

Martin Andre Førreisdahl
August Johansen Berg
Vegard André Hansen
Andreas Knutli
Bjørg Solem

Solsensor design og testing til Orbit NTNUs CubeSat

Bacheloroppgave i Elektroingeniør

Veileder: Cuong Phu Le

Medveileder: Andreas Westre og Martin Mostad

Mai 2021

Martin Andre Førriisdahl
August Johansen Berg
Vegard André Hansen
Andreas Knutli
Bjørg Solem

Solsensor design og testing til Orbit NTNUs CubeSat

Bacheloroppgave i Elektroingeniør
Veileder: Cuong Phu Le
Medveileder: Andreas Westre og Martin Mostad
Mai 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for elektroniske systemer



Kunnskap for en bedre verden

Bacheloroppgave

Oppgavens tittel: Solsensor design og testing til Orbit NTNUs CubeSat Project title: Sun sensor design and testing for Orbit NTNU's CubeSat	Gitt dato: 16.01.21
	Innlevingsdato: 20.05.21
	Gradering <input checked="" type="checkbox"/> åpent <input type="checkbox"/> lukket <input type="checkbox"/> åpent fra _____
	Antall sider/bilag: 133/27
Gruppedeltakere: August Johansen Berg, Martin Andre Førreisdahl, Vegard André Hansen, Andreas Knutli og Bjørg Solem	Veileder internt (navn/email): Cuong Phu Le/ cuong.le@ntnu.no
Studieretning: Elektronikk og industriell instrumentering	Prosjektnummer: E2116
Oppdragsgiver: Orbit NTNU	Kontaktperson hos oppdragsgiver (navn/tlf.): Andreas Westre/99387440

Sammendrag (norsk og engelsk) Bachelor-gruppen har hatt som mål å designe og kalibrere en prototype solsensor med Field of View på 60°. Solsensorer er kommersielt sett veldig kostbare, av denne grunnen Orbit å designe og produsere disse in-house. For å oppnå dette har gruppen forsket på teorien bak forskjellige tilnæringer til problemet, og gjennom vårsemesteret 2021 designet en solsensor som er basert på en fire-kvadrants fotodiode. Selve sensoren kalibreres med en kalibreringsrigg med to frihetsgrader som er av gruppens eget design, og i tillegg til dette har gruppen en rekke andre kostnadsbesparende løsninger. Gruppen har simulert forventet resultater i MATLAB, og legger fram de faktiske målingene fra sensoren. Deretter diskuteres og sammenlignes disse, og gruppen konkluderer med at oppgaven er fullt gjennomførbar, men at noen uheldige omstendigheter gjorde at målet ikke ble fullstendig nådd. Som siste del av rapporten skrives det om framtidig arbeid, og hva gruppen mener Orbit bør gjøre som neste steg i prosessen.		The bachelor group was tasked with designing and testing of a prototype sun sensor with a Field of View of 60°. Commercial sun sensors are a considerable expense, therefore Orbit wanted the option to produce them in-house. To accomplish this goal the group had to research different approaches on how to solve the problem, and throughout the spring semester of 2021 they designed a sun sensor based on a four quadrant photodiode. The sensor is tuned by a calibration rig with two degrees of freedom of the groups own design. They have in addition to this devised a number of other cost saving measures. The group has simulated expected values from the sensor through MATLAB, and provided actual data from the sensor. These datasets are then analyzed and compared, and the group reaches the conclusion that the thesis is fully feasible, and that unfortunate circumstances lead to the goal not being fully met. The rest of the thesis is dedicated to future work, and the groups thoughts on Orbits path forward regarding sun sensor design.	
Stikkord: Solsensor, satellitt, software, hardware, kalibrering Quad-photodiode, PCB	Keywords: Sun sensor, satellite, software, hardware, calibration, quad-photodiode, PCB		

Forord

Denne rapporten er blitt gjennomført av en gruppe studenter i sitt siste semester for bachelorgraden i elektroingeniør ved NTNU Trondheim. Prosjektet er det siste som gruppen blir bedømt på, og fullførelsen av emnet utgjør 20 studiepoeng for hver student. Dette er et prosjekt som gruppen har jobbet med siden sent januar 2021 frem til mai 2021 og det er beregnet at hver student skal bruke rundt 500 timer på oppgaven.

Opgaven som gruppen har valgt er å designe en solsensor for Orbit NTNU. Orbit NTNU er en studentorganisasjon som gjør prosjekter relatert til satellitter i verdensrommet. Et spennende prosjekt organisasjonen har aspirasjoner om å gjennomføre er "SelfieSat-prosjektet". Solsensorer er svært dyre og Orbit ønsker i sine framtidige prosjekt, etter "SelfieSat", å produsere disse in-house. For å kunne gjennomføre denne oppgaven måtte gruppen hente informasjon fra artikler som omhandler dette, og sette seg inn i prosessen av å designe og produsere en sensor som oppfylte de kriteriene som ble satt. Målet med oppgaven var å ende opp med et resultat som organisasjonen kan ta inspirasjon fra når de har tenkt til å produsere sine egne.

Bachelor-gruppen ønsker å takke oppdragsgivere Andreas Westre og Martin Mostad fra Orbit NTNU. Begge to har bidratt med masse teknisk hjelp og har vært tilgjengelig for gruppen gjennom hele prosjektet. Gruppen ønsker også da å takke Orbit NTNU for muligheten til å arbeide med en lærerik og interessant oppgave.

En stor takk går også til veileder Cuong Phu Le som har hjulpet til med rådgivning og som har bistått med svar på diverse spørsmål gruppen har hatt i forhold til praktiske deler av oppgaven.

Tusen takk til Microchip Technology Inc. for de to AVR128DB48 Curiosity Nano Evaluation Kitene gruppen fikk i forbindelse med gjennomførelsen av oppgaven.

Gruppen ønsker også å takke MAKE NTNU og Elektronikk og prototypelaboratoriet. MAKE NTNU sitt verksted har vært svært nyttig for gruppen da 3D-printerene og loddeboltene deres har blitt brukt mye. Elektronikk og prototypelaboratoriet har hjulpet med lodding av komponenter slik at gruppen kunne bruke mer tid andre aspekter av oppgaven.

Sammendrag

Bachelor-gruppen har hatt som mål å designe og kalibrere en prototype solsensor med Field of View på 60° . Solsensorer er kommersielt sett veldig kostbare, av denne grunnen ønsker Orbit å designe og produsere disse in-house. For å oppnå dette har gruppen forsket på teorien bak forskjellige tilnærminger til problemet, og gjennom vårsemesteret 2021 designet en solsensor som er basert på en firekvadrant lysdiode. Selve sensoren kalibreres med en kalibreringsrigg med to frihetsgrader som er av gruppens eget design, og i tillegg til dette har gruppen en rekke andre kostnadsbesparende løsninger. Gruppen har simulert forventet resultater i MATLAB, og legger fram de faktiske målingene fra sensoren. Deretter diskuteres og sammenlignes disse, og gruppen konkluderer med at oppgaven er fullt gjennomførbar, men at noen uheldige omstendigheter gjorde at målet ikke ble fullstendig nådd. Som siste del av rapporten skrives det om framtidig arbeid, og hva gruppen mener Orbit bør gjøre som neste steg i prosessen.

Summary

The bachelor group was tasked with designing and testing of a prototype sun sensor with a Field of View of 60° . Commercial sun sensors are a considerable expense, therefore Orbit wanted the option to produce them in-house. To accomplish this goal the group had to research different approaches on how to solve the problem, and throughout the spring semester of 2021 they designed a sun sensor based on a four quadrant photodiode. The sensor is tuned by a calibration rig with two degrees of freedom of the groups own design. They have in addition to this devised a number of other cost saving measures. The group has simulated expected values from the sensor through MATLAB, and provided actual data from the sensor. These datasets are then analyzed and compared, and the group reaches the conclusion that the thesis is fully feasible, and that unfortunate circumstances lead to the goal not being fully met. The rest of the thesis is dedicated to future work, and the groups thoughts on Orbits path forward regarding sun sensor design.

Innhold

1	Innledning	1
1.1	Problemstilling	1
1.2	Bakgrunn	1
1.3	Rapportens struktur	2
1.4	Begrensninger til oppgaven	3
1.4.1	Budsjett	3
1.4.2	Pandemi	3
1.4.3	Integreringen av sensoren på satellitt	4
2	Metoder og kilder	5
2.1	Litteratursøk	5
2.2	Verktøy	5
2.3	Systemdesign	6
2.4	Testing og kalibrering	7
3	Teoretisk innføring og alternative løsninger	8
3.1	Solsensorer	8
3.1.1	Kriterier	9
3.1.2	Forskjellige teknologier	9
3.1.3	Ulike fotodetektoroppsett	10
3.2	Valg av design	12
3.3	Komponenter	12
3.3.1	Fotodioder	12
3.3.2	Operasjonsforsterker	14
3.3.3	Mikrokontroller	15
3.4	Matematikk	15
3.4.1	Utregning av vektor	15
3.4.2	Konvertering for overføring	17
3.4.3	Kalibrering	17
3.5	Lookup-table	18
3.6	Alternative løsninger i hardware	19
3.6.1	Black sun effect	19
3.6.2	Rolling Shutter Effect vba. CMOS-kamera	19

3.6.3	Multiplekser	19
3.6.4	Fototransistorer: Fotodioder vs. fototransistorer	20
3.6.5	Spenningsregulator og lavpassfilter	21
4	Hardware	22
4.1	Komponentspesifikasjoner	22
4.1.1	Quad-Fotodiode: QP5.6-TO5	22
4.1.2	Mikrokontroller: AVR128DB48	24
4.1.3	Quad-Operasjonsforsterker: OPA4188AIDR	25
4.2	Blokkskjema	26
4.3	LTSpice	27
4.3.1	Skjematikk og simuleringer	27
4.3.2	Parameterverdier	28
4.3.3	Feil i kretsen	29
4.4	Altium Designer	30
4.4.1	Prototype 1	30
4.4.2	Støybehandling	34
4.4.3	Prototype 2	35
5	Software	39
5.1	Krav til software	39
5.2	ADC	39
5.3	I2C	40
5.4	Interrupts	42
5.5	Programkoden	42
5.5.1	ADC	43
5.5.2	Utregninger og databehandling	43
5.5.3	I2C	45
5.5.4	ISR og oppstart	47
5.6	Lookup-table	49
6	Kalibreringsrigg	51
6.1	2-DOF-stasjon	52
6.2	Servomotorer	54
6.3	Lyskilde	60

6.4	Ferdigstilt kalibreringsrigg	61
7	Deksel for fotodiodekomponent	63
7.1	Apertur	63
7.2	Endelig deksel-design	69
7.3	Alternative løsninger	71
8	Testing og kalibrering	73
8.1	Lodding og feilsøking	73
8.2	Funksjonstesting	74
8.2.1	Defekt fotodiode	75
8.3	Testing av software	76
8.4	Akkumulering av sampler	76
8.5	Kalibreringsprosess	77
9	Resultater	78
9.1	Gjennomførte målinger	78
9.1.1	Måling med ulike deksel	78
9.1.2	Akkumulering av sampler	79
9.2	Simuleringer	79
9.2.1	Matlab scripts for forventede målinger	79
9.2.2	Målinger fra Matlab script	80
10	Diskusjon	83
10.1	Resultater	83
10.2	Feilkilder	84
10.2.1	Fotodiode	84
10.2.2	Feil i prototype 1	85
10.2.3	Forsinkelser med kretskort til prototype	85
10.2.4	Kalibreringsrigg	86
10.3	Reviderte prosjektmål	86
11	Videre arbeid	88
11.1	Endringer i prototype 2	88
11.1.1	Motstander	88
11.1.2	Fotodioden	88

11.1.3	Kalibreringsrigg	89
11.1.4	Deksel	89
11.2	Software	90
11.3	Videre testing	90
11.3.1	Sammenligning med kommersiell sensor	91
11.3.2	Vakuum-testing	91
11.3.3	Termisk testing	92
11.3.4	Termisk vakuumtest	92
11.3.5	Vibrasjonstesting	92
11.3.6	Strømforbruk	93
11.4	In-orbit kalibrering	93
12	Konklusjon	94
A	Solens strålespektrum	I
B	Spektral respons fotodioder	II
C	Prototype 1	III
D	Prototype 2	IV
F	Måling med ulikt antall sampler	IX
G	Servo program blokkdiagram	X
H	Matlab simuleringer	XIV
I	Datablad for QP5.8-6	XXIII
J	Definisjoner	XXVI

Figurer

3.1	Quad-fotodiode 2D og 3D	16
4.1	QP5.6-TO5 spesifikasjoner[24]	23
4.2	QP5.6-TO5[24]	24
4.3	QP5.8-6-TO5[25]	24
4.4	AVR128DB48 Curiosity Nano Board[28]	25
4.5	Pinnekonfigurasjoner for operasjonsforsterker7[29]	26
4.6	Blokkskjema for første og endelig design	27
4.7	Simulering av hele kretsen.	30
4.8	OPAMP-kobling.	30
4.9	PCB uten komponenter	34
5.1	I2C overføring [33]	40
5.2	I2C overføring av to vinkler fra slave til master	41
5.3	Programvare flytskjema	42
6.1	Rotasjon med posisjonsskift	53
6.2	Avsluttende design for 2-DOF-stasjonen.	54
6.3	Datsett etter kalibrering	55
6.4	Datsett som viser ulineariteten i servoene	55
6.5	Servo Program Front Panel	56
6.6	Illustrasjon av Scan funksjon	58
6.7	Kalibrerings program State chart	59
6.8	Kalibreringsstasjon forfra	62
7.1	Figur som viser 60° FOV med apertur tykkelse 0,3mm	64
7.2	Figur som viser 45° FOV med aperturtykkelse 0,3mm	65
7.3	Figur som viser forhold mellom apertur tykkelse og åpning	66
7.4	Figur som viser forhold mellom apertur tykkelse og sollys som går tapt ved maks vinkler	67
7.5	Målt spenning for varierende dekseltyper	68
7.6	3D-render av dekselet.	70
9.1	Måling med innfallende sollys.	81
A.1	Solstrålingsspektrum[40]	I
B.1	Spektral respons QP5.6-TO5[24]	II
B.2	Spektral respons QP5.8-6-TO5[25]	II

C.1	Skjematikk tilhørende revisjon 1	III
C.2	PCB revisjon 1	III
D.1	Skjematikk tilhørende revisjon 2	IV
D.2	PCB revisjon 2	IV
F.1	Måling med ulike sampler pr. ADC konvertering	IX
G.1	Control State blokk diagram del 1	X
G.2	Control State blokk diagram del 2	XI
G.3	Scan State blokk diagram del 1	XII
G.4	Scan State blokk diagram del 2	XIII
H.1	Script for kvadrant A	XIV
H.2	Script for kvadrant B	XV
H.3	Script for kvadrant C	XVI
H.4	Script for kvadrant D	XVII
H.5	Script for målinger del.1	XVIII
H.6	Script for målinger del.2	XIX
H.7	Måling for kvadrant A	XIX
H.8	Måling for kvadrant B	XX
H.9	Måling for kvadrant C	XX
H.10	Måling for kvadrant D	XXI
H.11	Måling for phi	XXI
H.12	Måling for theta	XXII
I.1	Datablad for QP5.8-6 side 1[25]	XXIII
I.2	Datablad for QP5.8-6 side 2[25]	XXIV
I.3	Datablad for QP5.8-6 side 3[25]	XXV

1 Innledning

1.1 Problemstilling

Problemstillingen ble introdusert av Orbit NTNU. Det som skulle gjennomføres var å designe, produsere og teste en solsensor, som er en av hovedsensorene som brukes for “attitude control” i romfartøy. Slik kunne Orbit NTNU senere ha muligheten til å produsere disse selv, og på dette viset ville de ha muligheten til å bli mer selvstendige i fremtiden. Sensoren må overholde visse tekniske krav, slik at det i den forstand er lagd en liste med kriterier sensoren må oppfylle.

1.2 Bakgrunn

Orbit NTNU er en non-profit studentorganisasjon som arbeider med design og produksjon av mindre satellitter. De oppholder seg på campus Gløshaugen i Trondheim, og har studenter fra mange forskjellige studieprogram. Et viktig aspekt i deres arbeid er å gi medlemmene sine relevant erfaring i avanserte romprosjekter, for å kunne utdanne fremtidens romfartsingeniører. For tiden jobber de mot oppskytningen av SelfieSat, som vil bli deres første funksjonelle satellitt. Målet er i den forbindelse å bli den første norske studentorganisasjonen som presterer å bygge en operativ satellitt.

SelfieSat er en cubesat, en liten satellitt med standardiserte dimensjoner. Satellitten skal være en standard ”2 unit” (2U) satellitt. En satellitt på 1 unit har dimensjonene 10x10x10 cm, og veier typisk under 1,33kg. [1] Hensikten med SelfieSat er at den skal ha en skjerm som skal vise en selfie fra en vilkårlig person på jorden, som skal lastes opp fra bakkestasjonen ved NTNU. Deretter skal et kamera som er festet på en utstrakt arm kunne fotografere bildeskjermen med jorden i bakgrunnen, for så å sende bildet tilbake til personen som lastet opp selfien. [2]

På denne satellitten har Orbit NTNU tatt i bruk fem kommersielle solsensorer, dette er navigasjonsinstrumenter som skal kunne detektere solretninga og som brukes til å orientere satellitten. En kommersiell solsensor er ofte ganske kostbar, den kan ha høyt effektforbruk og den kan være stor i størrelsen. Orbit NTNU fant derfor ut at det kunne være fordelaktig å produsere egne solsensorer in-house, slik at de kommer seg nærmere målet sitt om å kun benytte seg av egendesignede deler. Ved å

produsere egen solsensor vil de kunne redusere kostnader på sikt, påvirke størrelsen på sensoren og det kan øke prosesseffektiviteten. En annen positiv side ved prosjektet er også at de kan endre sensoren dersom de får behov for andre funksjoner, uten betydelige ekstrakostnader.

1.3 Rapportens struktur

Rapporten inneholder 13 “seksjoner”, som beskriver selve oppgaven, metodene og fremgangsmåtene gruppen har benyttet og hvilke løsninger det er blitt kommet frem til.

- Seksjon 1, “Innledningen” beskriver de fastsatte rammene rundt oppgaven.
- Seksjon 2, “Metoder og kilder”, forteller om hvilke typer kilder og metoder gruppen har benyttet seg av.
- Seksjon 3, “Teoretisk innføring og alternative løsninger” inneholder det teoretiske som er nødvendig å vite i forbindelse med rapporten, samt alternative løsninger gruppen hadde for å løse problemstillingen.
- Seksjon 4, “Hardware” beskriver fremgangsmåte og beslutninger tatt i forbindelse med produksjon av det fysiske kretskortet.
- Seksjon 5, “Software” forteller om programkoden som er skrevet, og hvordan denne skal fungere.
- Seksjon 6, “Kalibreringsrigg” beskriver oppbygningen av hele kalibreringsriggen.
- Seksjon 7, “Deksel” omhandler matematikk og designvalg rundt dekselet til solsensoren.
- Seksjon 8, “Kalibrering og testing” beskriver selve test- og kalibreringsprosessen.
- Seksjon 9, “Resultater” blir resultatene lagt frem.
- Seksjon 10, “Diskusjon” inneholder diskusjon rundt resultater og feilkilder.
- Seksjon 11, “Videre arbeid” legger frem muligheter for å arbeide videre med problemstillingen.
- Seksjon 12, “Konklusjon” legger frem konklusjonen gruppen har kommet frem til.
- Seksjon 13, “Vedlegg” inneholder vedleggene til rapporten. Som vedlegg J ligger en liste over orddefinisjoner.

1.4 Begrensninger til oppgaven

I forbindelse med gjennomføringen av prosjektarbeidet, er det satt en del grenser til hva som skal gjøres og ikke, og hva som faktisk kan gjennomføres. Disse begrensningene er basert både på selve oppgavebeskrivelsen, altså hva gruppen har ansvar for med tanke på satellitten, og andre ytre kriterier som ellers kan påvirke hvordan fullførelsen av produktet faktisk blir til slutt, som blant annet pris og leveringstid.

1.4.1 Budsjett

Til å begynne med fikk gruppen innvilget et budsjett på 3000 kr fra Orbit NTNU. Dette var ikke et konkret beløp, så dersom det ble nødvendig, kunne det være muligheter for å søke om mer senere i prosjektforløpet. En del av hensikten bak prosjektet for Orbit NTNU sin del var at de skulle kunne tjene inn på dette i forhold til å benytte kommersielle sensorer. Dette ville bety at sensoren ikke kunne bli for dyr å lage i første omgang eller for å bestille opp igjen ved senere anledning.

Et fokus for gruppen har vært å forsøke å holde seg innenfor budsjetttrammene, såfremt det lot seg gjøre. Samtidig var det viktig å prøve å nå alle prosjektmålene, slik at ytelsen til sensoren helst ikke skulle bli påvirket av budsjettbegrensninger, men samtidig jobbet gruppen for å holde seg til et beløp som kunne bli fordelaktig for oppdragsgiver. Med det budsjettet som var innvilget skulle det være mulig å produsere en velfungerende sensor. Likevel er det klart at noen av valgene, som blant annet komponentvalg og testoppsett, ble påvirket av gruppens ønske om å fullføre prosjektet med så lave kostnader som mulig. Dermed var det klart fra begynnelsen av at sensoren ikke nødvendigvis ville bli perfekt.

1.4.2 Pandemi

Det som kanskje forårsaket mest problemer i forbindelse med oppgaven var nok Covid-19 pandemien. Denne førte til at mange fasiliteter endte opp med å være stengte, eller hvertfall ikke allment tilgjengelige, slik at gruppen måtte bruke en del tid på andre kreative løsninger for å kunne gå videre i prosessen. Noe måtte også bare legges til side, fordi det rett og slett ikke var gjennomførbart på grunn av restriksjonene.

Det faktum at så mye var stengt problematiserte også samarbeidet ved at vanlige møteplasser ikke lenger var tilgjengelige, slik at mesteparten av gruppearbeidet måtte foregå på andre måter, som via nettbaserte metoder. Det ble da en del ekstraarbeid med koordinering og planlegging bare for å møtes. Dette kan ha påvirket resultatet, fordi det førte til at det ble vanskeligere å gjennomføre gruppearbeidet på en god måte.

Dette førte også til at det var vanskelig å få tak i komponentene gruppen opprinnelig hadde tenkt å bruke, på grunn av komponentmangel i verden som følge av pandemien. Blant annet hadde gruppen en type quad-fotodiode som egentlig var tenkt å bruke i prosjektet, men den var kun tilgjengelig for bestilling fra et nettsted der frakten ble alt for høy. På andre nettsteder var den utsolgt, så det eneste alternativet ble å velge en annen fotodiode.

1.4.3 Integreringen av sensoren på satellitt

Det ble lagt frem i starten at denne solsensoren ikke skulle brukes på det førstkomende prosjektet til Orbit NTNU, men at den kunne være tilgjengelig for bruk på senere satellitter. De ville nok mest sannsynligvis jobbe med å forbedre den etterhvert, for å tilpasse den i forbindelse med etterfølgende satellitter. Derfor ble det klargjort at gruppen ikke skulle håndtere selve integreringen av solsensoren på en satellitt.

I sammenheng med at gruppen ikke hadde ansvar for integreringen av solsensoren var det dermed en del ting gruppen ikke behøvde å tenke på. Dette omfattet blant annet andre systemer på satellitten, slik at det ble bestemt at det ikke var nødvendig med komponenter for filtrering av ekstern støy fra andre systemer på satellitten. Derav var det da ikke viktig å sette inn for mye tanke på de andre delene, eller at det skulle være 100% optimalt for verdensrommet. Etter at satellitten som bruker solsensoren er sendt opp, burde det også foretas en ny in-orbit kalibrering, noe som gruppen da følgelig ikke kan ha ansvar for å gjennomføre.

2 Metoder og kilder

For å lære om hvordan oppgaven kan gjennomføres har gruppen tatt i bruk ulike metoder. Stoffet som gruppen samlet individuelt ble så tatt opp og diskutert under gruppemøter for å kunne planlegge og samkjøre oppgavegjennomføringen.

2.1 Litteratursøk

Det viktigste for gruppen under informasjonshevingen var å prøve å finne kilder som virket relevante og som faktisk opprettholdt en viss faglig standard. Dersom det skulle bli tatt i bruk tidligere utførte oppgaver, måtte disse være godt gjennomførte, pluss at de også skulle være baserte på faglig stoff.

I starten av prosessen ble mye av tiden brukt til å gjennomføre litteratursøk og å undersøke bruken og oppbygning av selsensorer, for å kunne samle mest mulig informasjon rundt temaet. Orbit NTNU sendte en del relevante artikler og dokumenter som kunne benyttes for å lese seg opp rundt temaet. Gruppen har også selv funnet flere artikler, og lignende oppgaver gjort tidligere.

Mye av startprosessen gikk også med til selve forprosjektet som er sammenknyttet med prosjektgjennomføringen. Mesteparten av oppgaven som omfattet forprosjektet gikk ut på planlegging av hvordan alt skulle gå for seg fremover med tanke på tid, budsjett og oppgavefordeling.

2.2 Verktøy

Under prosjektet ble det brukt en rekke forskjellige verktøy for design av hardware og software, og for testing av produktet.

- **Altium Designer** - program for å designe kretskort. Et kretsskjema tegnes først opp som så brukes til å designe selve kretskortet i et miljø som tilsvarer hvordan det fysiske kortet vil se ut.
- **Microchip studio** - tidligere **Atmel Studio**, er et program for utvikling av programvare for mikrokontrollere. Det brukes også til å last opp kode til mikrokontrolleren.
- **LTSpice** - program for å designe og simulere elektriske kretser.
- **MATLAB** - matematisk program for å kontrollere ligningene som brukes. Har

også blitt brukt til å simulere.

- **LabVIEW** - program for å utvikle kontrollpanel som ble brukt i forbindelse med kalibreringsriggen.
- **GeoGebra** - matematisk program for grafisk fremstilling av geometri, brukt for å lage grafikk.
- **SolidEdge** - 3D-design; brukes ofte til å lage modeller som senere kan slices og printes av en 3D-printer.
- **Ultimaker Cura** - Et program som tar 3D-modeller og slicer de sånn at de kan printes av en 3D-printer.
- **Adobe Illustrator** - Et program som brukes for å tegne. Det har i denne oppgaven blitt brukt til å lage grafikk.

Makerverkstedet til MAKE NTNU ble benyttet til lodding av komponenter og 3D-printing. I tillegg ble noe lodding gjort av Elprolabben ved NTNU og noe ble gjort med personlig loddebolt. Det ble benyttet et multimeter for testing av kretskortet, og Orbit bidro med en iiglo LED Lommelykt 3500 LM som simulerte sollyset for å teste sensoren.

2.3 Systemdesign

Designet av selve systemet kan deles inn i to deler som videre blir beskrevet i egne seksjoner i rapporten. Disse omfatter hardwaredesign, som er design av selve kortet med komponenter, og softwaredesign, som er fremstillingen av programmet som driver systemet.

Den første delen beskriver kretsdesignet, noe som hovedsaklig foregikk i LTSpice i første runde, for å kunne konstruere og simulere kretsdesignet. Slik var det mulig å få en viss tanke på hvordan systemet kunne fungere i praksis. Store deler av kretsen som ble lagd var inspirert av samme type skjematikk som kunne finnes i relevante kilder, da spesielt et design fra Alvaro Enrich Carmona [3]. Videre ble kretsdesignet og selve kretskortdesignet lagt inn og produsert i Altium Designer.

Samtidig som gruppen ventet på komponenter og holdt på med design og produksjon av kalibreringsrigg ble det designet en ny revisjon av kortet. Visjonen for dette kortet var at det skulle være en del mindre, med mikrokontrolleren direkte påmontert selve kortet og med en bryter for å forhindre OPAMP fra å konstant trekke strøm. Denne revisjonen ble designet i samme program som den første.

Neste del omfatter da softwaredesignet, som skulle inneholde kommunikasjonen mellom kortet og mikrokontrolleren, og omgjøringen fra analogt til digitalt signal. Dette foregikk parallelt med hardwaredesignet. Programmeringen av mikrokontrolleren foregikk hovedsaklig i Microchip Studio, og det skulle blant annet inneholde beskrivelsen av selve signalprosesseringen for å gi ut en god estimering av vinklene.

2.4 Testing og kalibrering

Etter å ha mottatt og loddet på komponentene på prototypekortet ble det jobbet en del med å teste og feilsøke kortet for å forsikre at det fungerte på en tilfredsstillende måte. Dette ble gjennomført ved å sjekke spenning på inn- og utganger til hovedkomponentene med et multimeter, for å se om verdiene stemte overens med de forventede resultatene.

For å kunne gjennomføre kalibrering av kortet måtte gruppen designe og produsere en egen kalibreringsrigg bestående et 3D-printet oppsett som skulle kunne rotere i x- og y-akse ved hjelp av to servomotorer, samt en lyskilde som kunne etterligne sollys. For å kunne styre servomotorene som skulle benyttes for å rotere oppsettet ble det lagd et program i LabVIEW. Kalibreringsdelen omfatter også design av dekslet som skulle ligge over solsensoren. Dette ble designet og deretter produsert ved bruk av 3D-printing. Selve kalibreringen ble ikke praktisk gjennomført, på grunn av en feil i fotodioden som ble benyttet, men selve oppsettet og fremgangsmåten ble ferdigstilt som planlagt.

3 Teoretisk innføring og alternative løsninger

Valget av typen solsensorerdesign er avhengig av hva som trengs fra den i forhold til satellittoppdraget. I forbindelse med dette satte Orbit NTNU spesifikke krav til sensorens nøyaktighet, field of view og samplingstid som måtte oppfylles som designet og komponentene måtte velges ut ifra.

3.1 Solsensorer

Det en solsensor gjør er å bestemme satellittorienteringen ved å registrere egen retning mot sola. Solsensorer brukes mye i romfartøy for å fastslå orienteringen til romfartøyet basert på solens plassering. Dette er viktig med tanke på holdningskontrolleringen av satellitten og for å passe på at de går i riktig bane. De kan også benyttes i navigasjonssystemer i for eksempel ubemannede kjøretøy eller bakkebaserte metrologiske systemer som værstasjoner. Andre mulige bruksområder er også luftfartøy som ballonger eller droner.[4]

Solsensorer kan deles inn i “fine sun sensors” og “coarse sun sensors”, der førstnevnte vil være den mest nøyaktige av de to. Fine sun sensors gir oss informasjon om både høyde og asimut, mens coarse kun behandler høyden. Dette vil også da medføre at fine sun sensors kan være mer komplekse.[5]

Måten en fine sun sensor fungerer på er at lyset fra sola skinner gjennom små åpninger i et deksel, slik at det dannes lysflekker på fotodetektorene under, som dermed gir ut strømmer basert på hvor mye av lyset som treffer hver kvadrant. Slik kan man detektere posisjonen til lysflekkene, også videre arbeide frem vinklene som beskrevet i seksjon 3.4 “Matematikk”. En coarse sun sensor derimot måler strømmen på utgangen. Dermed kan man finne vinkelen fordi strømmen vil være proporsjonal til cosinus av vinkelen mellom sola og fotocellen sin normal. [6]

Det som da ville være mest naturlig å jobbe videre ut ifra var solsensorer av typen “fine”. Det er denne som er mest nøyaktig, og den gir informasjon om både høyden og asimut, noe som er ønskelig. Ved hjelp av hvilken type utgangsverdi man vil få ut, samt hvilket fotodetektoroppsett som blir valgt kan man videre arbeide frem informasjon om solvektoren. De forskjellige valgene under disse kategoriene blir videre diskutert i henholdsvis seksjon 3.1.2 og 3.1.3.

3.1.1 Kriterier

I forbindelse med dette prosjektet spesifiserte Orbit NTNU egne kriterier denne solsensoren måtte oppfylle. Det å redusere kostnader var et sentralt tema i oppgaven som måtte tas hensyn til, men det ble ikke stilt noen spesifikke krav rundt prisen på sensoren, annet enn det første budsjettet som ble lagt frem. Siden sensoren skulle sitte på en CubeSat kunne ikke størrelsen på sensoren bli for stor, men her ble det heller ikke stilt noen spesifikke krav. Tabell 3.1 viser hvilke spesifikke krav som ble stilt til sensoren. Disse kravene var basert på sensoren NanoSense fra Gomspace, som Orbit nå bruker i sine prosjekter, men ble noe modifiserte for prosjektoppgaven.

	Bachelor	Gomspace NS
Nøyaktighet, FOV < 45	$\pm 2^\circ$	$\pm 0.5^\circ$
Nøyaktighet, FOV < 60	$\pm 4^\circ$	$\pm 2^\circ$
FOV	60°	60°
Max sample periode	2 ms	10 ms
Supply Voltage	3.3V	3.3V
Current draw, sampling	10mA	4mA
Current draw, not sampling	<10uA	10uA

Tabell 3.1: Krav til sensoren fra Orbit NTNU

3.1.2 Forskjellige teknologier

Det finnes så ulike “teknologier” solsensorer kan benytte, som gjør at de gir ut forskjellige utgangssignal. Basert på dette kan de splittes inn i disse kategoriene[4][5]:

- Digitale solsensorer, som gir ut diskrete utgangssignal.
- Sun presence sensor, som gir ut en konstant binær output, basert på om solen er innenfor sensorens FOV.
- Analoge solsensorer, som gir ut et kontinuerlig outputsignal som en funksjon av solvinkelen.

Digitale solsensorer opererer ved integrering av en todimensjonal lyssensor og ved signalprosessering for å skille mellom direkte og reflektert sollys.[6] Denne typen passer best til konstruksjoner som krever høy nøyaktighet. En ulempe med disse er at de ofte bruker mye effekt på grunn av omfattende signalprosessering. [7]

Digitale solsensorer har også komplekse design[7] som krever kompetanse utover det gruppen innehar. På grunn av at de benytter digitale bildedetektorer er også designet for stort til å kunne brukes på små satellitter.[8] På grunn av størrelsen og kompleksiteten vil de også være dyrere å produsere. Løsningen med digitale solsensorer ble dermed utelukket ganske tidlig i prosessen.

En sun presence sensor vil gi ut en konstant utgangsverdi så lenge solen er innenfor dens field of view. [4] Det var vanskelig for gruppen å finne videre opplysninger rundt denne typen, men siden den uansett ikke kan gi ut en vinkel som utgangssignal, så gruppen at denne typen ikke ville være tilstrekkelig for oppgaven.

Analoge solsensorer er mindre komplekse enn digitale sensorer, men de har som regel også lavere nøyaktighet.[7] Disse fungerer med lyssensitive komponenter som endrer strøm, spenning eller motstand avhengig av hvor mye lys som treffer de. Det brukes som regel flere slike komponenter, sammensatt i ulike fotodetektoroppsett, der alle utgangssignalene brukes til å beregne posisjonene til solen.[9] I og med at disse er relativt enkle passer de gruppens kompetanse godt.

Det skulle også være mulig å oppnå alle kravene som ble stilt fra oppdragsgiver med en analog solsensor. Orbit ville at det skulle bli tatt utgangspunkt i et lignende design som den typen sensor de allerede benyttet hadde, som var analog. Sammen ble det da konkludert med at det var det beste alternativet for dette prosjektet.

3.1.3 Ulike fotodetektoroppsett

Det finnes også en del forskjellige muligheter å velge mellom når det kommer til designet av solsensoren, basert på fotodetektoroppsettet og design på dekslet. Noen design fokuserer på å øke nøyaktigheten i systemet, mens andre design kan ha som hensikt å øke hastighet. Videre i denne seksjonen vil det bli forklart mer i dybden om hvilke fotodetektoroppsett som ble vurdert av gruppen.

3.1.3.1 Matrix based

Designet benytter en “active pixel sensor” som lysdetektor, dette er en type bildesensor der man vil ha matriser av fotodioder, der hver rad typisk inneholder veldig mange piksler. Dette kan dermed medføre en økt mulighet for høy nøyaktighet, basert på hvilket design man bruker på åpningen i dekslet, samt signalprosesseringen. [9] Dersom man velger å lese av den analoge verdien fra en og en piksel i denne typen

design vil det selvsagt ta veldig stor plass, det vil ta lenger tid og dataprosesseringen vil tilslutt bli veldig kompleks. Alternativt bruker “active pixel sensors” ofte digitale grensesnitt som krever hardware-omformere eller spesialiserte kontrollere for prosessering, noe som da vil gjøre at denne utgår i gruppens tilfelle.

3.1.3.2 Array based

Dette vil være en forenkling av forrige designtype, der det kun blir brukt to fotodiode-arrays som ligger vinkelrett på hverandre, i to akser, slik at man ikke lenger tar i bruk alle pikslene. Dette gjør at denne typen design er mye enklere slik at man kan fjerne behovet for komplekse grensesnitt. Systemet har heller ikke behov for like stort strømtrekk når de fleste pikselene fjernes. Den største ulempen med dette designet er overflatearealet som kreves for å implementere det. [9] Med tanke på at solsensoren som skulle lages i dette prosjektet skal brukes på en CubeSat med relativt lite overflateareal, var denne ulempen ganske betydelig for prosjektgjennomførelsen.

3.1.3.3 Quadrant based

Denne typen sensor bruker fire separate fotoceller som er monterte ganske tett i forhold til hverandre i en 2x2 matrise, slik at en lysfleck som kommer inn gjennom et hull på et deksel vil kunne delvis dekke alle cellene. [10] Designtypen har ganske enkel utforming, og dette kunne forminske det nødvendige arealet sensoren måtte oppta. Samtidig ville også da kostnadene kunne minke samtidig som arealbruken gjorde det. Nøyaktigheten på denne typen design vil øke med distansen mellom detektoren og dekselet, og den samme distansen er omvendt proporsjonal med field of view, noe som vil bety at det er vanskelig å kunne få både en høy nøyaktighet sammen med bred field of view.[9] En annen mulig negativ side med denne typen design er at responsen til fotodioden kan endres sakte over tid, slik at de kanskje må stilles inn igjen mens de er i bruk. [10]

Det er ganske tydelig ut i fra de ulike typene fotodetektoroppsett som kan benyttes, at det var “quadrant-based” som egnet seg best til prosjektet, med tanke på kostnad og kompleksitet. Det er da snakk om en enkel utforming, som lett kan gi et lite areal. Det var også enklere å sette seg inn i dette oppsettet på grunn av all informasjonen som var å finne om det. Det kunne også se ut til at dette oppsettet kan gi gruppen tilstrekkelig field of view og nøyaktighet.

3.2 Valg av design

Solsensoren som Orbit NTNU tidligere har brukt på sine prosjekter var en “fine” solsensor med analog “teknologi” og “quadrant-based design”, dette var dermed noe som veide ganske sterkt i beslutningen for gruppen. Orbit ville gjerne ha noe ganske likt sensoren de tidligere benyttet.

På grunnlag av dette, ble det besluttet at det beste alternativet ville være å lage en analog, “fine” solsensor med et design basert på et kvadrant-oppsatt. Dette var også det valget som passet gruppens kompetanse best, pluss at det var mulig å finne mye informasjon rundt dette.

