

Jørgen Johannessen  
Martin Engen Holmstad  
Vetle Foslien

# Utnyttelse av IoT-Teknologi for overvåking og rapportering av snødybde i veinettet

Bacheloroppgave i BIELEKTRO  
Veileder: Halgeir Leiknes  
Mai 2023



Jørgen Johannessen  
Martin Engen Holmstad  
Vetle Foslien

# Utnyttelse av IoT-Teknologi for overvåking og rapportering av snødybde i veinettet

Bacheloroppgave i BIELEKTRO  
Veileder: Halgeir Leiknes  
Mai 2023

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



Kunnskap for en bedre verden



# Sammendrag

Oppgavens tittel:	Dato:	22.05.23	
Utnyttelse av IoT-teknologi for overvåking og rapportering av snødybde i veinettet	Antall sider:	48	
	Bacheloroppgave		X
Navn: Jørgen Johannessen <a href="mailto:joergen.johannessen@ntnu.no">joergen.johannessen@ntnu.no</a> Martin Holmstad <a href="mailto:marehol@ntnu.no">marehol@ntnu.no</a> Vetle Foslien <a href="mailto:vetlefo@ntnu.no">vetlefo@ntnu.no</a>			
Veileder: <a href="mailto:halgeir.leiknes@ntnu.no">halgeir.leiknes@ntnu.no</a>			
Oppdragsgiver: Gjøvik kommune: May Kristin Bøe (kontaktperson): <a href="mailto:may-kristin.boe@gjovik.kommune.no">may-kristin.boe@gjovik.kommune.no</a> Pål Godard (Kontaktperson)			
Sammendrag: Vedlikehold av vei, sommer som vinter, er kritisk for at vegnettet skal kunne tas i bruk av aktører innenfor transport, nødutrykning, sivile osv. For at kommunene skal kunne sette riktig mengde bemanning utover hele veinettet, kreves det at de finner en anvendelig måte de kan ha kontinuerlig oversikt over status på veinettet. Prosjektet vi har inngått med Gjøvik kommune bygger på behovet for en digitalisert måte å måle hvor mye snøfall som ligger på veiene. Produktet vi har designet samler informasjon om status på veinettet og sender denne informasjonen over et medium til kommunen som da kan ta viktige avgjørelser om hvor det er mest kritisk å gjøre vedlikehold. Produktet bruker en ultralyd-basert sensor for å måle snødybde, bruker en temperatursensor for å kunne avgjøre om det kan foreligge is på veien, og en vindsensor for å avgjøre om eventuell snø på veien kommer fra snødrift i veikanten. Produktet bruker en NB-IoT modul som sender informasjon over 4G og 5G nettet, til en server med et brukervennlig grensesnitt. Da mobilnettet er i konstant utvikling, vil denne innretningen kunne tas i bruk over utstrakte områder nasjonalt.			

Stikkord: Sensorsystemer, IoT, Veivedlikehold, brøyting, Digital overføring

# Abstract

Title: Utilization of IoT technology for monitoring and reporting snow depth in the road network	Date:	22.05.23
	Number of pages	48
	Bachelor's thesis	X
Names: Jørgen Johannessen <a href="mailto:joergen.johannessen@ntnu.no">joergen.johannessen@ntnu.no</a> Martin Holmstad <a href="mailto:marehol@ntnu.no">marehol@ntnu.no</a> Vetle Foslien <a href="mailto:vetlefo@ntnu.no">vetlefo@ntnu.no</a>		
Supervisor: <a href="mailto:halgeir.leiknes@ntnu.no">halgeir.leiknes@ntnu.no</a>		
Employer: Gjøvik kommune: May Kristin Bøe (Contact person): <a href="mailto:may-kristin.boe@gjovik.kommune.no">may-kristin.boe@gjovik.kommune.no</a> Pål Godard (Contact person)		
Abstract: Road maintenance, summer as winter, is critical for the use of the road network by companies within transport, emergency services, as well as civilians. In order for the municipalities to be able to put the right amount of staffing over the entire road network, it is required that they find an applicable way to have a continuous overview of the status of the road network. The project we have engaged with the department for maintenance in Gjøvik Kommune is based on the need for a digitized way to measure how much snowfall is on the roads. The product we have designed collects information about the status of different roads, and sends this information over a medium to the interested parties, where they can then make important decisions about where it is most critical to carry out maintenance. The product uses an ultrasound-based sensor to measure snow depth, a temperature sensor to determine whether there may be ice on the road, and a wind sensor to determine whether any snow on the road comes from snow drifts on the roadside. The product uses an NB-IoT module that sends information over the 4G and 5G network to a server with a user-friendly interface. As the mobile network is constantly developing, this device will be applicable over large areas nationally.		

