GetValue(&hadc1);
5   sprintf(msg, "%hu\r\n", adc_verdi);
6   adcFlag = RESET;
7
8 }
9
10 if (adc_verdi >= 3800) {
11   peak = adc_verdi;
12   current_time = HAL_GetTick();
13   uint16_t time_R_interval = current_time - last_toggle_time;
14   HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
15   last_toggle_time = current_time;
16   pulsen = (60.0 / time_R_interval) * 1000;
17   sprintf(bpm, "%hu", pulsen);
18   HAL_UART_Transmit(&huart2, (uint8_t*)bpm, strlen(bpm), HAL_MAX_DELAY);
19
20
21   // Sender streng til skjermen
22   fill_rectangle(0, 0, WIDTH, HEIGHT, COLOR_BLACK);
23   draw_string(20, 60, COLOR_RED, 2, "BPM:<3");
24   // puls verdien er lagret i char bpm[400];
25   draw_string(70, 90, COLOR_WHITE, 3, bpm);
26
27
28 }
29
30
31 else if (adc_verdi >= 3500 && adc_verdi <= 3800) {
32   peak = adc_verdi;
33   current_time = HAL_GetTick();
34   HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
```

Kode

```
35     uint16_t time_R_interval = current_time - last_toggle_time;
36     last_toggle_time = current_time;
37     pulsen = (60.0 / time_R_interval) * 1000;
38     sprintf(bpm, "%hu", pulsen);
39
40     HAL_UART_Transmit(&huart2, (uint8_t*)bpm, strlen(bpm), HAL_MAX_DELAY);