Videre skulle det settes på et deksel med åpning for å slippe inn sollys på fotodetektoren. Kretskortet som utgjorde selve sensoren skulle også kobles til en mikrokontroller som skulle ta inn data fra fotodetektorene, og dermed gjennomføre signalprosessering av informasjonen for å fastslå retningen mot solen. Det ble også valgt at det beste alternativet for fotodetektor var å benytte fotodioder, selv om det ble spekulert i blant annet muligheten rundt å bruke fototransistor, som diskutert i seksjon 3.6.4.

3.3 Komponenter

Denne seksjonen forteller generelt om virkemåten til de tre viktigste komponentene den typen solsensor som ble lagd i oppgaven består av. Det blir også lagt frem videre informasjon som er viktig å vite om disse i forhold til prosjekt av denne sort, samt hvordan planen var for hvordan de skulle brukes.

3.3.1 Fotodioder

Fotodioder er en type sensorer, basert på halvlederdiodes, som fungerer ved at de genererer en strøm dersom de blir utsatt for lys, altså at den konverterer mottatt lysenergi om til elektrisk energi. Dette skjer ved at når lys treffer PN-overgangen i dioden, vil det skje en ionisering av de kovalente båndene, noe som igjen danner hull og elektronpar i depletion layer. Videre vil det elektriske feltet forflytte de frie, negative elektronene over til n-siden og de positive hullene over til p-siden, som vil gi en økning i omvendt strøm gjennom PN-overgangen. [11] Enheten er typisk forspent i sperreretning, slik at at strømmen er liten i mørket, også øker den proporsjonalt

med mengden lys som treffer den. [12]

Fotodioder har to forskjellige operasjonsmodus, fotokonduktiv- og fotovoltaisk modus. Forskjellen mellom de to er at i fotokonduktiv modus er det lagt på en ekstern spenning på PN-overgangen.[13] Denne modusen gir en økning i responstid og en bedre linearitet for et bredere spekter av lysintensiteter. Den gjør også at fotodioden vil være mer sensitiv.[14] Når den negative forspenningen øker vil også shot noise øke, sammen med at det blir en økning i dark current. [15]

I fotovoltaisk modus har vi null dark current, som resultat av det ikke blir satt på noen forspenning, dette fører til at det kun er termisk støy i kretsen. Når det ikke vil gå noe dark current i komponenten vil den også ha mulighet for mye bedre sensitivitet for lave lysnivåer. Det negative ved denne type modus er at den har ganske lav responstid, siden kapasitansen til fotodioden ligger på max. [13]

I gruppens applikasjon ble det da besluttet å benytte fotokonduktiv modus. Dette vil da gjøre det mulig å få raskere responstid, samt bedre linearitet. Det ble i den forbindelse valgt at katodene skulle kobles til spenningsplanet på kortet, mens anodene skulle kobles inn på inngangene til operasjonsforsterkeren.

Denne typen komponent har spesifikke bølgelengdeområder de er mest sensitive for, såkalt peak wavelength, slik at de er mest effektive i disse områdene. Derfor er det viktig å finne en type fotodiode som fungerer optimalt i samme bølgelengdeområder som sola sin utstråling er mest konsentrert rundt, gitt i figur A.1 i vedlegg A, for å kunne få best mulig utgangsverdi fra fotodioden.

Det ble valgt at fotodiodene skulle kobles opp i en 2x2-matrise, for å oppnå et “quadrant-based” design. Slik vil man dermed kunne finne ut lysets posisjon i kvadranten, i forhold til midten av mønsteret, ved å se på hvor mye strømsignal hver fotodiode gir ut. [16] I samband med dette bestemte gruppen seg for å bruke en såkalt quad-photodiode, en komponent der fire små fotodioder er plassert tett inntil hverandre. Årsaken til denne beslutningen var at det ville forenkle designet, samt muligens gjøre kostnadene lavere. Det forminsket også arealbruken, og det var den beste metoden for å gjøre mellomrommet mellom kvadrantene minst mulig.

For å velge fotodiode hadde gruppen flere kriterier i tankene. Noe av det viktigste i første omgang var at den ikke var for dyr, men at den likevel hadde spesifikasjoner som tilsvarte kravene som ble satt. Blant annet var gruppen ute etter en fotodiode

som hadde mulighet til å oppnå en FOV på 60 grader, samt at det var nødvendig å velge en type fotodiode som hadde riktig spektral respons og liten dark current slik at støymengden forårsaket av dette kunne reduseres.

3.3.2 Operasjonsforsterker

Etter å ha fått ut et gunstig utgangssignal fra fotodiodene, må dette mest sannsynlig konverteres til spenning og forsterkes før det tas videre inn til mikrokontrolleren, da det er snakk om veldig små strømmer som går fra utgangen av fotodiodene. Dette kan gjøres ved å sette inn en operasjonsforsterker, en type integrert krets som kan forsterke svake elektriske signaler. Denne typen komponent kan også brukes som en transimpedance converter[17], som i dette tilfellet vil si at den skal gjøre om strømsignalene til spenninger som mikrokontrolleren kan ta inn.

Siden det ble valgt at et quadrant-based design skulle brukes, ble det klart at det måtte finnes en mulighet å forsterke alle fire utgangssignalene. Det ble lagt frem ulike forslag om blant annet å bruke fire OPAMP-er, en quad-OPAMP, eller en mikrokontroller med innebygde OPAMP-er. De mulige løsningene og endelig valg av hvordan det skulle gjøres blir videre diskutert i senere avsnitt.

Gruppens forståelse var at de fleste fotodioder kobles i revers forspenning, dvs. at de da vil generere negative strømmer på utgangen. For å få positivt signal inn til mikrokontrolleren, var det altså tenkt operasjonsforsterkerne kobles inverterende slik at strømmen ut fra operasjonsforsterkeren ville bytte fortegn. For å avgjøre forsterkningen ble det da lagt inn en inngangsresistans og en feedback-resistans, slik at det negative forholdet mellom de to ville gi størrelsen på forsterkningen. Se formel 3.1, R_f vil i dette tilfellet være R5, R8, R9 og R12, mens R_{in} vil gjelde R6, R7, R10 og R11 i skjematikken til prototype-kortet i vedlegg C.

Dermed var det viktig for gruppen å vite hvor stor spenning mikrokontrolleren trenger å få inn, for å kunne avgjøre hvilken forsterkning operasjonsforsterkeren bør ha. Når man da vet hvilken forsterkning signalet må løftes til, kan man finne logiske verdier på de to resistansene ut i fra denne forsterkningen og standard resistansverdier som finnes på markedet. Gruppen bestemte seg til slutt for å bestille opp flere ulike resistansverdier slik at det ville bli mulig å endre forsterkningen. Det ble valgt å bruke $2k\Omega$ sammen med disse mulige feedback-resistansene: $3.6k\Omega$, $36.5k\Omega$, $390k\Omega$ og $3M\Omega$. På dette viset ville det bli mulig å teste ut de forskjellige

forsterkningene.

$$A = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}} \quad (3.1)$$

3.3.3 Mikrokontroller

I utgangspunktet lå kravet om en implementert mikrokontroller under “kan gjøres” i oppgavebeskrivelsen. Dette ble tidlig endret til “skal gjøres” siden sensoren må kunne utføre beregninger og kommunisere med satellitten.

En mikrokontroller er en liten datamaskin som inneholder en mikroprosessor, minne og flere periferer som lar mikrokontrolleren interface med andre komponenter eller komputere. Den brukes som regel til å styre en prosess eller andre elektriske komponenter i en integrert krets.

Mikrokontrollerens oppgave i solsensoren er å lese av verdiene fra OPAMP-ene, regne disse om og sende de til satellitten. For å oppnå dette kreves en *Analog to Digital Converter* (ADC) til å konvertere verdiene fra OPAMP-ene slik at mikrokontrolleren kan behandle verdiene. Det kreves også en form kommunikasjon, og det ble lagt frem et ønske fra oppdragsgiver om at sensoren skulle kommunisere ved hjelp av I2C.

Gruppen hadde fra før erfaring med programmering av mikrokontrolleren ATmega328P som sitter på Arduino UNO, og det var derfor ønskelig med en mikrokontroller som lignet på denne.

3.4 Matematikk

Matematikken som ble valgt for å beregne verdiene fra diodene er relativt enkel, og beregner en vektor som peker fra sensoren mot sola.

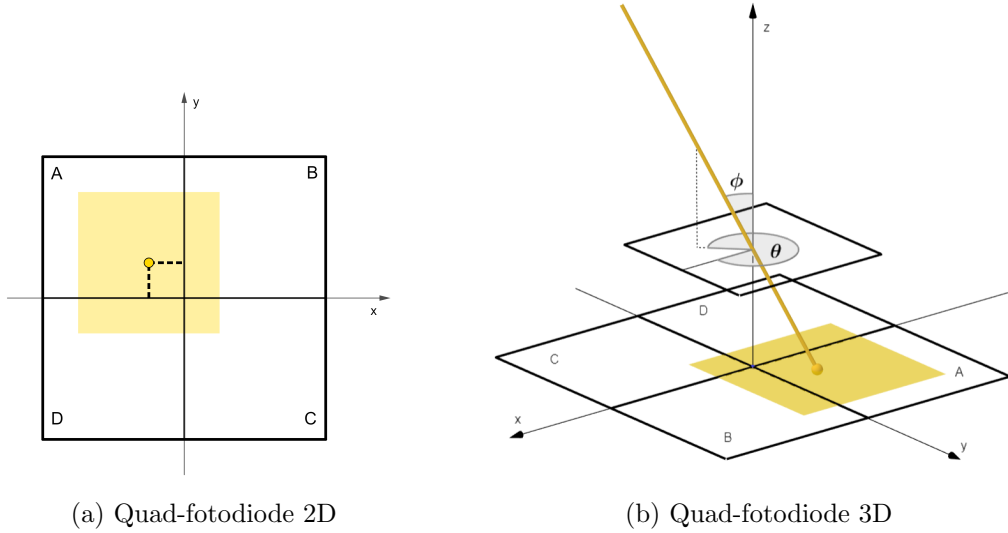
3.4.1 Utregning av vektor

Utregningen av vektoren fra solsensoren til sola foregår i tre steg. Først finnes ett sett med x- og y-koordinater. Disse representerer hvor sentrumet av lyset som treffer fotodioden befinner seg som vist i figur 3.1a. Disse verdiene finnes fra de to ligningene 3.2 og 3.3[18] der r er avstanden mellom origo og kanten på dioden

langs horisontalen. Ligningene baserer seg på forhold mellom de ulike strømmene fra diodene og ikke selve størrelsen på strømmen.

$$x = \frac{B + C - A - D}{A + B + C + D} * r \quad (3.2)$$

$$y = \frac{A + B - C - D}{A + B + C + D} * r \quad (3.3)$$



Figur 3.1: Quad-fotodiode 2D og 3D

Neste steg er å regne x- og y-koordinatene om til vinklene ϕ og θ som vist i figur 3.1b. Dette er to av tre komponenter til de sfæriske koordinatene til vektoren. ϕ er vinkelen mellom z-aksen og vektoren som går fra origo til sola. θ er vinkelen mellom x-aksen og linjen som vektoren fra origo til sola projeksjonerer på xy-planet. For å dekke alle innfallsvinkler med en FOV på 60° må ϕ gå fra 0° til 30° og θ gå fra 0° til 360° . Omregningen fra x- og y-koordinater vises i ligning 3.4 og 3.5[18] der h er høyden fra diodens overflate til linsen på sensoren.

$$\phi = \arctan \frac{\sqrt{x^2 + y^2}}{h} \quad (3.4)$$

$$\theta = \arctan \frac{y}{x} \quad (3.5)$$

Siden θ skal gå fra 0° til 360° må vinkelen korrigeres avhengig av hvilken kvadrant sentrum av lyskilden befinner seg i. Dette er på grunn av symmetriegenskapene til arctan funksjonen, og gjøres ved å legge 180° til vinkelen i 2. og 3. kvadrant, og ved å legge til 360° i 4. kvadrant. Det siste steget er å konvertere de to vinklene til de tre vektorkomponentene x_v , y_v og z_v . Disse finnes fra de 3 likningene i 3.6.

$$x_v = \sin \phi \cos \theta \quad y_v = \sin \phi \sin \theta \quad z_v = \cos \phi \quad (3.6)$$

Ligningene i 3.6 vil gi en enhetsvektor der alle komponentene i teorien vil ha verdier fra -1 til 1, men siden ϕ går fra 0° til 30° vil verdiene være begrenset. Utregningene i 3.7 viser at x_v og y_v vil gå fra -0.5 til 0.5, og z_v vil gå fra 0.8660 til 1.

$$\begin{aligned} x_{v(min)} &= \sin 30^\circ \cos 180^\circ = -0.5 & x_{v(max)} &= \sin 30^\circ \cos 0^\circ = 0.5 \\ y_{v(min)} &= \sin 30^\circ \cos 270^\circ = -0.5 & y_{v(max)} &= \sin 30^\circ \sin 90^\circ = 0.5 \\ z_{v(min)} &= \cos 30^\circ = 0.8660 & z_{v(max)} &= \cos 0^\circ = 1 \end{aligned} \quad (3.7)$$

3.4.2 Konvertering for overføring

Vektorkomponentene er dårlig egnet til å overføres fra solsensoren til satellitten når de er desimaler med negativt eller positivt fortegn. Det er derfor ønskelig å konvertere disse til merkede heltall.

Dette gjøres ved å gange resultatene opp til en større verdi og avrunde de til heltall. Hvor store tallene multipliseres opp til avhenger av hvor stor oppløsning som er ønskelig. Tallene må lagres som en variabeltype som håndterer både negative og positive tall.

3.4.3 Kalibrering

To faktorer som kan korrigeres med kalibrering er avviket mellom de fire kvadrantene på diodene og avvik mellom de faktiske dimensjonene på sensoren og de utregnede dimensjonene.

Et avvik mellom kvadrantene oppstår dersom alle kvadrantene blir belyst med lik mengde lys, men gir ut ulike strømmer. Dette kan korrigeres ved å lyse på dioden med en lyskilde som er sentrert og vinkelrett på dioden slik at både ϕ og θ er 0° . Deretter regner man ut x- og y-verdier og kaller disse x_0 og y_0 . Ligning 3.2 og 3.3 kan så endres til å inkludere disse nye verdiene og blir henholdsvis til ligning 3.8 og 3.9.

$$x = \frac{B + C - A - D}{A + B + C + D} * r - x_0 \quad (3.8)$$

$$y = \frac{A + B - C - D}{A + B + C + D} * r - y_0 \quad (3.9)$$

Et avvik mellom de faktiske dimensjonene på sensoren og de beregnede dimensjonene vil bare påvirke vinkelen ϕ . Dette kan ses fra ligningene 3.2, 3.3 og 3.5, der h ikke inngår i ligningen til θ og r vil kansellere seg selv.

Dersom den avleste ϕ -verdien fra sensoren avviker i forhold til den faktiske innfallsvinkelen til lyset kan dette korrigeres. En ny variabel k introduseres som “endrer forholdet” mellom h verdien i ligning 3.4 og r verdien i ligning 3.2 og 3.3.

Beregning av k -verdien vises i ligning 3.10 til 3.13, der ϕ_1 er den faktiske innfallsvinkelen, ϕ_2 er den avleste innfallsvinkelen og $\Delta\phi$ er avviket mellom de to.

$$\phi_1 = \arctan\left(\frac{\sqrt{x^2 + y^2}}{h}\right) \quad (3.10)$$

$$\phi_2 = \arctan\left(\frac{\sqrt{x^2 + y^2}}{h} * \mathbf{k}\right) \quad (3.11)$$

$$\Delta\phi = \phi_2 - \phi_1 \quad (3.12)$$

$$k = \tan(\phi_1 - \Delta\phi) * \frac{h}{\sqrt{x^2 + y^2}} \quad (3.13)$$

Kalibreringen kan dermed utføres ved å lyse på sensoren med en kjent innfallsvinkel, lese av avviket og beregne en verdi for k .

3.5 Lookup-table

En alternativ løsning til å bruke likninger for å regne ut vinkelene er å ha et lookup-table der data fra kalibrering ligger klar til å leses av. Mikrokontrolleren kan da lese av verdiene fra dioden, og ekstrapolere vinklene ut i fra dette. Denne løsningen er

veldig presis, men har en stor bakside; lagringsplass i RAM-en. I kapittel 6 beskrives en løsning der vi bruker to servomotorer til å styre rotasjonen til dioden, her vil det utdypes litt mer om hvorfor vi valgte bort en løsning som kun er basert på Lookup-table.

3.6 Alternative løsninger i hardware

3.6.1 Black sun effect

En gruppe fra Korea som hadde samme problemstilling som oss designet en solsensor som tar en ellers uønsket effekt kalt “Black sun effect” og bruker den til sin fordel.[19] Denne effekten skjer når pikslene i midten av et CMOS-kamera blir overmettet på grunn av solens sterke lys, som fører til et fenomen der elektroner “renner over” sånn at det på utgangen leses som null sollys. Ut fra dette ekstrapolerer de hvor sola er gjennom analyse av bildet, og dette fører til en høy FOV og relativt høy nøyaktighet.

Denne løsningen ble valgt bort etter å ha presentert den for arbeidsgiver, som virket skeptiske, men også fordi arbeidet som kreves for å gjøre denne løsningen mest sannsynlig ville sprengt tidsbudsjettet vårt.

3.6.2 Rolling Shutter Effect vba. CMOS-kamera

En gruppe fra Kina hadde også samme problemstilling som oss, og tok også i bruk et CMOS-kamera.[20] Til forskjell fra den andre gruppa så valgte de å ta i bruk noe som alle digitale kamera bruker - Rolling Shutter Effect (RSE). Denne effekten får man av at et bilde ikke tas på en gang, men dukker opp etter en veldig rask scan fra den ene siden av kameraets FOV til den andre. Ved bruk av denne effekten så scanner de kontinuerlig etter sola, og ekstrapolerer vinkelen til sola ut i fra dette.

Også denne løsningen ble bortvalgt, mye av samme grunn som forrige løsning. Det har høye kostnader med tanke på tid, og det ville vært usikkert om det var realistisk å få et ferdig resultat innen fristen.

3.6.3 Multiplekser

Det var nødvendig å forsterke signalene som kom fra utgangene på fotodiodene. I og med at det var snakk om et kvadrant-oppsatt ville det være nødvendig med fire

forsterkere. For å da forsøke å unngå dette ble det lagt frem forslag om et design med en krets som tok i bruk en multiplekser (MUX). Idéen var at det ikke var behov for input fra alle kvadrantene samtidig, og at det da kunne brukes kun én OPAMP. AVR-DB, mikrokontrolleren det ble bestemt skulle brukes, har tre innebygde OPAMP-er, så det var tenkt å bruke en eller flere av disse. Planen var å koble opp disse på en måte slik at systemet kontinuerlig kunne lese av ett og ett signal fra hver kvadrant. Ved en multiplekserløsning ville det være mulig å kunne bytte mellom hvilken kvadrant som skulle leses av med et fast mellomrom.

Slik ville det bli mulig å spare både penger og plass på kortet, fordi mikrokontrolleren skulle uansett bestilles, og ved en slik metode ville det ikke vært nødvendig å legge inn fire eksterne OPAMP-er som ville tatt opp plass. Problemet med denne løsningen derimot var at strømmen man får fra fotodioden er såpass liten, og at en analog MUX ofte kan “lekke” strøm når de er i bruk, slik at strømmen til slutt ville blitt for liten. Dette ville gjort denne løsningen vanskeligere å gjennomføre enn å bare legge inn eksterne OPAMP-er. Det kunne også endt med små forsinkelser, siden alle utgangene ikke kunne blitt lest av samtidig. Noen utgangsverdier kunne også gått tapt på tiden det ville tatt å svitsje mellom de fire signalene.

3.6.4 Fototransistorer: Fotodioder vs. fototransistorer

I stedet for å ta i bruk fotodioder hadde gruppen lest at man alternativt kunne benytte fototransistorer i stedet, noe som ville gitt et ganske unikt design, da de ikke ofte blir tatt i bruk i sånne konstruksjoner. En fototransistor er en komponent som vil konvertere lysenergi over til en elektrisk energi, men størrelsen på strømmen vil variere med lysintensiteten som treffer transistoren. [21]

Forskjellen mellom de to er at en fototransistor genererer både strøm og spenning, og den er mer sensitiv enn en fotodiode på grunn av deres forsterkende virkning. [22] Dersom man sammenligner en fototransistor med en lignende fotodiode vil man også kunne finne at fototransistoren ofte kan gi ut en strøm 50 til 100 ganger større enn det fotodioden kan gi ut, den produserer mer strøm når de blir utsatt for lik mengde lysenergi. [23] Hovedgrunnen til at en løsning med fototransistorer ble lagt på is var at utgangsresponsen til en fototransistor visstnok skulle være mye tregere enn den for fotodiode. Fotodioden produserer et utgangssignal mye raskere etter at lyset har nådd overflaten til dioden enn det en fototransistor gjør. Det samme

gjelder også operasjonshastigheten, pluss at en fotodiode vil være mer responsiv i henhold til innkommende lys.

Etter at det ble valgt at gruppen skulle bruke fotodiode sto valget så mellom å bruke fire separate fotodioder, eller en quad-fotodiode. I den forbindelse ble det besluttet at det beste alternativet ville være å bruke en quad-fotodiode. Dette var fordi det ville kunne minke avstanden mellom kvadrantene, og dermed også påvirke arealet på kretskortet. Mellomrommet mellom elementene kan også påvirke summen av utgangssignalene, dersom mellomrommet er veldig stort, vil lysflekken treffe mindre av diodekvadrantene og det vil bli en lavere utgangsstrøm.

3.6.5 Spenningsregulator og lavpassfilter

I utgangspunktet tenkte gruppen å legge inn en egen spenningsregulator som forsynte sensoren med 3.3V for å holde spenningen til en konstant verdi. Etter å ha lagt frem forslaget for Orbit NTNU ble ideen droppet siden satellittens *Attitude Determination and Control System* (ADCS) hadde en spenningsregulator som kunne forsyne sensoren med konstant spenning.

Det ble også lagt inn forslag om et lavpassfilter, bestående av en kondensator og en spole, for å unngå interfererende signal fra andre høyfrekvente systemer i satellitten. Etter diskusjon med Orbit ble det vedtatt at det ikke var nødvendig å legge inn lavpassfilter, fordi det mest sannsynligvis ikke ville være andre systemer på satellitten som kunne påvirke sensoren. Dermed ble også lavpassfilteret fjernet.

4 Hardware

Hardware-seksjonen omfatter den fysiske delen av kortet. Det første som ble gjort etter research-fasen var å implementere det gruppen hadde lært inn i et schematic-design. Utfordringen rundt dette var at det fantes svært lite informasjon å jobbe ut ifra. Alvaro [3] var blant de eneste kildene som kunne finnes som viste et bilde av skjematikken til kretskortet. Vårt design er derfor ganske inspirert av hans løsning. Videre i hardware-seksjonen vil det bli lagt frem komponentspesifikasjoner tilhørende de delene som ble valgt. Det vil også bli diskutert rundt parameterverdier, samt informasjon rundt selve designet av kortene.

4.1 Komponentspesifikasjoner

Etter at gruppen hadde lest seg opp rundt stoffet, ble det besluttet at det var tre hovedkomponenter som var avgjørende for designet. Det er disse hovedkomponentene som utgjør de tre hovedblokkene i skjematikken, og som faktisk er de avgjørende delene i kretsen. Disse tre omfatter da quad-fotodioden, QP5.6-TO5, mikrokontrolleren, AVR128DB48 og quad-operasjonsforsterkeren, OPA4188AIDR. Seksjonen under vil gå litt dypere inn på spesifikk informasjon om nettopp disse tre komponentene.

4.1.1 Quad-Fotodiode: QP5.6-TO5

Markedet for quad-fotodioder så ikke ut til å være særlig stort, og de fleste så ut til å ligge utenfor gruppens budsjett. Siden det var et såpass begrenset antall å velge mellom sto gruppen tilslutt igjen med to komponenter, QP5.6-TO5 og OPR5911. Begge disse har likt spektral-område, fra 400nm til 1100nm. Det er likevel en veldig liten forskjell i peak wavelength, som for førstnevnte er 900nm, mens for OPR5911 er den 890nm. En annen mer vesentlig forskjell mellom de to er at OPR5911 er surface mountable, mens QP5.6-TO5 er through hole. De har også forskjell i driftstemperaturen, da OPR5911 fungerer ved både 15°C lavere temperatur og 25 °C høyere temperatur enn QP5.6-TO5, som kun fungerer mellom -40 °C og 100 °C. Oppbevaringstemperaturen er den samme for begge diodene.

Et problem som kunne oppstått med OPR5911 er at det er varslet i databladet at frontvinduet er trykksensitivt, og at man ikke skal påføre vinduet trykk eller

høyt vakuum. Dette kunne vært problematisk, siden systemet som designes må tåle vakuum. Noen som jobbet med et lignende prosjekt, Álvaro Enrich Carmona, brukte denne dioden. Han la i den forbindelse frem spekulasjoner rundt at dioden kanskje ikke er testet i forbindelse med miljøet i verdensrommet. Videre forklarte han at dioden tålte hans testing i vakuum. [3] Dermed kan det antas at den vil fungere tilstrekkelig, såfremt den testes skikkelig.

Begge komponentene så ut til å være brukbare i henhold til oppgaven, og på andre områder er de forholdsvis like, så valget mellom dem ble til slutt avgjort basert på budsjettet. OPR5911 var ikke tilgjengelig for bestilling på de sidene gruppen var kjent med fra før av, så på de sidene den var tilgjengelig endte den opp med en veldig høy frakt slik at den til slutt ble veldig dyr. QP5.6-TO5 endte på en lavere sluttsum, så det ble bestemt at denne skulle bestilles. Når man ser på de spesifikke karakteristikkene til quad-fotodioden, figur 4.1, ser man blant annet at den har en ganske lav dark current, noe som er ønskelig. Komponentens dark current ligger på 0.2nA.

Electro-optical characteristics @ 23 °C

Symbol	Characteristic	Test Condition	Min	Typ	Max	Unit
	Number of elements		4 quadrants			
	Active area		diameter 2520 (total)			μm
	Active area	per element	1.22			mm ²
	Gap	between elements	24			μm
I_D	Dark current	$V_R = 10\text{ V}$; per element		0.2		nA
C	Capacitance	$V_R = 0\text{ V}$; per element		20		pF
		$V_R = 10\text{ V}$; per element		3		pF
	Responsivity	$\lambda = 632\text{ nm}$		0.4		A/W
		$\lambda = 900\text{ nm}$		0.64		A/W
	Uniformity of responsivity	$V_R = 10\text{ V}$; $\lambda = 880\text{ nm}$; $R_s = 50\ \Omega$		±1	±2	%
t_R	Rise time	$V_R = 0\text{ V}$; $\lambda = 850\text{ nm}$; $R_s = 50\ \Omega$		2000		ns
		$V_R = 10\text{ V}$; $\lambda = 850\text{ nm}$; $R_s = 50\ \Omega$		20		ns
	Shunt Resistance	$V_R = 5\text{ mV}$; per element	250	500		MΩ
	N.E.P.	$V_R = 5\text{ V}$; $\lambda = 900\text{ nm}$; per element		1.3 E-14		W/√Hz
V_{BR}	Breakdown voltage	$I_R = 2\ \mu\text{A}$	20	50		V

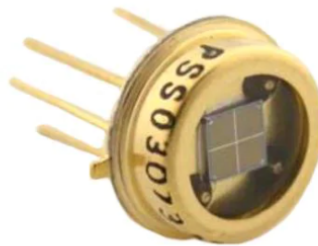
Figur 4.1: QP5.6-TO5 spesifikasjoner[24]

Noe gikk desverre galt med bestillingen av komponenten, slik at da fotodioden ble levert var det feil type. Komponentforsendelsen tok veldig lang tid, slik at når de først kom frem, hadde ikke gruppen mulighet til å prøve å få riktig fotodiode til rimelig tid. Komponentens gruppen mottok var av typen QP5.8-6-TO5, men siden de to komponentene hadde samme footprint og forholdsvis ganske like spesifikasjoner bestemte gruppen seg for å forsøke å jobbe ut i fra denne komponenten i stedet. Hovedforskjellen var at deksel sin utforming måtte endres på grunn av at elementene

var satt sammen til en kvadratform i stedet for en sirkel, se fig. 4.2 og fig. 4.3.



Figur 4.2: QP5.6-TO5[24]



Figur 4.3: QP5.8-6-TO5[25]

Andre forskjeller så ut til å være at QP5.8-6-TO5 hadde et litt større mellomrom mellom elementene, litt større aktivt område og en litt større dark current, men det virket ikke som at dette ville kunne påvirke oppgavegjennomførelsen i særlig stor grad. De to fotodiodene har forøvrig ganske lik spektral respons, se figurer i vedlegg B. Dersom disse ses i sammenheng med solens spektrum illustrert i figuren i vedlegg A, ser de ut til å ha forholdsvis grei spektral respons. Det var uansett vanskelig å finne andre fotodioder med bedre spektral respons. Dette var en viktig del av valget for fotodioden, så dette, pluss at størrelsesmål og andre viktige spesifikasjoner var like, gjorde det mulig å likevel ta i bruk denne fotodioden.

Videre hadde QP5.6-TO5 en peak DC current på 10mA, så dette fungerte som et slags utgangspunkt under simuleringen av kretsen i LTSpice. Gruppen var klar over at det var ganske usannsynlig at strømmen kom til å bli så stor, så dette var et ytterpunkt. Det var heller mer sannsynlig å ta utgangspunkt i 10 μ A, noe som viste seg å være en ganske solid antakelse, for på spesifikasjonene til QP-5.8-6-TO5 er photocurrent opplistet som nettopp 10 μ A. [25][26]

4.1.2 Mikrokontroller: AVR128DB48

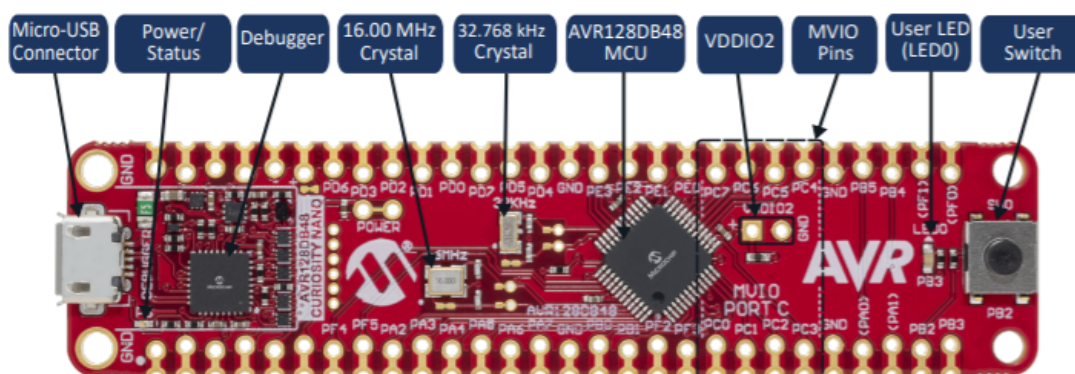
Gruppen planla i utgangspunktet å bruke Microchip's ATmega328P som mikrokontroller, men etter en diskusjon med en av veilederne ble det bestemt å bruke AVR128DB48(AVR). Dette er en nyere mikrokontroller produsert av samme bedrift.

AVR-en har en prosessor som kjører på opptil 24MHz, med 128 KB Flash-minne, 16 KB SRAM og 512B EEPROM. Den har lavt effektforbruk og et

bredt temperaturområde som ligger mellom -40°C og 125°C . [27] For utvikling av programvare og på prototypen brukte gruppen AVR128DB48 Curiosity Nano Evaluation Kit, vist i figur 4.4, som er et utviklingskit som gjør det enkelt å laste opp kode til AVR'en og som inneholder en justerbar spenningsregulator.

Det største argumentet for å benytte AVR-en var at den har 3 innebygde OPAMP-er, og det var i utgangspunktet meningen å benytte disse til å forsterke signalene fra diodene. Dette ble senere forkastet siden sensoren benytter 4 dioder. Det ble derfor bestemt å bruke 4 eksterne OPAMP-er istedet da dette ville forenkle programvaren.

AVR-en har også en 12-bit differentiell ADC som kan konfigureres til 10-bit og single ended. For kommunikasjon har den 5 USART kanaler, 2 SPI kanaler og 2 TWI/I2C kanaler.



Figur 4.4: AVR128DB48 Curiosity Nano Board[28]

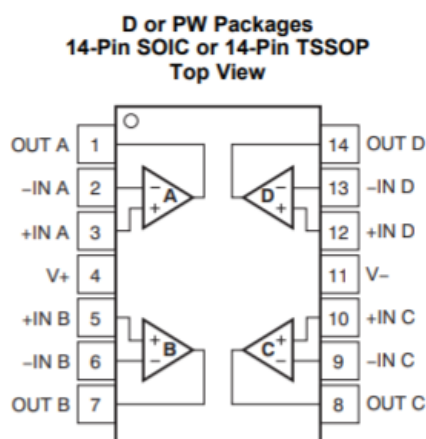
4.1.3 Quad-Operasjonsforsterker: OPA4188AIDR

Valget for operasjonsforsterker falt på OPA4188AIDR, en quad-operasjonsforsterker, som vil si at den fungerer som fire operasjonsforsterkere i en enhet. Denne typen ble valgt i forbindelse med at en quad-fotodiode også ble satt inn, dermed ville det være fire utgangsverdier som måtte behandles. For å forminske arealforbruket var en quad-OPAMP dermed det beste alternativet i stedet for å sette inn fire individuelle operasjonsforsterkere. Pakningen som ble valgt er en 14-pin *Small Outline Integrated Circuit* (SOIC). Pinnekonfigurasjonene er gitt i figur 4.5. Illustrasjonen viser hvordan de fire OPAMP-ene ligger inne i pakningen, med innganger, utganger og spenningsreferansene.

I forbindelse med valget av operasjonsforsterkeren var det viktig å ha en forsterker

som hadde stor nok voltage gain, med andre ord mulighet for å forsterke signalet til en tilfredsstillende størrelse. Denne forsterkeren hadde open-loop gain på 136dB, altså det er mulig å oppnå denne forsterkningen dersom ingen tilbakekobling brukes. Kravet til selve forsterkningen var ikke særlig stort, da det mest sannsynligvis ikke ville bli nødvendig å forsterke signalet veldig mye. Det eneste kravet var at mikrokontrolleren skulle klare å detektere signalendringene.

Andre viktige spesifikasjoner var at quiescent current per amplifier ikke var for høy, dette vil si strømmnivået i forsterkeren når den produserer en utgangsverdi lik null. Dersom dette er ganske høyt vil forsterkeren trekke mye effekt, selv når den ikke brukes. For denne operasjonsforsterkeren ligger dette på 385 til 465 μA for spenningsverdier mellom $\pm 2\text{V}$ til $\pm 4\text{V}$. Videre kan operasjonsforsterkeren operere på spenninger mellom $\pm 2\text{V}$ til $\pm 18\text{V}$. [29] Temperaturområdet til operasjonsforsterkeren ligger mellom -40°C og 125°C , noe som er gunstig med tanke på forholdene i verdensrommet.

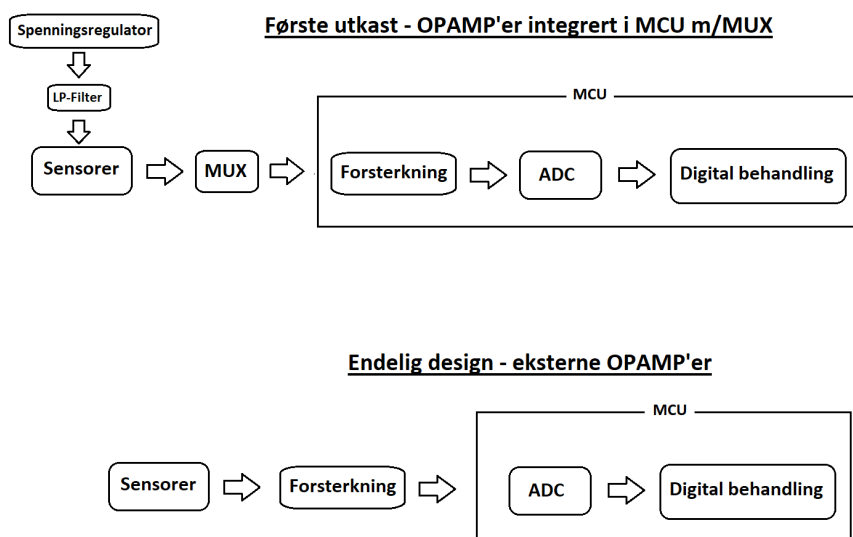


Figur 4.5: Pinnekonfigurasjoner for operasjonsforsterker [29]

4.2 Blokkskjema

Det første utkastet av kretsdesign som gruppen la frem tok i bruk flere av de alternative løsningene som ble diskuterte i seksjon 3.6. Det ble først planlagt å benytte innebygd OPAMP i mikrokontrolleren, og kontrollere hvilke av signalene som skulle leses av fra fotodiodene til enhver tid vha. en multiplekser. I forbindelse med skisseringen av denne modellen, ble det også illustrert et enkelt blokkskjema,

se øverste skisse i fig. 4.6. Dette tidlige utkastet var også basert på at gruppen skulle legge inn både spenningsregulator og lavpassfiltrering, noe som i ettertid også ble kuttet ut av designet. I den nederste illustrasjonen i figur 4.6 kan det endelige blokkskjemaet sees. Dette viser hvordan utgangsverdiene fra fotodiodene skal sendes inn i eksterne OPAMP-er. Dermed skal det analoge signalet så bli videresendt til mikrokontrolleren, omgjort til digitalt og så videre signalbehandlet.



Figur 4.6: Blokkskjema for første og endelig design

4.3 LTSpice

4.3.1 Skjematikk og simuleringer

LTSpice-simuleringene vil kunne gi gruppen informasjon om hvordan elementene i kretsen fungerer med hverandre. Viktige momenter fra simuleringene vil være å lage et godt grunnlag for PCB-designing og teste forskjellige verdier på komponenter for å se hva som gir et ønskelig resultat. For å lage en skjematikk som skal teste ut kretsen på en realistisk måte har det blitt lagt til en strømkilde som skal forestille en fotodiode fra det fysiske kortet. Det er totalt fire strømkilder for å simulere de fire fotodiodene fra komponenten, mer om verdien til strømkildene blir diskutert i et senere avsnitt. Denne strømmen vil kunne gi et spenningsfall over R12, se figur 4.7 og 4.8, og vil til sammen med R8 avgjøre hvor stor forsterkning OPAMP-en vil gi.

OPAMP-en i LTSpice-skjematikken er en helt standard OPAMP komponent som finnes i biblioteket til LTSpice. Dette er et viktig moment i skjematikken da prototypen skal benytte en Quad-OPAMP, slik at fire forsterkere må brukes for å etterligne egenskapene til komponenten.

Punktene hvor det vil være fornuftig å teste spenningsverdier vil være ved inngangen og utgangen til OPAMP-en. Inngangsspenningen ved V- porten til OPAMP-en vil endre fortegn på grunn av inverterende modus og vil bli forsterket med forholdet mellom R12 og R8 slik at utgangsspenningen blir større enn spenningen ved V-.