Keywords:	
Sensor Systems	IoT
Road maintenance	Plowing
Digital Transmission	

## Forord

Ved levering av denne rapporten avslutter gruppen vår, bestående av 3 medstudenter, nesten et års arbeid med prosjekt og en 3-årig bachelorgrad innenfor elektronikk. Det har vært 3 spennende og givende år, med oppturer og nedturer, men som alt i alt har gitt et enormt utbytte for oss. Produktet vi har designet er en sammensetning av alt lærestoff vi har tilegnet oss i løpet av årene på NTNU. Emnet som produktet omfatter er av særegen interesse for oss, og er et godt eksempel på mulige felt og oppgaver vi kunne tenke oss å jobbe med etter endt bachelor. Vi har alle interesse for spesielt IoT, sensorsystemer og digital overføring av informasjon, noe som gjorde at denne oppgaven appellerte veldig til oss.

Innledningsvis vil vi rette en stor takk til Halgeir Leiknes, vår veileder i denne oppgaven. For å ha tillit og tro på oss som gruppe, og for å utfordre oss til å angripe oppgaven med forskjellige vinklinger. På veiledningsmøtene våre fikk vi satt lys på utfordringer og mulige løsninger med nye øyne og innså flere ganger alternativer vi kanskje ikke ville oppdaget på egen hånd. Din kunnskap på feltet har gitt oss god innsikt og gode bidrag ved gjennomføringen av denne oppgaven. Oppfølgingen og innspill du har kommet med har vært høyt satt pris på i gruppen.

Vi vil også gi en takk til Gjøvik kommune, avdeling veg for muligheten og tilliten til å gjennomføre denne oppgaven. Vi takker for en god dialog og et godt samarbeid, og for de veiledningene og innspillene dere hadde i hvilken retning dere ønsket prosjektet skulle strekke seg.

Et prosjekt av en viss størrelse sier seg umulig å gjennomføre uten bidrag på tvers av fagfelt. Derfor vil vi videre takke dere som har bidratt oss på veien, med både større og mindre bidrag. En takk til Steffan Kjenslie som viste interesse for prosjektets innkapsling og bidro i stor grad ved å tilegne sin kompetanse innenfor 3D-printing til gruppen og står blant annet bak design og printing av innkapslingen til produktet, selvfølgelig med innspill fra gruppen. En takk også til Adel Holm Gundersen som, med sitt fagbrev innenfor webutvikling, bidro med å skrive koden til en nettside for fremvisning av informasjon, etter ønsker om design fra oss. Vi vil også takke Elektroimportøren Gjøvik, for bidrag i form av enkelte komponenter som vi trengte underveis i prosjektperioden.

## Innholdsfortegnelse:

Sammendrag.....	1
Abstract.....	2
Forord.....	3
Innholdsfortegnelse:.....	4
Figurliste:.....	6
1. Innledning.....	1
1.1. Problemstilling.....	1
1.2. Begrunnelse.....	1
1.3. Avgrensing.....	1
2. Bakgrunn.....	2
2.1. Bakgrunn for oppgaven.....	2
2.2. Bruksområde.....	2
2.3. Markedsanalyse.....	2
2.4. Utvikling videre.....	4
2.5. Risikovurdering.....	5
3. Teori.....	6
3.1. MB7374-100.....	6
3.2. ESP32: Trådløs Kommunikasjon.....	7
3.3. NB-IoT.....	7
3.4. PT1000 Temperaturføler.....	8
3.5. Cotech sensor for Vindhastighet.....	9
3.6. HY-DI-2S2P-S1.....	10
3.7. Ezviz Solar panel.....	10
3.8. LM7805.....	11
3.9. LM2621.....	11
3.10. LM2904DT.....	11
3.11. ICM-20948.....	12
3.12. Wheatstone målebro.....	12
4. Metode.....	13
4.1. Planlegging.....	13
4.2. Fremdriftsplan.....	15
4.3. Valg av type sensor.....	16
4.4. Gjennomføring og bestilling av komponenter.....	17
4.5. Oppkobling av løsning.....	17
4.6. Kretskort.....	18
4.7. Kalibrering av temperatursensorer.....	20
4.8. Innkapsling.....	21
4.9. Grensesnitt.....	22
4.10. Kode.....	23
5. Resultater.....	32
5.1. Kostnader og Budsjett for løsningen.....	32
5.2. Funksjonalitet og Effektivitet.....	32
6. Diskusjon.....	34



<b>6.1. Problemer underveis.....</b>	<b>34</b>
<b>6.2. Krav og ønsker fra arbeidsgiver.....</b>	<b>35</b>
<b>6.3. Forbedringer.....</b>	<b>36</b>
<b>7. Konklusjon.....</b>	<b>37</b>
<b>8. Litteraturliste.....</b>	<b>38</b>