41
42
43     fill_rectangle(0, 0, WIDTH, HEIGHT, COLOR_BLACK);
44     draw_string(20, 60, COLOR_RED, 2, "BPM:<3");
45     draw_string(70, 90, COLOR_WHITE, 3, bpm);
46
47
48 }
49
50
51
```


Tillegg B

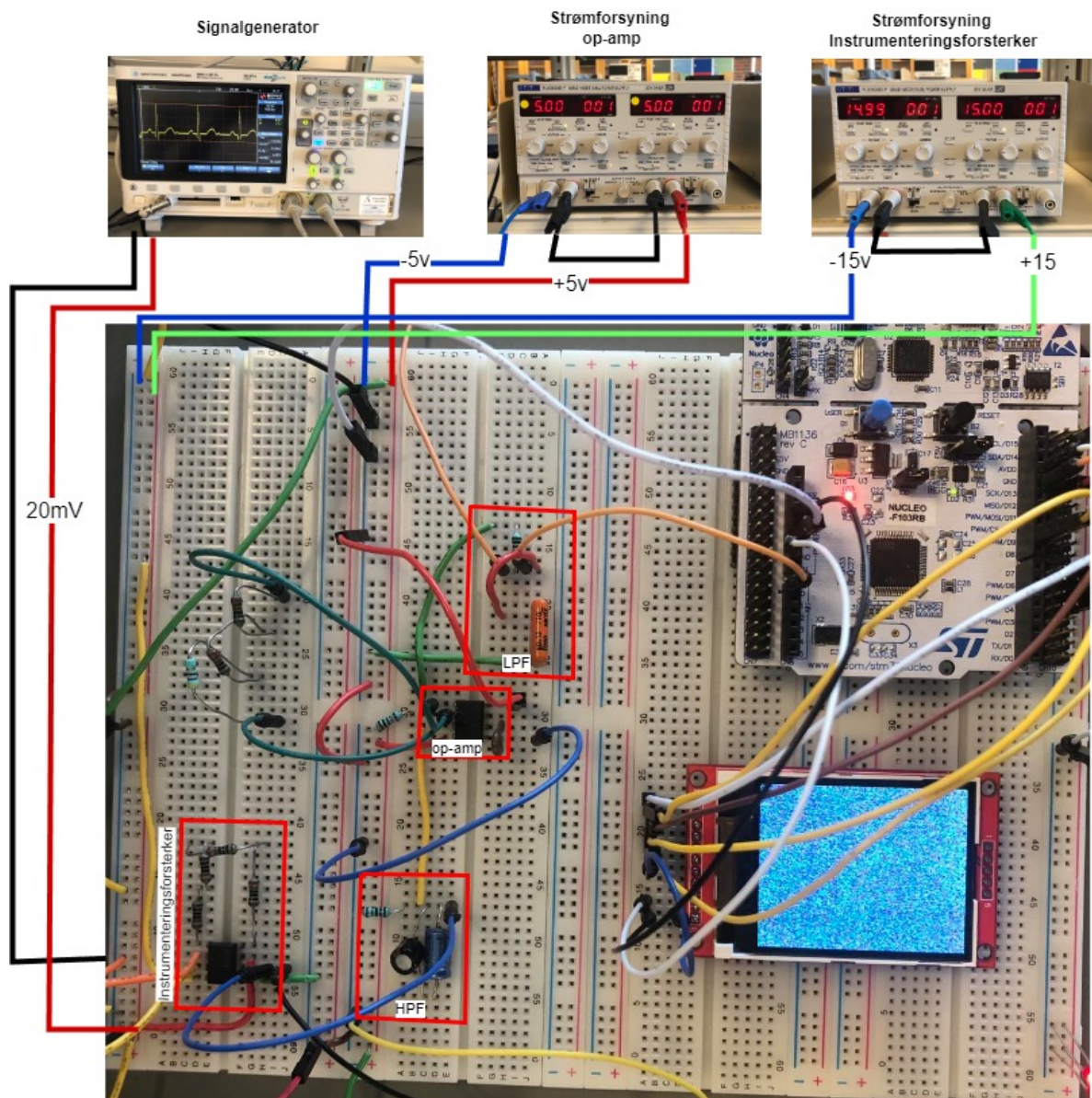
Komponentliste

Komponenter	Type	Verdi	Pris
Instrumenteringsforsterker	INA128		74kr
Operasjonsforsterker	TLV272		
Motstand	RG	20KOmega	
Kondensator	C1	32uF	
Kondensator	C2	1.8uF	
Motstand	R1	10KOmega	
Motstand	R2	10KOmega	
Motstand	R3	10KOmega	
Microkontroll	STM32F103RB		
Display-TFT	LCD ILI9225 SPI		300kr
Oscilloskop	MSO-X 3012A		
Strømforsyning X2	PL303QMD-P		

Tabell B.1: Liste av komponentene som ble bruk i oppgaven. Komponenter som ikke oppgitt pris er fått av universitetet.

Tillegg C

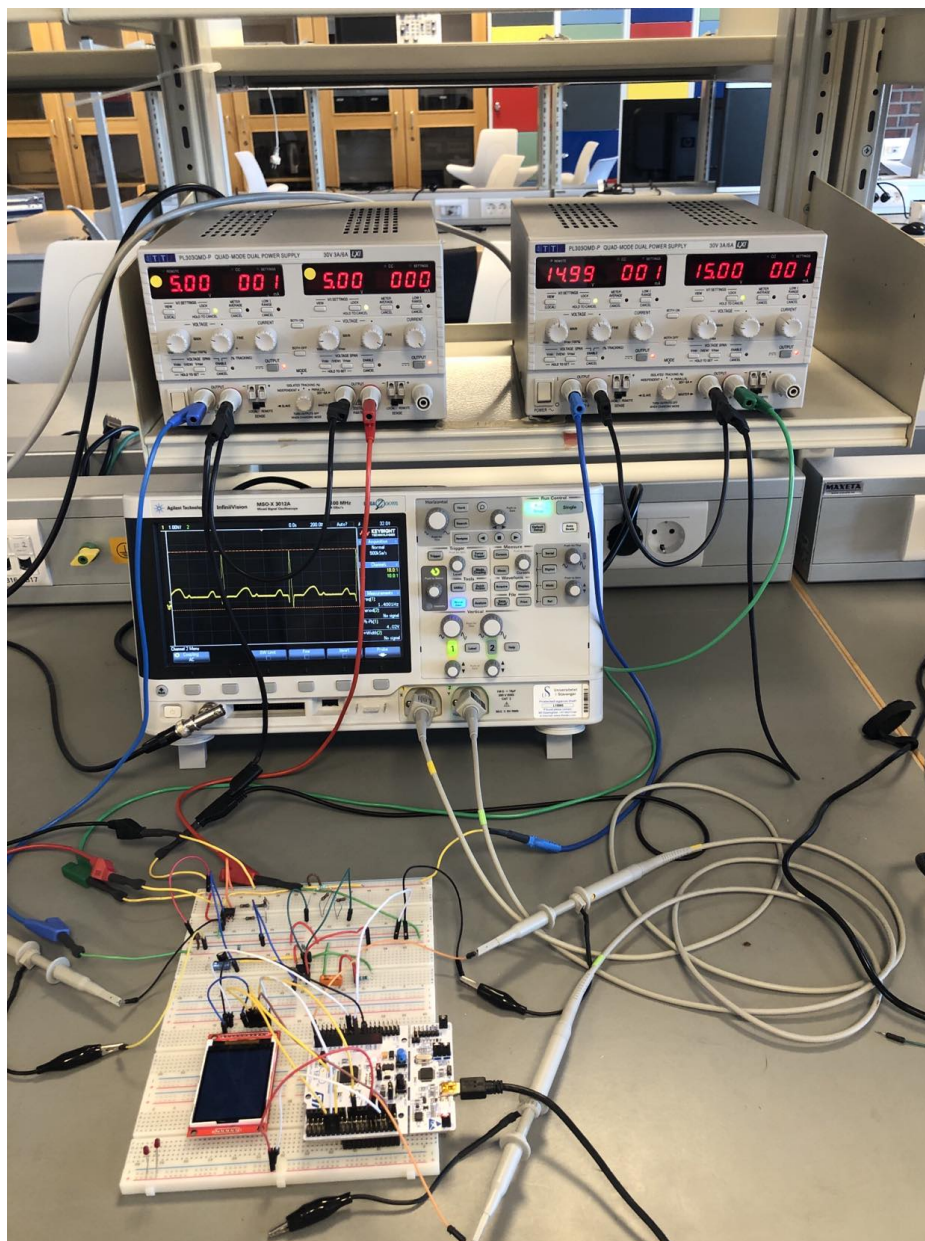
Oversikt av fysisk kretsskjema



Figur C.1: Komplette kretsdiagram tilkobling av hele EKG-systemet

Tillegg D

Laboratoriet



Figur D.1: Viser oversikt av laboratoriet

Bibliografi

- [1] Analog.com. In-AMP basics. <https://www.analog.com/media/en/training-seminars/design-handbooks/designers-guide-instrument-amps-ch1.pdf>. Nettside; besøkt, 18.02.2023.