4.3.2 Parameterverdier

Verdier for komponentene i kretsen har blitt valgt med hensyn til hvor stor strømmen som kan forventes fra fotodioden er. Dette avhenger av følgende faktorer:

- Responsivitet, avhengig av bølgelengde
- Størrelsen på elementene i hver kvadrant
- Solstråling

En kan bruke følgende formler til å behandle disse variablene:

$$(Solar\ Radiation) \times (kvadrant\ element) \times (Responsivitet)$$

Ettersom to av disse verdiene kan finnes i databladet, se sider i vedlegg I, er det kun solstråling som avgjør hvor stor strømmen fra hver kvadrant blir. Selv med denne kunnskapen er det vanskelig å avgjøre hvilken strøm vi kan forvente. For å sette en øvre grense på hvor stor strømmen kan være, fant vi ut at solstrålingskonstanten i verdensrommet er **1380W/m²**. [30].

En typisk verdi for responsivitet finner vi i databladet, som spesifiserer mellom 0.4A/W og 0.64A/W Verdien 0.5 A/W ble valgt. Arealet til de sensitive elementene er 1.44mm².

$$\begin{aligned} 1380W/m^2 &= 1.38W/mm^2 \\ 1.38W/mm^2 \times 1.44mm^2 &= 1.9872mW \\ 0.5A/W \times 1.9872mW &= 0.9936mA \approx 1mA \end{aligned}$$

En strømverdi på 0,8 mA ble valgt for simuleringen, grunnen til dette er at det ble tilsendt feil diode som hadde litt større areal (1.44mm² i stedet for 1.22mm²) enn

den gruppen simulerte med, og simuleringen ble gjort før diodene ankom. Ettersom dette er en øvre grense som vi sammen med teknisk rådgiver anser som usannsynlig valgte vi å beholde kretsen slik den var.

Som nevnt tidligere er det vanskelig å avgjøre hva slags strøm man kan forvente fra fotodioden slik at en lavere forsterkning vil være bedre i dette tilfellet da det ikke er ønskelig med en utgangsspenning som er høyere enn V_{cc} som skal være 3,3 V. OPAMP-en vil kunne miste forsterkningsegenskapene eller bli ødelagt om utgangsspenningen er høyere enn V_{cc} . For å kunne få et fornuftig spenningsfall ved inngangen til OPAMP-en ble en motstandsverdi på 2k ohm valgt slik at motstand R6, R7, R10 og R11 har denne verdien. Spenningsfallet over disse motstanden blir da 1.6V. Motstandsverdiene til R5, R8, R9 og R12 er valgt for å oppnå en forsterkning på 1.8, det vil si at de må være

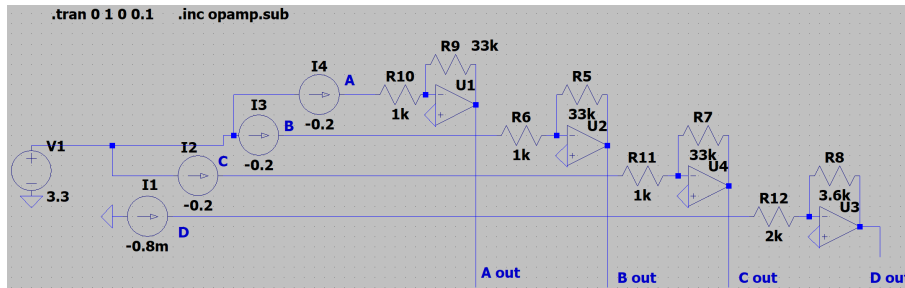
$$1.8 \times 2k\Omega = 3.6k\Omega$$

Den forsterkede utgangsspenningen vil være $1.8 \times 1.6V = 2.88V$ noe som er lavere enn V_{cc} . I tillegg til disse motstandsverdiene har det blitt bestilt andre motstander med høyere verdier slik at det er mulig å få en større forsterkning om den nåværende forsterkningen blir for lav.

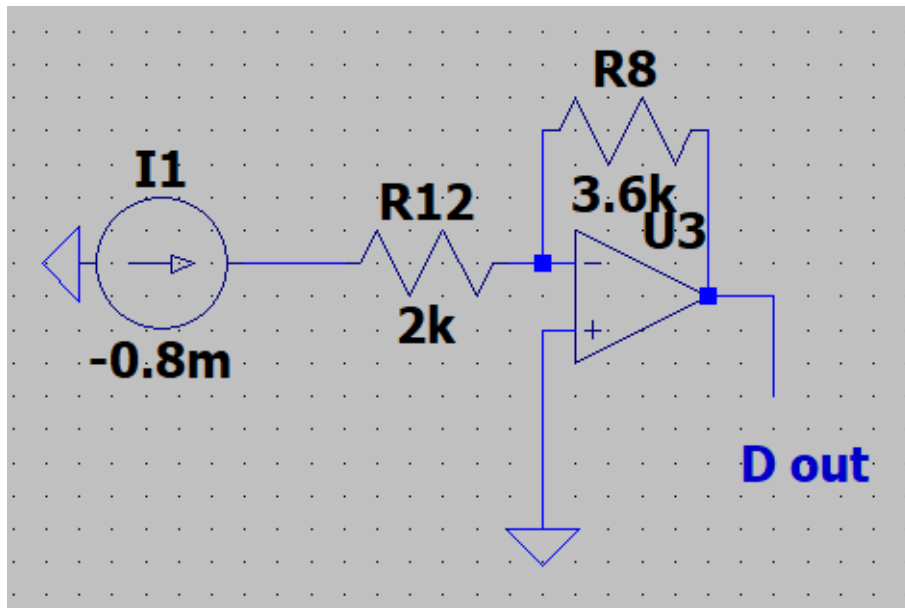
Kondensatorverdien på 100nF er valgt ettersom det er standardisert å ha en avkoblingskondensator for OPAMP-en. En kondensator som er koblet slik vil være ønskelig da den vil begrense støyen i kretsen i tillegg til å holde spenningen som føres inn i OPAMP-en mer stabil.

4.3.3 Feil i kretsen

Etter å ha gjennomgått LTSpice-skjematikken i etterkant av kretskortbestillingen gikk det opp for gruppen at kretsen hadde noen feil. Den første var at hver av $In+$ -polene til OPAMP-en var kobla til jord. Dette fører til at referansespenninga ligger ved jord, og når OPAMP-en prøver å forsterke spenninga inverst så leser man “negativ” spenning på utgangen. Dette klarer ikke ADC-en å lese, så kretsen blir ubrukelig. Den andre feilen var at gruppen koblet en motstand mellom dioden og $In-$ -polene, noe som førte til varierende spenning over dioden og er lite gunstig ettersom det fremmer de ulineære egenskapene til dioder. Videre diskusjon rundt denne feilkilden, pluss informasjon om hvordan gruppen håndterte den etter testfasen står under seksjon 8.1: “Lodding og feilsøking”.



Figur 4.7: Simulering av hele kretsen.



Figur 4.8: OPAMP-kobling.

4.4 Altium Designer

Altium Designer er et program som brukes til å designe kretskort. Man legger inn skjematikken, og hvis alle komponenter har en footprint kan en gå over til PCB-design-modus og lage en visuell modell for hvordan kortet kommer til å se ut når det er bestilt. Dette innebærer blant annet plassering av komponenter, kobberbaner, loddeputer, silk screen, m.f. Under denne seksjonen blir fremgangsmåten og valgene for de to kretskortene diskutert videre.

4.4.1 Prototype 1

Prototype 1 ble laget som et test-kort. Hensikten bak dette var at komponentene skulle være spredt ut slik at det ville bli enkelt å gjennomføre lodding, testing og

feilsøking. Derfor ble dette kortet intensjonelt lagd mye større enn det i realiteten hadde trengt å være. Ettersom en CubeSat kun har sider som er 100cm^2 sprenger disse dimensjonene målene for størrelse på solsensoren, men dette er et valg gruppen tok i samarbeid med teknisk veileder. Videre under denne seksjonen blir det diskutert rundt både skjematikk og PCB-design av den første prototypen i Altium Designer.

4.4.1.1 Skjematikk

Etter at gruppen og teknisk veileder var fornøyd med resultatet fra simuleringen i LTSpice var gruppen klar til å begynne design-prosessen for prototypen til kretskortet. Først ble skjematikken tegnet, som i stor grad ligner på skjematikken fra LTSpice, dog til forskjell fra skjematikken i LTSpice så fantes footprints/modeller for fotodioden og quad-operasjonsforsterker allerede som CAD-modell på Mouser sine nettsider.

Det ble også valgt å legge inn 3 headere i kretsen. To av disse brukes til koble sammen kretskortet og AVR DB Nano-kitet. Fordelen med dette er at vi enkelt kan bruke kitet som spenningskilde, og direkte lese av utgangspinnene uten å permanent feste mikrokontrolleren til kretskortet. Den siste headeren brukes til å gi en utgangsverdi for I²C-koblingen som diskuteres videre under kapittel 5 Software.

Det var et fokus rundt å gjøre skjematikken enkel og oversiktlig, slik at det skulle bli lett å forstå den, pluss at dette forenklet PCB-designingen. Det var viktig å bruke mye plass, slik at den skulle bli tydelig. Dette punktet gikk også videre over til PCB-delen for denne prototypen. Det har blitt forsøkt å lage sporene slik at de går så direkte som mulig, for at det ikke skal oppstå noen forvirring rundt hvor de skal. Dette viste seg også å være hjelpsomt i forbindelse av skjematikk-design til neste revisjon. NTNU Orbit vil også ha stor nytte av en oversiktlig skjematikk, da det gir dem muligheten til å enkelt gjøre endringer ved behov.

Som nevnt tidligere var planen i utgangspunktet å koble katodene i fotodioden til VCC, for så å koble anodene videre til en inverterende operasjonsforsterker. På dette viset trodde gruppen at utgangsverdiene fra fotodioden kom til å være negative, slik at dette kunne inverteres ved å benytte operasjonsforsterkeren inverterende. Derfor ble det også lagt inn tilbakekoblingsnettverk med motstandsverdier som var tiltenkt å gi ut forsterkninger basert på forholdet mellom resistansverdiene som ble loddet på.

4.4.1.2 PCB

Selve kretskortdesignet av det første kortet var enkelt å gjennomføre, siden skjematikken ble jobbet gjennom på en oversiktlig og ryddig måte. For å lage PCBen til prototype 1 ble det lagt til footprints til alle komponenter, også ble de plassert utover i et mønster gruppen mente ville være oversiktlig. Det viktigste var å legge operasjonsforsterkeren nær den tilhørende avkoplingskondensatoren. Det var også viktig å ha operasjonsforsterkeren nær fotodioden, slik at det skulle bli minst mulig tap i kobberbanene mellom komponentene, da strømmen som generes i fotodiodene kan forventes å være lav. For de resterende komponentene jobbet gruppen med å få kortet symmetrisk, samt at det skulle være god plass mellom komponentene for å forenkle loddeprosessen.

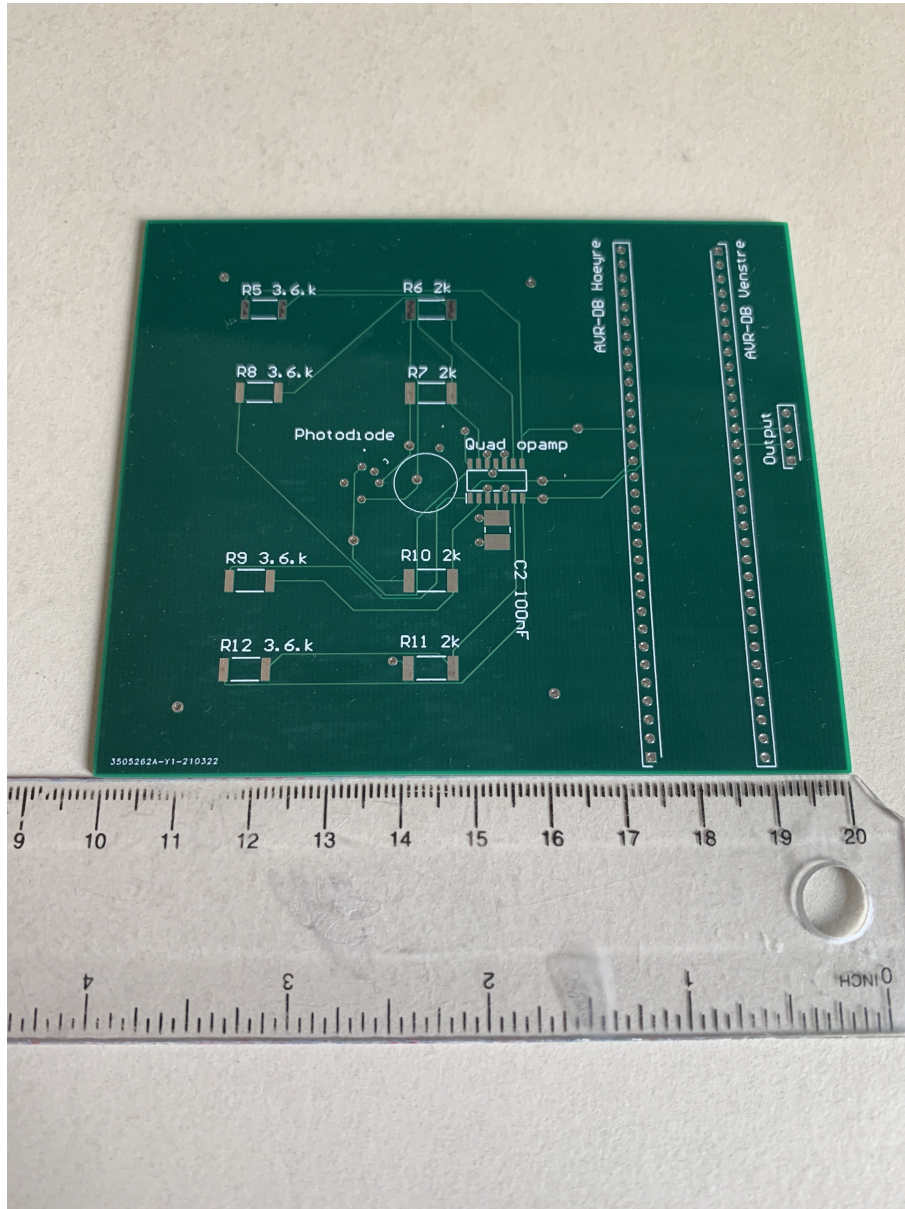
Videre designing ble hovedsaklig gjort ved hjelp av Auto Route-funksjonen i Altium. Denne lager spor for brukeren basert på det som er designet ut ifra skjematikken, slik at det eneste som behøves er at skjematikken har riktige tilkoblinger i utgangspunktet. Denne lager da kobberbaner ut ifra koblingen i skjematikken og markerer hvor komponentene er blitt plassert på selve kortet, samt driller “through holes” der det trengs. Auto Route i et program som Altium vil ikke alltid gi det beste resultatet, men da dette var en prototype som hadde et ekstra tidspress med tanke på at gruppen måtte bruke denne modellen til å teste hvordan kretsen fungerer, var dette godt nok.

I og med at dette var en prototype var det mer fokus på funksjon enn utseende, og av den grunn ble det valgt å bruke mikrokontrolleren som spenningskilde. Kortet ble designet slik at VTG-pinnen på AVR DB mikrokontrolleren ble koblet til det 2. laget i PCB-en. Dette gjorde at kortet fikk et eget VCC-plan, slik at alle VCC-tilkoblinger kunne gå til dette planet ved hjelp av VIA-hull i PCBen. For det endelige designet er det ønskelig å legge inn spenningsforsyningen på et annet vis, da kretskortet skal kunne gå inn i en CubeSat, slik at det skal få egen spenningsforsyning fra satellitten. Denne spenningsforsyningen hadde selvsagt ikke gruppen tilgjengelig, så denne metoden ble valgt for enkelhetens skyld, pluss at det ga riktig utgangsspenning.

Siden det ble lagt inn en spenningsplan, var det også logisk å legge inn et jordingsplan, som skulle ha samme funksjonalitet som spenningsplanet, ved at alle punkter som skulle jordes kunne kobles direkte til dette. Eneste forskjellen er da at dette ligger på 0V istedet for 3.3V. Grunnen til at dette ble gjort er fordi PCBer

med et multiplum av 2 antall lag er fordelaktig da dette er standarden for PCB-produsenter, i tillegg til at man kan få problemer med den strukturelle integriteten til kortet dersom det ikke er lagt opp slik. Det kan forekomme bøyinger dersom man ikke har et multiplum av 2 antall lag etter at man har loddet på komponenter.

Dette med å bruke et spenningsplan og et jordplan kan gjøre kretskortet mer oversiktlig, og det kan bli lettere for Orbit å lage nye kort eller endre på det i fremtiden. Det har også forminsket lengdene på ledningsbanene. Erfaringen til gruppe medlemmene med tanke på lodding er ganske lav, så det å bruke stor plass og store komponenter gjorde det litt lettere. I dette kortet ble det brukt surface-mounted resistanser med komponentstørrelse på 2512. Det endelige PCB-designet til prototype 1 kan sees i vedlegg C, og i figur 4.9 kan kortet sees uten komponenter sammen med en linjal for størrelsessammenligning.



Figur 4.9: PCB uten komponenter

4.4.2 Støybehandling

Et viktig konsept i forbindelse med kretskortdesign er å arbeide for å begrense støydannelsen i kortet. Metoder for å begrense mengden støy er blant annet å unngå å bruke 90° vinkler i ledningsbanene, da disse kan føre til Elektromagnetisk forstyrrelse (EMI). Det er også fordelaktig å korte ned lengden på banene, slik at det ikke er så lang avstand mellom komponentene.

Dermed ble det lagt fokus på det med å forsøke å redusere støy. I den forbindelse ble

det tatt forhåndsregler ved å blant annet forsøke å plassere komponenter som burde ligge sammen nærme hverandre, deriblant fotodioden og operasjonsforsterkeren. Dette gjorde da at noen av ledningsbanene betydelig kortere, noe som kan redusere støydannelse. Også det med å ha flere lag på kortet gjorde at ledningsbanene kunne kortes ned en del, fordi det var mulig å bruke via-hull direkte ned til jord eller power-planet.

Det var også nødvendig å legge inn en støydempende avkoplingskondensator fra $V+$ til jord i forbindelse med quad-OPAMP-en. Det var oppgitt i databladet i komponenten at det burde koples opp 0.1- μF kondensatorer mellom hver spenningsforsyning og jord, men siden det kun ble brukt en spenningsforsyning holdt det da med en kondensator. Denne ble plassert så nærme operasjonsforsterkeren som mulig.

Andre muligheter for å redusere støy, som blant annet å legge inn et analogt filter ble diskutert med oppdragsgiver, men det ble til slutt avgjort at det ikke ville være nødvendig. Det ble også bestemt at det også ville være best å prøve å redusere antall komponenter på prototypen, for å prøve å få den så billig og liten som mulig, slik at andre komponenter eventuelt kunne bli lagt til på senere revisjoner. I sammenheng med dette ble også innleggelse av flere avkoplingskondensatorer ansett som unødvendig, da det ikke var flere komponenter som spesifikt behøvde det.

4.4.3 Prototype 2

Det var planlagt at det neste designet skulle gjøres mindre. Hensikten med den første prototypen var å kunne lage et design som var optimalt for å få enkelt loddet det sammen, og for testingen sin del. Dermed var det meningen at etter at kortet hadde blitt testet skulle det optimaliseres. Det kom uforutsette hendelser som problematiserte muligheten for å rekke å bestille opp kort nummer to og sette det sammen innenfor tidsskjemaet. Etter å ha diskutert det med oppdragsgiverne kom det frem at det var tilstrekkelig med prototypen.

En mulighet ville vært å bestilt opp et kort før prototypen var satt sammen, men dette ville ført til at det ville vært svært lite endringer fra en revisjon til den neste. Det eneste som kunne blitt forbedret ville isåfall vært størrelsen på kortet. Derfor besluttet gruppen at det var mulig å designe et nytt kort med modifikasjoner etter å ha satt sammen og testet prototypen. Orbit kunne dermed få skjematikk og

tilhørende dokumenter til dette desginet, uten at gruppen faktisk bestilte kortet og komponentene. Navnet på hvilke komponenter som er lagt inn står ved siden av komponentene i skjematikken, vedlegg D, dette skal gjelde alle bortsett fra resistansene, der det er mulig å bestille opp andre verdier.

Det er likevel en del som kanskje burde finpusses på dette kortet også, for å få enda bedre resultater. I og med at testfasen til gruppen ble forkortet på grunn av flere uforutsette forhold, kunne ikke alt testes så nøye som gruppen hadde sett for seg. Disse mulige endringene som kan måtte gjøres blir diskutert videre i seksjon 11.1 “Endringer i prototype 2”. Dette var med andre ord ikke et endelig resultat, men skulle heller fungere som en prototype 2.

4.4.3.1 Skjematikk

På det nye kortet var det viktig å få arealet mindre, en god måte for å redusere størrelsen på kortet var å legge mikrokontrolleren AVR-DB direkte på kortet, i stedet for å sette inn hele kitet, som da selvsagt er mye større. Ellers har det blitt lagt inn en switch, og rettet opp i feilen som ble oppdaget i revisjon 1. Resten av skjematikken er veldig lik den fra revisjon 1.

Det at kretsen ikke fungerte som tenkt på forrige kort, var noe som måtte fikses i denne revisjonen. Det som ble gjort var å koble opp skjematikken på den nye versjonen på samme vis som vår eksterne veileder, Martin Mostad, gjorde for å fikse forrige PCB. Det som ble gjort var å koble inn VCC inn på OPAMP-ens positive terminaler, og å koble fotodiodens utganger direkte inn på negativ terminal på OPAMP. Den nye skjematikken kan sees i vedlegg D.

Gruppen ble også gjort oppmerksomme på at operasjonsforsterkeren som ble brukt i revisjon 1 ikke har tilstrekkelig spenningsforsyningsområde. Dette skjedde fordi gruppen leste databladet feil. Siden det sto $\pm 2V$ til $\pm 18V$, tolket gruppen det som at det ville fungere med 3.3V spenningsforsyning. Men, fordi kun den positive power supply-pinnen blir brukt, er det i realiteten nødvendig med 4V spenningsforsyning. Hadde derimot begge blitt brukt, ville det vært tilstrekkelig med så lite som 2V på hver. Derfor valgte gruppen å sette inn en LM124D på prototype 2 istedet. Dette er en quad-OPAMP med samme case som OPA4188AIDR, den har godt temperaturområde og trenger kun 3V spenningsforsyning hvis man bruker single supply.[31] Et annet alternativ kan være å bruke fire operasjonsforsterkere istedet.

En annen del med det forrige designet som mulig var litt problematisk var selve oppkoblingen til operasjonsforsterkeren med tanke på strømtrekk. Slik den ble koblet inn vil den konstant trekke strøm, selv om den ikke brukes. Et forslag til denne revisjonen var dermed å legge inn en switch for å kunne bryte strømtilførselen når den ikke skal være i bruk. Det var da tenkt å legge inn en bryter ved strømforsyningen til operasjonsforsterkeren, som kan åpnes og lukkes på kommando fra mikrokontrolleren.

I forbindelse med dette var nok det beste alternativet å legge inn en NPN-transistor som kunne fungere som en bryter. Det ble besluttet å legge inn en PEMD13 fra Nexperia USA Inc., en komponent som ble foreslått til gruppen av ekstern teknisk veileder. Transistoren har innebygde bias-reistanser,[32] noe som kommer godt med i forbindelse med arealbruken, da det ikke er nødvendig å legge inn flere resistanser.

4.4.3.2 PCB

Det nye designet ble lagt opp til å være en del mindre, og dimensjonene er: 35.433mm i X-retning og 27.686mm i Y-retning. Grunnen til at det var mulig å gjennomføre forminskingen av kortet er fordi alle resistansene på kortet og avkoplingskondensatoren før operasjonsforsterkeren ble valgt med komponentstørrelse 0603. Denne størrelsen for disse komponentene er mindre enn det som tidligere ble benyttet for disse komponentene.

Implementeringen av selve mikrokontrollerchipsen istedenfor hele kitet gjorde det mulig å forminske arealet på kortet enda mer. For å kunne gjøre dette måtte minimumsavstanden mellom pads i Altium Designer gjøres om fra 0.254mm til 0.20mm, fordi dette er den virkelige avstanden mellom padsene på mikrokontrolleren. Dette gjaldt også da avstand mellom tracks, siden pins som ligger rett ved siden av hverandre på mikrokontrolleren blir tatt i bruk i designet.

Det er også forsøkt å lagt opp kortet slik at fotodioden er i midten av kortet, samt at den er rotert 45° for å få katoden pekende nedover. Dette gjør at en senere kalibreringsprosess skal bli enklere å jobbe med og gjennomføre. På dette viset vil det ikke lenger være nødvendig å legge inn ekstra materiale i kalibreringsriggen for å få fotodioden i midten. Det er også blitt lagt inn en switch i kortet, mellom operasjonsforsterkeren og dens strømforsyning, som forklart tidligere. Skjematikken og PCB-designet kan sees i figurer D.1 og D.2 i vedlegg D.

For å prøve å forminske støy i denne versjonen ble det arbeidet med å forsøke å gjøre ledningsbanene så korte som mulig. I den forbindelse ble det også lagt inn flere plan på dette kortet, slik at det var et firelags kort med Top layer, VCC, GND og bottom layer. Dette så ikke ut til å øke kostnaden stort på forrige bestilling, pluss at det minket lengden på ledningsbanene betraktelig, slik at det kanskje kan begrense støydannelsen.

På dette kortet ble det ikke benyttet auto-routing, for å prøve å sikre at det ville gi det beste mulige resultatet. Slik kunne igjen ledningsbanene forhåpentligvis kortes ned, og det gjorde det enklere å følge med om alt ble riktig koblet. Dette er klart litt mer tidkrevende, men det utgjorde ikke en signifikant forskjell. I og med at kortet uansett ble utviklet mens gruppen måtte vente på å få gjort ferdig andre deler hadde det uansett ikke noe å si for fremgangen i oppgaven. Dette ble kun gjort for å være på den absolutt sikre siden, da bruken av auto-route i forrige kort uansett ikke viste seg å medføre noen feil.

5 Software

For å digitalisere verdiene fra dioden og for å bestemme retningen til sola ut fra disse ble det utviklet programvare som skulle kjøre på mikroprosessen. Oppgaven til programmet er å regne ut og overføre en vektor som peker mot sola til satellitten når den ber om det. All programkoden er skrevet fra bunnen av i C, og Microchip Studio ble brukt for å skrive koden og laste den opp til AVR'en.

5.1 Krav til software

Kravene til software fra Orbit var relativt løse og er bare oppgitt som digitalisering, filtrering og støydemper. I tillegg er det et konkret krav som sier at maks samplingstid skal være 2ms og det er ytret ønske om at kommunikasjon skal foregå ved hjelp av "Inter-Integrated Circuit" (I2C) protokollen.

5.2 ADC

En "Analog-to-Digital-Converter" (ADC) konverterer det analoge signalet fra dioden til et digitalt signal som mikrokontrolleren kan behandle. AVR-en har en innebygd ADC som benyttes. Denne kan konfigureres som en 10-bits eller en 12-bits ADC med mulighet for både single-ended og differensiell inngang og den har flere interne spenningsreferanser å velge mellom i tillegg til eksterne[27].

Den kan også konfigureres til å akkumulere opp til 128 sampler under en konvertering. Dette kan brukes til å redusere påvirkningen av støy på signalet. Hvor mange sampler som kan akkumuleres begrenses av kravet til samplingstid som ble spesifisert fra Orbit. Ligning 5.1 viser hvordan samplingstiden til ADC'en er endres med n antall sampler, der $f_{CLK_{ADC}}$ er samplingshastigheten til ADC-en og $f_{CLK_{PER}}$ og hastigheten på klokken til mikrokontrolleren.

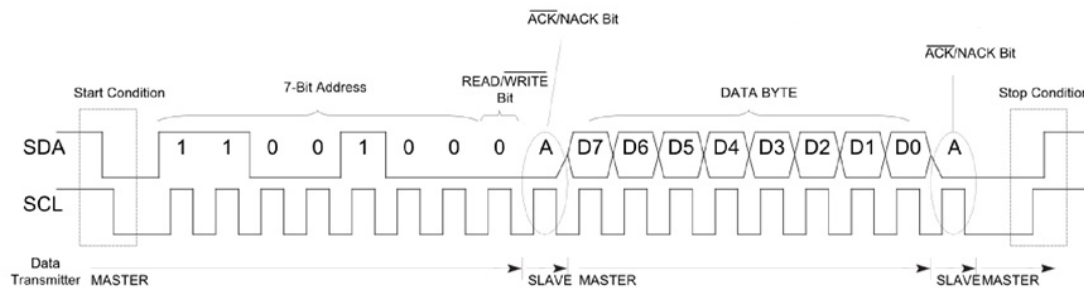
$$Samplingstid_{ADC} = \frac{2}{f_{CLK_{PER}}} + n \left(\frac{13.5 + 2}{f_{CLK_{ADC}}} + \frac{2}{f_{CLK_{PER}}} \right) \quad (5.1)$$

ADC-en benytter en "Analog Comparator" og to referanser til å bestemme verdien på inngangssignalet. For "single-ended" operasjon er den nedre referansen 0V og den øvre referansen kan bestemmes etter ønske. Ved differensiell operasjon kan både nedre og øvre referanse bestemmes.

Inngangssignalet sjekkes opp mot midtpunktet mellom den øvre og den nedre referansen; $(V_{ref+} - V_{ref-})/2$. Dersom inngangssignalet er større enn midtpunktet settes den første bit'en i resultatet til 1, og dersom det er negativt settes den til 0. Deretter flyttes midtpunktet opp eller ned avhengig av om første bit er 1 eller 0. Dersom det var 1 flyttes det oppover til $(V_{ref+} - V_{ref-})/2 + (V_{ref+} - V_{ref-})/4$, og inngangssignalet sjekkes opp mot denne nye verdier og et nytt bit skrives til resultatet. Dette fortsetter til resultatet har 10 eller 12 bit, avhengig av oppløsningen til ADC-en.

5.3 I2C

Dataoverføringen mellom sensoren og satellitten blir gjort ved hjelp av I2C protokollen. Det er en synkron kommunikasjonsform som foregår over to linjer; en datalinje(SDA) og en klokkelinje(SCL), som begge er høye så lenge overføring ikke pågår. Enheten som styrer overføringene kalles master, mens de andre enhetene kalles slaver. Alle slavene i systemet må ha sin egen unike 7-bits adresse slik at masteren kan kommunisere med de, og det kan være opp til 127 slaver i ett system. Alle overføringene er på 8-bit, og etter hver overføring blir det sendt en "Acknowledged-bit" (ACK) eller en "Not Acknowledged-bit" (NACK) av enheten som mottar data over datalinjen for å indikere om overføringen var vellykket eller om overføringen skal stanse. I denne oppgaven vil satellitten fungere som master og sensoren som



Figur 5.1: I2C overføring [33]

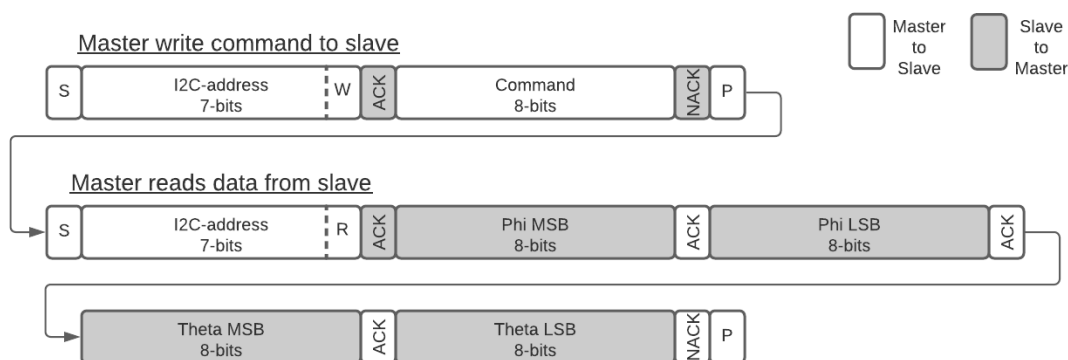
Masteren bestemmer når kommunikasjonen skal starte. Det gjør den ved å trekke SDA ned til jord samtidig som SCL er høy. Deretter sender den en 7-bits adresse fulgt av en bit som indikerer om masteren vil overføre data fra eller til slaven. Slaven må deretter bekrefte at den har mottatt sin egen adresse med en ACK, og enten

sende data til masteren eller forberede seg på å motta data fra masteren. Figur 5.1 viser hvordan SDA og SCL trekkes ned for å overføre data.

Dersom slaven skal sende data til masteren, sender den dataene over SDA linjen og venter på ACK eller NACK fra masteren. Dersom masteren sender en ACK sender slaven neste del av datene over SDA linjen. Dersom masteren sender en NACK indikerer dette at masteren har mottatt all data og overføringen avsluttes.

Dersom slaven skal motta data, lagrer den dataene som sendes over SDA og sender ACK etter hver byte den mottar. Når slaven har mottatt alle dataene sender den en NACK og overføringen avsluttes.

Figur 5.2 viser hvordan satellitten kan hente vinklene ϕ og θ fra sensoren. Først sender satellitten en kommando til sensoren som forteller sensoren at den skal regne ut en ny vinkel og hvilket format masteren ønsker vinkelen på. Så avsluttes overføringen for å gi sensoren tid til å regne ut vinkelen. Deretter startes en ny overføring hvor masteren leser 4 bytes fra sensoren. Tabell 5.1 viser en oversikt over alle kommandoene masteren kan sende til slaven.



Figur 5.2: I2C overføring av to vinkler fra slave til master

Kommandonavn	Kommando	Beskrivelse
setI2Caddress	0x02	Gir sensoren en ny I2C adresse
getAngles	0x04	Beregner og returnerer vinkler
getVector	0x05	Beregner og returnerer vektor
getADCvalues	0x06	Leser av og returnerer ADC verdiene

Tabell 5.1: I2C kommandoer

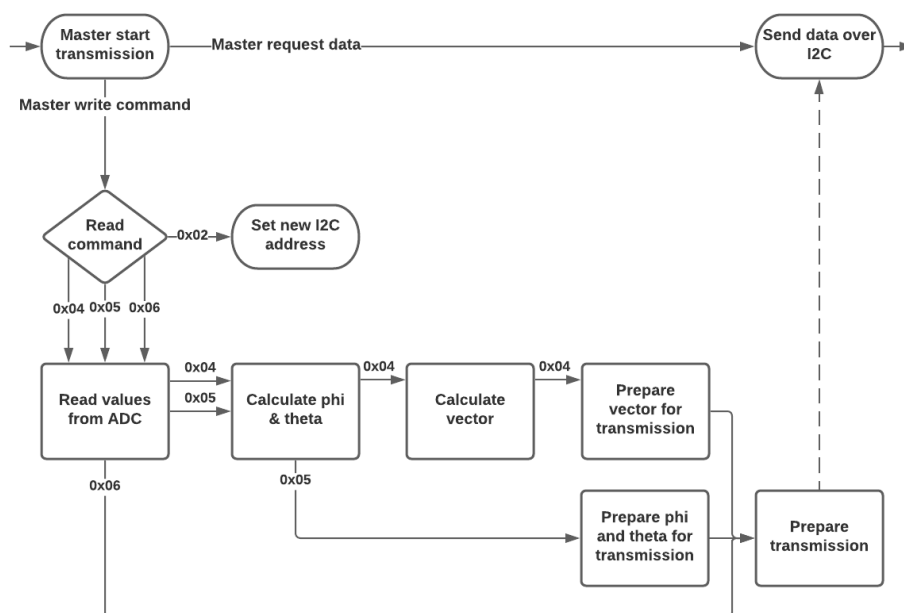
5.4 Interrupts

Programmet er interrupt basert, som betyr at istedet for å hele tiden sjekke om masteren prøver å kontakte slaven vil selve oppropet fra masteren til slaven aktivere koden som skal kjøres.

Dette gjøres ved hjelp av en “Interrupt service routine”(ISR) som kjører ved to anledninger. Den første er en ‘address interrupt’ som oppstår hver gang AVR’en oppdager en startkondisjon på I2C linjene etterfulgt av adressen til den spesifikke slaven. Den andre er en ‘data interrupt’ som oppstår hver gang slaven mottar data fra masteren eller hver gang den sender data til masteren.

5.5 Programkoden

Programmet som leser av de fire kvadrantene på dioden, regner de om til vinkler eller en vektor og overfører disse ved hjelp av I2C protokollen. Programmet kan sende ut verdier direkte fra ADC’en, de to vinklene θ og ϕ , eller en vektor. De to første brukes kun for testing og kalibrering, siden ønsket format fra oppdragsgiver var en vektor. Utdrag fra programkoden er presentert under, og hele programkoden ligger under vedlegg E. Figur 5.3 viser et flytskjema over programmet.



Figur 5.3: Programvare flytskjema

5.5.1 ADC

Programmet inneholder to funksjoner i forbindelse med ADC'en. Den første, `ADC_init`, konfigurerer ADC'en ved oppstart ved å velge referansespenning, stiller inn ADC klokken, velger oppløsning og antall sampler per konvertering.

```
void ADC_init(void){
    //Set ADC reference voltage
    VREF.ADCOREF |= VREF_REFSEL_VDD_gc;
    //Choose number of samples for each conversion
    ADC0.CTRLB |= ADC_SAMPNUM_ACC8_gc;
    //Select pre-scaler to 20 (CLK_ADC = 4Mhz/20 = 200kHz)
    ADC0.CTRLC |= ADC_PRESC_DIV20_gc;
    //Set ADC resolution to 12-bit and Enable ADC
    ADC0.CTRLA |= ADC_RESSEL_12BIT_gc; | ADC_ENABLE_bm;
}
```

Den andre funksjonen, `ADC_read`, tar en av de kvadrantene på dioden som argument og leser av verdien på den. Først setter den inngangen på ADC-en til den valgte dioden, deretter starter den konverteringen, venter til konverteringen er ferdig og returnerer resultatet.

```
uint16_t ADC_read(diode_t diode){
    //Set ADC input
    ADC0.MUXPOS = diode;
    //Start conversion
    ADC0.COMMAND = ADC_STCONV_bm;
    //Wait until conversion finished
    loop_until_bit_is_set(ADC0.INTFLAGS,ADC_RESRDY_bp);
    //Return result
    return ADC0.RES;
}
```

5.5.2 Utregninger og databehandling

Funksjonen `calculateAngles` bruker `ADC_read` til å lese av alle kvadrantene. Deretter bruker den likningene fra kapittel 3.4 til å regne ut vinklene ϕ og θ .

```

void calculateAngles(void){
    //Reading values from all diodes
    A = ADC_read(DIODE_A);
    B = ADC_read(DIODE_B);
    C = ADC_read(DIODE_C);
    D = ADC_read(DIODE_D);
    //Calculating x and y from diode values
    x = (float)(B+C-A-D)/(A+B+C+D)*r;
    y = (float)(A+B-C-D)/(A+B+C+D)*r;
    //Calculating phi
    phi = atan(sqrt(pow(x,2)+pow(y,2))/h);
    //Calculating theta and correcting angle to range from 0 - 2pi radians
    theta = atan2(y,x);
    if(theta<0){theta+=(2*M_PI);}
}

```

Resultatene fra utregningen av vinklene er radianer på desimalform. Funksjonen `setUpAngles` konverterer de til grader og legger de inn i en array med 8-bits verdier som kan overføres videre fra sensoren. Vinklene ganges også med 100 for å få med to desimalsiffer. Vinkelene må deles på 100 igjen etter de har blitt overført.

```

void setUpAngles(void){
    /*
    Converting phi and theta to degrees and loading into array
    as 16-bit values like this: angles[phiMSB, phiLSB, thetaMSB, thetaLSB]
    */
    angles[0] = ((uint16_t)round(phi*180/M_PI*100) >> 8) & 0xFF;
    angles[1] = ((uint16_t)round(phi*180/M_PI*100) >> 0) & 0xFF;
    angles[2] = ((uint16_t)round(theta*180/M_PI*100) >> 8) & 0xFF;
    angles[3] = ((uint16_t)round(theta*180/M_PI*100) >> 0) & 0xFF;
}

```

Programmet har også to funksjoner som ligner `setUpAngles`; `setUpVector` og `setUpADCvalues` som respektivt klargjør en vektor og ADC-verdiene for overføring.