## Figurliste:

[1]: Vurdering av forskjellige risikofaktorer i prosjektet og alvorlighetsgrad av disse .....	5
[2]: Konseptet ultralydsensoren er basert på. Tar inspirasjon fra biomimikk, her fra blant annet flaggermus sin ekkolokalisering. ....	6
[3]: MB7374-100, ultralydsensoren brukt i prosjektet. Her med kjegleformet innkapsling som indikerer måleområdet.....	6
[4]: ESP32-modul, mikrokontrolleren som styrer den trådløse kommunikasjonen i prosjektet.....	7
[5]: NB-IoT modul brukt for å kommunisere mellom blant annet mange sensorer. Bruker 4G og 5G nettet.....	8
[6]: PT1000 Temperaturfølerelement. Metallbasert med høy nøyaktighet og robusthet.....	8
[7]: Vindsensor som måler styrke og retning på vind i 360 grader.....	9
[8]: Vindsensoren oppmontert for kalibrering og testing av funksjonalitet.....	9
[9]: Vindsensoren åpnet og modifisert for å kunne direkte tilkobles til produktet.....	9
[10]: HY-DI-2S2P oppladbar batteripakke.....	10
[11]: Enkeltkrystallinsk silisium-solcellepanel fra EzViz, her montert på en loddrett vegg.....	10
[12]: Benene på en LM7805.....	11
[13]: En typisk krets bygget rundt LM2621 .....	11
[14]: To operasjonsforsterkere i LM2904DT.....	12
[15]: Bevegelsessensoren innebygget med akselerometer, gyroskop og magnetometer.....	12
[16]: Oppkoblingen av en målebro.....	13
[17]: De forskjellige forsyningsalternativene sammenlignet. Med fordeler og ulemper.....	13
[18]: Tidlig skisse av montering for produkt. Her skissert i Tinkercad.....	14
[19]: Endelig revisjon av fremdriftsplan. Enkelte deler forskjøvet pga. sen komponentlevering.....	15
[20]: Sammenligning mellom de ulike typene avstandssensorer som ble vurdert.....	16
[21]: Første testing og kalibrering av ultralydsensoren MB7374-100.....	17
[22]: Hvordan kretsen ser ut ved endelig revisjon, kun med prosjektert oppkobling her da PCB ikke var ferdig produsert.....	17
[23]: Viser hvordan ledningsbaner og komponenter plasseres på kortet.....	18
[24]: Hvordan selve kretskortet ser ut, uten komponenter påloddet og montert. Platene ved hvert komponent viser loddeplatene som komponentene festes til.....	18
[25]: Oppkobling av Wheatstone målebro og komparator.....	19
[26]: Kalibrering av temperatursensor opp mot referansepunkter med tydelige temperaturer.....	20
[27]: Innkapslingen for produktet under produksjon.....	21
[28]: Oversikt over alle sensorer utplassert over et visst område. Gir mulighet for å endre fokus fra sensor til sensor.....	22
[29]: Informasjonen fra sensoren fremvist under den ene mappen på nettsiden. F.v. Total snødybde, vindstyrke, innvendig temperatur og utvendig temperatur.....	22
[30]: Tabell oversikt over målinger sendt til server. Utklipp hentet fra supabase server.....	32
[31]: Endelig kretskort plassert i innkapslingen.....	32
[32]: Lukket kapsling med snødybdesensor foran.....	33
[33]: Lukket kapsling bakfra. Strømforsyning innkoblet.....	33
[34]: Den nye ESP32 modulen fjernet fra kretskortet, i påvente av en eldre ESP32 med riktige pins.....	34

# 1. Innledning

## 1.1. Problemstilling

Designe, planlegge og konstruere en sensor som kan kontinuerlig måle mengde snø på en valgt vei, måle temperatur, vind, så brukeren kan avgjøre om det er fare for driftsnø eller is på strekningen. Deretter skal informasjonen fremvises på et lett anvendelig grensesnitt for de ansvarlige for veivedlikehold.

## 1.2. Begrunnelse

Oppgaven vekket interesse for gruppen da dette ligger innenfor det arbeidsområdet og de temaene vi har jobbet med i løpet av studiet, noe som gjorde at vi kunne utnytte kompetansen vi har oppbygget så langt. Sensorer, IoT og trådløs kommunikasjon for å fremvise informasjon er aktuelt i dagens arbeidsmarked, noe som gjør oss forberedt på arbeidsoppgaver som kan møte oss i yrket vi jobber mot på studiet.

## 1.3. Avgrensing

Med en oppgave som denne, kreves det at vi setter tydelige retningslinjer og grenser for oppgavens omfang. Dette produktet har mulighet til å utvides til et stort system, med mange funksjoner og høy grad av kompleksitet. Gruppen har, sammen med arbeidsgiver, blitt enige om å begrense seg til én enhet, med sensorer for måling av dybde, temperatur og vind. Enheten forsynes med strøm fra solcellepanel og et BMS (Battery Management System). Informasjonen enheten samler inn, vises i et grafisk grensesnitt, i form av en browserbasert applikasjon.