- [2] Dr. Rahul Kher. Signal Processing Techniques for Removing Noise from ECG Signals. (this work is licensed under the creative commons attribution international license.). <http://www.jscholaronline.org/articles/JBER/Signal-Processing.pdf>. Nettside; besøkt, 16.02.2023.
- [3] Harald Arnesen. EKG. <https://sml.sn1.no/EKG/>. Nettside; besøkt, 3.02.2023.
- [4] ILITEK Enrich visual experience. ILI9225. <https://datasheetspdf.com/pdf-file/653902/ILITEK/ILI9225/1>. Nettside; besøkt, 22.05.2023.
- [5] John R. Giudicessi and Michael J. Ackerman. Potassium-channel mutations and cardiac arrhythmias—diagnosis and therapy. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6590894/>. Nettside; besøkt, 4.06.2023.
- [6] Larisa G. Tereshchenko, Mark E. Josephson. Frequency content and characteristics of ventricular conduction. <https://doi.org/10.1016/j.jelectrocard.2015.08.034>. Nettside; besøkt, 3.3.2023.
- [7] life.augmented. Product overview. <https://www.st.com/en/evaluation-tools/nucleo-f103rb.html>. Nettside; besøkt, 15.05.2023.
- [8] life.augmented. STM32F103RB. <https://www.st.com/en/microcontrollers-microprocessors/stm32f103rb.html>. Nettside; besøkt, 16.05.2023.
- [9] Lovdata. Forskrift om medisinsk utstyr. <https://lovdata.no/dokument/SF/forskrift/2021-05-09-1476?q=forskrift%20om%20medisinsk%20utstyr>. Nettside; besøkt, 5.06.2023.
- [10] Marius Dalby, Knut Gjesdal. Hva sier hvilepulsen om helse og sykdom. <https://tidsskriftet.no/2012/06/oversiktsartikkel/hva-sier-hvilepulsen-om-helse-og-sykdom>. Nettside; besøkt, 4.06.2023.
- [11] Radek Martinek, Martina Ladrova, Michaela Sidikova, Rene Jaros, Khosrow Behbehani, Radana Kahankova, and Aleksandra Kawala-Sterniuk. Advanced bioelectrical signal processing methods: Past, present, and future approach—part iii: Other biosignals. *Sensors*, 21(18), 2021.

- [12] Morten Tengesdal. Frå transistor til datamaskin. https://uis.brage.unit.no/uis-xmlui/bitstream/handle/11250/2494937/Notat_39.pdf?sequence=1&isAllowed=y. Nettside; besøkt, 29.05.2023.
- [13] Norsk Helseinformatikk. Hjertets ledningssystem - animasjon. <https://nhi.no/animasjoner/hjerte-og-kar/hjertets-ledningssystem/>. Nettside; besøkt, 23.01.2023.
- [14] Norsk Helseinformatikk. start av elektisk impuls. <https://nhi.no/sykdommer/hjertekar/hjertearytmier/hjertets-elektriske-system/>. Nettside; besøkt, 24.01.2023.
- [15] Robert Oshana. Nyquist Theorem. <https://www.sciencedirect.com/topics/engineering/nyquist-theorem>. Nettside; besøkt, 19.05.2023.
- [16] Texas instrument. INA12x Precision, Low-Power Instrumentation Amplifiers. https://www.ti.com/lit/ds/symlink/ina128.pdf?ts=1678459205320&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FINA128. Nettside; besøkt, 29.05.2023.
- [17] The heart.org. Normal Electrocardiography (ECG) Intervals. https://emedicine.medscape.com/article/2172196-overview?icd=login_success_email_match_norm®=1#a1. Nettside; besøkt, 1.2.2023.
- [18] Tosin Jemilehin, Michael Adu . Design and Simulation of Electrocardiogram Circuit with Automatic Analysis of ECG Signal (2016). https://www.researchgate.net/publication/334455337_Design_and_Simulation_of_Electrocardiogram_Circuit_with_Automatic_Analysis_of_ECG_Signal. Nettside; besøkt, 7.06.2023.