5.5.3 I2C

I2C_init konfigurerer AVR'en som en slave ved å gi den en adresse. Den aktiverer også interrupts og smart mode. Smart mode forenkler koden en del ved å klarere data interrupt flagget etter at data leses fra eller skrives til data registeret. Når adressen til I2C slaven skrives venstreskiftes den ett hakk siden bit'et helt til høyre registeret avgjør om slaven skal respondere på alle adresser, såkalt 'general call address'.

```
void I2C_init(uint8_t slaveAddress){
    //Enable I2C, data and address interrupts and smart mode
    TWI0.SCTRLA = TWI_DIEN_bm | TWI_APIEN_bm | TWI_ENABLE_bm | TWI_SMEN_bm;
    //Set slave address
    TWI0.SADDR = slaveAddress << 1;
    //Initialize counters for transmitting data
    I2C_isFirstByte = true;
    I2C_byte_counter = 0;
}
```

Siden sensoren returnerer flere ulike sett med data ble det laget to funksjoner for å forenkle prosessen med å overføre disse fra sensoren. Den første funksjonen, I2C_setupData tar to argumenter; en array med data og lengden på arrayen. Dataene lastes så inn i en ny array; I2C_data.

```
void I2C_setupData(uint8_t data[], uint8_t data_size){
    //Set up data to be transmitted
    I2C_data_size = data_size;
    for(uint8_t i = 0; i < data_size; i++)
    {
        I2C_data[i] = data[i];
    }
}
```

Den andre funksjonen, I2C_sendData, sender dataene som ble satt opp ved hjelp av I2C_setupData helt til masteren indikerer at den har mottatt all dataene med en NACK. To variabler benyttes i denne funksjonen, en boolsk variabel som er sann frem til først byte har blitt sendt og en tellevariabel som inkrementerer etter hver byte har blitt sendt.

Etter hver byte har blitt sendt sjekker programmet om masteren responderer med en ACK eller NACK. Dersom den responderer med en ACK sendes neste data. Den boolske variabelen kreves for første dataoverføring siden slaven ikke mottar ACK foran først overføring som vist i figur 5.2. Når slaven mottar en NACK fra masteren avsluttes overføringen og variablene resettes.

```
void I2C_sendData(void){
    //if first byte of transmission
    if(I2C_isFirstByte)
    {
        //Load data for master to read
        TWIO.SDATA = I2C_data[I2C_byte_counter%I2C_data_size];
        //Increment counter and update variable
        I2C_isFirstByte = false;
        I2C_byte_counter = (I2C_byte_counter + 1)%I2C_data_size;
        //Wait until data has been transmitted. This will trigger new
        interrupt
        loop_until_bit_is_set(TWIO_SSTATUS,TWI_DIF_bp);
    }
    //if ACK received from master
    else if(!(TWIO.SSTATUS & TWI_RXACK_bm))
    {
        TWIO.SDATA = I2C_data[I2C_byte_counter%I2C_data_size];
        I2C_byte_counter = (I2C_byte_counter + 1)%I2C_data_size;
        loop_until_bit_is_set(TWIO_SSTATUS,TWI_DIF_bp);
    }
    //if NACK received from master
    else
    {
        //Complete transaction and release I2C lines
        TWIO.SCTRLB = TWI_SCMD_COMPTRANS_gc;
        //Reset counters
        I2C_isFirstByte = true;
        I2C_byte_counter = 0;
    }
}
```

Dersom masteren etterspør mer data enn det som er tilgjengelig sender slaven data fra starten av arrayen på nytt. For eksempel dersom master ber om vinklene ϕ og θ må den lese av 4 bytes, 2 for hver vinkel. Dersom den leser av 8 bytes istedet vil den motta de 4 bytes-ene 2 ganger.

5.5.4 ISR og oppstart

ISR rutinen aktiveres når AVR-en gjenkjenner adressen sin og hver gang den sender eller mottar data. For å avgjøre hvilken kondisjon som aktiverer ISR-en, kontrolleres statusregisteret til I2C.

Dersom det er en adress interrupt sender slaven tilbake en ACK for å bekrefte at den har mottatt adressen sin. Dette lar masteren vite at slaven er klar til å kommunisere.

Etter en adresse interrupt vil det oppstå en data interrupt, og programmet må sjekke om masteren vil sende eller motta data. Dersom masteren ønsker å skrive til AVR-en er det en kommando den vil skrive til slaven. Denne lagres i en variabel kalt `cmd`, og variabelen `checkCmd` settes til sann for å indikere at slaven må undersøke hvilke kommando den mottok fra masteren etter at ISR-rutinen har kjørt. Deretter sendes det en NACK tilbake til masteren for å indikere at slaven ikke ønsker mer data.

Når masteren ønsker å lese data fra slaven, aktiveres `I2C_sendData` kommandoen som sender data helt til masteren stopper overføringen med en NACK.

```
ISR(TWIO_TWIS_vect){
    //Address interrupt
    if((TWIO.SSTATUS & TWI_APIF_bm) && (TWIO.SSTATUS & TWI_AP_bm))
    {
        I2C_sendAck();
    }
    //Data interrupt
    else if(TWIO.SSTATUS & TWI_DIF_bm)
    {
        //Master read from slave
        if(TWIO.SSTATUS & TWI_DIR_bm)
        {
            I2C_sendData();
        }
    }
}
```



```

    //Master write command to slave
else
{
    cmd = TWIO.SDATA;
    checkCmd = true;
    I2C_sendNack();
}
}
}

```

Når AVR-en starter settes adressen til slaven, og variablene h og r defineres. I2C_init og ADC_init kjøres for å konfigurere ADC-en og I2C, og aktiverer interrupts på AVR-en.

```

int main(void){
    slaveAddress = 0x08;
    //Initializing parameters for calculations
    h = 1.76e-3;
    r = 1.20e-3;
    //Initializing I2C, ADC and interrupts
    I2C_init(slaveAddress);
    ADC_init();
    sei();

    while (1){
        //If new command received from master
        if(checkCmd){
            //Disable interrupts temporarily
            cli();
            executeCmd(cmd);
            checkCmd = false;
            //Re-enable interrupts
            sei();
        }
    }
}

```

I while løkken til programmet aktiveres `executeCmd` dersom sensoren har mottatt en ny kommando fra masteren. Her kontrolleres det hvilken kommando som ble mottatt, og de passende tiltakene blir utført. Interrupts deaktiveres med `cli` funksjonen mens programmet utfører kommandoen og reaktiveres igjen når kommandoen har blitt fullført med `sei` funksjonen. Dette for å garantere at kommandoen ikke blir avbrutt mens den beregner nye verdier.

```
void executeCmd(uint8_t cmd){
    switch(cmd)
    {
        //Master requests angles phi and theta
        case getAngles:
            calculateAngles();
            setUpAngles();
            I2C_setUpData(angles,4);
            break;
        //Master requests vector R[x,y,z]
        case getVector:
            calculateAngles();
            setUpVector();
            I2C_setUpData(R_vect, 6);
            break;
        //Master requests ADC values
        case getADCvalues:
            calculateAngles();
            setUpADCvalues();
            I2C_setUpData(ADC_values, 8);
            break;
    }
}
```

5.6 Lookup-table

En løsning som tok i bruk et lookup-table ble foreslått, vi ønsket da å styre lysets innfallsvinkel fra $(-30^\circ, -30^\circ)$ til $(30^\circ, 30^\circ)$, med en oppløsning på 1 grad, det vil da si $61 \cdot 61 = 3721$ mulige innfallsvinkler, og lagre strømverdiene som samsvarer med

denne posisjonen. Problemet oppstår ved at det blir for mye data å lagre i RAMen hvis vi skulle tatt i bruk et fullstendig look-up-table med en oppløsning på 1 grad. Hvis vi regner med at strømmene lagres i 16-bit register, og innfallsvinklene lagres i 8-bit register, så ender vi opp med $16 \text{ bits} * 4 + 8 \text{ bits} * 2 = 80 \text{ bits}$ per rad, og det er 3721 rader, da ender vi opp med $297680 \text{ bits} = 37,21 \text{ kB}$. Med tanke på spesifikasjonene om hastighet er det ugunstig å lagre som flashminne, derfor må det lagres i SRAM-en som kun har lagringsplass for opp til 16kB. En løsning på dette problemet er å redusere oppløsningen, for eksempel kun lagre hver fjerde posisjon. Da reduseres størrelsen til 9,3 kB, som gir oss stort slingringsmonn til variabler, stack, og annet som lagres på SRAM.

6 Kalibreringsrigg

Hovedutfordringen med kalibreringsprosessen er å finne brukbare løsninger for de to hoveddelene i systemet, solsimulatoren og rotasjonsmekanismen. Kravet til solsimulatoren, som skal produsere innfallende lys på sensoren, er at den må kunne representere sollyset. Det er derfor viktig at lyset har riktig bølgelengde og tilstrekkelig styrke. Det andre hovedpunktet i oppsettet er et type mekanisk system som kan brukes for å rotere sensoren, slik at lyset kan treffe sensoren i forskjellige vinkler. Kravet til rotasjonsmekanismen er at den må kunne bevege solumsensoren i samme oppløsning som sensorkravet tilsier. Kalibreringsprosessen er særdeles viktig for å få en nøyaktig solumsensor, og løsningen som har blitt valgt å bruke er en original idé som gruppen ikke har sett vært brukt i andre oppgaver som omhandler solumsensorer.

Det ble også vurdert alternative løsninger for kalibrering. Under researchfasen kom gruppen frem til at de fleste som hadde utført lignende oppgaver allerede hadde testoppsett tilgjengelig. Flere hadde laboratorier designet til satellittbygging, lignende oppsett på skoler eller de kunne ta i bruk oppsett lagd i forbindelse med tidligere bachelor- eller masteroppgaver. Disse hadde da ordentlige solsimulatorer og stasjoner for å rotere sensoren.

Gruppen kikket dermed på løsninger for dette som kunne bestilles. Det var nødvendig å benytte seg av en mekanisme som roterte i to akser, og en slags type solsimulator. Solsimulatorer gir ut lys som skal etterligne naturlig sollys. Hensikten med dem er å gjøre det mulig å drive tester som krever sollys inne i et laboratorium, med kontrollerbare omstendigheter. [34] Desverre er dette veldig store enheter, som både er dyre og vanskelige å få tak i. Noen mindre alternativer ble selvsagt også vurdert, som f.eks “Pico LED small area solar simulator” fra G2V optics, men også her var det likevel usikkert med tanke på pris og leveringstid.

Det ble også sett på en del billigere alternativer, som for eksempel xenonpærer som løsning for solsimulatoren. Det viktigste med dette er å velge en lampe som har spektrum som ligger i nærheten av AM1.5G solspektret, som vist i vedlegg A.1, slik at lampen sitt lys faktisk ligner sollys. LED-lamper ble også vurderte, da også disse kan ha mye til felles med sollys. De kan etterligne både intensiteten og fargetonen til sola, fordi komponentene i LEDs gir mulighet for finere justering for farge, lysstyrke

og intensitet. [35] Ingen av disse mulighetene ble valgt fordi Orbit NTNU hadde en lyskilde tilgjengelig, slik at det var det beste alternativet med tanke på tid og budsjett.

De mer grundige og robuste systemene for rotasjon av sensoren som kunne bestilles så også ut til å ligge godt utenfor prosjektbudsjettet. Dermed ble det besluttet at den beste løsningen var å lage en stasjon selv, som er videre beskrevet i denne seksjonen.

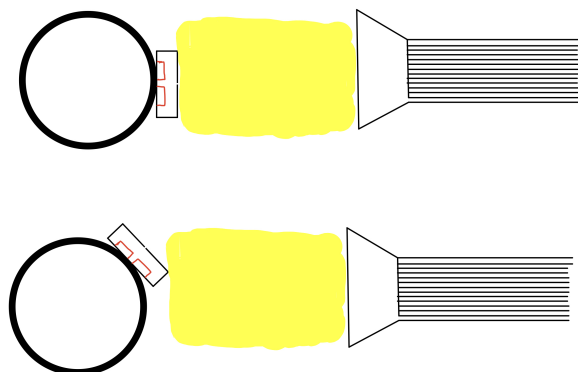
Utstyret som gruppen har tilgjengelig for å lage kalibreringsriggen er 3D-printerene til MAKE NTNU og diverse programvare som brukes for å lage 3D-modeller. Programmet som har blitt brukt for å designe modellene er Solid Edge. Dette er et program som er gratis for studenter, og det gjør prosessen av å designe realistiske 3D-modeller svært enkelt. MAKE NTNU har et verksted bestående av flere Ultimaker 2+ printere. Printerene har en standard oppløsning på 0,1mm og bruker PLA-filament som materiale for modellene som blir printet ut.

Denne høye oppløsningen fører til at resultatet vil kunne bli veldig nøyaktig, men et problem som oppstår med filament 3D-printere er at de blir upresise når størrelsen på modellen nærmer seg oppløsningen. Dette vil si at dersom modellen bare inneholder lengder på rundt 0,1mm vil printeren kunne bomme på plasseringen av filament og resultatet kan bli ubrukelig. Delene som gruppen skulle printe ut ville ha varierende størrelser, men de minste delene av modellene ville tilhøre dekslet til fotodiode-komponenten. Denne komponenten består av flere sirkulære former hvor den største har en diameter på 9,2mm. Høyden til komponenten er 3,2mm. Disse størrelsene skulle være store nok slik at printeren unngikk problemer med dårlig presisjon.

6.1 2-DOF-stasjon

Gruppen hadde en idémyldring med en kontakt der hardware-delen av kalibreringsprosessen ble diskutert. Flere løsninger ble deliberert, og det som var målet for alle var en enkel prosess med få bevegelige deler. Første spørsmål gruppen stilte seg selv var om posisjonen til dioden i forhold til lyskilden var relevant under rotasjon (Figur 6.1). Hvis det var tilfellet at posisjonsskift under rotasjon *ikke* påvirket resultatet så ville vår første og enkleste løsning fungere; den omhandlet å bruke to syllindere, og feste kortet på den midtereste, og rotere rundt syllinderen. Det ble så valgt å kontakte Dag Roar Hjelme, som har mer kunnskap rundt akkurat dette, og høre med han om dette kunne stemme. Dessverre ble tilbakemeldingen at

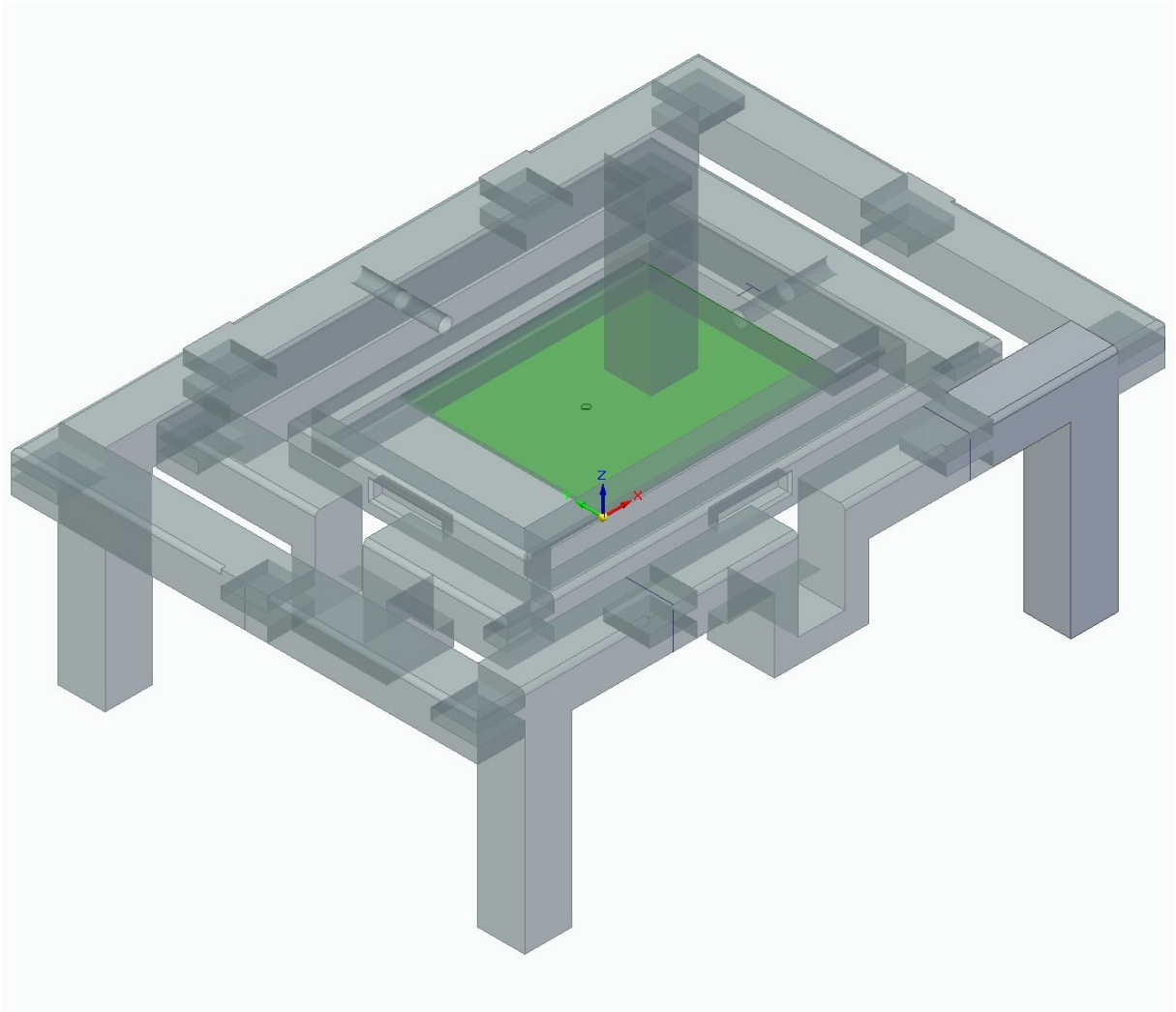
det avhenger for mye av variabler som er vanskelig for oss å kontrollere. Av denne grunnen ble denne løsningen valgt bort.



Figur 6.1: Rotasjon med posisjonsskift

Neste steg i prosessen var å finne en måte for å kunne fjerne så mange variabler som mulig i kalibreringsprosessen, så gruppen lett kunne ha kontroll på de variablene som påvirker FOV og nøyaktigheten til sensoren. Et viktig moment her er at lyskilden alltid stilles inn med samme lysstyrke, da elimineres også dette som variabel. Gruppen hadde mange forskjellige idéer, bl.a. å bruke en laser til å presist lyse på sensoren fra forskjellige kontrollerte vinkler. Denne idéen ble forkastet fort, men den ga inspirasjon til nye løsninger. Det neste forslaget var den første utgaven av det som ble vårt endelige design. I denne løsningen skal sensoren vendes om x-aksen for å styre vinkelen i y-retningen, og lyskilden svinger om y-aksen for å styre vinkelen i x-retning. Da er det kun to variabler som fysisk trengs å styres.

Den endelige versjonen av kalibrerings-riggen er en revidert versjon av den forrige diskuterte løsningen. Den kan sammenlignes med et “labyrintspill” (avbildet i figur 6.2) der en kan separat styre kortet langs x-og-y-aksene. For å gjøre denne styringa presist ble det valgt å bruke to MG996R servo-motorer koblet til et 6V-batteri styrt av en Arduino. Selve stasjonen ble designet i Solid Edge og består av tre hoveddeler; en indre ramme som holder kretskortet og kan vendes om y-aksen, en ytre ramme som kan vendes om x-aksen og roterer den indre rammen, og en statisk struktur som holder de bevegelige elementene i en avstand unna bakken så de kan rotere uten å bli forhindret, i tillegg til å holde servoen som vender den ytre rammen.



Figur 6.2: Avsluttende design for 2-DOF-stasjonen.

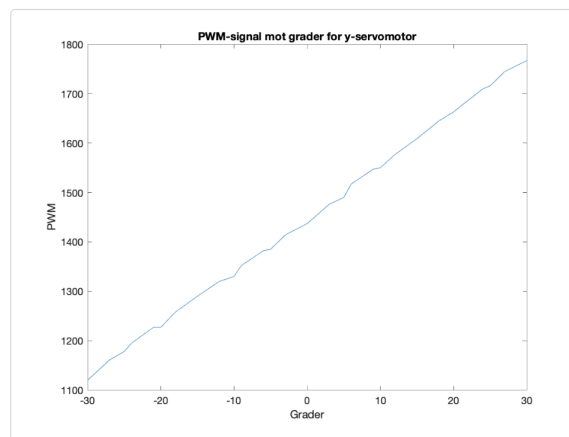
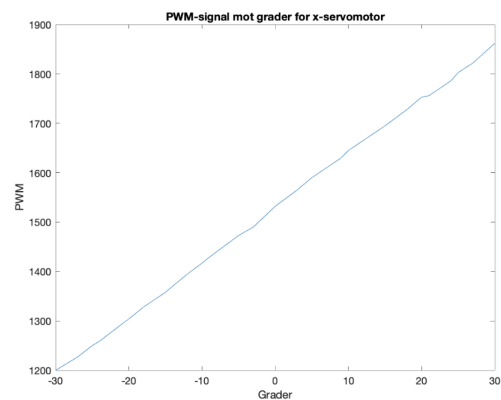
6.2 Servomotorer

Servomotorer styres av pulsbreddemodulerte (PWM) signal der de fleste har sitt “nullpunkt” (posisjonen der man kan rotere like mye til hver side) ved en pulsbredde på rundt $1500 \mu\text{s}$. En utfordring ved bruk av servomotorer er at de som regel er ulineære. Hvis man regner med at $1500 \mu\text{s}$ tilsvarer 0 grader, og $1800 \mu\text{s}$ tilsvarer 30 grader, kan man ikke gå ut i fra at pulsbredden øker med $(1800-1500)/30 = 10$ for hver grad. Det er fullt mulig at den øker med alt fra $5-15 \mu\text{s}$ mellom en grad og neste. Dette gjør motorene svært upresise hvis man styrer med grader, og har som sin verste egenskap at den lander i nye posisjoner etterhvert som man kjører den i en syklus. Derfor må servomotorens respons kartlegges i en kalibreringsprosess. Dette

kan gjøres på mange forskjellige måter, bl.a. kan man bruke et akselerometer og regulere med en regulator. Denne løsningen har mulighet til å være veldig presis, og fjerner mye av behovet for kalibrering av servoene, men det kreves også en del ressurser for å få den til å fungere. En alternativ løsning gruppen fant på var å bruke en smarttelefon sitt innebygde akselerometer og manuelt kalibrere servomotorene. Dette var gjort ved å plassere en telefon oppå kortholderen for kalibreringsriggen, som da kunne måle helningen på planet, og gjennom labview stille inn pulsbredden som tilsvarte målt helning på akselerometeret. Disse verdiene var så skrevet i et lookup table (Figur 6.3) som programmet senere har tilgang til, og det kan da interpolere verdier ut i fra de testpunktene vi valgte å måle. Verdiene er plottet i figur 6.4, så en enkelt kan se ulineariteten i servoene.

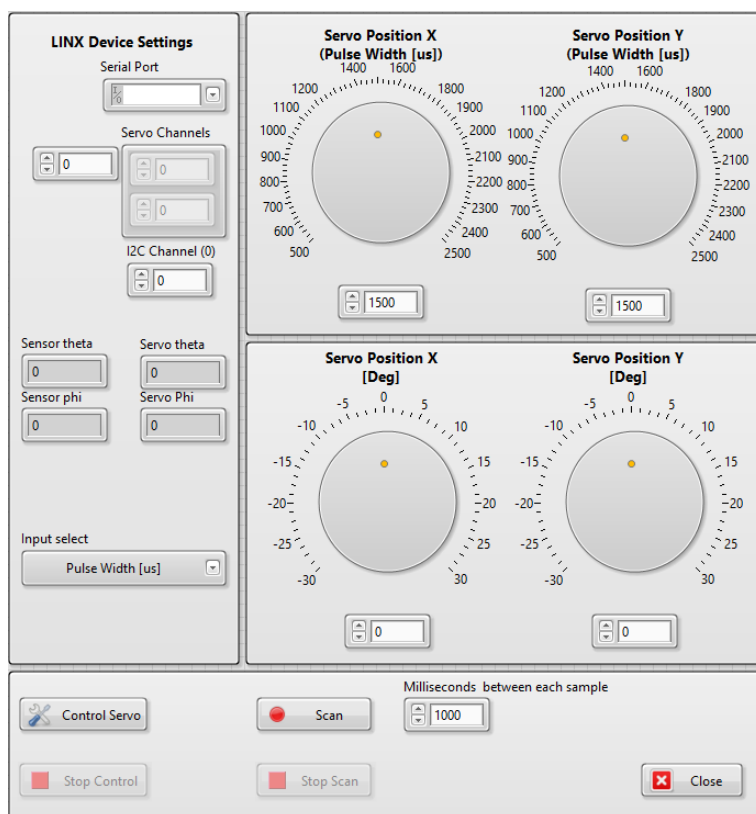
Servox Degrees	PWM[us]	Servoy Degrees	PWM[us]
-30	1200	-30	1120
-27	1227	-27	1161
-25	1250	-25	1178
-24	1259	-24	1195
-21	1293	-21	1227
-20	1304	-20	1227
-18	1328	-18	1258
-15	1358	-15	1290
-12	1395	-12	1320
-10	1417	-10	1330
-9	1429	-9	1352
-6	1462	-6	1382
-5	1473	-5	1385
-3	1490	-3	1414
0	1532	0	1437
3	1565	3	1476
5	1590	5	1490
6	1600	6	1517
9	1630	9	1547
10	1645	10	1550
12	1665	12	1577
15	1695	15	1609
18	1728	18	1645
20	1753	20	1663
21	1756	21	1675
24	1786	24	1710
25	1803	25	1716
27	1822	27	1745
30	1862	30	1767

Figur 6.3: Datasett etter kalibrering



Figur 6.4: Datasett som viser ulineariteten i servoene

Programmet som styrer servomotorene er kodet i LabView. Gruppen valgte dette programmet fordi en enkelt kan lage et oversiktlig og brukervennlig grensesnitt som man kan kontrollere servoene ifra. En Arduino ble brukt til å koble servoene og sensoren opp mot PC'en som LabView koden ble kjørt på. For å få Arduino til å kommunisere med LabView er man nødt til å benytte seg av en add-on som heter LINX. Denne gjør det mulig å styre eksterne systemer som f.eks Raspberry Pi og Arduino fra LabView. Programmet er programert som en state machine og har to hoved 'states'. *Servo Control* og *Servo Scan*. Et bilde av frontpanelet til programmet kan sees nedenfor.



Figur 6.5: Servo Program Front Panel

Før man starter programmet er man nødt til å spesifisere hvilken Com-port som Arduinoen er koblet til, hvilke pinner på Arduinoen som servoene er koblet til og hvilken I2C kanal som sensoren er koblet til. *Servo Control* lar brukeren manuelt bestemme posisjonene til servoene, og kan velge mellom å styre de ved å modulere pulsbredden, eller spesifisere ønsket helning i grader på hver akse. For at brukeren skal kunne styre servoene med grader, er det først nødvendig å lage et lookup table

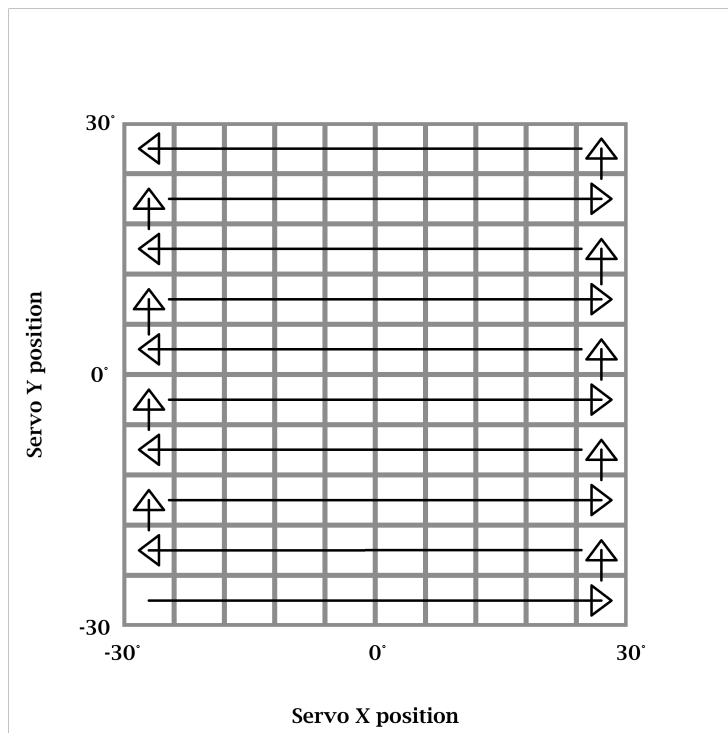
som angir hvilke pulsbredder som tilsvarer hvilke helningsvinkler for hver servo. Disse verdiene kan f.eks lagres i en tekstfil, som da blir importert inn i programmet og verdiene blir interpolert. I interpolerings funksjonen er ønsket helningsvinkel input og den tilsvarende pulsbredden er output, denne blir deretter skrevet til servomotoren. I *Servo control* staten vises også θ og ϕ verdien til planet. Disse verdiene regnes ut med følgende formler:

$$\theta = \arctan \frac{y}{x} \qquad \phi = \sqrt{x^2 + y^2} \qquad (6.1)$$

Der x og y er posisjonene til servoene i grader

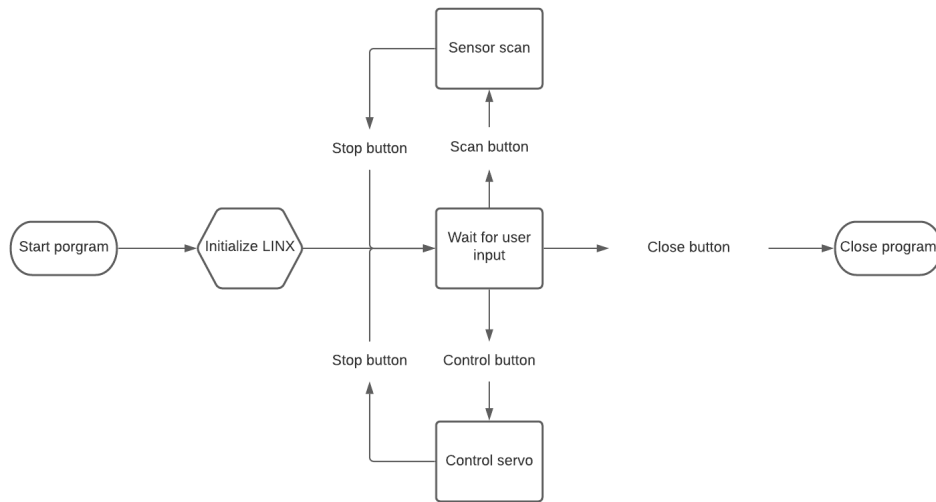
Programmet vil også vise de målte θ og ϕ i vinklene fra sensoren dersom den er koblet til. *Servo Control* brukes hovedsakelig til å kalibrere servoene før man kjører *Servo Scan*, slik at man kan kontrollere om servoene er nøyaktige nok og riktig kalibrert.

I *Servo Scan* funksjonen gjøres det målinger for hver posisjon på x og y-aksen, som lagres når målingen er ferdig. Programmet er laget slik at endringen i posisjonen er på en grad for hver måling, slik at man ender opp med en 61x61 matrise med verdier når målingen er ferdig. Verdiene som lagres er begge servoene sin posisjon, θ og ϕ til planet, og de målte θ og ϕ verdiene fra sensoren. Når målingen er ferdig lagres all data i en tekstfil slik at man senere kan bruke målingene til å undersøke hvor nøyaktig sensoren faktisk er. Nedenfor kan man se en illustrasjon som viser hvordan Scan funksjonen går gjennom målepunktene.



Figur 6.6: Illustrasjon av Scan funksjon

Det er viktig at målingene blir gjort etter at servoene er ferdige med å stille seg inn i riktig posisjon, slik at man ikke gjør målinger imellom to målepunkter. Det er derfor lagt inn mulighet for å justere tiden mellom hver måling, avhengig av hvor lang til det tar for servoene å stille seg inn i riktig posisjon.



Figur 6.7: Kalibrerings program State chart

For at sensoren skal kunne kalibreres nøyaktig, er det også viktig at servoene som brukes er nøyaktige nok. Etersom servoene er tannhjul drevne, vil det alltid være noe mekanisk dødgang som kan føre til unøyaktig posisjonering av servoene i kalibreringsprosessen. Det kan derfor være interessant å kartlegge hvor mye dødgang det er, eller lage en oversikt over hysteres-egenskapene til servoene. Hvor mye dødgang som kan tillates vil da avhenge av kravene til hvor nøyaktig sensoren skal være.

Etersom 2-DOF-stasjonen er 3d-printet ligger det en viss feilmargin i strukturen, noe som har ført til at servoene ikke stod helt statisk under omdreining, fordi det var et ørlite mellomrom mellom veggene som skulle holde den på plass. Løsningen på dette var å brette en bit av et A4-papir to ganger. Etter dette hadde motorene ingen posisjonsskift i noen retning.

6.3 Lyskilde

Et element fra kalibreringsriggen som vil være viktig å ta hensyn til da man skal simulere sollys vil være lyskilden. Graden av hvor mye gruppen kan etterligne sollys vil bestemme hvordan sensoren vil kunne prestere i verdensrommet som er et viktig poeng for Orbit. Da det har blitt bestemt at prototype 1 ikke skal ut i verdensrommet vil det viktigste momentet være å finne en lyskilde som gir utslag på fotodiodene.

Lyskilden som gruppen bruker i kalibreringsriggen, er en iiglo LED Lommelykt 3500 LM. Denne lommelykten har blitt lånt ut fra Orbit NTNU, og er svært kraftig da den kan gi ut mye lys. Lyskilden vil kunne gi et godt utslag på diodene, slik at en strøm kan genereres. I hvilken grad denne lommelykten vil simulere sollys er vanskelig å si, men da løsningen gruppen har kommet opp med for kalibreringsriggen er viktigere, har det ikke stor betydning om den har samme egenskaper som sollys. Om sensoren skal kalibreres i forhold til sollys må man finne en mer avansert løsning som vil kunne simulere dette.

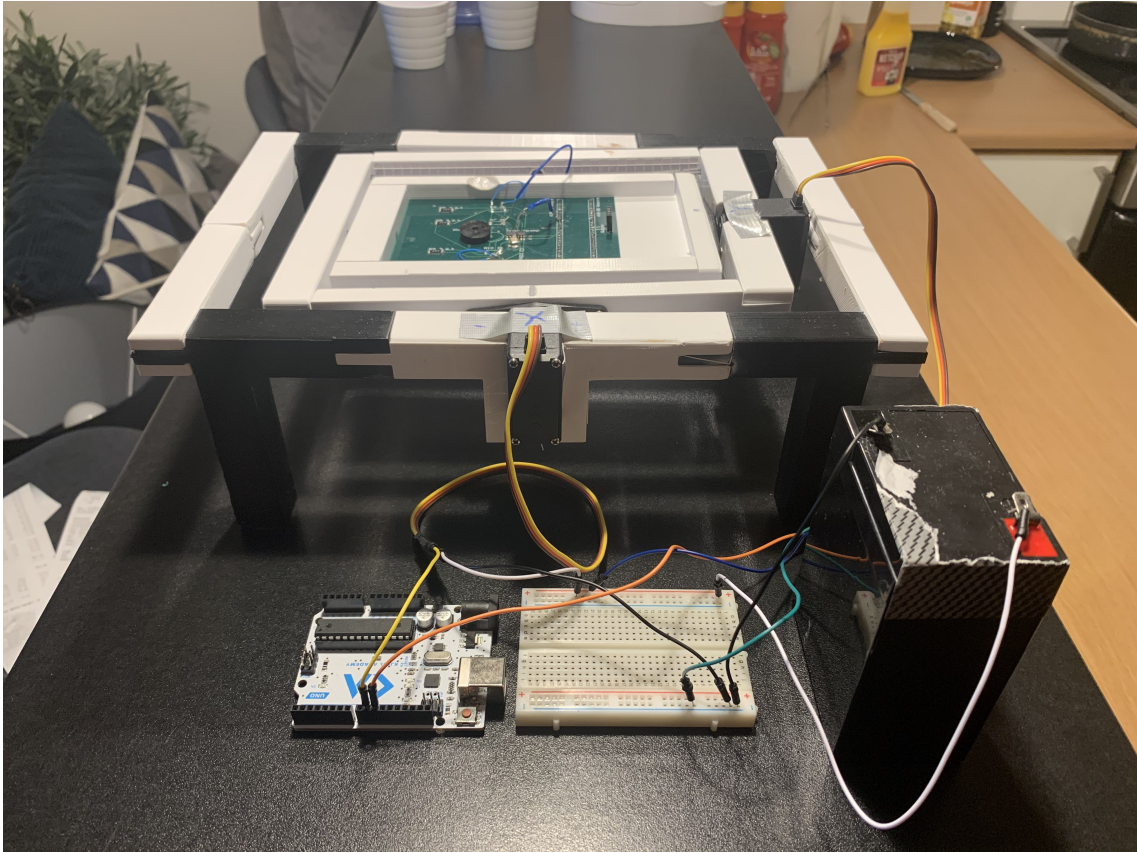
For å kunne kalibrere sensoren på en fornuftig måte er det viktig at lyskilden sentreres så godt som mulig og holdes stabilt, da det er stasjonen som kretskortet sitter på som skal rotere for å ta opp vinklene som skal testes. Denne lyskilden som skrevet om ovenfor, er en kraftig lommelykt som har en relativt stor masse slik at gruppen må ta hensyn til dette. For å løse dette problemet har gruppen i sitt oppsett satt opp et par stoler hvor det var en stasjon mellom stolene. I denne

stasjonen var det muligheter for å feste lyskilden slik at den holdte seg stabil og kunne lyse vinkelrett på det som stod nedenfor. Lommelykten var festet fast med gaffateip og satt fast nok til at den ikke bevegede på seg eller at det var fare for at den ville falle ned.