## 2. Bakgrunn

### 2.1. Bakgrunn for oppgaven

Med utbyggingen av veinettet og de klimaendringene samfunnet står overfor i dag, er det relevant å gjennomføre en digitalisering av mange offentlige funksjoner som i dag innebærer mye manuelt arbeid, deriblant varsling av behov for vedlikehold. I tillegg er det mange ønsker om å redusere kostnader og å avlaste arbeidsmengden. En effektivisering av vedlikeholdsvarsling ville spart produktinnehaver for mye tid og arbeidskraft som til nå har blitt brukt til å manuelt reise ut på befaring for å vurdere vedlikeholdsbehovet. Produktet som gruppen ønsker å utvikle, vil også spare arbeidsgiver for økonomiske utgifter siden de ikke trenger å ha like mange på jobb som venter på å reise ut, men heller hjemme i beredskap.

### 2.2. Bruksområde

Produktet er anvendelig i de situasjonene brukeren ønsker å vite hvor mye snø som har falt på en strekning, og hvilke forhold man bør vente seg. Hovedsakelig vil produktet være nødvendig i områder hvor man ikke har overvåkning av veien og ikke ferdes så ofte, slik som fjelloverganger og områder kjent for mye uregelmessig snøfall. Med dette produktet kan man likevel få en oversikt over behovet for å brøyte eller utføre vedlikehold på veien, uten å reise ut til stedet. Ved perioder med uforventet snøfall, nedbemanning eller andre uforventede hendelser, vil man lettere kunne prioritere hvilke områder det er spesielt behov for vedlikehold.

### 2.3. Markedsanalyse

I analysen av markedspotensial og mulig etterspørsel vil vi i utgangspunktet ta for oss hovedintensjonen med produktet, å måle snødybden. Dette for å kunne måle det eksakte formålet opp mot andre alternativer. Markedet for dybdemålere av snø er et marked med mye potensial, da det ikke er overfylt med tilbud, men naturen i Norge tilsier at det er et produkt som kan ha stort potensial. Potensielle kunder er inndelt i brukersegmentering, da det vil være mest relevant å inndele bruksområder og relevans inn i hva slags formål kundene har. For privatpersoner vil det være mer relevant å for eksempel ha en enhet stående på hytteveggen på fjellet, hvor man kan ha oversikt over dybden før man reiser. I denne situasjonen vil kunden kunne få informasjon på telefonen, og deretter ta en avgjørelse om å bestille brøyting, forberede seg på å brøyte selv, eller i verste fall unngå å reise hvis det har vært et veldig stort snøfall. Den andre aktøren dette produktet er relevant for, er kommuner og industriaktører. Et eksempel på dette er vår arbeidsgiver, som har som intensjon å ta i bruk produktet. Aktører i

offentlig sektor, som har ansvar for vedlikehold på kommunale veier og fylkesveier, kan sette opp denne sensoren og få oppdateringer på behovet for vedlikehold. I en ideell situasjon, ønsker arbeidsgiver å sette opp en mengde enheter på forskjellige strekninger hvor de ikke har oversikt over snøfall og føre. Dermed kan de varsles når snøfallet overskrider de verdiene bestemt for vedlikehold av veien.

Ved oppstartsmøte av prosjektet med arbeidsgiver, kom det frem at arbeidsgiveren tidligere har hatt et lignende prosjekt, men på kontrakt av en ekstern aktør. Dette prosjektet ble kalt Smart Vinterveg og faller under et samarbeid med de andre kommunene rundt Mjøsa, som heter Smarte Mjøsbyer (Smarte Mjøsbyer, 2020). Samarbeidet går ut på å effektivisere mange funksjoner, spesielt rundt vedlikehold. Prosjektet ble outsourcet til et selskap som heter Betelo, som stilte med sensorer og IT-løsning for overføring av informasjon. Oppstart for dette prosjektet var våren 2019, og ble avsluttet 28. februar 2022, da kommunene ikke oppnådde den mengden av innovasjon og fremgangen de ønsket innenfor de tidsrammene som ble satt, samtidig som at Betelo ikke helt fikk til å implementere alle funksjonene i sitt eget system inn i kommunen sitt system. Betelo tok også med seg produktene da kostnaden for disse var høye, noe kommunene ikke kunne ta seg råd til. Betelo er å kunne anse som gruppen sin største konkurrent i et eventuelt marked da de allerede har godt etablerte sensorer og et ferdig oppbygd IT-system. Ønskene fra arbeidsgiver var et lignende system som de oppnådde med Betelo, men til en lavere pris.

Det ferdige produktet gruppen vil produsere, er arbeidsgivers eiendom og dersom behovene deres er tilfredsstilt vil det bli tatt i bruk. Ideen er likevel mulig å bygge en markedsstrategi etter. For å kunne lykkes i et slikt marked, kreves det at man har verdier og egenskaper ved produktet sitt som overstiger konkurrentene. Produktet gruppen skal designe bygger på en allerede pålitelig og lett anvendelig programmeringsplattform, Arduino. Produktet tar i bruk komponenter som er lette å få tak i og, ikke minst, rimelige i kostnad. Dette er antakelig den største styrken til produktet. Ved å velge komponenter som er pålitelige, men rimelige, og open-source programvare som ikke er lisensiert, kan man holde kostnader nede og tilby et produkt til en lavere pris. I tillegg til dette virker det av nettsiden for samarbeidet at Betelo ikke har en direkte oversikt og måling av temperatur og vind. (Smarte Mjøsbyer, 2020) Dette gir en fordel i form av flere tilgjengelige funksjoner. Likevel har Betelo mulighet for GPS-lokasjon på brøytemaskiner, noe som gir dem en fordel for enkelte kunder.

(Smarte Mjøsbyer, 2020)

## 2.4. Utvikling videre

Produktet er i utgangspunktet laget for å enkelt kunne ekspanderes til et større system, med mulighet for implementering av flere enheter. I bedriftsperspektiv er det aktuelt for større aktører å kunne ha flere enheter samlet på samme grensesnitt. I offentlig sektor vil det være relevant å kunne plassere flere av disse innenfor kommunegrensene og kunne oversiktlig følge med på de ulike nivåene på forskjellige enheter på samme grensesnitt. Med slike muligheter vil man effektivt kunne prioritere hvilke strekninger som er mest kritisk, i tillegg til at man kan gardere bemanning til riktige områder, uten å måtte sende noen for å utføre målinger i et annet område.

I dag blir plasseringen til produktet lagt inn manuelt. Ved å utvikle en GPS-løsning montert i produktet vil man automatisk kunne få status på hvor sensoren befinner seg, som igjen vil være praktisk ved ekspansjon og flere sensorer å ha oversikt over. For å kunne være mer konkurransedyktig mot det private kundemarkedet, kunne systemet også inneholde en løsning hvor man får oversikt over hvor de som brøyter befinner seg, når de er ute på oppdrag. Da vil privatkunder kunne få en oversikt over status på veistrekningene, istedenfor å kun se på verdiene fra sensorene. I tillegg vil dette gjøre det lettere for offentlige aktører å kunne planlegge ruter og gi beskjeder om hvem som skal rykke ut eller endre ruten.

Grensesnittet kan utvides med historikk for antall alarmer innenfor et visst tidsrom og mengde som da har falt. Ved oppstart av prosjektet ble det også diskutert at kommunen ønsket et produkt med en egen applikasjon de kunne ha på telefonen eller nettbrett i brøytemaskinene. Gruppen har kun utviklet en nettbasert applikasjon.

## 2.5. Risikovurdering

Det kreves i en oppgave å diskutere risikoer knyttet til dens art. Tre enkle spørsmål er kjernen i risikovurderingen illustrert i figur 1:

- Hva kan gå galt?
- Hva kan vi gjøre for å forhindre dette?
- Hva kan vi gjøre for å redusere konsekvensene dersom det skjer?

I forhold til et prosjekt hvor man skal produsere et produkt, vil det være andre risikofaktorer som kan oppstå, enn ved en analyseoppgave. Risikofaktorene for denne oppgaven er beskrevet og vist i vedlegget, prosjektplanen for oppgaven. På høyre side i figuren er risikofaktorene illustrert. Dette er hendelsene med høyest risiko som kan skje under perioden. Sannsynlighetsfaktoren opererer på en skala fra 1-5, Konsekvensfaktoren på en skala fra 1-10 og Risikofaktoren på en skala fra 1-25. Hendelser med risikofaktor mellom 1-4 er lave og vil ikke påvirke oppgaven særlig om det skulle skje. Hendelser med risikofaktor mellom 5-12 har middels risiko og kan påvirke oppgaven, men burde være løsbare. Over 12 er høy risikofaktor og kan være ødeleggende for om det endelige produktet møter de kravene satt i oppgaven, dersom hendelsen skulle inntreffe.

I dette prosjektet er det større sjans for at svakhet i komponenter, problemer i oppkobling eller andre praktiske problemer inntreffer. Gruppen har spesielt tatt hensyn til risikoer relatert til robusthet og hvordan produktet tåler varierende klima, da produktet skal monteres i omgivelser som kan påvirke eller ødelegge hvordan produktet fungerer. Hvilke risikofaktorer som i løpet av perioden viste seg å være reelle, skal diskuteres i kapittel 6, som inneholder diskusjonsdelen av rapporten.

(Arbeidstilsynet, u.d.)

Hendelse	Sannsynlighet	Konsekvens	Risiko	Strategi	Tiltak
Kapslingsgrad På system	2	7	10	Holde oss til løsninger med riktig kapslingsgrad	Pålitelige leverandører.
Temperaturmotstand	1	9	18	Velge løsninger med høy/lav nok angitt temperaturmotstand	Løsningsvalg.
Alvorlig Sykdom	2	6	12	Andre medlemmer må steppe inn	Skifte arbeidsområde
Tidsfrister ikke overholdt	2	9	20	Føre en god tidsplan for frister	Skal ikke forekomme
Intern konflikt	1	6	12	Løse konflikten på en diplomatisk og rettferdig måte	Være positive og selvkritiske ved uenigheter.
Forsinket leveringstid på komponenter	2	7	14	Sørge for bestilling i god tid, og obs. på angitt leveringstid fra leverandør	Planlegging.