6.4 Ferdigstilt kalibreringsrigg

Den endelige kalibreringsriggen besto av alle delene som er nevnt i dette kapittelet. Delene var nærmest satt sammen, med unntak av lyskilden som var noe ekstern på grunn av løsningen som er nevnt tidligere. Lyskilden vil kunne lyse i retningen nedover mot bakken, slik at kretskortet med stasjonen og dekselet måtte settes under denne lommelykten. Da det er en viss høyde på stolene som er brukt til å sette opp stasjonen til lyskilden ville det kunne være problematisk å sentrere kretskortet rundt lyskilden. For å få til denne sentreringen på en best mulig måte ble det brukt en string med tråd. Denne tråden ble plassert slik at den var omtrent i midten av lyskilden og tuppen av stringen som trekkes ned mot kretskortet på grunn av gravitasjon vil kunne hjelpe oss med å sentrere kortet. Rigger ble så flyttet slik at fotodiodene skulle være i sentrum av tråden slik at riggen ble nokså sentrert rundt lyskilden. En naturlig feilkilde rundt denne riggen vil være hvorvidt sentreringen av lyskilden ble vellykket. Det ble gjort på øyemål slik at nøyaktigheten for denne operasjonen er bare så nøyaktig det er mulig å gjøre det ved å se.

I tillegg til at kretskortet må være sentrert rundt lyskilden vil det også være viktig at dekselaperturen sitt midtpunkt er det samme som midtpunktet til fotodiodene. Dette er viktig for å kunne få et bra utgangspunkt da det vil være like mye lys på alle diodene når lyskilden står vinkelrett på kortet. Dekselet er også designet slik at det skal være litt slingringsmonn for at prosessen av å sette det på fotodioden skal være så lett som mulig. Denne egenskapen vil føre til at etuiet sklir mot en side etterhvert som hellingen til stasjonen blir mer og mer drastisk. For å motvirke dette ble dekselet teipet fast til kretskortet slik at det skulle holde seg stabilt så mye som mulig. Dette er ikke den mest optimale løsningen, da teipeprosessen vil kunne gjøre at man dytter dekselet fra et sentrert punkt til et ikke-sentrert punkt. Nøyaktighet og presisjon vil være viktig for denne løsningen da det vil kunne påvirke resultatet.



Figur 6.8: Kalibreringsstasjon forfra

7 Deksel for fotodiodekomponent

En viktig del av sensoren er dekselet som skal dekke fotodioden. Dette dekselet har som oppgave å beskytte komponenten samtidig som det skal la sollyset treffe fotodiodene slik at en strøm kan genereres. Dekselet må designes slik at det kan festes på komponenten sånn at det ikke faller av under kalibreringsprosessen, og at det skal kunne tillate sollyset på vinkler mellom -30° og 30° å komme igjennom en apertur i toppen av overflaten til dekselet. På grunn av størrelsen til komponenten og viktigheten for å oppnå et nøyaktig resultat har det blitt bestemt at dekselet skal 3D-printes.

7.1 Apertur

For å kunne bestemme hvilke vinkler fra sollyset som skal kunne treffe dioden må det designes en apertur som tar hensyn til dette. Hvilken form og dimensjoner den har, i tillegg til hvor den er plassert, vil kunne bestemme hvor mye sollyset som kommer inn gjennom åpningen, og hvor mye sollyset som kan treffe kvadrantene ved en bestemt vinkel og FOV til sensoren. Dersom man ønsker en FOV på 60° må man finne høyden til overflaten av dekselet over fotodioden og hvor tykk aperturen skal være. Etter en del testing har gruppen funnet ut av at printerens ser ut til å kunne klare å lage en aperturtykkelse på 0,3mm. Når det gjelder formen til aperturen ser det ut til at det finnes to fornuftige løsninger: sirkel- og kvadrat-formet.

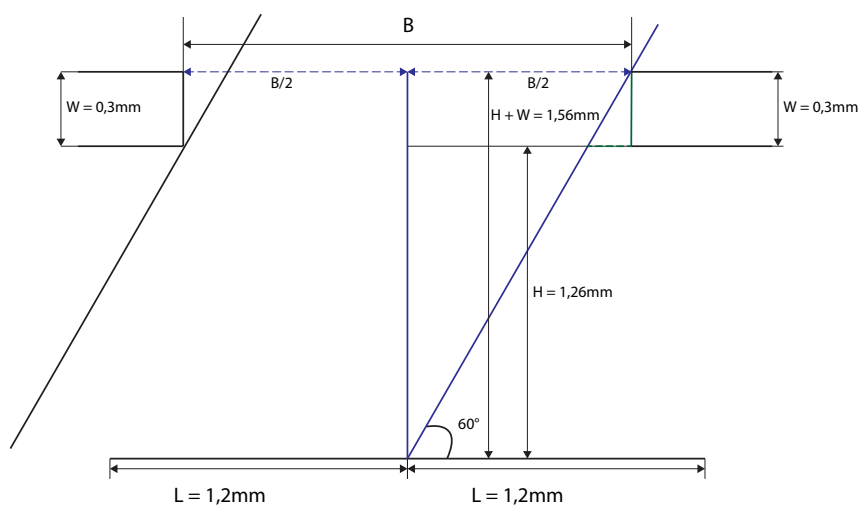
Fordelen med en sirkelform er at det er lettere for printerens å lage sirkler når det er snakk om små størrelser. Det å lage 90° vinkler for et kvadrat med lav lengde/bredde blir veldig vanskelig for printerens slik at man kan ende opp med sirkulære kanter istedenfor en full 90° . Dette vil kunne påvirke mengden sollyset som kommer inn i aperturen. Diameteren for sirkelen vil også være lik overalt sånn at man ikke får en diagonal som er lenger enn lengden/bredden, noe man får om man velger en kvadratisk løsning. Dersom det er ønsket å sentrere dekselet rundt fotodiodene er dette også lettere ved hjelp av en sirkulær form. Da trenger ikke dekselet å roteres for å ta hensyn til kvadrantene.

Den største fordelen med en kvadratisk løsning vil være at man kan etterligne formen til fotodiodene som lyset skal treffe, slik at man kan få inn mer sollyset, som vil gjøre at fotodiodene genererer en strøm av høyere verdi.

Quad-fotodioden som er brukt er laget slik at det ligger et glassbelegg på overflaten som har i oppgave å beskytte dioden som ligger inne i kapselen. Glasset ligger 1,26mm over fotodioden. Dette vil si at høyden over fotodioden vil bli høyden til glasset + tykkelsen til aperturen, så dersom det blir brukt en tykkelse på 0,3mm får man en høyde på 1,56mm over diodene. Lengden/bredden til en kvadrant er på 1,2mm i tillegg til en glippe på $50 \mu\text{m}$ mellom nabokvadranter. Disse dimensjonene er viktige for å kunne regne ut en apertur som skal kunne tillate en gitt vinkel. I dette tilfellet er det ønskelig med en FOV på 60° . For å oppnå dette må man gjøre utregninger med fokus på at man skal kunne få inn vinkler fra -30° til 30° som til sammen blir 60° .

Illustrerer en maks inngangsvinkel på 30° ved 60° FOV, apertur tykkelse = 0,3mm, høyde over fotodiodene = 1,26mm og lengde på fotodiode = 1,2mm

FOV = 60°
W = tykkelse
B = Apertur åpning
H = høyde over diodene
L = lengde på fotodiode



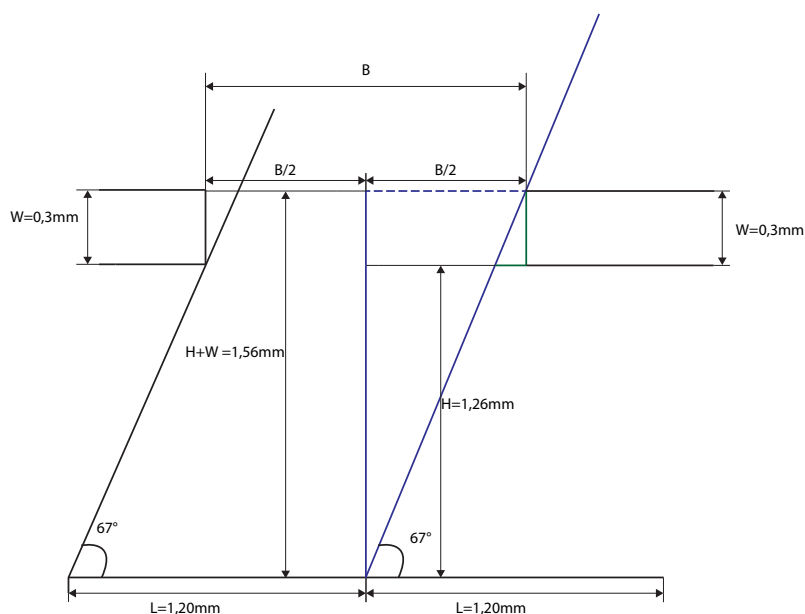
Figur 7.1: Figur som viser 60° FOV med apertur tykkelse 0,3mm

Figur 7.1 viser et forhold mellom lengden/bredden eller diameter, avhengig om man bruker kvadratisk eller sirkulær apertur, og tykkelsen til aperturen. Dersom man øker tykkelsen vil åpningen bli større og mer sollys vil kunne komme inn. Figur 7.2 viser forholdet om man har en ønsket FOV på 45° . Denne figuren er bygd opp likt, men

forholdet mellom tykkelse og åpning blir annerledes på grunn av ny inngangsvinkel på $22,5^\circ$.

Illustrerer en maks inngangsvinkel på $22,5^\circ$ ved 45° FOV, apertur tykkelse = $0,3\text{mm}$, høyde over fotodiodene = $1,26\text{mm}$ og lengde på fotodiode = $1,2\text{mm}$

FOV = 45°
W = tykkelse
B = Apertur åpning
H = høyde over diodene
L = lengde på fotodiode



Figur 7.2: Figur som viser 45° FOV med aperturtykkelse $0,3\text{mm}$

En sammenheng mellom tykkelse og åpning for apertur og FOV for sensor kan finnes ved å se de tidligere figurene, men den er enda enklere vist ved figurene 7.3 og 7.4. Man kan se at dersom man øker tykkelsen har dette en direkte påvirkning på åpningen da høyden over fotodioden blir lenger. Dersom man har en ønsket vinkel må dette tas hensyn til. For å kunne slippe til vinkelen må åpningen gjøres større.

Det finnes også en sammenheng med hvor mye sollys som kan komme inn i forhold til hvor stor åpningen er. Dette tilfellet vil ligne på det med aperturåpning, men etterhvert som apertur blir tykkere vil mindre sollys komme inn i dekkelet i forhold til åpningen ved de mest ekstreme vinklene. Dette vil si at dersom tykkelsen hadde vært uendelig tynn ville det kommet inn like mye sollys i lengde som åpningen er, men dette er åpenbart ikke mulig å få til praksis.

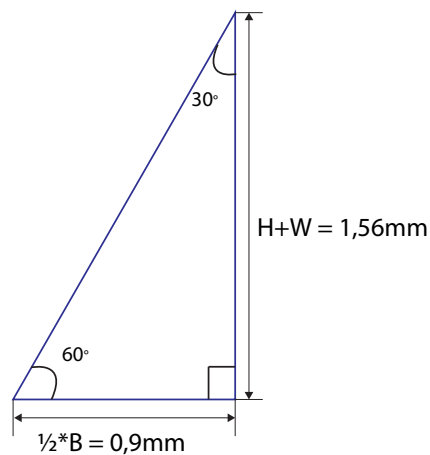
Sammenheng mellom tykkelse og åpning

FOV = 60°

H = Høyde fra fotodioder til glassbelegg = 1,26mm

W = Tykkelse til apertur

B = Lengde/diameter(avhengig om man bruker kvadratisk eller sirkulær form) for apertur åpning



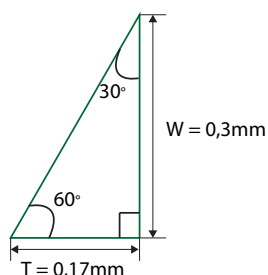
$$\tan(60^\circ) = (H+W)/(B * 1/2)$$

$$B = (2*(H+W))/\tan(60^\circ)$$

Figur 7.3: Figur som viser forhold mellom apertur tykkelse og åpning

Sammenheng mellom tykkelse og
sollys som går tapt

W = Tykkelse = 0,3mm
T = Tapt mengde sollys
S = Sollys størrelse i lengde
B = Lengde til åpning = 1,8mm



$$\tan(60^\circ) = W/T$$

$$T = W/\tan(60^\circ) = 0,17\text{mm}$$

$$S = B - T = 1,63\text{mm}$$

Figur 7.4: Figur som viser forhold mellom apertur tykkelse og sollys som går tapt ved maks vinkler

Tykkelsen til aperturen er et punkt ved deksel-designet som forårsaket mye usikkerhet. Tidligere ble det nevnt at det må være en viss tykkelse for at 3D-printeren skal klare å printe oppover i høyden, men dette problemet har blitt løst ved å øke minste tykkelsen til 0,3mm. Et annet problem som kunne forekomme med valg av tykkelse var faren for støy. Med støy så menes det utslag fra fotodioden når sollys kommer seg igjennom uønskede åpninger. Dersom man har en tykkelse på 0,3mm

kan man se at materialet ikke er helt tett. Det er antydning til at sollys kan komme seg igjennom tykkelsen fordi den er såpass tynn. For syns skyld kan man se dette om man holder et ferdiglagd deksel opp mot en lyskilde som for eksempel en lysspotte i taket. Man kan klart se at det kommer mer lys igjennom åpningen, men det ser ut som om det også kommer lys igjennom den delen av overflaten som skal være helt tett.

For å teste hvor mye støy som kan forekomme har det blitt testet å bruke deksler som ikke har åpning med varierende tykkelser i overflate og deksel farge for å se mengden støy. Figur 7.5 viser en oversikt over spenninger som er målt med forskjellige deksel. Fotodioden er konfigurert slik at utgangsspenningen til OPAMP-en er lav når fotodioden får mye lys på seg. Dette fører til at i en støymåling er man ute etter målinger hvor spenningsverdiene er høye. Resultatene fra denne testingen vil avgjøre hvilket apertur-design gruppen bruker. En oversikt over målingene finnes i figur 7.5. Hvordan testingen har blitt gjort forklares i kapittel 9.1.1.

	Tykkelse	Lys	Ingen lys
Grå	0,3	3,8V	
	0,5	4,18V	
	0,7	4,22V	
Svart	0,3	4,21V	
	0,5	4,23V	
	0,7	4,23V	
Ingen deksel		0V	4,17V

Figur 7.5: Målt spenning for varierende dekseltyper

Som man kan se fra denne oversikten var det noen store forskjeller ved forskjellige deksel. Fargen viste seg å utgjøre en større forskjell enn først antatt. Gruppen hadde en formening om at tykkelsen var det som kom til å utgjøre mesteparten av tykkelsen, slik at fargene, svart og grå, kom til å ha omkring samme verdier ved alle tykkelser. Dette viste seg til å være feil. 0,3mm tykkelsen for det gråe dekselet viste seg å være det designet som genererte mest uønsket strøm, slik at denne varianten er dårlig

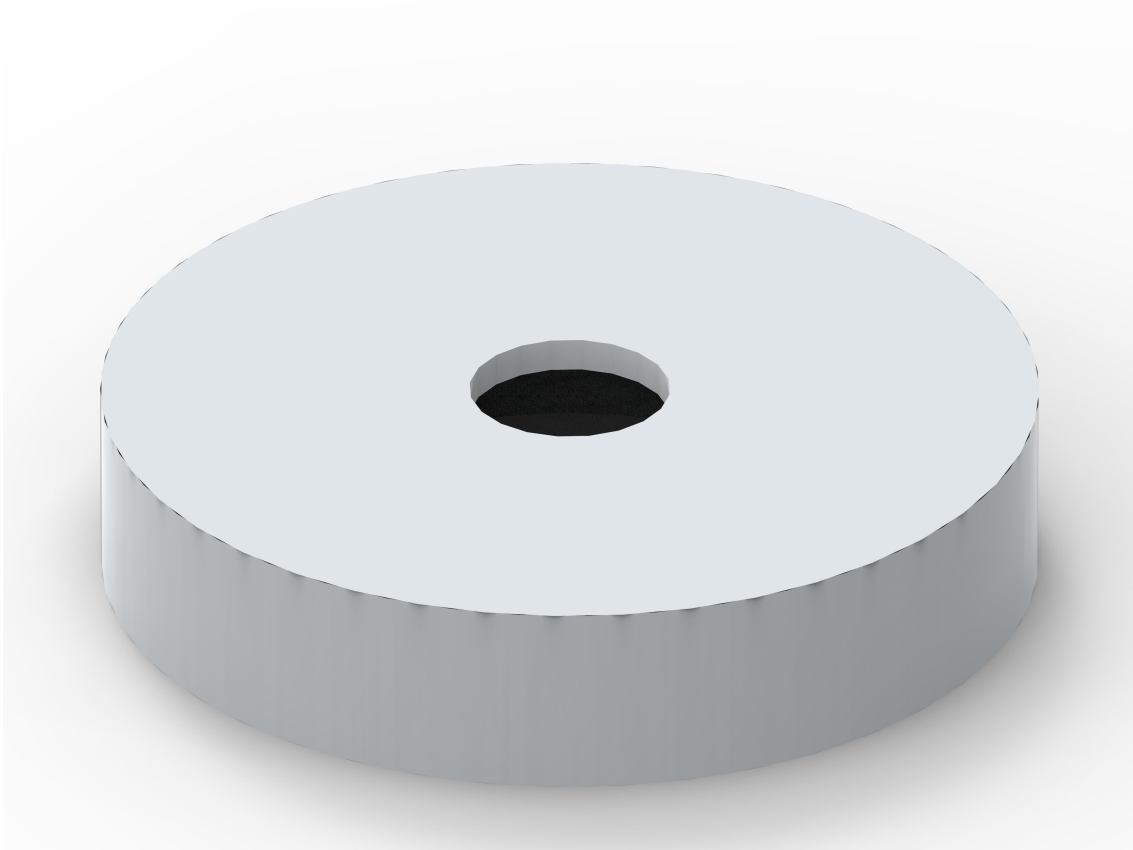
egnet for å bruke i sensoren.

Endring i tykkelsen utgjør en viss forskjell i lys som slipper inn. Ved 0,3mm tykkelse vil noe lys kunne slippe inn for begge fargene, men ved 0,5mm vil bare den grå varianten kunne slippe inn uønsket lys. 0,5mm og 0,7mm variantene for svart slipper inn like lite lys, så den ekstra tykkelsen på 0,2mm utgjør ingen forskjell. Grensen hvor tykkelsen har noe å si forhold til støynivå stopper på omkring 0,5mm dersom man bruker svart farge.

Etter å ha drøftet rundt dette med aperturen er det visse valg som gir mer mening enn andre. Den aperturen som gruppen har valgt å bruke burde ha god støydemping. En klar favoritt vil da være en aperturtykkelse på 0,5mm sammen med fargen svart. 0,5mm tykkelsen vil være bedre å bruke enn 0,7mm-varianten, da sensoren vil være mer presis med en mindre tykkelse fordi åpningen blir mindre. Dette vil føre til at en større andel av lyset som kommer inn i dekselet treffer kvadrantene til fotodioden og ikke veggene til innsiden av dekselet. Fra formlene som har blitt forklart tidligere vil denne tykkelsen føre til en åpning på 2mm for aperturen. Formen som vil være best å bruke for sensoren vil være sirkulær.

7.2 Endelig deksel-design

Det endelige designet som gruppen har valgt har en løsning som følger formen til komponenten for å oppnå et deksel som sitter tett på, men samtidig vil begrense og tillate visse vinkler å treffe kvadrantene. Et sylindrerformet design er derfor valgt fordi det etterligner formen til fotodioden slik at man skal få denne tettsittende funksjonen.



Figur 7.6: 3D-render av dekselet.

Tykkelsen til aperturen har vært et punkt hvor 3D-printeren kunne ende opp med problemer, men det finnes enda en dimensjon hvor dette kan være et problem. For å kunne gjøre sylindere høyere er man avhengig av at den har tykke nok vegger for at den skal kunne printes oppover i høyden. I likhet med prosessen av å finne minimum aperturtykkelse har det også blitt testet forskjellige tykkelser på veggene til sylindere for å se hva som er mulig å printe ut. Det ble kommet fram til at vegger på omkring 3 til 4 mm er det minste printeren vil tillate uten at strukturen har en sjanse for å falle sammen.

Siden det er et visst krav om en strukturell integritet for veggene til sylindere ble det gått for en løsning som har 5mm tykke vegger. Det utgjør ingen forskjell resultatmessig om man bruker 5mm tykke vegger eller tynnere, slik at det da virker mer fornuftig å bruke 5mm. Oppbygningen til komponenten lar den sylindereformede delen ligge over kretskortet når den har blitt loddet på, slik at hvis man velger 5mm tykke vegger påvirker ikke dette noen andre momenter ved kretskortet da den

og komponenten ligger over kretskortets overflate. Komponenten har diameter på 9mm hvor det er aktuelt å feste designet, mer om akkurat hvorfor vil bli beskrevet i neste avsnitt. For at dekselet skal kunne tres på, men samtidig sitte godt fast inntil komponenten har det blitt bestemt at en diameter på 9mm er fornuftig. Som forklart ovenfor skal veggene være 5mm tykke slik at den ytre diameteren blir $9mm + 5mm = 14mm$.

På grunn av viktigheten mellom tykkelse for apertur og høyden over dioden for å oppnå en ønsket FOV har det blitt bestemt at sylindredesignet gruppen lagde skulle ha en høyde på 2mm. Denne verdien er valgt på grunn av høyde-dimensjonene til komponenten. Fotodioden har en slags knagg ved bunnen av der den begynner å få en sylinder-form. Det er ønskelig at dekselet sin overflate skal ligge på overflaten til komponenten, slik at dersom høyden til dekselet er for stor så vil overflaten bli dyttet opp, og man vil ende opp med en for stor høyde over kvadrantene. 2mm høyden vil gjøre at dekselet ikke kommer seg helt ned til disse knaggene, noe som er en fordel da det ikke finnes noe som kan dytte dekselet over den ønskede høyden som har blitt brukt til å utregne inngangsvinkler. På grunn av denne lave verdien vil det være fare for at dekslet ikke kan tres så langt inn på dioden slik at det vil være en liten sjanse for at dekselet kan skli av under kalibreringsprosessen. Denne faren vil være usannsynlig på grunn av den tettsittende egenskapen til dekselet som er beskrevet i avsnittet over, men det kan likevel være en viss fare for at det kan skje.

Dekselet som brukes for kalibreringsprosessen vil være designet som er forklart ovenfor sammen med den mest optimale aperturen som også har blitt diskutert tidligere. Denne kombinasjonen vil føre til at gruppen ender opp med et deksel som er godt egnet for sensoren.

7.3 Alternative løsninger

Andre løsninger som gruppen har kommet opp med er lignende varianter av designet som er beskrevet ovenfor, men forskjeller i dimensjoner. Det var alltid vært sikkert at det skulle brukes en hul sylinder slik at denne formen har vært konstant gjennom hele designprosessen.

Et design som var ønsket å få til var en variant hvor dekselet tok graden av å etterligne komponenten enda lenger. Dette betyr at dekselet hadde alle de samme

formene som utsiden av fotodioden, men med litt større dimensjoner slik at det skulle kunne tres på. I dette designet lignet dekslet på den sylinderveformede delen av komponenten hvor de knaggene som var nevnt i et tidligere avsnitt også med. Grunnen til at dette designet ble forkastet når det så ut til å være en litt mer avansert variant med bedre festeegenskaper, var fordi at det var urealistisk å 3D-printe disse knaggene da de er så små i størrelse.

Da det var usikkerhet rundt dette med å kunne lage en tynn og liten nok apertur ble det foreslått at en laserkutter kunne løse problemet på en fornuftig måte. Omega Verkstedet ved NTNU Gløshaugen har en maskin som bruker en laser for å kunne kutte nøyaktige hull i overflater av forskjellige materialer. Kutteren bruker filer av typen Scalable Vector Graphics (SVG) for å styre hvor laseren skal kutte i materialet. Tanken med å bruke denne kutteren var for å forsikre at det kunne oppnås en nøyaktig apertur hvor tykkelsen kunne bestemmes til omkring 0,1mm og en diameter/lengde med lave dimensjoner slik at den kan kutte mer nøyaktig enn en 3D-printer vil klare å printe. Designet ville vært en kombinasjon av 3D-printing og laserkutting. Denne sylinderveformen som har blitt beskrevet tidligere ville blitt utsiden av dekslet, og en overflate som har blitt kuttet av laseren ville festes eller limes på overflaten av sylinderveformen. Dette designet vil ligne veldig mye på det som allerede har blitt gått for, men det vil være mulig å kunne bruke mindre dimensjoner for aperturen da laserkutteren er mer nøyaktig. En apertur tykkelse på omkring 0,1mm ville være mulig å få til da et A4-ark kan brukes som overflate. Grunnen til at dette designet ikke har blitt valgt er fordi et A4-ark vil kunne slippe igjennom mer sollys enn dekslene som har blitt printet ut slik at mer støy vil kunne oppstå om man bruker denne løsningen. Dersom man bruker en annen type materiale som er tykkere blir det mindre viktig å bruke laserkutteren da 3D-printeren klarer å lage et dekslet med apertur tykkelse på 0,3mm. For at det skal være noe poeng å bruke laseren burde et materiale som er tynnere enn hva minimum tykkelsen til 3D-printeren velges. Laseren vil dog ha en høyere presisjon enn 3D-printeren, så det kan likevel vurderes å bruke denne løsningen hvis man skulle trenge å lage en apertur med svært høy presisjon.

8 Testing og kalibrering

8.1 Lodding og feilsøking

Kretskort versjon 1

For å lodde komponentene på kortene ble Makerverkstedet til Make NTNU brukt, da dette var fritt tilgjengelig for gruppen. Det var vanskelig å finne alternativer for loddeverksted, siden de fleste var stengte eller hadde begrenset adgang av på grunn av smitterestriksjoner.

Makerverkstedet hadde ikke mikroskoper som kunne benyttes til loddingen, noe som gjorde det vanskeligere å få til loddingen. Quad-OPAMP-en var veldig liten og hadde ganske små bein med lite mellomrom, slik at det var en utfordring å få til å lodde på denne uten mikroskop. Gruppen fikk loddet OPAMP-en på kretskortet, men det var en del usikkerhet om den var snudd riktig vei. Resistorer, kondensatoren og fotodioden ble loddet på uten problem.

Det ble oppdaget at footprintet til connectorene for mikrokontrolleren var snudd feil vei på kretskortet. Dette ble fikset ved å snu den og feste mikrokontrolleren på baksiden av kortet. Dette kan være en potensiell ulempe med tanke på festing av kortet på kalibreringsriggen, men det fungerte likevel som en rask fiks foreløpig.

Første test av det ferdigloddete kortet viste at det var en feil en plass i kretsen og tre av fire utganger på OPAMP-en visste 0V. Alle loddingene ble gått over og kontrollert, men etter endel feilsøking ble det funnet ut at OPAMP-en var loddet på feil vei.

Kretskort versjon 2

Et nytt kort ble så levert inn til Elektronikk og prototypelaboratoriet (Elpro-labben) ved NTNU for å få loddet på OPAMP-en rett vei og for å være sikker på at loddingen ble utført riktig. Dioden som var montert på det første kortet ble også avloddet her og flyttet over på det andre kortet.

Andre test av kortet ga ikke målinger som var forventet. Utgangene på OPAMP-en ga verdier, men disse var veldig lave og varierte ikke med lysintensiteten som traff dioden. Det ble forsøkt å endre forsterkning på OPAMP-en ved å endre motstander

men det ga ingen resultater.

Det ble foreslått å isolere OPAMP-en, og dermed sette på spenning på denne alene, for å se om denne i seg selv var problemet. Det ble også oppdaget at databladet var blitt tolket feil, slik at den faktisk trengte 4V som forsyningsspenning og ikke 3.3V. Det ble videre forsøkt å sende ut høyere spenning fra spenningsregulatoren på Curiosity Nano kitet, for å se om operasjonsforsterkeren behøvde mer spenningsforsyning for å fungere.

Problemet kunne se til å ligge ved ADC i mikrokontrolleren. Da utgangene på operasjonsforsterkeren ble koblet inn på de fire ADC-pinnene på mikrokontrolleren virket det som om de var jordet, slik at det ikke var mulig å få lest av verdier. Dette viste seg å ikke stemme, og etterhvert ble det oppdaget at problemet lå i OPAMP-en og hvordan den var koblet opp. Etter litt videre feilsøking ble det besluttet at dette var på grunn at utgangsverdien på fotodioden var en positiv strøm i stedet for en negativ. Som tidligere nevnt i seksjon 4.3.3 er OPAMP-en i kretsen koblet feil, ettersom det gruppen forventet var å få en negativ strøm. Dette gjorde da målingene uleselige for ADC-en.

Kretskort versjon 3

Kortet ble da levert til en av veilederne på gruppa for å modifiseres. Motstandene R6, R7, R10 og R11 (se figurer i vedlegg C.1) ble fjernet og koblingen der de sto ble kortsluttet. De positive inngangene på OPAMP-en som i utgangspunktet var koblet til jord ble koblet til VCC. Den nye koblingen har dermed utgangene på diodene koblet direkte til negativ inngang på OPAMP-en, med en feedback-motstand og VCC koblet til positiv inngang. Etter endringene ble signalet invertert, slik at lavere utgangsverdier betydde mer lys inn på fotodiodene.

8.2 Funksjonstesting

Etter at koblingen på kortet ble modifisert og fungerte ble det utført noen enkle funksjonstester for å kontrollere at hardware og software fungerte sammen. Det ble lest av verdier på utgangene på OPAMP-en som varierte med lysmengden som traff diodene. Når diodene var dekket til var utgangsverdiene 5V, og når det ble lyst på diodene gikk spenningen ned til 0V. Det ble også vist at de forskjellige kvadrantene som fungerte ga et litt forskjellig utslag når de mottok like mye lys.

Forskjellene mellom spenningene som ble målt var på omkring 20mV. Dette tyder på at kvadrantene i fotodioden har litt forskjellig egenskaper.

Det ble forsøkt å lese av vinkler fra sensoren ved å lyse på diodene med en håndholdt lommelykt med etuiet over dioden og rotere den rundt diodene. Det ble observert at θ endret seg når man beveget seg rundt diodene, men siden disse målingene bare ble gjort for hånd var ikke innfallsvinkelen kjent og målingene var ikke-konklusive.

8.2.1 Defekt fotodiode

Videre ble det lest av ADC verdier direkte fra mikrokontrolleren. Det ble da oppdaget at en av kvadrantene på dioden oppførte seg ulikt som de andre, ved at den viste 0V når de andre lå rundt 4V. Det ble dermed iverksatt en større prosess med feilsøking for finne ut hvor problemet lå.

Det ble oppdaget at dersom man holdte fingeren på overgangen mellom utgangen av den aktuelle kvadranten og inngangen av OPAMP-en gikk spenningen på utgangen fra 0V til 5V, men den endret seg ikke dersom lysintensiteten på dioden endret seg. Det ble forsøkt å lodde denne overgangen på nytt, men det løste ikke problemet. Det ble antatt at når man holdte fingeren på overgangen ble kretsen manipulert på en ukjent måte som ikke var forbundet med det reelle problemet.

For å finne ut om feilen lå i dioden eller på kretskortet ble kortet testet for dårlige baner eller brudd. Dette ble utført med et multimeter i kontinuitetsmodus, men det ble ikke funnet noen brudd som kunne forklare feilen. Det ble etter dette antatt at feilen lå i selve dioden.

Det ble også observert at dersom det ble påført et fysisk trykk ned på dioden, virket den som den skulle. Alle kvadrantene ga da ut omtrent samme spenning og alle reagerte likt når det ble lyst på de. Dersom trykket ble fjernet sluttet den ene kvadranten å fungere igjen og falt ned til 0V. Det ble etter det antatt at en av kvadrantene på dioden hadde blitt skadet og defekt.

Gruppen hadde bestilt to dioder, men begge to hadde blitt montert på kretskort og montert av igjen. Det ble forsøkt å bytte dioder på det modifiserte kretskortet men det lot seg ikke gjøre, da gruppen ikke hadde tilstrekkelig med loddeutstyr for oppgaven. Den andre dioden som også hadde blitt loddet av ble senere undersøkt og visste seg å ha samme problemet som kunne tyde på at avloddningen fra kretskortet

var en for stor belastning.

8.3 Testing av software

Programvaren ble testet fortløpende mens den ble utviklet. Det ble brukt en enkel spenningsdeler for å teste at ADC-en på mikrokontrolleren leste av rett spenning. Når kretskortet var klart til testing ble det utført tester med alle 4 inngangene samtidig. Et multimeter ble brukt til å kontrollere den avleste verdien.

Overføring av data over I2C ble testet med både en Raspberry PI og en Arduino UNO der det ble skrevet enkle program i Python og “Arduino C” for å sende og motta data. For å teste utregningene ble det lagt inn simulerte ADC verdier, deretter ble vinklene og vektoren overført over I2C og resultatene ble sammenlignet med resultatene fra et program i MATLAB.

Debuggeren på Curiosity Nano brettet ble også brukt aktivt under utviklingen av programvaren for å feilsøke koden ved å lese av verdier på variabler og lese av aktuelle registre på mikrokontrolleren.

8.4 Akkumulering av sampler

Et virkemiddel som ble undersøkt for å redusere støy var å ta flere sampler under en ADC konvertering og så ta et gjennomsnitt av disse samplene. Det ble tatt målinger med 1, 8 og 16 sampler per konvertering der alle forhold ble forsøkt å holdes konstante. Amplituden på målingene var derfor forventet å være like.

Ved å bruke ligning 5.1 i kapittel 5.2 beregnes den totale samplingstiden ved ulike antall sampler med en klokkehastighet på ADC-en på 200kHz. Siden kravet til total samplingstid fra sensoren er på 2ms ble det ikke tatt målinger med 32 sampler siden kravet ikke tillater så lang samplingstid.

- 1 sample: 0.0785 ms
- 8 sampler: 0.6245 ms
- 16 sampler: 1.2485 ms
- 32 sampler: 2.4965 ms

Det ble tatt 720 målinger for hver av de tre konfigurasjonene på en enkelt kvadrant på dioden med 2ms mellom hver måling.

8.5 Kalibreringsprosess

Kalibreringsprosessen gjennomføres for å sikre at sensoren fungerer optimalt og som tiltenkt. Hensiktet bak det er å lage kalibreringsdata for å møte kravene til sensoren. Det er også nødvendig å gjennomføre dette for å kunne korrigere for feil som kunne oppstått under produksjonen av sensoren, som kan komme fra ulike kilder som for eksempel mekaniske og elektriske feil. Førstnevnte kan omfatte feiljusteringer i sensordesignet, som blant annet høyden mellom deksel og sensor, menneskelige eller tekniske feil under testutførelsen eller størrelse/form på apertureåpningen i dekselet. Elektriske feil kan omfatte signalforvrenging og støyproblemer, feil på grunn av analog/digital konvertering eller feil introdusert under signalprosesseringen.[36]

Det ble planlagt en kalibreringsprosedyre som skulle gjennomføres etter at det ble bekreftet at kretsen, dioden og software fungerte som planlagt. Prosedyren skulle utføres når sensoren var satt i kalibreringsriggen.

Kalibreringsprosedyre:

1. Ta noen enkle målinger ved å stille inn kalibreringsriggen til en bestemt vinkel og bekrefte at man får ut noenlunde riktige verdier fra sensoren.
2. Eventuell korrigerende av sensoren dersom det ble funnet store avvik i steg 1.
3. Kjøre LabVIEW programmet som belyser sensoren med lys fra alle vinkler innen FOV.
4. Analysere resultat fra steg 3 og utfør eventuelle kalibreringer som ble beskrevet i kapittel 3.4.3.
5. Kjøre LabVIEW programmet en gang til for nye avlesninger etter kalibrering.
6. Sammenlign resultatene før og etter kalibrering.

Dersom resultatene etter trinn 6 i prosedyren viser seg å ikke være tilfredsstillende må grundigere analyser av sensorens oppbygging utføres. Punkter som burde undersøkes inkluderer fysiske mål på dioder og deksel, plassering av sensor i kalibreringsriggen, plassering av lyskilde i forhold til kalibreringsriggen, mm.

9 Resultater

De resultatene som gruppen fikk ut i fra sensoren vil være ganske manglende da sensoren ikke fungerte helt som den skulle. Noen målinger har blitt gjort og vil fortsatt være nyttige, men den endelige funksjonen til sensoren vil ikke fungere for prototypen.

9.1 Gjennomførte målinger

9.1.1 Måling med ulike deksel

I kapittel 7.1 er det en tabell som viser forskjellige spenningsverdier ved bruk av forskjellige varianter for deksel. Dette var en av de få målingene som kunne gjennomføres på en fornuftig måte da man er bare avhengig av at en kvadrant på fotodioden fungerer. Målingene ble bare gjort ved å lese av en utgang. Grunnen til at dette ble gjort er fordi man er ute etter å se spenningsverdiene ved utgangen til OPAMP-en etterhvert som fotodioden blir belyst, men da man kan lese av spenningen til en spesifikk utgang virket det fornuftig.

De forskjellige kvadrantene i fotodioden gir ut litt forskjellige strømmer når det er samme lys over dem slik at målingene blir mer nøyaktig om man bare ser på samme kvadrant. Denne målingen foregikk da ved å måle den samme kvadranten sin spenning ved utgangen til OPAMP-en ved forskjellige dekselvarianter. Gruppen brukte deksel tykkelser på 0,3mm, 0,5mm og 0,7mm. De forskjellige fargene for deksler som ble brukt var grå og svart.

Prosessen av denne målingen foregikk ved at gruppen systematisk leste av spenningsverdiene fra kvadranten på fotodioden med et spesifikt deksel og skrev ned verdien. Dekslene ble byttet ut mellom målingene. Det ble også lest av målinger i tilfeller hvor lyskilden var av. Dette er interessant fordi man kan finne ut av hvor mye støy det naturlige lyset i rommet tilføyer. Ved å gjøre en måling hvor lyskilden var av og fotodioden ikke var dekket, i tillegg til en måling hvor lyskilden var av og fotodioden var dekket, og dermed se på forskjellen mellom disse, kan man si noe om støynivået lyset i rommet står for. Tabellen i kapittel 7.1, som er basert på disse målingene, viser en 0,06V forskjell fra disse målingene. Dette utgjør ikke noen stor forskjell, men da dette ble testet i et miljø hvor det var lite naturlig lys vil man

kunne få andre resultater om man tester i et bedre belyst miljø.

9.1.2 Akkumulering av sampler

Resultatene ved forsøkene med ulike antall sampler per ADC avlesning ligger som vedlegg F. En sammenligning som viser høyeste spenning avlest, laveste spenning avlest, differansen mellom de to og standard avviket ved de ulike antall samplene kan sees i tabell 9.1.

	Min	Max	Differanse	Standard avvik
1 sampele	4.1809V	4.1931V	0.0122V	0.0031V
8 sampler	4.1736V	4.1907V	0.0171V	0.0031V
16 sampler	4.1858V	4.1968V	0.0110V	0.0028V

Tabell 9.1: Minimum og maksimum verdi ved ulikt antall sampler

9.2 Simuleringer

Da målingene fra sensoren vil være brukbare i forskjellig grad ble det bestemt at gruppen skulle lage en liste over resultater som var forventet fra sensoren. Disse resultatene har blitt regnet ut ved å se på hvor mye av arealet til hver kvadrant i fotodioden som blir dekket av sollys, og forholdet mellom de opplyste arealene. Denne måten å regne ut resultater vil ligne veldig mye på formlene som kan finnes i kapittel 3.4.1, men her vil man se på forholdet mellom opplyst areal for kvadrantene og ikke strømforhold. Det vil ikke være mulig å bruke strømforholdene for å regne ut verdiene på grunn av manglende målinger, men sammenhengen mellom opplyst areal og generert strøm i fotodiodene antas å være proporsjonale. Det opplyste arealet til hver enkelt kvadrant vil være mulig å regne ut med en 60° FOV. Målingene har blitt gjort med et kvadratisk dekselåpning som strider i mot det endelige designet til gruppen da en sirkulær dekselåpning ble brukt. Matlab har blitt brukt for å lage script og målinger for disse verdiene. Scriptene og målingene kan finnes i vedlegg H.

9.2.1 Matlab scripts for forventede målinger

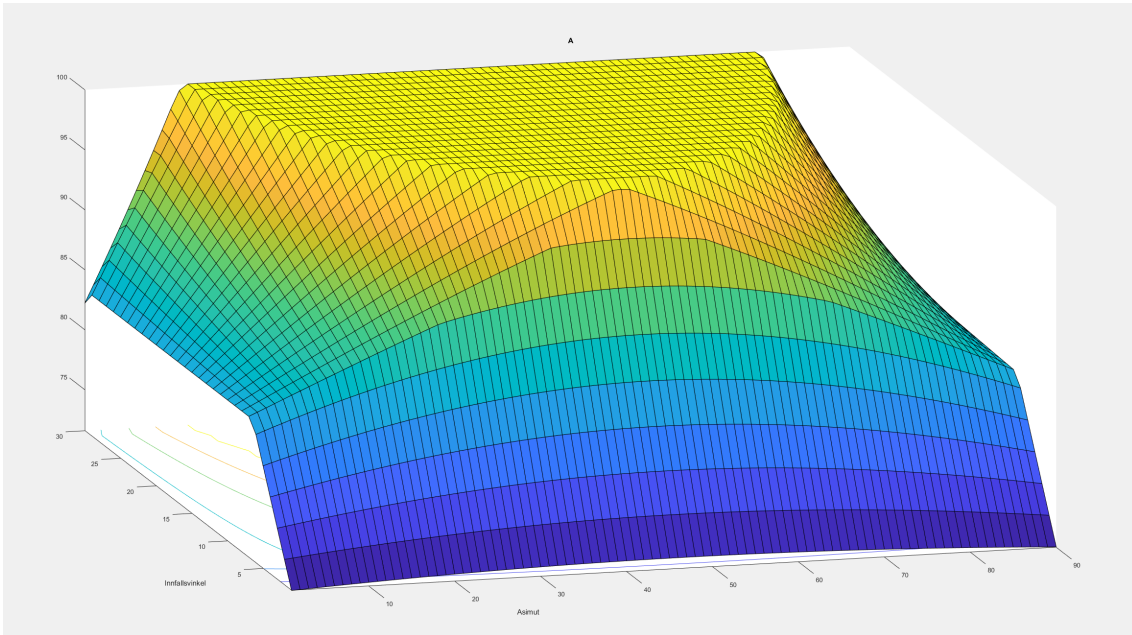
Scriptene som er laget i Matlab illustrerer hvor mye av hver enkelt diode som blir lyst opp ved en varierende vinkel theta (asimut), og en varierende phi (innfallsvinkel

til lyset). De øverste erklæringene i alle matlab scriptene er konstante størrelser som gruppen har kommet fram til. Formler for å regne ut tappt sollys og total bredde til sollys kan finnes ved å se på figurene i kapittel 7.1. Disse scriptene tar for seg vinkel theta fra 0° til 89° , og en vinkel phi fra 0° til 30° . Dette vil føre til at kvadrant A vil være mest belyst ved forskjellige verdier for theta og phi. Theta og phi vil kunne ha andre verdier hvor det vil være mulig å hente fornuftige målinger, men dette vil bare føre til at forholdene mellom opplyst areal i diodene blir forskyvet. Dersom man bruker en theta på 90° til 180° vil det være kvadrant D som får mest lys, slik at denne blir som kvadrant A var for en theta verdi fra 0° til 90° . Nullpunktet for theta er beregnet til å være i bunnen, mellom kvadrantene C og D.

Figurene H.1, H.2, H.3 og H.4 viser script for å regne ut opplyst areal i prosent for kvadrant A, B, C og D respektivt. Figurene H.5 og H.6 viser to deler av et script. Det har blitt gjort om til to figurer fordi scriptet er såpass langt. Dette scriptet kaller på verdiene til de andre scriptene og regner ut vinkel theta (asimut) og vinkel phi (innfallsvinkel til lyset). I tillegg plotter den flere figurerer som viser opplyst areal i hver kvadrant i tillegg til plot for utregnet theta og phi. Som nevnt tidligere har simuleringene blitt gjort med en kvadratisk apertur. Dette vil føre til forskjeller i målingene i forhold til dersom simuleringene hadde blitt gjort med en sirkulær apertur. Scriptet i figur H.5 og H.6 har derfor blitt tilpasset for å passe et deksel med kvadratisk apertur.

9.2.2 Målinger fra Matlab script

Etter å ha kjørt det siste scriptet blir seks figurerer laget som viser forskjellige plot. figurene H.7, H.8, H.9 og H.10 viser hvor stor prosentandel av kvadratens areal som er opplyst ved de forskjellige verdiene for phi og theta. H.7 viser for kvadrant A, H.8 for kvadrant B, H.9 for kvadrant C og H.10 for kvadrant D. Figurene H.11 og H.12 viser utregninger av vinkel phi og vinkel theta respektivt.



Figur 9.1: Måling med innfallende sollys.

Simuleringene som gruppen har gjort i Matlab ser ut til å gi mye mening. Hvis man ser på det opplyste arealet i kvadrant A, så vil det være mer av dette arealet som er lyst opp enn noen av de andre figurene. Når theta og phi er begrenset fra 0° til 90° og 0° til 30° respektivt vil det være kvadrant A som får mest sollys ved de forskjellige verdiene. Etersom at theta har disse begrensningene vil opplyst areal i kvadrant B og D være like, men ved forskjellige verdier av theta. Dersom theta = 45° vil figurene for B og D være identiske med tanke på det opplyste arealet fordi lyset vil komme fra det høyre hjørnet i kvadrant C. Ved en theta vinkel på 30° vil det være mer lys som treffer i kvadrant B slik at dette arealet blir mer opplyst. Dersom man ser på en theta vinkel på 60° vil kvadrant D være mer opplyst enn kvadrant B. Ut i disse tre tilfellene for vinkel theta kan en sammenheng finnes for kvadrant B og D. Ved en theta $< 45^\circ$ vil kvadrant B få mer sollys, ved en theta $> 45^\circ$ vil D få mer sollys og ved en theta = 45° vil det være likt med sollys for begge kvadranter. Denne sammenhengen vil gjelde så lenge vinkel phi er større enn 0° . C vil være den kvadranten som får minst sollys ved de forskjellige verdiene til theta og phi. Dersom phi er 0° vil alle kvadrantene få like mye lys, men etterhvert som innfallsvinkelen blir større vil lyset bli forskjøvet lenger unna denne C kvadranten.

Plot av utregningene til vinkel phi og vinkel theta finnes kan sees i figurene H.11

og H.12, og disse i likhet med de andre simuleringene, ser ut til å stemme ganske bra. Overflatene burde være lineære, men da disse simuleringene er gjort med en kvadratisk åpning vil det bli noen forskjeller i utregnede verdier. Simuleringene er nøyaktige nok til at man kan si at konseptet for sensoren gruppen har laget fungerer.

10 Diskusjon

Under denne seksjonen har gruppen dratt frem aspekter rundt oppgavegjennomførelsen og diskutert nærmere rundt disse. Slik er det blitt lagt frem hva som eventuelt kunne blitt gjort annerledes, eller hvilke fremgangsmåter gruppen benyttet for å forsøke å rette opp i situasjoner som kunne hatt negativ innvirkning på sluttresultatet.

10.1 Resultater

Det gruppen fant ut ifra testprosessen og kalibreringen, viste seg å ha mulighet til å gi veldig gode resultater. Det ga uttrykk for at dersom fotodioden hadde fungert som den skulle, ville solsensoren virke akkurat som gruppen hadde planlagt. Desverre ble ikke dette tilfellet, men det kan vise til at et slikt prosjekt absolutt er gjennomførbart.

Det at fotodioden ga ut forventet utgangssignal på alle kvadrantene dersom det ble lagt press på den kan indikere at det var noe feil med komponenten, ikke selve kretsen. Dersom det hadde vært tid nok til å bestille opp og sette sammen en revisjon to, ville nok denne gitt ut riktige kalibreringsresultater.

Resultatene som gruppen hentet fra sensoren vil desverre ikke være bra nok for å gi to nøyaktige vinkler for sensor i forhold til lyskilde. Som nevnt tidligere var det bare tre av fire kvadranter som viste en fornuftig spenning. I de fleste tilfellene viste den siste dioden 5V. Dette vil føre til at man ikke kan se forskjellen på resultatene hvor lyskilden står rett over kretskortet og når den er vinklet x-antall grader fra retningen hvor den defekte kvadranten er plassert. Det vil være umulig å skille disse vinklene da resultatet vil kunne være omtrent det samme for disse tilfellene. Den defekte kvadranten i fotodioden vil kunne gi en spenning på 5V hele tiden, som betyr at den ikke blir belyst, uavhengig av hvor mye lys som faktisk treffer kvadranten. Dersom forholdet mellom de tre andre kvadrantene er det samme ved forskjellige vinkler vil resultatet bli det samme.

Resultatene fra ADC-en med forskjellige antall sampler viste ingen merkbare forskjeller mellom 1 og 8 men for 16 sampler ble standardavviket til avlesningene noe redusert som er forventet. Årsaken til at det ikke var noen forskjell mellom 1 og 8 sampler kan skyldes at målingene ble utført på feil måte eller at det ikke ble tatt nok sampler. Disse målingene burde utføres på nytt da de strider med forventningene

om at standardavviket burde gå ned med økt antall sampler.

Ulempen med å ta flere sampler er økt samplingtid og økt strømforbruk, der samplingstiden begrenset antall sampler i denne oppgaven til 16. Det burde merkes at antall sampler kan økes ved å øke klokkehastigheten til ADC-en, noe som ikke ble undersøkt i denne oppgaven. Hvordan strømforbruket påvirkes ble heller ikke undersøkt.

10.2 Feilkilder

I forbindelse med prosjektgjennomførelsen har det skjedd en del som kunne påvirket sluttresultatet på ulike vis. I seksjonen under blir det tatt opp de forskjellige scenarioene som oppsto underveis og det blir videre diskutert rundt hvilke måter gruppen har valgt å håndtere disse situasjonene for å prøve å forbedre resultatet.

10.2.1 Fotodiode

Som nevnt før ble det enighet om å kun bestille opp en av de to fotodiodene som ble vurdert, på grunn av fraktkostnadene. Dessverre ble feil fotodiode tilsendt slik at dioden gruppen faktisk bestilte ikke nådde frem. Det så derimot ut til at de var ganske like, derfor ble det bestemt at den feilsendte likevel skulle bli tatt i bruk, på tross av mulige små forskjellelr.

Under kalibreringen av kretsen viste det seg at det mest sannsynligvis var en feil i fotodioden, slik at denne kun gav ut tre riktige utgangsverdier istedet for fire. Dette påvirket resultatet så mye at det ikke ville blitt mulig å få logiske vinkelverdier. Det så ut til at det kunne være en feil inni selve dioden, fordi utgangsverdiene så riktige ut når det ble lagt press på fotodioden.

Det var ikke mulig å stå og legge konstant press på fotodioden under kalibreringsprosessen, siden stasjonen sensoren var montert inni ville rotere. Det ble forsøkt å legge på mer loddetinn på pinnene på fotodioden, noe som fungerte for en liten periode, men med en gang fotodioden ble berørt falt den ene kvadranten ut igjen. Gruppen kikket også på andre muligheter for å legge press på denne mens kalibreringsprosessen foregikk, men på grunn av dekslet som skulle ligge over fotodioden var det ikke mulig å få det til på en fornuftig måte.

Vurderingen fra gruppen videre ble at det beste alternativet var å forsøke å lodde

på den andre fotodioden på kortet, siden den som var montert på fra før allerede var defekt. Det var ikke sikkert at denne fungerte heller, for den hadde blitt loddet av et annet kort, men det ble sett som en siste utvei.

Det ble spekulert i at fotodiodene ikke tålte å bli avloddet, fordi begge hadde mest sannsynligvis vært igjennom hver sin avloddingsprosess. Siden ingen av de fungerte og det da ikke var mulig å teste denne hypotesen, er dette selvsagt ingen fast konklusjon på det, men heller en spekulasjon i hva som kan ha gått galt.

10.2.2 Feil i prototype 1

Ved en feil i design-prosessen fungerte ikke gruppens første utkast av kretskort, som mer grundig forklart i seksjon 4.3.3. Dette gjorde at kretskortet ikke fungerte, og gruppen mistet dyrebar tid i samarbeid med veileder for å finne problemet. Dette i seg selv er ikke uventet at kan skje i et studentprosjekt, men i sammenheng med at gruppen er blitt forhindret i å teste kortet grunnet treghet i komponentforsendelsen ble det mye mer utfordrende å få prototypen til å levere resultat. Som et resultat av forsinkelser utenfor gruppens kontroll ble derfor ikke nye kretskort bestilt opp i tide.

Det ble tidligere nevnt at Auto Route-funksjonen ble tatt i bruk under design av PCB. Det har ikke vist seg noe ulemper ved å bruke dette. Etter at kortet ble reparert med tanke på feilen, fungerte selve kortet akkurat slik gruppen hadde tenkt.

10.2.3 Forsinkelser med kretskort til prototype

Gruppen hadde diverse problemer med kretskortbestillingen for den første prototypen da Aisler, den tyske produsenten som det først var tenkt å bruke, ikke kunne lage kortet slik det var designet. Det så ut til at produsentens nettsider ikke kunne lage et firelags kort, gruppens kretskort ble vist slik det var, bare at det manglet to av planene. I og med at det selvsagt var viktig å ha det som et kort med fire lag ble det bestemt at det skulle bestilles fra JLCPCB i stedet.

I og med at det ble en del problematikk med bestillingen ble det brukt en del lenger tid på dette enn først tiltenkt. Først tok selve bestillingen ganske lang tid, deretter kom også komponentene ganske lang tid etter kortet. På nettsiden til JLCPCB var det oppgitt at det kunne ta lenger tid enn forventet til kortene kom frem pga. Covid-19 pandemien. Likevel kom kortene ganske kort tid etter, komponentene

derimot, hadde veldig lang leveringstid. Disse ble bestilt separat fra kretskortet, og brukte rundt 3 uker på å komme frem. Dette gjorde at lodde- og testedelen av oppgaven ble en del forsinket. Gruppen benyttet mye av denne tiden til å ferdigstille kalibreringsriggen, men likevel ble det en liten stopp i prosessen. Årsaken til at komponentene brukte så lang tid er foreløpig ukjent, men det kan spekuleres i at det kan ha sammenheng med både Covid-19, og komponentmangel som resultat av kinesisk nyttår.

10.2.4 Kalibreringsrigg

Som tidligere nevnt, ville det beste alternativet for kalibreringen vært hvis gruppen hadde hatt et laboratorium med kalibreringsrigg tilgjengelig. Et slikt oppsett ville trolig gjort kalibreringsresultatene enda mer nøyaktige, fordi det ville gitt bedre testomstendigheter. Selv om gruppen hadde klart å finne laboratorium i nærheten, ville det mest sannsynligvis blitt vanskelig å gjennomført kalibrering der, på grunn av restriksjonene som følge av pandemien. Det ble også vurdert å bestille opp deler, fordi dette kunne gitt resultater av en bedre standard.

I et ideelt scenario ville kalibreringen dermed foregått på en veldig annerledes måte. Det ble tidlig besluttet at det ikke ville være mulighet for å bestille opp deler til kalibreringsriggen, da dette ville overgått budsjettet ganske fort. Dermed sto gruppen igjen med å enten lage et enkelt kalibreringsoppsett som kanskje ikke ville holde til formålet, eller å designe og 3D-printe et mer robust system. Gruppen gikk for det siste alternativet, noe som da selvsagt var mye mer tidskrevende, og førte til et mye større tidspress under kalibreringen enn først forventet. I et slikt prosjekt er det klart at andre løsninger kunne gitt bedre resultater, men metoden som ble valgt var den beste løsningen gruppen hadde mulighet til å faktisk gjennomføre.

10.3 Reviderte prosjektmål

Etter å ha jobbet en del med oppgaven ble det klart at den var mye mer teknisk utfordrende enn først antatt. Orbit NTNU informerte om at de ikke hadde spikret oppgavemålene, fordi de selv ikke var helt sikre på hva som var oppnåelig. Som den del av prosessen har oppgaven blitt spisset ned til noe som er realistisk å gjøre med tiden og ressursene vi har fått tildelt. Målene har gått fra å potensielt lage noe som kan brukes på en satellitt, som innebærer da riktig størrelse på

sensoren, space-grade kvalitet, og som er kvalitetsikret i et vakuumkammer og under verdensromomstendigheter, til å ha et kort som fungerer og er kalibrert, og å strømlinje kalibreringsprosessen for fremtidige versjoner av kortet.

11 Videre arbeid

11.1 Endringer i prototype 2

For senere revisjoner av kortet er det en del ting som kan endres for å få bedre resultater. Deriblant er det mest sannsynligvis noe som kan endres etter å ha gjennomgått grundig testing i forbindelse med støy eller omgivelsene i rommet. Det er også mulig at komponenter bør byttes ut, dersom det senere kommer inn bedre muligheter på lager etter pandemien.

11.1.1 Motstander

Motstandene som er lagt på kortet for revisjon 2 ha samme verdier som revisjon 1. Eneste som er endret er footprintet, slik at andre resistanser med mindre case må bestilles for å få plass. I forbindelse med dette kan det være bedre med andre resistansverdier enn det som ble brukt i første revisjon, dersom andre verdier kan gi en mer egnet forsterkning. Det er da ikke spesifisert akkurat hvilke motstander som bør bestilles.

11.1.2 Fotodioden

Fotodioden som ble brukt på prototypekortet gjorde selve etui-designet problematisk for å kunne oppnå riktig field of view. Dette var på grunn av at glasset som ligger over diodekvadrantene er i veien, slik at avstanden mellom åpningen i etuiet og diodekvadrantene blir for stor i forhold til det som er nødvendig for å oppnå 60° FOV. Dette glasset er tilsynelatende heller ikke mulig å fjerne.

Det andre alternativet som ble vurdert, OPR5911, ville kanskje ikke ført til det samme problemet med etuiet. En annen ulempe som derimot kan oppstå med denne er trykk. Det er oppgitt i databladet til komponenten at den kan være sensitiv ovenfor høyt trykk eller høyt vakuum. Gruppen har derimot sett lignende prosjekt som har tatt i bruk denne komponenten, der det har gått fint å bruke den. Dermed er det rimelig å anta at denne fotodioden kan være en mulig bedre erstatning for fotodioden så lenge den passerer testing i vakuumkammer.

Det virket også for at fotodioden som ble brukt i prototype 1 ikke tålte å bli loddet av, så dersom denne likevel skal brukes i fremtidige versjoner, er dette noe som

må tas hensyn til. Gruppen anbefaler derimot å velge en annen fotodiode, som for eksempel OPR5911 eller OPR5925. Disse er de billigste forslagene, det finnes også flere dyrere alternativer som QP50-6-SM. Alle disse mulighetene er surface mounted devices, og de ligger ganske flatt inntil kortet, så det vil være enklere å lage et deksel til sensoren ved bruk av slike fotodioder.

11.1.3 Kalibreringsrigg

Klare mål for fremtiden er å videreutvikle kalibreringsprosessen, som kan gjøres ved å installere et akselerometer og en regulator som i “real time” leser av og korrigerer helningen for hvert av plana gjennom servomotorene. Et annet alternativ hvis man ønsker å spare penger er å lodde seg på potensiometeret i hver servomotor, og kartlegge ulineariteten til hver av motorene, og basere seg på disse resultatene under kalibrering av solsensorene.

11.1.4 Deksel

Da sensoren er tenkt seg ut i verdensrommet er det viktig å kunne beskytte kretskortet med komponenter slik at de skal kunne overleve å være i verdensrommet. Et deksel som skal fungere som en kapsel for kretskortet er da nødvendig. Deksel-designet for prototype 1 står beskrevet detalj i et senere kapittel, men momenter rundt etui for prototype 2 må forklares da løsningen blir annerledes. Formen til dette designet vil være nesten lik kretskortet da det er ønskelig å dekke hele kortet. Design muligheter for dette dekselet vil være å enten lage en slags boks som kretskortet kan legges inn i eller et tettsittende belegg som kortet kan tres inn i. Den første løsningen er nok den enkleste da man ikke trenger å være like nøye med å følge formen til kretskortet. Man vil kunne slippe å følge omrissene til visse komponenter avhengig av hvordan boksen, men en slags lukke/åpne funksjon vil være nødvendig da man må kunne legge kortet inn i boksen. Løsningen med det tettsittende dekselet vil kreve mer nøyaktighet med tanke på omrissene til komponenter på kretskortet, slik at denne løsningen vil kreve mer arbeid. Det burde ikke være noen forskjell mellom designene med tanke på prestering, så lenge begge modellene er laget slik at fotodiodene ikke er helt tildekket da man er avhengig av at sollys skal kunne treffe komponenten.

11.2 Software

Programvaren som ble utarbeidet ser ut til å fungere bra, men siden kretskortet fungerte ble det ikke mulig å teste det til sitt fulle. Programmkoden har alle nødvendige implementeringer med unntak av muligheten for å endre I2C adresse som er nødvendig dersom man skal kommunisere med flere sensorer samtidig. Det ble heller ikke undersøkt hvor lang samplingstid programmkoden bruker. I tillegg vil det alltid være rom for optimaliseringer som for eksempel mulighet til å aktivere 'sleep mode' når sensoren ikke brukes.

11.3 Videre testing

Det er en del tester en solsensor må gjennom for å sikre at den er i stand til å takle forholdene i rommet. Disse testene gjennomføres i simulatorer som er til for å lage forskjellige miljø som skal tilsvare de faktiske omstendighetene sensoren blir møtt med i verdensrommet. Det ble avgjort at disse testene ikke skulle gjennomføres av gruppen, spesielt på grunn av tidsskjemaet og fordi de fleste av disse testfasilitetene ikke kunne slippe inn alle i gruppen grunnet smitterestriksjoner. De neste seksjonene beskriver mer detaljert om hva disse testene er og hva de gjør, og hvordan gruppen hadde planlagt å faktisk gjennomføre de i utgangspunktet.

I verdensrommet er det ikke den samme atmosfæren som på jorden, jo lenger ut i verdensrommet man går, desto lavere blir tettheten til atmosfæren. Langt nok ut i rommet kan mangelen på atmosfære føre til enorme temperaturforskjeller. I direkte sollys kan temperaturene ligge rundt 120°C , mens tilskyggede objekter kan nå ned til -100°C . Dette er noe gruppen har prøvd å kompensert for i forbindelse med komponentvalg.

Det er også en betydelig større mulighet for strålingsskade, hovedsaklig fra sola. Dette kommer av at den atmosfæriske beskyttelsen ikke lenger er tilstede. Et annet aspekt, som også er påvirket av mangelen på atmosfære, er at mye av de ytre delene av verdensrommet fungerer nesten som et vakuum. Dette er fordi det nesten er helt tomt, noe som resulterer i veldig lavt trykk i verdensrommet. [37]

11.3.1 Sammenligning med kommersiell sensor

Gruppen tenke i utgangspunktet å samle inn resultatene fra sensoren, og dermed sammenligne disse med samme type resultater fra en kommersiell sensor. Det var planlagt å sammenligne med sensoren Orbit NTNU har tilgjengelig, NanoSense fra Gomspace, for å se om resultatene fra den selvproduserte sensoren var gode nok til at den kunne erstatte den kommersielle. I og med at gruppen ikke klarte å gjennomføre all testing, er dette noe som bør gjennomføres etter at ny sensor er ferdigprodusert.

11.3.2 Vakuum-testing

Det lave trykket som er i verdensrommet, er noe forskere har klart å tilnærme i simuleringskammer. Et vakuumkanter er et kammer der gasser fjernes ved hjelp av en vakuumpumpe, slik at det ender som et lavtrykksmiljø. Et viktig aspekt var dermed å velge komponenter som best kunne passe omgivelsene i verdensrommet, med tanke på det lave trykket. Selve kortet må også være laget av materiale som håndterer dette. Disse delene ville Orbit NTNU heller stå for selv når de skulle endre på og bestille opp kort.

Tanken bak det med vakuum-testingen var at gruppen kunne teste kortet sitt ved fasiliteter ved Simpro. Det ble derimot informert allerede veldig tidlig i prosessen om at det kunne bli vanskeligheter dersom det ikke ble lettelser i restriksjonene i forbindelse med pandemien. Disse har ikke endret seg stort over tidsforløpet, så det ble avgjort at gruppen ikke skulle bruke unødvendig tid på å planlegge for dette, siden det mest sannsynligvis ikke ville bli mulig å gjennomføre uansett.

Dersom gruppen skulle testet for dette, ville det viktigste vært å ha fokuset på komponenter som tåler denne typen omgivelser. I det tilfellet at det hadde latt seg gjøre å gjennomføre testene, ville gruppen forsøkt å fått notert ned trykket brukt i kammeret, utgangsverdiene fra alle diodekvadrantene, den digitale verdien fra mikrokontrolleren, og om diodene hadde vært opplyst eller ikke. Testen skulle blitt gjennomført flere ganger, med ulike verdier og omstendigheter, for å få et bredt resultatspekter. Deretter skulle disse resultatene blitt satt opp i en tabell for sammenligning, slik at det ville vært mulig å se om fotodioden, og resten av kretskortet hadde fungert som optimalt i vakuumkanteret.

11.3.3 Termisk testing

I sammenheng med de store temperaturforskjellene som kan oppstå ute i rommet, er det også nødvendig å gjennomføre termisk testing for å se at sensoren tåler det. Gruppen har forsøkt å velge komponenter med stort sprang i temperaturområder, slik at de kan fungere i omgivelsene uten at de blir påvirket negativt på noe vis. Det ble forsøkt å finne komponenter med tilstrekkelig temperaturområde, såfremt det faktisk lot seg gjøre.

Testene skulle i utgangspunktet gjennomføres ved fasiliteter for det, dersom omstendighetene hadde talt for det. Forutsatt at det hadde vært mulighet for det, ville gruppen gjennomført tester i forskjellige temperaturområder, og notert ned hvilke utgangsverdier sensoren hadde hatt. Det viktigste ville vært å teste ytterpunktstemperaturene, og sjekket om sensoren likevel hadde fungert optimalt i disse tilstandene. Dette er dermed noe som må testes i fremtidig viderearbeid med sensoren.

11.3.4 Termisk vakuumtest

I et termisk vakuumkanne er det mulig å simulere slik at miljøet er enda nærmere det faktiske miljøet i verdensrommet. Det kammeret gjør er at det fjerner luft og senker trykket, samtidig som at det går gjennom veldig høye og veldig lave temperaturer. [38] Ved å gjennomføre denne typen test, vet man mer sikkert at sensoren tåler de realistiske omgivelsene. Hovedmålet er da at den skal få til å håndtere både vakuum og de store temperaturendringene samtidig.

11.3.5 Vibrasjonstesting

Under oppskytningen blir romfartøy utsatt for sterke påkjennelser som blant annet intense vibrasjoner, noe som kan påvirke deler og komponenter på fartøyet. For å vite at sensoren overlever oppskytningen er det derfor nødvendig at den blir testet for slike påkjennelser ved hjelp av vibrasjonstesting. Dette gjøres for å se hvordan systemet vil reagere på vibrasjonene det blir utsatt for, og om det da bør forbedres som følge av dette. Det kan være det blir nødvendig å bytte ut komponenter som viser seg å ikke tåle kraftige vibrasjoner, og heller sette inn komponenter som kanskje er mer egnet.

11.3.6 Strømforbruk

En av kravene som ble satt at Orbit omhandlet strømforbruket til sensoren. De hadde tatt utgangspunkt i strømforbruket til Gomspace sensoren de tidligere hadde brukt, og justert kravene noe for oppgaven. Her var det hovedsakelig snakk om strømforbruk når sensoren samplet data og når den ikke samplet. Strømforbruket til sensoren vil være avhengig av hvilke komponenter som blir brukt og samplings frekvens. I kretsen som gruppen lagde ville det vært OPAMP-en og MCU-en som i hovedsak hadde bestemt strømforbruket. Gruppen gjorde ikke noen strømmålinger på deres prototype, da de hovedsakelig fokuserte på å få sensoren til å fungere. Dersom gruppen hadde sett at virkemåten til sensoren fungerte, ville det da blitt gjort målinger for å se hvilke tiltak som burde gjøres i neste versjon av sensoren mtp strømforbruk. En ide som gruppen har implementert i versjon to er en bryter til spenningsforsyningen til OPAMP-en, slik at den ikke trekker strøm kontinuerlig. Videre kan det da implementeres en funksjon som kun skrur på OPAMP-en når data skal samples. Andre tiltak som kan gjøres er optimalisering av kode, og eventuelt bytte ut nåværende MCU med en som har lavere strømforbruk.

11.4 In-orbit kalibrering

Etter at sensoren er ferdigprodusert og den skal sendes opp er det viktig å tenke på de miljømessige effektene sensoren kan bli utsatt for i verdensrommet. Målenøyaktigheten til sensoren kan bli påvirket av selve installeringen på romfartøyet, utskytningsprosessen eller miljøet i verdensrommet.[36] Dette er feil som ikke kan korrigeres for i særlig stor grad på bakkenivå. De må da rettes opp etter at solsensoren er montert på satellitten, og den er blitt utsatt for disse påkjennelsene. Dette går under miljømessige effekter, blant annet temperatur kan påvirke solsensorens målenøyaktighet. En økning i temperatur, samt påvirkning fra strålingsmengden i verdensrommet kan øke støynivået i utgangssignalet til sensoren. Dark current kan også være vesentlig temperaturavhengig [39], så en betydelig temperaturøkning kan også føre til en økning av mengden strøm som går, uten at fotodiodene faktisk detekterer lys. For å få økt målenøyaktigheten bør man dermed ta høyde for in-orbit kalibrering av sensoren etter at satellitten er blitt sendt opp i verdensrommet. Dette er noe gruppen ikke kan utføre, i og med at oppsendingen vil foregå etter at prosjektet er endt.

12 Konklusjon

Spørsmålet om gruppen klarte å løse oppgaven vil kunne ha forskjellige svar ut ifra hvilke aspekter av oppgaven man ser på. En sensor ble laget, men funksjonene fungerte ikke helt som de skulle. Som diskutert tidligere kan en dårlig kobling i fotodioden ha vært problemet som gjorde at sensoren ikke fungerte. Dersom den siste fotodioden hadde fungert som den skulle, noe den gjorde hvis man trykket ned på komponenten, ville sensoren kunne virket som ønsket. Om dette hadde vært tilfellet ville vi ha kunnet fullført oppgaven, fordi alle de resterende stegene i operasjonen hadde vi allerede bekreftet at fungerer. Dette vil si at selv om gruppens sluttresultat mangler en fysisk, fungerende løsning har arbeidet de har gjort banet vei for en revidert prototype.

Basert på dette tror gruppen at resultatene som er oppnådd vil være verdifulle for Orbit. Eksempelvis kan både kalibreringsrigg og software gjenbrukes. Informasjonen og erfaringene som gruppen har tilegnet seg om kalibreringsprosessen vil være svært viktig for Orbits fremtidige selvproduserte solsensorer. Kalibreringsriggen viser hvor enkelt det er å designe et oppsett for å roterer sensoren, og utregningene for dekselstørrelse og sammenhengene som er funnet i denne delen av rapporten vil også kunne gjøre prosessen av å designe en liknende løsning rask og enkel for Orbit NTNU.

Softwaren som har blitt brukt for å implementere avlesing fra mikrokontroller vil være nyttig for fremtidige prosjekter da softwaren for disse prosjektene kan ta inspirasjon fra det gruppen har gjort i sin oppgave. Endringer må selvfølgelig gjøres slik at man får laget en solsensor som har de egenskapene som er ønsket, men idéene og løsningene som gruppen har kommet på i denne oppgaven vil da være uvurderlige i fremtidige prosjekter. Oppgaven kan i den forstand anses som velykket ettersom dette baner vei for en betraktelig mye lettere oppgave for Orbit NTNU når de skal begi seg ut på produksjon av solsensorer.

Referanser

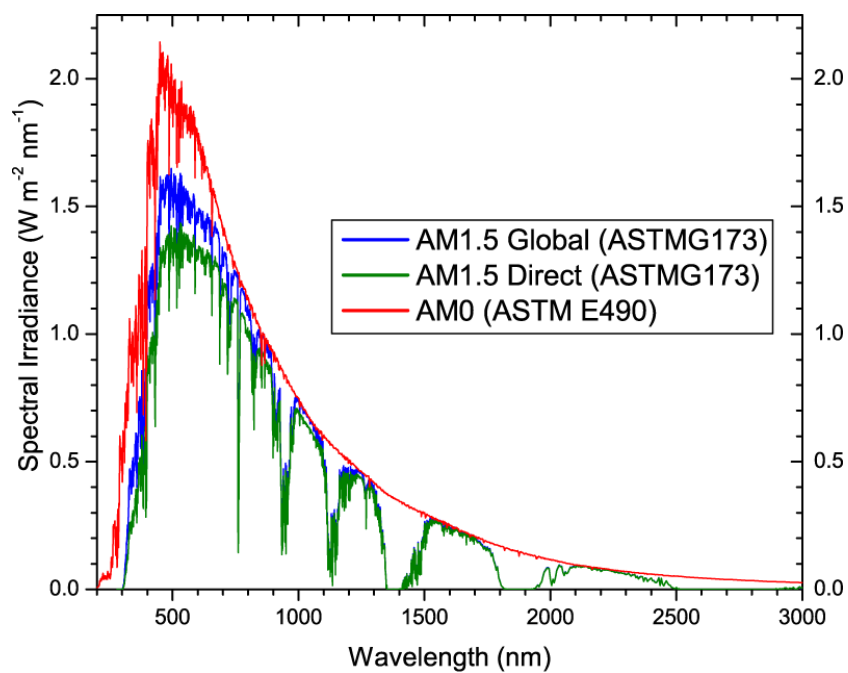
- [1] Sarah Loff. *CubeSats Overview*. URL: https://www.nasa.gov/mission_pages/cubesats/overview (sjekket 01.03.2021).
- [2] Mathias Støle. *Current Mission*. URL: https://www.orbitntnu.com/current_mission/ (sjekket 01.03.2021).
- [3] Álvaro Enrich Carmona. “Sun sensors for small satellites attitude determination systems”. I: 2017.
- [4] AZO Sensors. *What is a Sun Sensor?* 26. jun. 2013. URL: <https://www.azosensors.com/article.aspx?ArticleID=223> (sjekket 02.03.2021).
- [5] Faizullin Dmytro. “Improvement of Analog Sun Sensor Accuracy and Data Processing for Sun Vector Determination”. I: 2018, s. 17.
- [6] Hywel Curtis. *An overview of sun sensors available on the global marketplace for space*. 2020. URL: <https://blog.satsearch.co/2020-02-12-sun-sensors-an-overview-of-systems-available-on-the-global-marketplace-for-space> (sjekket 12.03.2021).
- [7] Ahad Ali og Fahad Tanveer. “Low-cost design and development of 2-Axis digital sun sensor”. I: *Journal of Space Technology vol.1*. 2011, s. 1.
- [8] Minsong Wei og Fei Xing og Zheng You. “A highly accurate wireless digital sun sensor based on profile detecting and detector multiplexing technologies”. I: *Mechanical Systems and Signal Processing* 82 (2017), s. 56–67.
- [9] Konstantin Bolshakov m.fl. “Array-based digital sun-sensor design for CubeSat application”. I: *Acta Astronautica* 178 (2021), s. 81–88. URL: <https://www.sciencedirect.com/science/article/pii/S0094576520304926>.
- [10] Henrik Rudi Haave. “Simulating sun vector estimation and finding gyroscopes for the NUTS project”. I: 2016, s. 36.
- [11] Adel S. Sedra og Kenneth C. Smith. *Microelectronic Circuits, seventh edition*. New York: Oxford University Press, 2015, s. 228.
- [12] Richard Rennie og Jonathan Law. “Photodiode”. I: *A Dictionary of Physics Eighth edition*. Oxford University Press, 2019. URL: <https://www.oxfordreference.com/view/10.1093/acref/9780198821472.001.0001/acref-9780198821472-e-2291> (sjekket 08.03.2021).
- [13] Wavelength electronics. *PHOTODIODE BASICS*. URL: <https://www.teamwavelength.com/photodiode-basics/> (sjekket 19.05.2021).

- [14] Robert Keim. *Understanding Photovoltaic and Photoconductive Modes of Photodiode Operation*. 20. des. 2020. URL: <https://www.allaboutcircuits.com/technical-articles/understanding-photovoltaic-and-photoconductive-modes-of-photodiode-operation/> (sjekket 07.05.2021).
- [15] Jacob Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications, Fifth edition*. Cham: Springer International Publishing AG, 2015.
- [16] AP Technologies. *Bi-Cell & Quadrant Photodiodes*. 2021. URL: <https://www.aptechnologies.co.uk/support/SiPDS/bicell-quad> (sjekket 08.03.2021).
- [17] Robert Keim. *Transimpedance Amplifier: Op-Amp-Based Current-to-Voltage Signal Converter*. URL: <https://www.allaboutcircuits.com/video-tutorials/op-amp-applications-current-to-voltage-converter/> (sjekket 19.05.2021).
- [18] Tahir Akram Ishtiaq Maqsood. “Development of a Low Cost Sun Sensor Using Quadphotodiode”. I: 2010.
- [19] Rashid Saleem; Sukhan Lee. “Accurate and Cost-Effective Micro Sun Sensor based on CMOS Black Sun Effect”. I: *MDPI* (2019).
- [20] Minsong Wei; Fei Xing; Zheng You. “An implementation method based on ERS imaging mode for sun sensor with 1 kHz update rate and 1 precision level”. I: ().
- [21] Martin H. Weik. “phototransistor”. I: *Computer Science and Communications Dictionary*. Springer US, 2001, s. 1273–1273. URL: https://doi.org/10.1007/1-4020-0613-6_14020 (sjekket 18.03.2021).
- [22] Circuit Globe. *Difference Between Photodiode and Phototransistor*. URL: <https://circuitglobe.com/difference-between-photodiode-and-phototransistor.html> (sjekket 18.03.2021).
- [23] Electronics Hub. *Basics of Phototransistor*. 23. mar. 2019. URL: <https://www.electronicshub.org/basics-of-phototransistor/> (sjekket 18.03.2021).
- [24] First Sensor. *QP5.6-TO5*. URL: https://no.mouser.com/datasheet/2/313/QP5-6_T0_501040-586454.pdf (sjekket 06.05.2021).
- [25] First Sensor. *QP5.8-6-TO5*. URL: https://no.mouser.com/datasheet/2/313/fsens01104_1-2279890.pdf (sjekket 06.05.2021).
- [26] Mouser Electronics. *QP5.8-6-TO5*. URL: <https://no.mouser.com/ProductDetail/First-Sensor/QP58-6-T05?qs=IxDs%252B9hDEuBqHy7FDZkWuA%3D%3D> (sjekket 09.04.2021).