[1]: Vurdering av forskjellige risikofaktorer i prosjektet og alvorlighetsgrad av disse.

## 3. Teori

### 3.1. MB7374-100

Denne ultralydsensoren blir brukt i prosjektet for å samle inn informasjon om lengde og spredning på snøfall innenfor et område bestemt av kjeglen på selve sensoren. Ultralyd vil si lyder med et frekvensområde over det mennesket kan høre, som vil si over 20 000 Hz.

Sensorens funksjonalitet er illustrert i figur 2. Den tar i bruk ultralyd, sender lydbølger (vibrerende bølger over et medium) som reflekteres tilbake fra en gjenstand og skaper ekko. Ved å måle tidsdifferansen mellom utsendt signal og mottatt refleksjon, kan man regne ut avstanden til gjenstanden. Denne sensoren, og andre lignende, er inspirert av biomimikk og bygger på naturlige konsepter.

I naturen bruker blant annet flaggermus og delfiner ultralyd for å anslå avstanden til f.eks. et byttedyr, kalt ekkolokalisering. Høyere frekvens på lydbølgene betyr kortere bølgelengde på signalet og høyere oppløsning (Valmot, 2015). Dette vil si at signalet har forflyttet seg over en kortere distanse. (Valmot, 2015)

MB7374-100 er en kostnadseffektiv sensor som ofte anvendes til værstasjoner, for å måle nivå på snøfall eller for å angi hvor mye av en mengde som befinner seg i en container. I figur 3 kan du se et bilde av sensoren.

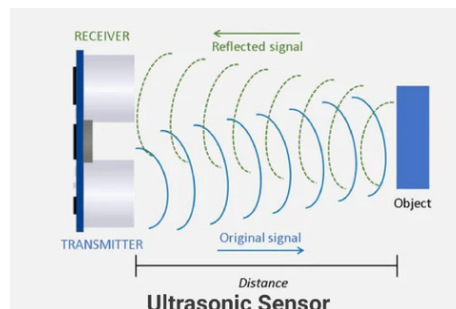
Sensoren er ikke sårbar for støy, lett monterbar og kan måle med opptil 5 meter avstand med en feilmargen på typisk 0.1% ved 1 meter måling (MaxBotix, u.d.).

Sensoren har en operasjonstemperatur på mellom -40 grader til +65 grader, noe som gjør den optimal å bruke på norske vinterveier. Sensoren har IP67 rating i

kapslingsgrad, som gir høy beskyttelse mot vann, støv og faste gjenstander. Sensoren opererer med en

inngangsspenning på mellom 2.7 volt og 5.5 volt, som gjør den anvendelig i en krets med f.eks. Arduino. Sensoren er kompatibel med RS232 og TTL serial kommunikasjon.

(MaxBotix, u.d.)



[2]: Konseptet ultralydsensoren er basert på. Tar inspirasjon fra biomimikk, her fra blant annet flaggermus sin ekkolokalisering. (What Is Ultrasonic Sensor, 2022)



[3]: MB7374-100, ultralydsensoren brukt i prosjektet. Her med kjegleformet innkapsling som indikerer måleområde. (DigiKey, u.d.)



### 3.2. ESP32: Trådløs Kommunikasjon

ESP32 er en mikrokontroller med innebygd Wifi og Bluetooth, ideell for IoT-systemer som krever at informasjon deles over en trådløs kanal. Det finnes flere forskjellige ESP32 moduler og i figur 4 er det illustrasjon av en ESPWROM-32S. Enheten er utstyrt med en 32-bit CPU, dual core som gir høy ytelsesevne på opptil 240 MHz (ESP32.net, u.d.). Dette gjør den kapabel selv med kompliserte og krevende oppgaver. Med smart strømstyring, kan enheten gå i dvale ved inaktivitet, som gir et lavt strømforbruk. Den er også ressurseffektiv i aktiv modus. ESP32 har en innebygd Real-Time-Clock med nøyaktig tidshåndtering, vesentlig i noen IoT-applikasjoner. Enheten støtter blant annet I2C, SPI og UART så den kan kommunisere med mange sensorer, enheter og systemer. I tillegg til å være lett programmerbar, støtter den både Arduino og Micro Python, og har støtte for ulike minnekort så den kan håndtere lagring av data. (ESP32.net, u.d.)



[4]: ESP32-modul, mikrokontrolleren som styrer den trådløse kommunikasjonen i prosjektet. (Surplustronics, u.d.)