- [27] Microchip Technology Inc. *AVR DB Family Datasheet*. 2020. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB28-32-48-64-Datasheet-DS40002247A.pdf> (sjekket 25.03.2021).
- [28] Microchip Technology Inc. *AVR128DB48 Curiosity Nano Hardware User Guide*. 2020. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB48-Curiosity-Nano-HW-UserG-DS50003037A.pdf> (sjekket 07.05.2021).
- [29] Texas Instruments inc. *OPA4188 0.03-uV/°C Drift, Low-Noise, Rail-to-Rail Output, 36-V, Zero-Drift Operational Amplifiers*. URL: https://www.ti.com/lit/ds/sbos641d/sbos641d.pdf?ts=1621440316099&ref_url=https%253A%252F%252Fwww.google.com%252F (sjekket 19.05.2021).
- [30] *Solar Radiation*. URL: <https://www.whoi.edu/science/AOPE/mvco/description/SolRad.html#:~:text=Above%20the%20earth's%20atmosphere%2C%20solar,noon%20in%20the%20summer%20months>.
- [31] Texas Instruments Inc. *LMx24, LMx24x, LMx24xx, LM2902, LM2902x, LM2902xx, LM2902xxx Quadruple Operational Amplifiers*. URL: https://www.ti.com/lit/ds/symlink/lm124.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1621362128099&ref_url=https%253A%252F%252Fn.o.mouser.com%252F (sjekket 18.05.2021).
- [32] Nexperia USA inc. *PEMD13; PUMD13: Product data sheet*. URL: https://assets.nexperia.com/documents/data-sheet/PEMD13_PUMD13.pdf (sjekket 18.05.2021).
- [33] Sal Afzal. *I2C Primer: What is I2C? (Part 1)*. URL: <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html> (sjekket 18.04.2021).
- [34] Stanford Engineering. *Solar Simulator*. URL: <https://uytengsuteachinglab.stanford.edu/equipment-supplies/chemical-and-material-sciences-instrument/solar-simulator> (sjekket 15.04.2021).
- [35] Katie Kailus. *The Health Benefits of Sunlight — And How LED Lighting Compares*. 5. apr. 2017. URL: <https://www.alconlighting.com/blog/newsfeed/health-benefits-natural-lighting-led-lighting-compares/> (sjekket 15.04.2021).
- [36] Shu-Fan Wu og W. H. Steyn. "In-Orbit Modelling and Calibration of the Sun Sensors on UoSAT-12 and Tsinghua-1 Satellites". I: 2001.

- [37] Donavyn Coffey. *Why is space a vacuum?* 2020. URL: <https://www.livescience.com/why-is-space-a-vacuum.html> (sjekket 19.04.2021).
- [38] Element Materials Technology. *Product qualification testing Thermal Vacuum Testing*. URL: <https://www.element.com/product-qualification-testing-services/thermal-vacuum-testing> (sjekket 22.04.2021).
- [39] Dr. Rüdiger Paschotta. *Dark Current*. URL: https://www.rp-photonics.com/dark_current.html (sjekket 07.04.2021).
- [40] PVEducation. *Standard Solar Spectra*. URL: <https://www.pveducation.org/pvcdrom/appendices/standard-solar-spectra> (sjekket 19.05.2021).

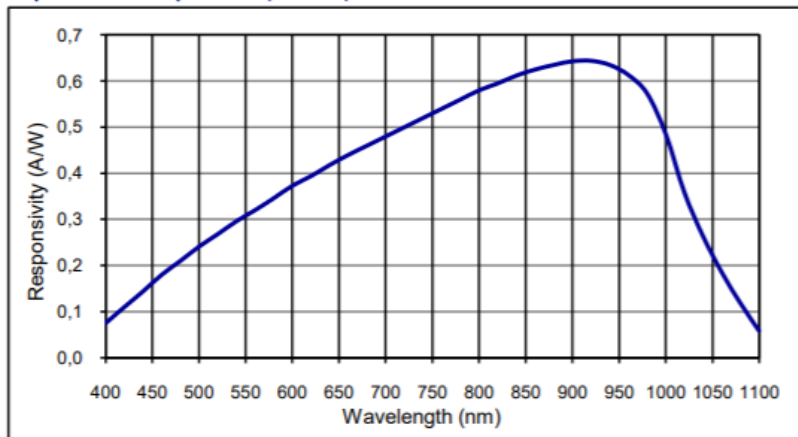
A Solens strålespektrum



Figur A.1: Solstrålingsspektrum[40]

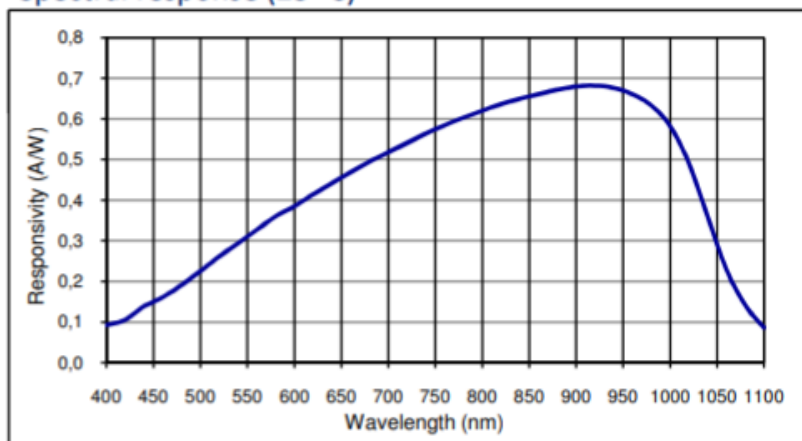
B Spektral respons fotodioder

Spectral response (23 °C)



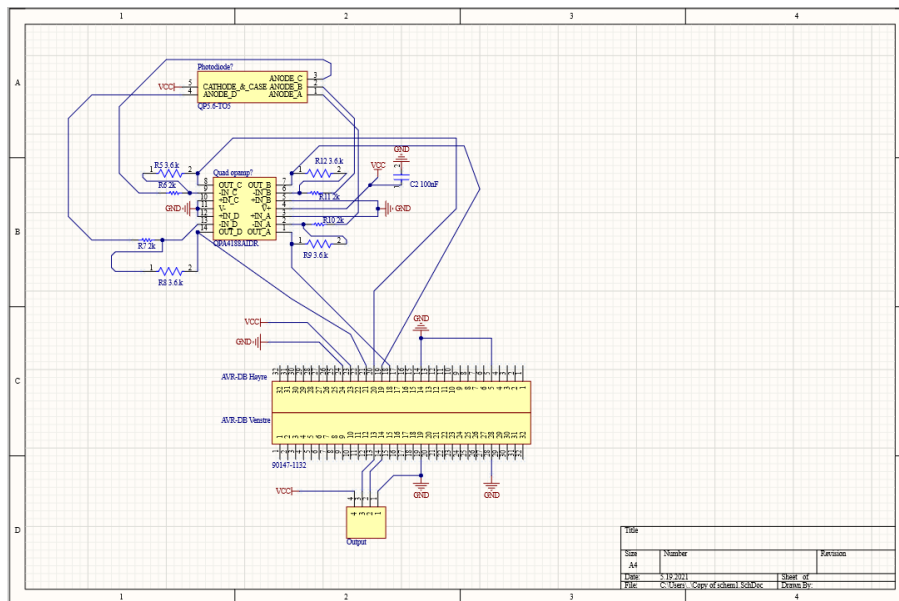
Figur B.1: Spektral respons QP5.6-TO5[24]

Spectral response (23 °C)

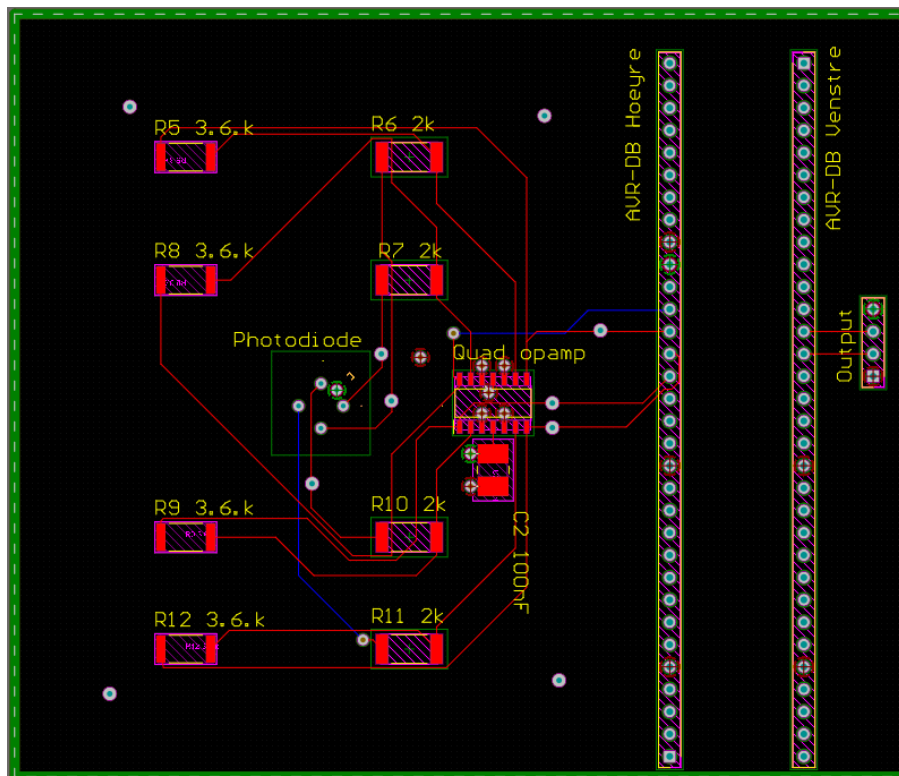


Figur B.2: Spektral respons QP5.8-6-TO5[25]

C Prototype 1

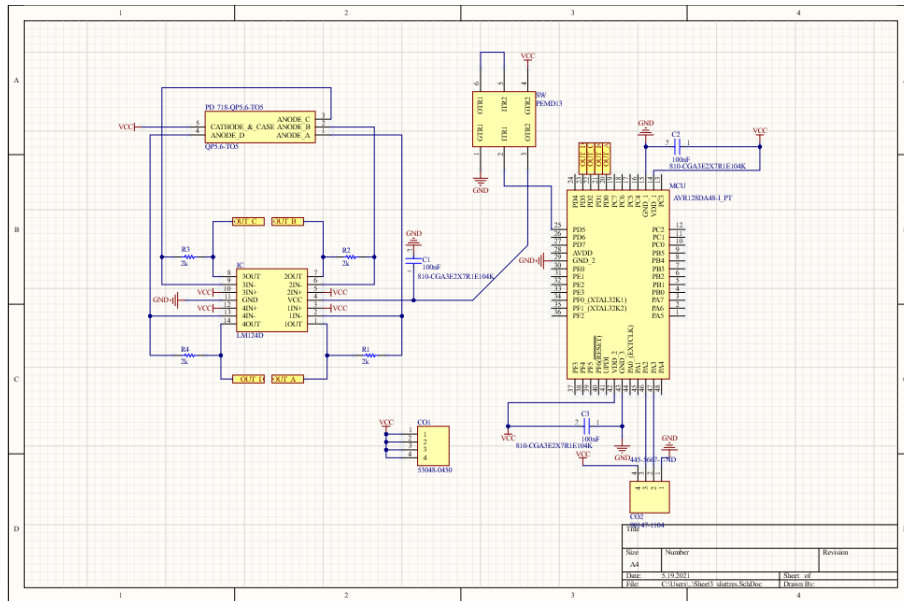


Figur C.1: Skjematikk tilhørende revisjon 1

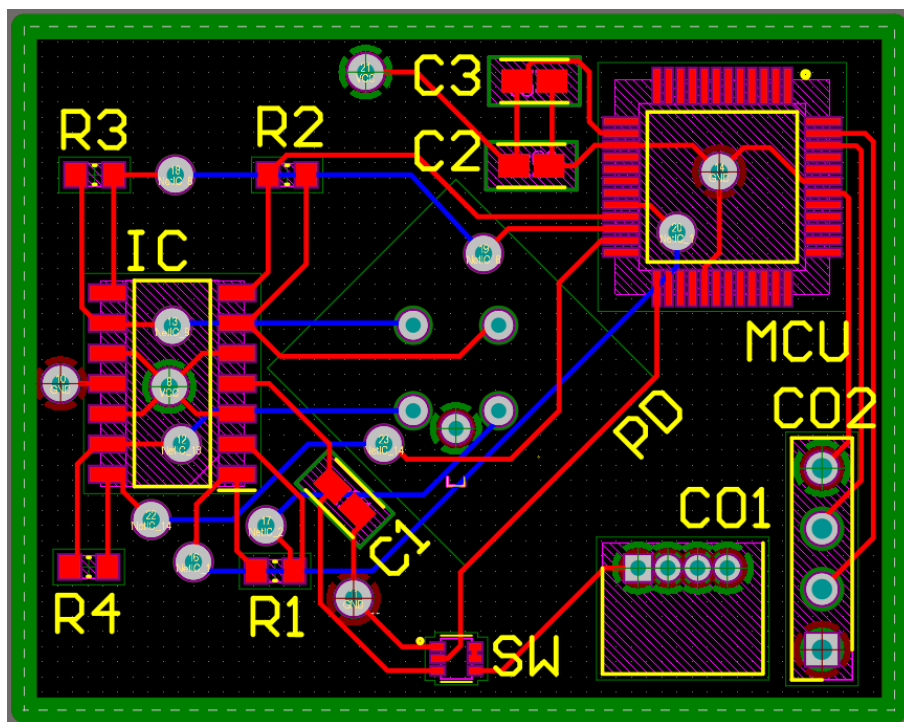


Figur C.2: PCB revisjon 1

D Prototype 2



Figur D.1: Skjematikk tilhørende revisjon 2



Figur D.2: PCB revisjon 2

E Programkoden

main.c

```
#include <avr/interrupt.h>
#include <avr/io.h>
#include <math.h>
#include <stdbool.h>
#include "include/ADC.h"
#include "include/I2C_slave.h"
#include "include/sun_sensor.h"

int main(void)
{
    //Set default slave address
    slaveAddress = 0x08;
    //Set paramaters for calculating angle
    h = 2.18e-3;
    r = 1.26e-3;
    //Initialize I2C, ADC and interrupts
    I2C_init(slaveAddress);
    ADC_init();
    sei();
    while (1)
    {
        //If command received from master
        if(checkCmd)
        {
            //Disable interrupts while measuring and calculating
            cli();
            //Execute command from master
            executeCmd(cmd);
            checkCmd = false;
            //Enable interrupts
            sei();
        }
    }
}
```

ADC.c

```
#include <avr/io.h>
#include "../include/ADC.h"

void ADC_init(void){
    //Set VREF
    VREF.ADCOREF = VREF_REFSEL_VDD_gc;
    //Select ADC resolution to be 12 bits
    ADC0.CTRLA |= ADC_RESSEL_12BIT_gc;
    //Set number of samples to be accumulated to 8
    ADC0.CTRLB |= ADC_SAMPNUM_ACC8_gc;
    //Set ADC clock to 200kHz (CLK_ADC = 4Mhz/20 = 200kHz)
    ADC0.CTRLC |= ADC_PRESC_DIV20_gc;
    //Enable ADC
    ADC0.CTRLA |= ADC_ENABLE_bm;
}

uint16_t ADC_read(diode_t diode){
    //Set ADC input
    ADC0.MUXPOS = diode;
    //Start conversion
    ADC0.COMMAND = ADC_STCONV_bm;
    //Wait until conversion finished
    loop_until_bit_is_set(ADC0.INTFLAGS,ADC_RESRDY_bp);
    //Return results
    return ADC0.RES;
}
```


I2C_slave.c

```
#include <avr/interrupt.h>
#include <avr/io.h>
#include <math.h>
#include <stdbool.h>
#include "../include/I2C_slave.h"
#include "../include/sun_sensor.h"

ISR(TWIO_TWIS_vect)
{
    //Address interrupt
    if((TWIO.SSTATUS & TWI_APIF_bm) && (TWIO.SSTATUS & TWI_AP_bm))
    {
        I2C_sendAck();
    }
    //Data interrupt
    else if(TWIO.SSTATUS & TWI_DIF_bm)
    {
        //Master read from slave
        if(TWIO.SSTATUS & TWI_DIR_bm)
        {
            I2C_sendData();
        }
        //Master write to slave
        else
        {
            cmd = TWIO.SDATA;
            checkCmd = true;
            I2C_sendNack();
        }
    }
}

void I2C_init(uint8_t slaveAddress)
{
    //Enable I2C, data and address interrupts and smart mode
    TWIO.SCTRLA = TWI_DIEN_bm | TWI_APIEN_bm | TWI_ENABLE_bm | TWI_SMEN_bm;
    //Set slave address
    TWIO.SADDR = slaveAddress << 1;
    //Initialize counters for transmitting data
    I2C_isFirstByte = true;
    I2C_byte_counter = 0;
}

void I2C_setUpData(uint8_t data[], uint8_t data_size)
{
    //Set up data to be transmitted
    I2C_data_size = data_size;
    for(uint8_t i = 0; i < data_size; i++)
    {
        I2C_data[i] = data[i];
    }
}

void I2C_sendData(void)
{
    //if first byte of transmission
    if(I2C_isFirstByte)
    {
        //Load data for master to read
        TWIO.SDATA = I2C_data[I2C_byte_counter%I2C_data_size];
        //Increment counter
        I2C_isFirstByte = false;
        I2C_byte_counter++;
        //Wait until data has been transmitted. This will trigger new interrupt
        loop_until_bit_is_set(TWIO.SSTATUS, TWI_DIF_bp);
    }
    //if ACK received from master
}
```

```

else if(!(TWIO.SSTATUS & TWI_RXACK_bm))
{
    //Load data for master to read
    TWIO.SDATA = I2C_data[I2C_byte_counter%I2C_data_size];
    //Increment counter
    I2C_byte_counter++;
    //Wait until data has been transmitted. This will trigger new interrupt
    loop_until_bit_is_set(TWIO.SSTATUS,TWI_DIF_bp);
}
//if NACK received from master
else
{
    //Complete transaction and release I2C lines
    TWIO.SCTRLB = TWI_SCMD_COMPTRANS_gc;
    //Reset counters
    I2C_isFirstByte = true;
    I2C_byte_counter = 0;
}
}

void I2C_sendAck(void)
{
    //Send ACK to master
    TWIO.SCTRLB = TWI_ACKACT_ACK_gc;
    TWIO.SCTRLB |= TWI_SCMD_RESPONSE_gc;
}

void I2C_sendNack(void)
{
    //Send NACK to master
    TWIO.SCTRLB = TWI_ACKACT_NACK_gc;
    TWIO.SCTRLB |= TWI_SCMD_RESPONSE_gc;
}

```

sun_sensor.c

```

#include <avr/io.h>
#include <math.h>
#include <stdbool.h>
#include "../include/ADC.h"
#include "../include/I2C_slave.h"
#include "../include/sun_sensor.h"

void calculateAngles(void){
    //Reading values from all diodes
    A = ADC_read(DIODE_A);
    B = ADC_read(DIODE_B);
    C = ADC_read(DIODE_C);
    D = ADC_read(DIODE_D);
    //Calculating x and y from diode values
    x = (double)(B+C-A-D)/(A+B+C+D)*r;
    y = (double)(A+B-C-D)/(A+B+C+D)*r;
    //Calculating phi
    phi = atan(sqrt(pow(x,2)+pow(y,2))/h);
    //Calculating theta and correcting angle to range from 0-2pi
    theta = atan2(y,x) + M_PI;
}

void setUpAngles(void){
    /* Converting phi and theta to degrees and loading into array
    as 16-bit values like this: angles[phiMSB, phiLSB, thetaMSB, thetaLSB] */
    angles[0] = ((uint16_t)round(phi*180/M_PI*100) >> 8) & 0xFF;
    angles[1] = ((uint16_t)round(phi*180/M_PI*100) >> 0) & 0xFF;
    angles[2] = ((uint16_t)round(theta*180/M_PI*100) >> 8) & 0xFF;
    angles[3] = ((uint16_t)round(theta*180/M_PI*100) >> 0) & 0xFF;
}

```

```

void setUpVector(void){
    /* Converting spherical coordinates into a vector R_vect[xv,yv,zv] */
    xv = sin(phi)*cos(theta);
    //Check if number is negative
    if(xv < 0){xv = xv*-1;xv_neg = true;}
    //Values over 0.5 are set to 0.49
    if(xv >= 0.5){xv = 0.49;}
    //Convert number from float to 8-bit value
    R_vect[0] = (uint8_t)round(xv*256);
    //if negative, mark 8-bit value as negative
    if(xv_neg){R_vect[0] |= 1 << 7;xv_neg = false;}

    yv = sin(phi)*sin(theta);
    if(yv < 0){yv = yv*-1;yv_neg = true;}
    if(yv >= 0.5){yv = 0.49;}
    R_vect[1] = (uint8_t)round(yv*256);
    if(yv_neg){R_vect[1] |= 1 << 7;yv_neg = false;}

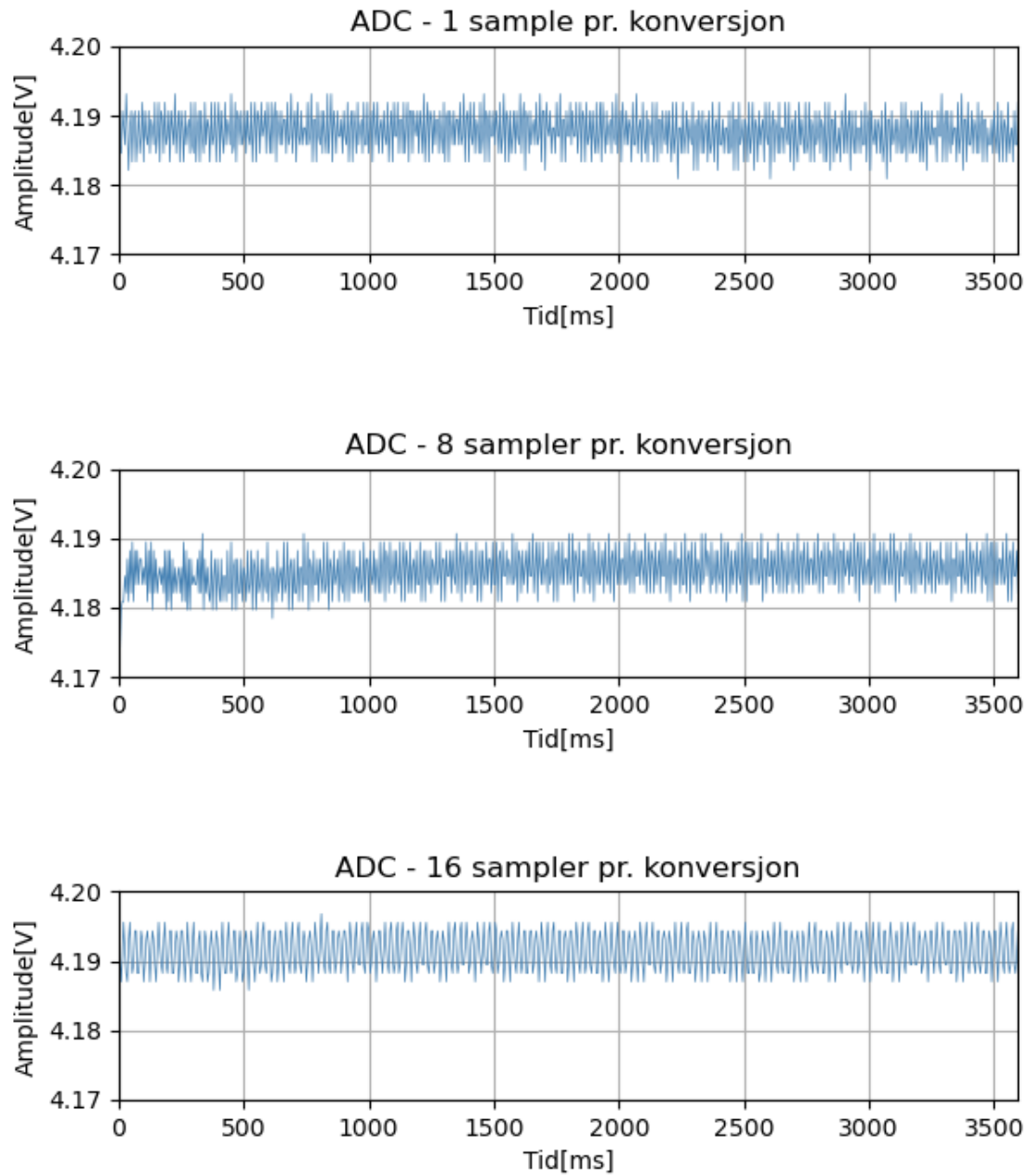
    R_vect[2] = (uint8_t)round(cos(phi)*256);
}

void setUpADCvalues(void){
    /* ADC_values[AMSB, ALSB, BMSB, BLSB, CMSB, CLSB, DMSC, DLSB] */
    ADC_values[0] = (uint16_t)(A >> 8) & 0xFF;
    ADC_values[1] = (uint16_t)(A >> 0) & 0xFF;
    ADC_values[2] = (uint16_t)(B >> 8) & 0xFF;
    ADC_values[3] = (uint16_t)(B >> 0) & 0xFF;
    ADC_values[4] = (uint16_t)(C >> 8) & 0xFF;
    ADC_values[5] = (uint16_t)(C >> 0) & 0xFF;
    ADC_values[6] = (uint16_t)(D >> 8) & 0xFF;
    ADC_values[7] = (uint16_t)(D >> 0) & 0xFF;
}

void executeCmd(uint8_t cmd){
    switch(cmd)
    {
        case getAngles: //Master requests angles phi and theta
            calculateAngles();
            setUpAngles();
            I2C_setUpData(angles,4);
            break;
        case getVector: //Master requests vector R[xMSB, xLSB, yMSB, yLSB, zMSB, zLSB]
            calculateAngles();
            setUpVector();
            I2C_setUpData(R_vect, 6);
            break;
        case getADCvalues:
            calculateAngles();
            setUpADCvalues();
            I2C_setUpData(ADC_values, 8);
            break;
    }
}

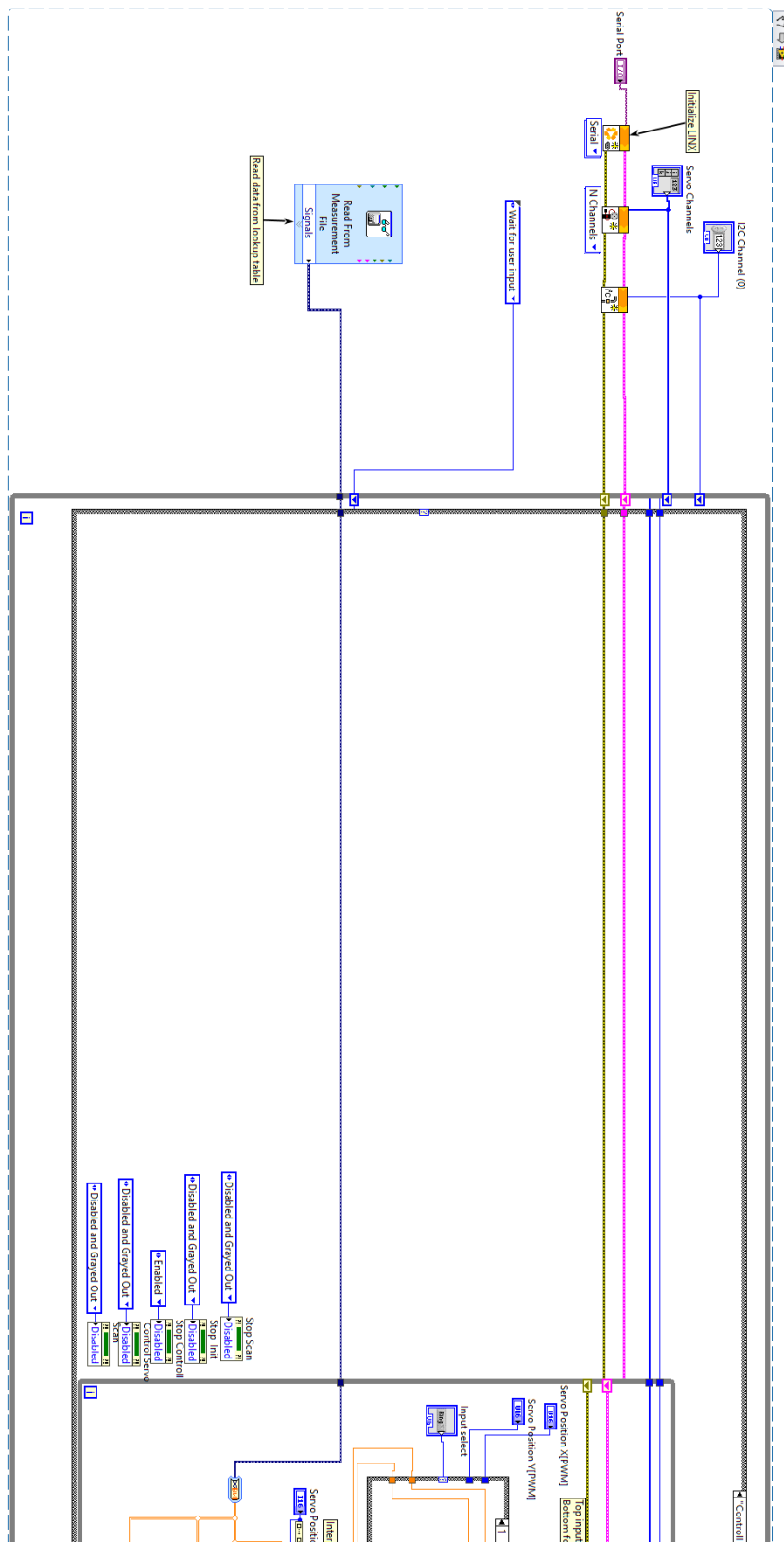
```

F Måling med ulikt antall sampler

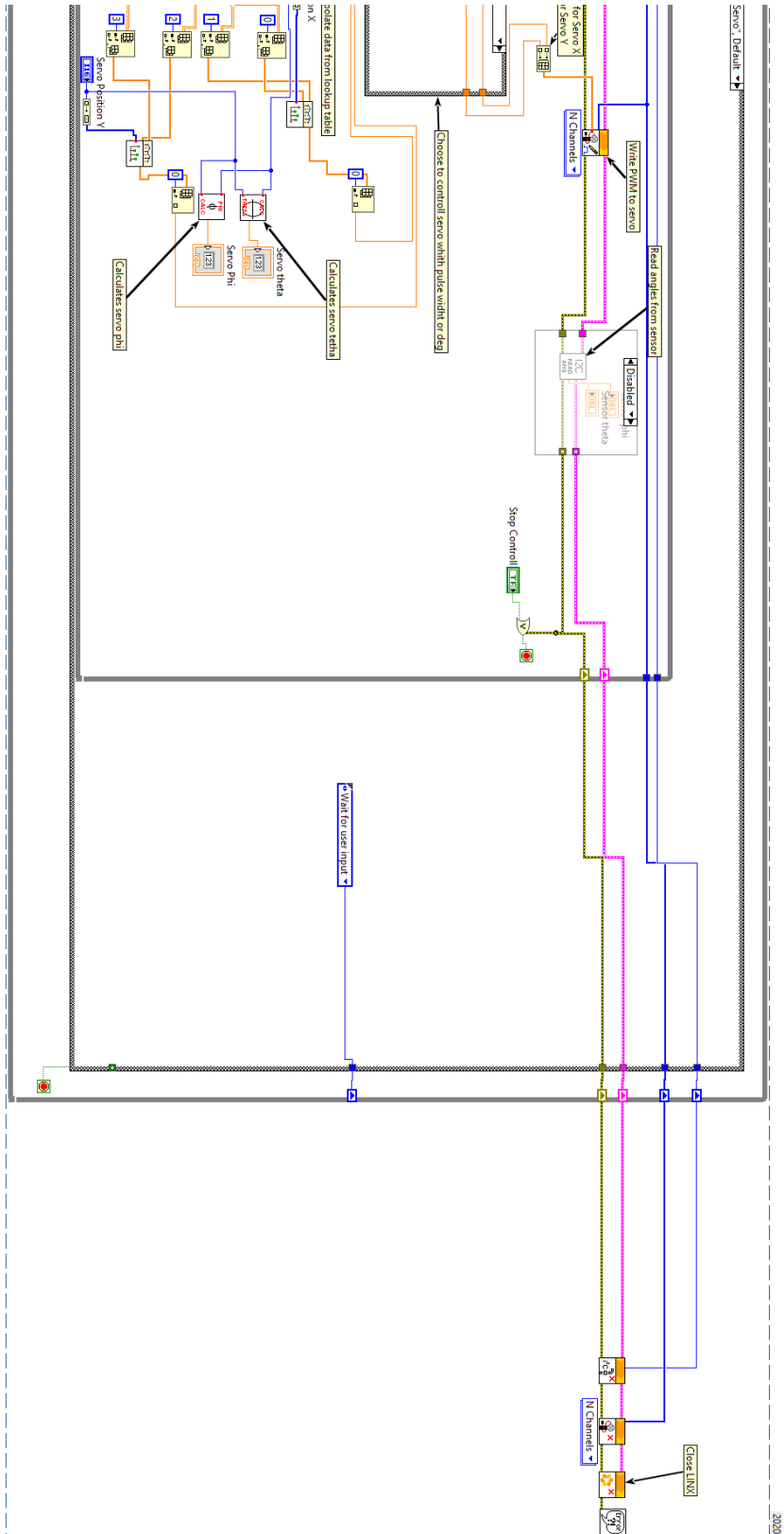


Figur F.1: Måling med ulike sampler pr. ADC konvertering

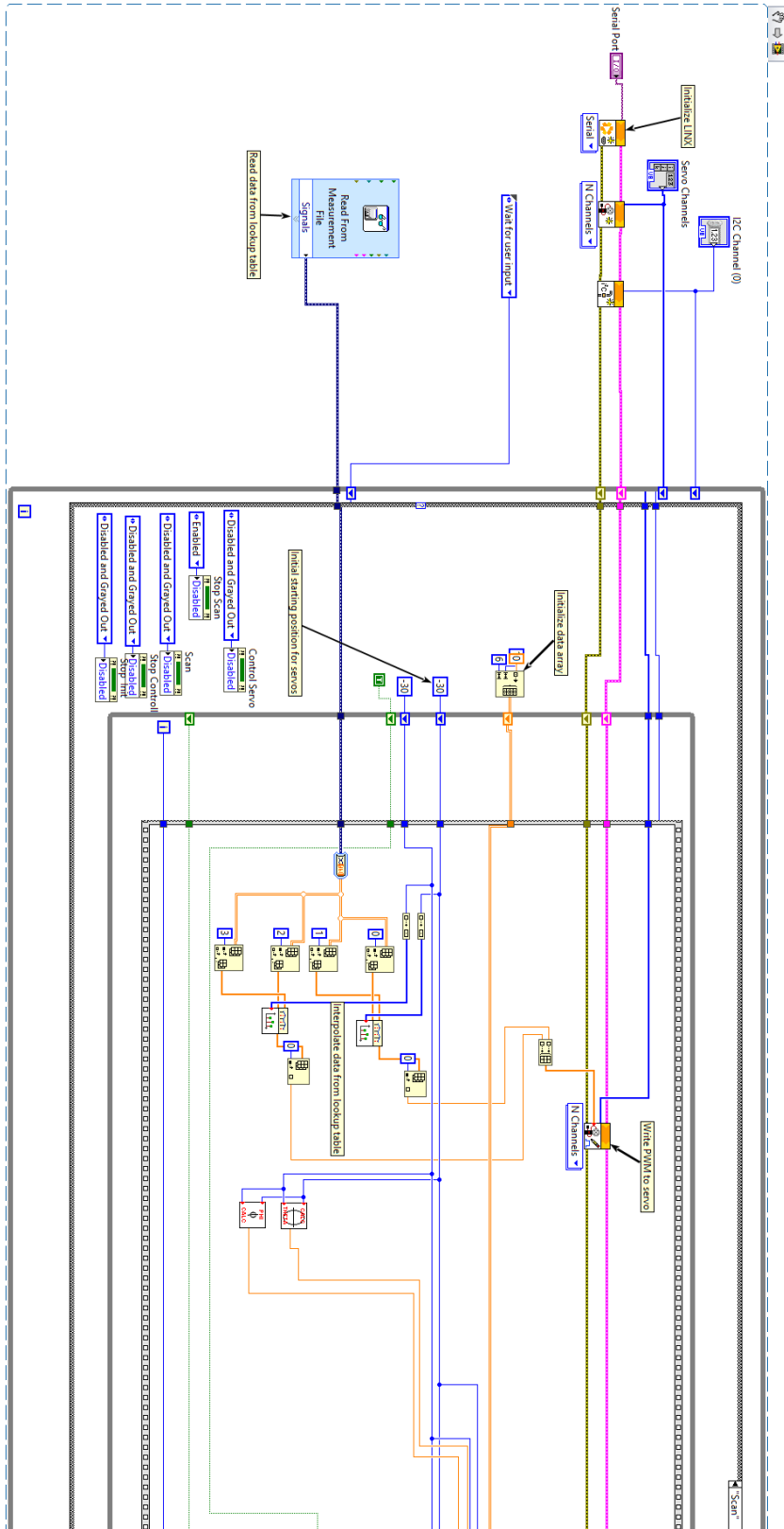
G Servo program blokkdiagram



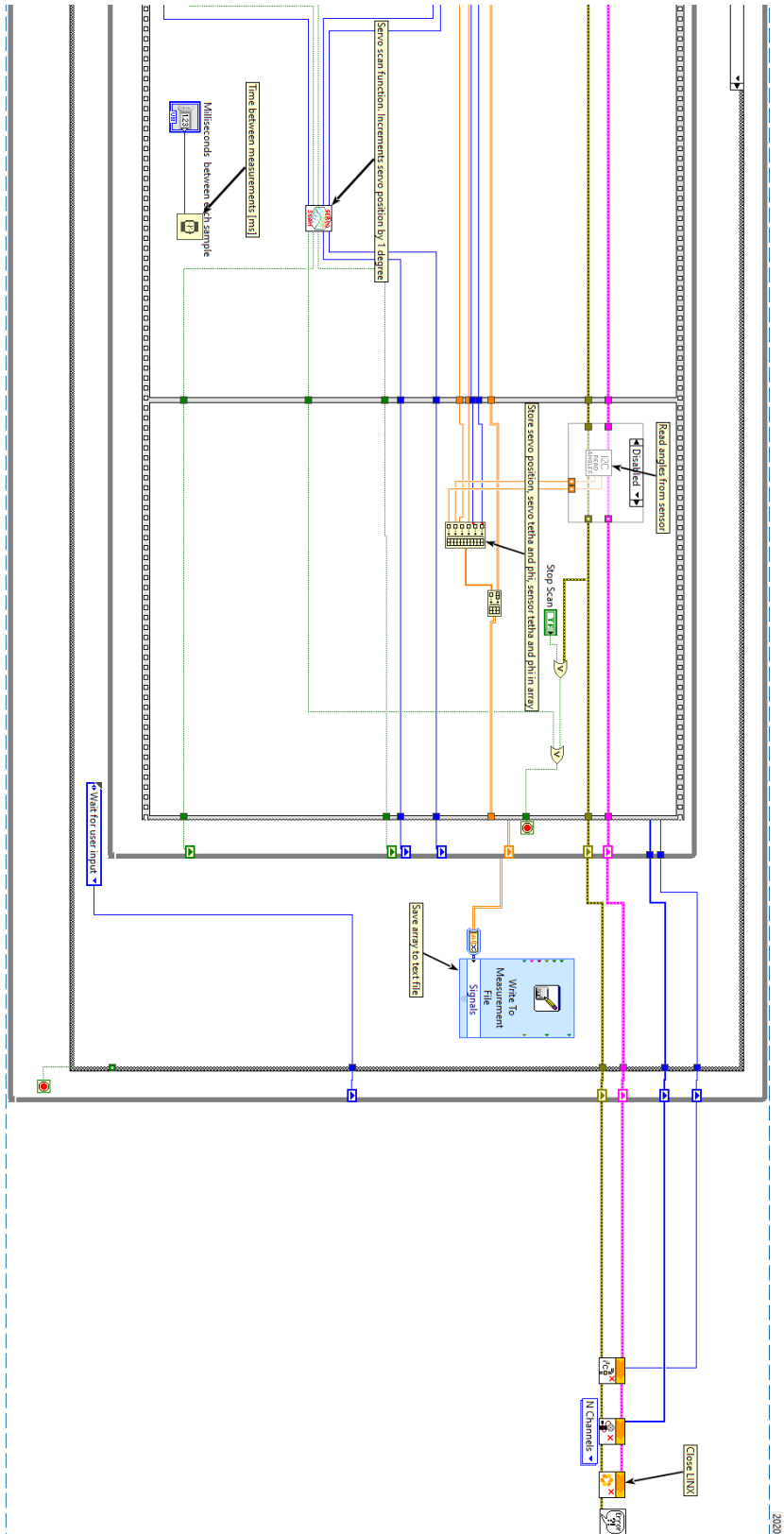
Figur G.1: Control State blokk diagram del 1