### 3.3. NB-IoT

Narrow-Band-Internet-of-Things er en 4G-basert kommunikasjonsteknologi, laget for å overføre små mengder informasjon over lengre avstander. Denne teknologien er skapt for IoT-enheter som ikke trenger en stor båndbredde for å overføre informasjon. Nettverket bruker en smal båndbredde, i tillegg til en lav bitrate. Dette gjør det mulig å overføre informasjon fra steder med dårlig dekning og over lange avstander. Dette gjør også at modulene bruker mye mindre strøm enn 4G moduler som er en fordel for oss som bruker batterier (Last\_Mile, u.d.). Enheter som bruker denne teknologien kan også gå i dvale når de er inaktive, noe som sparer mye strøm, ved for eksempel batteridrevne sensorer. NB-IoT er støttet av mobiloperatører over hele verden. (Rosencrance, 2020) (Last\_Mile, u.d.)

SIM7080G er kommunikasjonsenheten brukt i prosjektet og illustrert i figur 5. Denne enheten støtter CAT-M1 og NB-IoT teknologi, utviklet for applikasjoner som faller innenfor Internet Of Things. Enheten er kompatibel med ESP32, mikrokontrolleren ansvarlig for den trådløse kommunikasjonen fra Arduinoen. Enheten bruker mobilnettverk for å overføre informasjon, og kobler seg på 4G-nettverket. Enheten støtter GNSS. Dette vil si at den er kapabel til å bruke posisjoneringssystemer som GPS, GLONASS og Galileo(Elfa Distrelec, u.d.c). I dette prosjektet krever enheten overføring av informasjon og stedsbestemmelse, noe SIM7080G kan levere. Enheten er kapabel med forskjellige IoT-enheter ved hjelp av standard kommunikasjonsprotokoller som TCP, HTTP, MQTT og UDP. Dette gjør at den kan koble seg opp på allerede eksisterende systemer og plattformer uten noe videre utvikling. (Elfa Distrelec, u.d.c) (Telenor, 2019)



[5]: NB-IoT modul brukt for å kommunisere mellom blant annet mange sensorer. Bruker 4G og 5G nettet. (Elfa Distrelec, u.d.d)

### 3.4. PT1000 Temperaturføler

PT1000 er en type temperaturføler, ofte brukt i industrielle sammensetninger. Disse elementene brukes ofte i situasjoner hvor nøyaktigheten er prekær, som i matproduksjon, medisin, forskning og utvikling.

PT1000-elementet som illustrert i figur 6 er bygd opp av en platinatråd, viklet i en spiral rundt en keramisk kjerne eller glassfiberkjerne.

Platinatråden har en motstand på 1000 ohm ved 0°C. Motstanden endrer seg proporsjonalt med temperatur, denne kan leses av og

temperaturendringen kan regnes ut. Forholdet

mellom temperatur og motstand styres av en temperaturkoeffisient, definert som endring i motstand per grad celsius/fahrenheit. For ett PT1000-element med platinamateriale er temperaturkoeffisienten på 0.00385 ohm/ohm/°C, som vil si at for hver grad økning i temperatur, vil motstanden øke med 0.385% av motstanden ved 0°C.(MaxSiefert, u.d.)

Platinum er et optimalt materiale for dette bruksområdet da det har en mer lineær temperaturkoeffisient enn andre materialer og er mer stabilt. Følerne kan dermed gi nøyaktige målinger over et bredt temperaturområde. (Peak Sensors, u.d.) (MaxSiefert, u.d.)



[6]: PT1000 Temperaturfølerelement. Metallbasert med høy nøyaktighet og robusthet. (RS, u.d.)

Motstandstermometre finnes i flere forskjellige typer, som for eksempel PT100, PT200, PT500 og PT1000. Det kjemiske tegnet for platina (Pt) kjennetegner navnet på disse elementene, sammen med tallet som tilsier den elektriske motstanden i Ohm ved 0°C. Høyere elektrisk motstand ved 0°C tilsier lavere målefeil (Peak Sensors, u.d.). Det vil derfor i en situasjon hvor det er nødvendig med nøyaktighet være bedre å velge en føler med høy motstand. I prosjektet er det tatt i bruk et PT1000-element for å måle temperaturen på den aktuelle lokasjonen. Temperaturen vil kunne avgjøre om det er risiko for is på strekningen, og om man må ta en avgjørelse om strøing. Dette vil være avgjørende for hvordan mannskapet skal behandle varslinger og prioriteringer av utstyr. (Peak Sensors, u.d.)

### 3.5. Cotech sensor for Vindhastighet

Denne sensoren er i utgangspunktet tilbehør på en værstasjon, kalt FT0203 gitt av Cotech. Dette er en USB-tilkoblet vindsensor som bygger på trykkraften til vind og retningen den trykker. Vindsensoren måler vindhastigheten ved hjelp av hastigheten i vindskovlene på toppen av sensoren. Sensoren er illustrert i figur 7. Dette fungerer ved at vind skovlene er koblet til en sentral akse. Når denne sentrale akselen roterer rundt vil en magnet som er koblet til akselen inni sensoren også rotere. I denne sensoren er det også en reed-kontakt. Denne reed-kontakten bytter mellom åpen eller lukket ut fra posisjonen til magneten som spinner rundt når det er vind tilstede.