Figur G.2: Control State blokk diagram del 2



Figur G.3: Scan State blokk diagram del 1



Figur G.4: Scan State blokk diagram del 2

H Matlab simuleringer

```
1 function [AA_Venstre_Oppe] = Venstre_Oppe(phi, theta)
2
3 W = 0.5; %Apertur tykkelse = 0,5mm
4 h = 1.26; %Høyde til glassbelegg = 1,26mm
5 BB = 2; %Apertur åpning = 2mm
6
7 %phi = [1:1:30];
8 %theta = [10:10:80];
9
10 T = W/tand(90-phi); %Tapt sollys kan beregnes
11
12 Sollys = BB-T; %Beregner sollys som kommer inn i dekselet
13
14 D = (h+W)/(tand(90-phi)); %Beregner forskyving av lys fra apertur åpning
15
16 forskyv_x = D*sind(theta);
17 forskyv_y = D*cosd(theta);
18
19 x = 0; %Initierer x som variabel
20
21 if forskyv_x <= (1.2-BB/2)
22     x = BB/2 + forskyv_x;
23 elseif forskyv_x <= (BB/2)
24     x = 1.2;
25 elseif forskyv_x <= (1.2 +BB/2)
26     x = 1.2 +BB/2 - forskyv_x;
27 end
28
29 y = 0; %Initierer y som variabel
30
31 if forskyv_y <= (1.2-BB/2)
32     y = BB/2 + forskyv_y;
33 elseif forskyv_y <= (BB/2)
34     y = 1.2;
35 elseif forskyv_y <= (1.2 +BB/2)
36     y = 1.2 +BB/2 - forskyv_y;
37 end
38
39 A_Venstre_Oppe = x*y; %Regner ut arealet til venstre_oppe
40
41 AA_Venstre_Oppe = 100*A_Venstre_Oppe/1.44;
42 end
```

Figur H.1: Script for kvadrant A

```

1 function [AA_Hoyre_Oppe] = Hoyre_Oppe(phi, theta)
2
3 W = 0.5; %Apertur tykkelse = 0,5mm
4 h = 1.26; %Høyde til glassbelegg = 1,26mm
5 BB = 2; %Apertur åpning = 2mm
6
7 %phi = [1:1:30];
8 %theta = [10:10:80];
9
10 T = W/tand(90-phi); %Tapt sollys kan beregnes
11
12 Sollys = BB-T; %Beregner sollys som kommer inn i dekselet
13
14 D = (h+W)/(tand(90-phi)); %Beregner forskyving av lys fra apertur åpning
15
16 forskyv_x = D*sind(theta);
17 forskyv_y = D*cosd(theta);
18
19 x = 0;
20
21 if forskyv_x <= (BB/2)
22     x =BB/2 -forskyv_x;
23 end
24 y = 0;
25
26 if forskyv_y <= (1.2-BB/2)
27     y = BB/2 + forskyv_y;
28 elseif forskyv_y <= (BB/2)
29     y = 1.2;
30 elseif forskyv_y <= (1.2 +BB/2)
31     y = 1.2 +BB/2 - forskyv_y;
32 end
33
34 A_Hoyre_Oppe = x*y; %Regner ut arealet til venstre_oppe
35
36 AA_Hoyre_Oppe = 100*A_Hoyre_Oppe/1.44;
37 end
38

```

Figur H.2: Script for kvadrant B

```

Venstre_Oppe.m x Høyre_Oppe.m x Venstre_Nede.m x Høyre_Nede.m x Vinkel_Plot.m x Vinkel_Plo
1 function [AA_Hoyre_Nede] = Hoyre_Nede(phi, theta)
2
3 W = 0.5; %Apertur tykkelse = 0,5mm
4 h = 1.26; %Høyde til glassbelegg = 1,26mm
5 BB = 2; %Apertur åpning = 2mm
6
7 %phi = [1:1:30];
8 %theta = [10:10:80];
9
10 T = W/tand(90-phi); %Tapt sollys kan beregnes
11
12 Sollys = BB-T; %Beregner sollys som kommer inn i dekselet
13
14 D = (h+W)/(tand(90-phi)); %Beregner forskyving av lys fra apertur åpning
15
16 forskyv_x = D*sind(theta);
17 forskyv_y = D*cosd(theta);
18
19 x = 0;
20
21 if forskyv_x <= (BB/2)
22     x =BB/2 -forskyv_x;
23 end
24 y = 0;
25
26 if forskyv_y <= (BB/2)
27     y =BB/2 -forskyv_y;
28 end
29
30 A_Hoyre_Nede = x*y; %Regner ut arealet til venstre_oppe
31
32 AA_Hoyre_Nede = 100*A_Hoyre_Nede/1.44;
33
34 end

```

Figur H.3: Script for kvadrant C

```

1 function [AA_Venstre_Nede] = Venstre_Nede(phi, theta)
2
3 W = 0.5; %Apertur tykkelse = 0,5mm
4 h = 1.26; %Høyde til glassbelegg = 1,26mm
5 BB = 2; %Apertur åpning = 2mm
6
7 %phi = [1:1:30];
8 %theta = [10:10:80];
9
10 T = W/tand(90-phi); %Tapt sollys kan beregnes
11
12 Sollys = BB-T; %Beregner sollys som kommer inn i dekslet
13
14 D = (h+W)/(tand(90-phi)); %Beregner forskyving av lys fra apertur åpning
15
16 forskyv_x = D*sind(theta);
17 forskyv_y = D*cosd(theta);
18
19 x = 0;
20 if forskyv_x <= (1.2-BB/2)
21     x = BB/2 + forskyv_x;
22 elseif forskyv_x <= (BB/2)
23     x = 1.2;
24 elseif forskyv_x <= (1.2 +BB/2)
25     x = 1.2 +BB/2 - forskyv_x;
26 end
27
28 y = 0;
29
30 if forskyv_y <= (BB/2)
31     y =BB/2 -forskyv_y;
32 end
33
34 A_Venstre_Nede = x*y; %Regner ut arealet til venstre_oppe
35
36 AA_Venstre_Nede = 100*A_Venstre_Nede/1.44;
37
38 end
39

```

Figur H.4: Script for kvadrant D

```

1 - W = 0.5; %Tykkelse = 0,5mm
2 - BB = 2; %Apertur tykkelse = 2mm
3 - h = 1.26; %Høyde opp til glassbelegg er 1,26mm
4 - VP = [];
5 - VT = [];
6 - A = [];
7 - B = [];
8 - C = [];
9 - D = [];
10 - P = [];
11 - FL1 = [];
12 - FL2 = [];
13 - VOVO = [];
14 - HOHO = [];
15 - HNHN = [];
16 - VNVN = [];
17 - rb = BB/2; %Maks forskyving av sollys
18
19 - for theta = [0:1:89] %Setter opp for-løkke som varierer theta fra 0° til 90°
20
21 - RRR = []; INR = []; IND = []; BBB = [];
22 - FFVO = []; FFHO = []; FFHN = []; FFVN = [];
23 - SSS = []; TTT = []; UUU = [];
24 - RRR = []; INR = []; IND = []; BBB = []; B_BBB = []; B_TTT = [];
25 - FFVO = []; FFHO = []; FFHN = []; FFVN = []; B_ggg = []; G_TTT = [];
26
27 - for i = 1:1:30 %Setter opp for-løkke som varierer phi fra 0° til 30°
28 - FFVO(i) = Venstre_Oppe(i, theta);
29 - FFHO(i) = Hoyre_Oppe(i, theta);
30 - FFHN(i) = Hoyre_Nede(i, theta);
31 - FFVN(i) = Venstre_Nede(i, theta);
32 - BBB(i) = i;
33
34 - A = Venstre_Oppe(i, theta); %Diode A
35 - B = Hoyre_Oppe(i, theta); %Diode B
36 - C = Hoyre_Nede(i, theta); %Diode C
37 - D = Venstre_Nede(i, theta); %Diode D
38 - P(i) = i;
39 - T = W/tand(90-i); %Regner ut tapt sollys
40 - B;
41 - Sollys = BB-T; %Bredde til sollys
42 - pos_x = ((A+D)-(B+C))/(A+B+C+D); %Regner ut x-posisjon til sollys
43 - pos_y = ((A+B)-(C+D))/(A+B+C+D); %Regner ut y-posisjon til sollys
44
45
46 - Vink_Theta = atand(pos_x/pos_y); %Formel for å regne ut vinkel theta
47
48 - B_BBB(i) = Vink_Theta
49
50 - vartmp = Vink_Theta;
51 - if vartmp <= 45
52 -     Vink_Theta = vartmp*cosd(vartmp);
53 - elseif vartmp > 45
54 -     Vink_Theta = vartmp*sind(vartmp);
55 - end
56

```

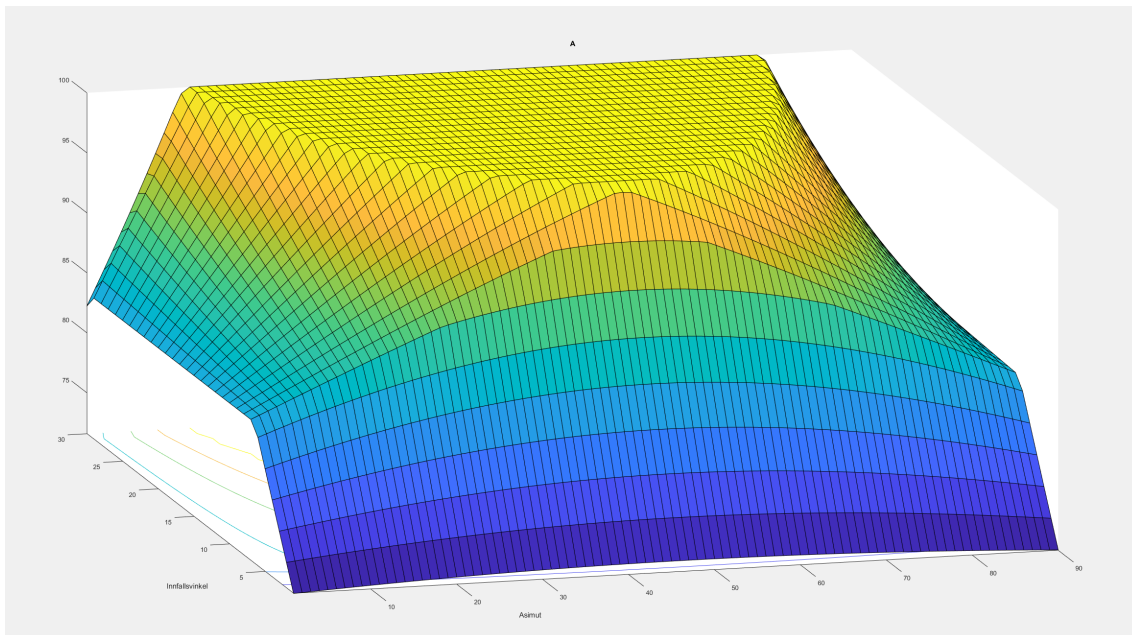
Figur H.5: Script for målinger del.1

```

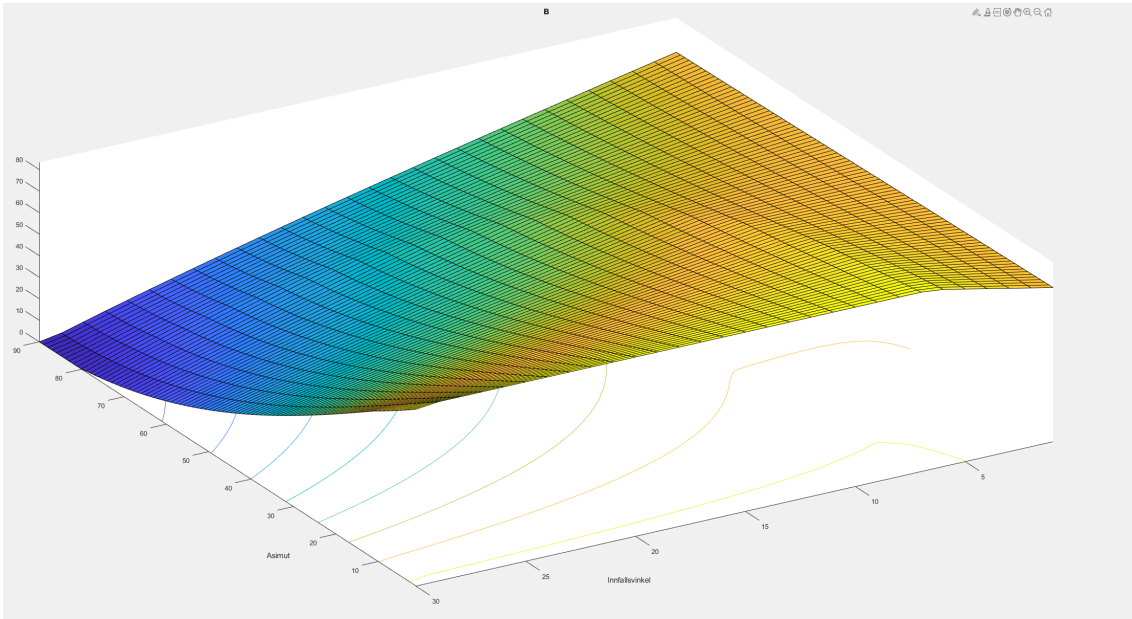
55 - end
56 -
57 - ggg = atand(pos_x/pos_y); %korrekt beta
58 -
59 - if Vink_Theta <= 45 %Tilpasser kvadratisk apertur ved å gjøre forskyving lenger
60 -     rb = (BB/2)/cosd(Vink_Theta);
61 - elseif Vink_Theta > 45
62 -     rb = (BB/2)/sind(Vink_Theta);
63 - end
64 -
65 - Vink_Phi = atand(rb*(sqrt(pos_x^2+pos_y^2))/(h+W)); %Formel for å regne ut vinkel phi
66 -
67 - B_ggg(i) = ggg;
68 - B_TTT(i) = Vink_Phi;
69 -
70 -
71 -
72 - i = i+1
73 - end
74 -
75 - VOVO = [VOVO,FFVO'];
76 - HOHO = [HOHO,FFHO'];
77 - VNVN = [VNVN,FFVN'];
78 - HNHN = [HNHN,FFHN'];
79 - FL1 = [FL1,B_TTT'];
80 - FL2 = [FL2,B_BBB'];
81 -
82 - end
83 -
84 - figure(156); surfc(FL1); xlabel('Asimut'); ylabel('Innfallsvinkel'); title('Innfallsvinkel'); %Figur av phi
85 - figure(157); surfc(FL2); grid; xlabel('Asimut'); ylabel('Innfallsvinkel'); title('Asimut'); %Figur av theta
86 - figure(158); surfc(VOVO); grid; xlabel('Asimut'); ylabel('Innfallsvinkel'); title('A'); %Figur av diode A
87 - figure(159); surfc(HOHO); grid; xlabel('Asimut'); ylabel('Innfallsvinkel'); title('B'); %Figur av diode B
88 - figure(160); surfc(VNVN); grid; xlabel('Asimut'); ylabel('Innfallsvinkel'); title('D'); %Figur av diode D
89 - figure(161); surfc(HNHN); grid; xlabel('Asimut'); ylabel('Innfallsvinkel'); title('C'); %Figur av diode C

```

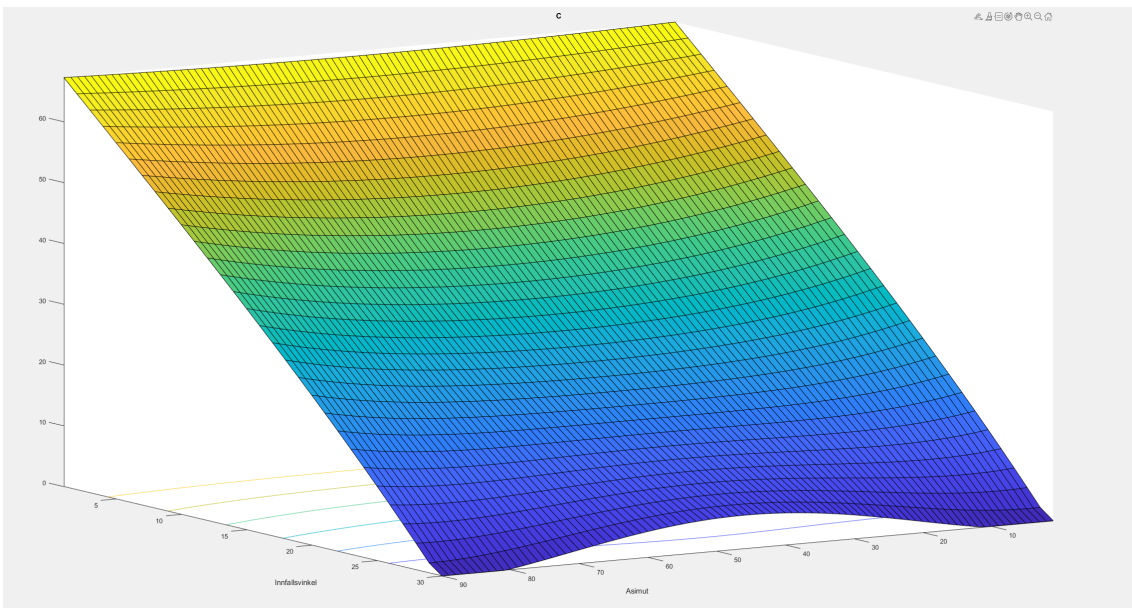
Figur H.6: Script for målinger del.2



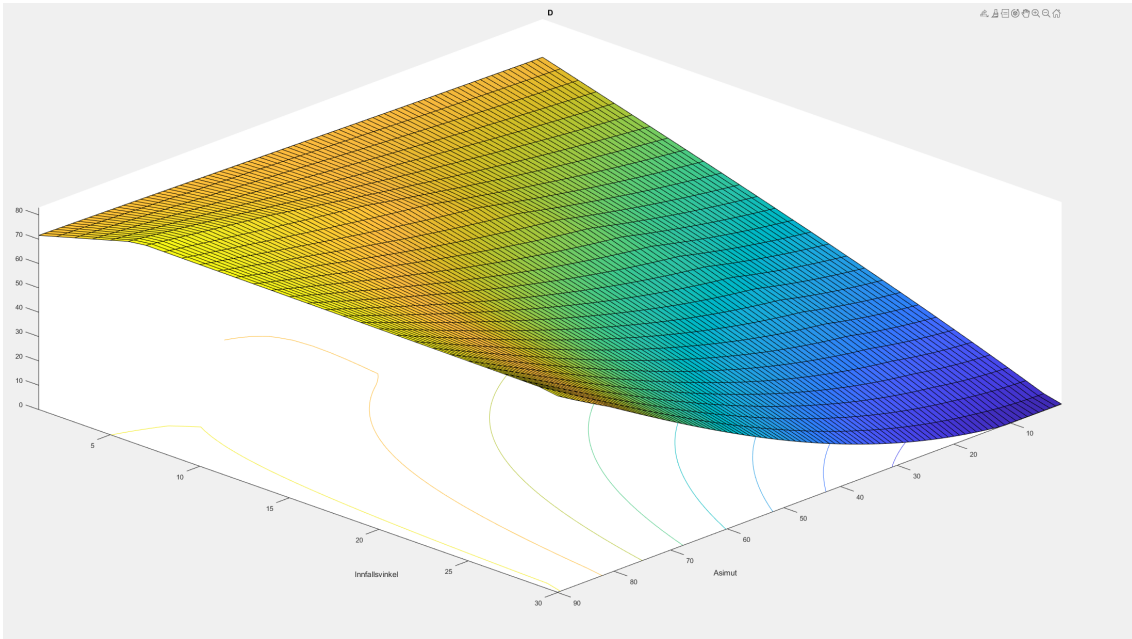
Figur H.7: Måling for kvadrant A



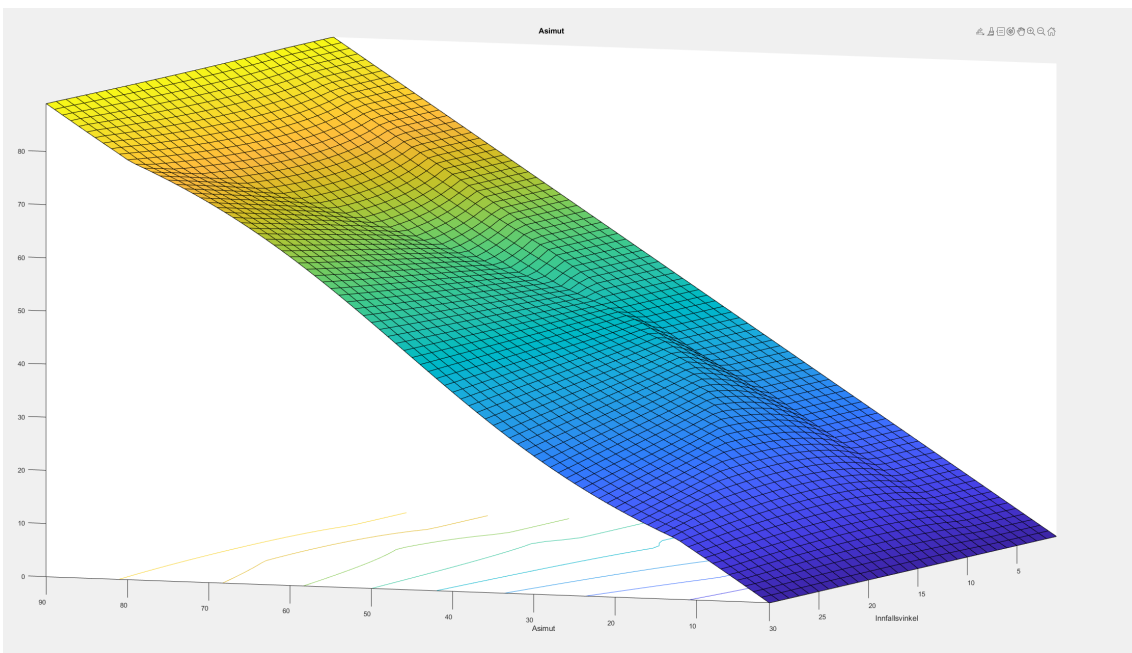
Figur H.8: Måling for kvadrant B



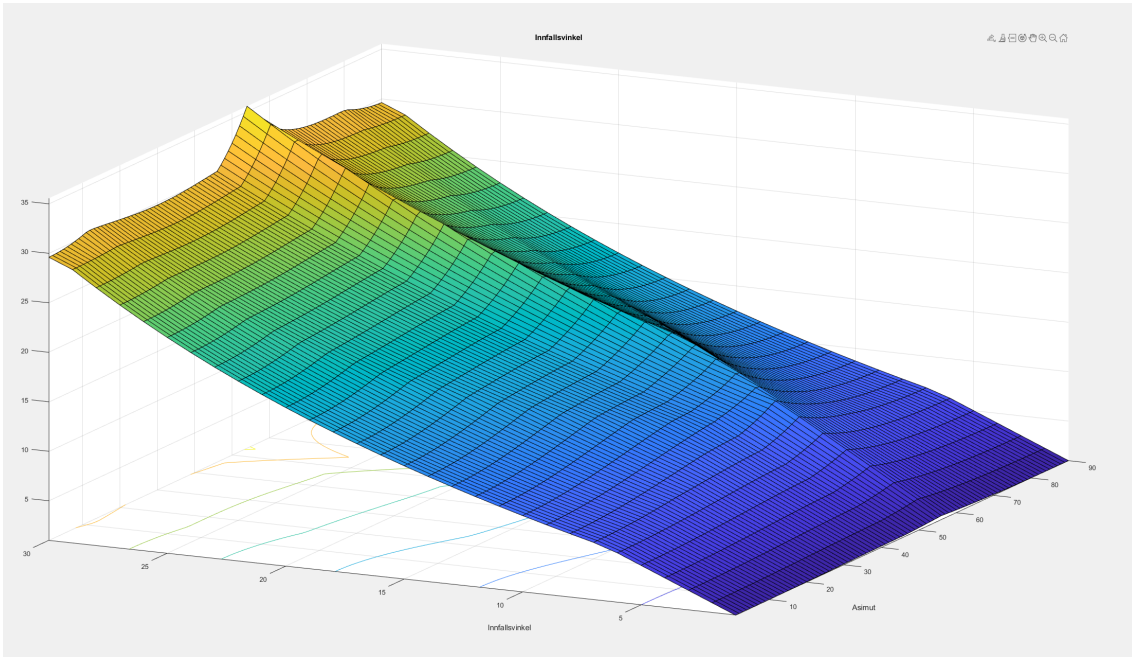
Figur H.9: Måling for kvadrant C



Figur H.10: Måling for kvadrant D



Figur H.11: Måling for phi

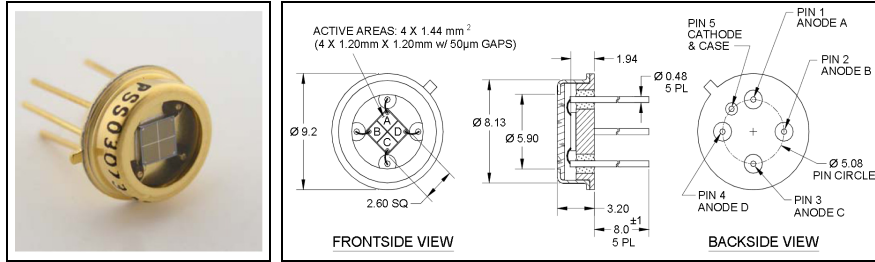


Figur H.12: Måling for theta

I Datablad for QP5.8-6



First Sensor QP Data Sheet
Part Description QP5.8-6 TO
Order # 501254
Version 24-11-11



Features

- 5.8 mm² Quadrant PIN detector
- High sensitivity
- Small gap
- Low dark current

Description

Low dark current circular active area quadrant PIN photodiode with 4 x 1.44 mm² active area. Metal can type hermetic TO5 package with clear glass window.

Application

- Laser beam position sensor
- Autocollimators
- Optical tweezers
- Ellipsometers

RoHS

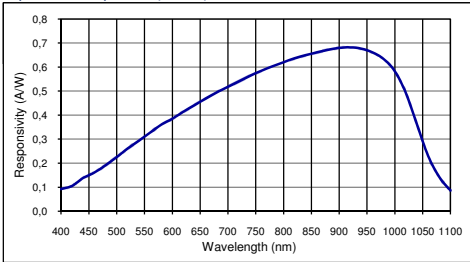
2002/95/EC



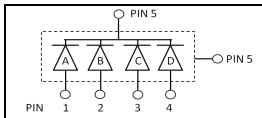
Absolute maximum ratings

Symbol	Parameter	Min	Max	Unit
T _{STG}	Storage temp	-55	125	°C
T _{OP}	Operating temp	-40	100	°C
V _{max}	Max reverse voltage		50	V
I _{PEAK}	Peak DC current		10	mA

Spectral response (23 °C)



Schematic



Electro-optical characteristics @ 23 °C

Symbol	Characteristic	Test Condition	Min	Typ	Max	Unit
	Number of elements		4 quadrants			
	Active area		2400 x 2400			µm
	Active area	per element	1.44			mm ²
	Gap	between elements	50			µm
I _D	Dark current	V _R = 10 V; per element		0.4		nA
C	Capacitance	V _R = 0 V; per element		20		pF
		V _R = 10 V; per element		3.5	5.5	pF
	Responsivity	λ = 632 nm		0.4		A/W
		λ = 900 nm		0.64		A/W
	Uniformity of responsivity	V _R = 10 V; λ = 880 nm; R _L = 50 Ω		±1	±2	%
t _r	Rise time	V _R = 0 V; λ = 850 nm; R _L = 50 Ω		2000		ns
		V _R = 10 V; λ = 850 nm; R _L = 50 Ω		20		ns
	Shunt Resistance	V _R = 5 mV; per element		250		MΩ
	N.E.P.	V _R = 5 V; λ = 900 nm; per element		1.8 E-14		W/√Hz
V _{BR}	Breakdown voltage	I _R = 2 µA	20			V

European, International Sales:



First Sensor AG
Peter Behrens-Strasse 15
12459 Berlin
Germany
Phone: +49-30-6399-2399
Fax: +49-30-6399-23752
E-Mail: sales.opto@first-sensor.com

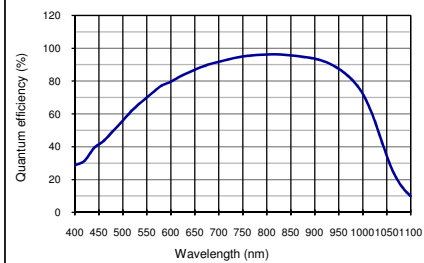
USA:



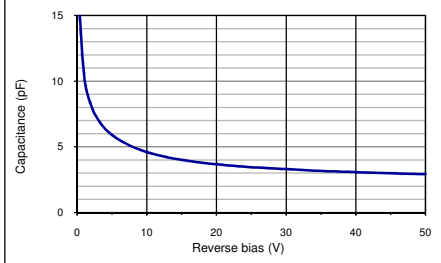
Pacific Silicon Sensor, Inc.
5700 Corsa Avenue #105
Westlake Village
CA 91362 USA
Phone: +1-818-706-3400
Fax: +1-818-889-7053
E-Mail: sales@pacific-sensor.com

Figur I.1: Datablad for QP5.8-6 side 1[25]

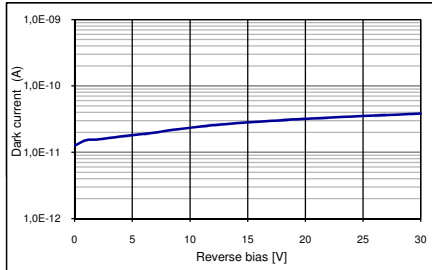
Quantum efficiency (23 °C)



Capacitance as fct of reverse bias (23 °C; per element)



Dark current as fct of bias (23 °C; per element)



Package dimension:

Small quantities: Foam pad, boxed (12 cm x 16.5 cm)

Handling precautions:

- Soldering temperature max. 260 °C for 10 s. The device must be protected against solder flux vapour.
- Minimum pin length is 2 mm.
- For ESD protection standard precautionary measures are sufficient
- For further questions please refer to document "Instructions for handling and processing"

Disclaimer: Due to our strive for continuous improvement, specifications are subject to change within our PCN policy according to JESD46C.

European, International Sales:



First Sensor AG
 Peter-Behrens-Strasse 15
 12459 Berlin
 Germany
 Phone: +49-30-6399-2399
 Fax: +49-30-6399-23752
 E-Mail: sales.opto@first-sensor.com

USA:



Pacific Silicon Sensor, Inc.
 5700 Corsa Avenue #105
 Westlake Village
 CA 91362 USA
 Phone: +1-818-706-3400
 Fax: +1-818-889-7053
 E-Mail: sales@pacific-sensor.com

Figur I.2: Datablad for QP5.8-6 side 2[25]

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[First Sensor:](#)

[QP5.8-6-T05](#)

Figur I.3: Datablad for QP5.8-6 side 3[25]

J Definisjoner

- **ADC**: Analog to Digital converter
- **Apertur**: En smal åpning som bestemmer hvor mye lys som kan komme inn
- **Asimut**: Vinkelen langs horisonten i horisontalkoordinater
- **Bit**: Minste dataenhet, kan være 1 eller 0
- **Byte**: En mengde bit's, som regel 8
- **Dark current**: Den lille strømmen som strømmer gjennom fotodioden uten at fotoner går inn i enheten.
- **DOF**: Degrees of freedom; Frihetsgrader
- **EMI**: Electromagnetic interference; Elektromagnetisk forstyrrelse
- **FOV**: Field of view, den observerbare verden som sensoren kan ta inn.
- **Hysteres**: Forskjellen i utgangsvariabel når inngangsvariabel øker vs når inngangsvariabel synker.
- **I²C**: Inter-Integrated Circuit
- **(ADC) konversjon**: En ADC operasjon, fra start til slutt. Kan bestå av flere sampler
- **OPAMP**: Operasjonsforsterker
- **PCB**: Printed circuit board
- **Peak wavelength**: De spesifikke bølgelengdeområdene fotodiodene er mest sensitive for
- **PWM**: Pulse Width Modulation; Pulsbreddemodulasjon
- **(ADC) sample**: En spesifikk måling fra ADC'en
- **SOIC**: Small outline integrated circuit.
- **State-machine**: En type programmerings teknikk som kan endre 'state' avhengig av input fra brukeren, der hver 'state' har en gitt funksjon

Solsensor design og testing til Orbit NTNUS CubeSat

Av August Johansen Berg, Martin Andre Førreisdahl, Vegard André Hansen, Andreas Knutli og Bjørg Solem

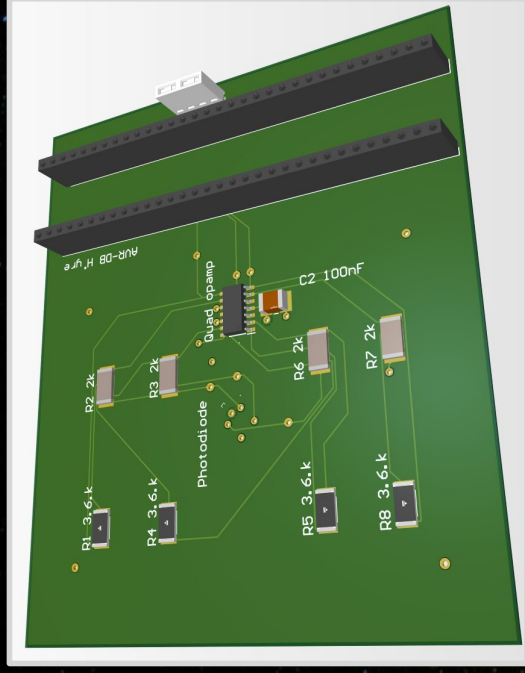
NTNU, Trondheim. 20. Mai 2021

Bakgrunn

På sitt nåværende prosjekt, SelfieSat, benytter Orbit NTNU seg av kommersielle solsensorer, som brukes for å detektere solens retning. Disse er ofte kostbare og kan ha høyt effektforbruk. I den forbindelse kom Orbit opp med ideen om å designe og produsere egne solsensorer, for å kunne redusere kostnadene på sikt.

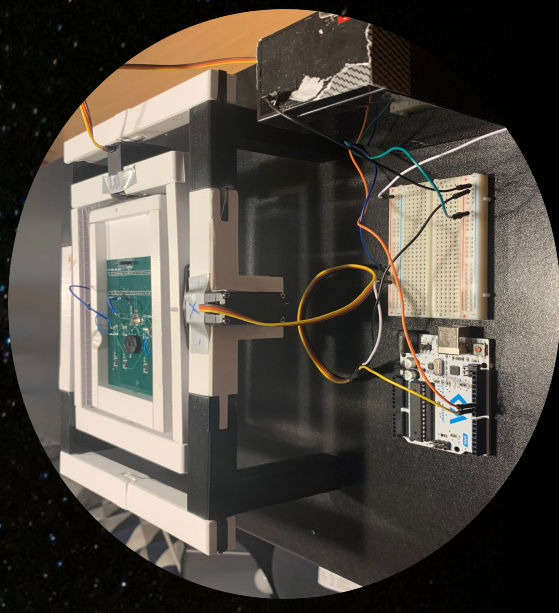
Metode

Bachelor-gruppen har gjort mye research om solsensorer ved hjelp av relevant faglitteratur, og har således designet og testet en prototype som kommer til å være grunnmuren som Orbits framtidige solsensorer vil være bygget på. Testingen blir gjennomført i en kalibreringsrigg som er av gruppens eget design.



Konklusjon

Avslutningsvis kunne det konkluderes med at det var fullt gjennomførbart å produsere en slik solsensor med de ressursene gruppen hadde tilgjengelig.



Resultater

Gruppen designet og lagde en solsensor, som videre ble testet i en kalibreringsrigg. De målte verdiene samsvarte til dels med de simulerte verdiene gruppen genererte.



NTNU

Kunnskap for en bedre verden