Magneten har to posisjoner hvor den vil gjøre at reed-kontakten er åpen. Vindsensoren kan måle fra 0-50 m/s med en nøyaktighet på  $\pm 3$  m/s opptil 10 m/s og  $\pm 10\%$  opptil 50 m/s (Manua.ls, u.d.). Retningen på vindsensoren avgjøres av et 8-punkts kompass, med 45 grader resolusjon

og 0-360 grader målområde. Vindsensoren bruker et 16-sekunders intervall mellom målinger. Sensoren er i dette prosjektet modifisert i den forstand at den kan kobles direkte på kretsen i produktet. (Manua.ls, u.d.) (clas ohlson, 23.02.22)



[7]: Vindsensor som måler styrke og retning på vind i 360 grader. (Clasohlson, 2022)



[8]: Vindsensoren oppmontert for kalibrering og testing av funksjonalitet. Foto: Jørgen Johannessen



[9]: Vindsensoren åpnet og modifisert for å kunne direkte tilkobles til produktet. Foto: Jørgen Johannessen

### 3.6. HY-DI-2S2P-S1

Dette er en oppladbar Li-Ion batteripakke med 7.2V og 6.67Ah kapasitet. Batteriet er illustrert i figur 10. Batteripakken består av 4 battericeller, med tilsvarende 4 dioder som viser til hvor stor ladning batteriet har. Batteriet kan levere i snitt 7A til kretsen, med en peak på 8.8A i totalt 10 minutter. Batteripakken har en anbefalt temperatur ved oppladning på 0°C til +45°C. Anbefalt temperatur ved utladning er -20°C til +60°C (Elfa Distrelec, u.d.b) . I dette prosjektet er denne batteripakken kombinert med et solcellepanel som gir muligheten for tidvis oppladning av batteriet, slik at man slipper å ha en direkte oppkobling på strømmettet. Da er produktet mer fleksibelt for plasseringer og lokasjoner rundt i veinettet. (Energy Systems, u.d.) (Elfa Distrelec, u.d.b)



[10]: HY-DI-2S2P oppladbar batteripakke. (Hy-Line, u.d.)

### 3.7. Ezviz Solar panel

Solcellepanelet fra Ezviz er et slitesterkt og robust panel, i utgangspunktet designet for å bruke sammen med et kamera fra samme produsent. Panelet er illustrert i figur 11. Panelet er bygget opp av enkeltkrystallinske silisiumsolceller. Denne typen solceller innebærer en høyere kostnad enn ved multikrystallinske silisiumsolceller, men med en effektivitet på 18-23% blir også mer sol tatt opp i hver solcelle. Solcellen kan per datablad utlevere en spenning på 6 volt  $\pm 5\%$  og en effekt på totalt 6.18 Watt. Med en arbeidstemperatur fra -20°C til +60°C, fuktighet på opptil 90% og en tetthetsgrad på IP65, er dette solcellepanelet tilstrekkelig for prosjektets formål. IP65 vil tilsi en total tetthet mot støv og faste gjenstander, i tillegg til sikker beskyttelse mot vannstråler projisert av en dyse på 6.3 mm fra alle vinkler. Ved testing viste det seg at solcellepanelet kunne levere en utspenning på max 9 volt, minimum 3 volt.

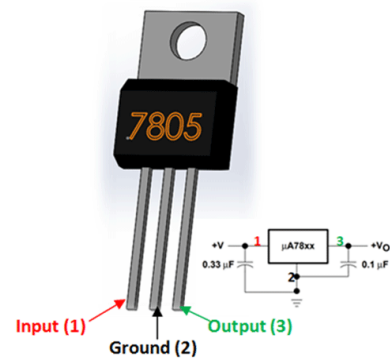


[11]: Enkeltkrystallinsk silisium-solcellepanel fra EzViz, her montert på en loddrett vegg. (EzViz, u.d.a)

(EzViz, u.d.b) (Haaland, 2020)

### 3.8. LM7805

LM7805 er en lineær spenningsregulator. Den er en del av LM78XX serien og de to siste sifrene 05 beskriver hva utgangsspenningen er. Dette er da en spenningsregulator som har en utgangsspenning på +5V. Dette passer bra til vår ESP32 og flere sensorer som har en 5V inngangsspenning. (LM7805 Voltage Regulator, 2019). I figur 12 ser man at den integrerte kretsen har tre bein. Bein 1 tar imot spenning mellom 7V og 35V. Bein 2 er jord og bein 3 er utgangen. På utgangen får man en spenning mellom 4.8V og 5.2V (7805 Voltage Regulator, 2020).

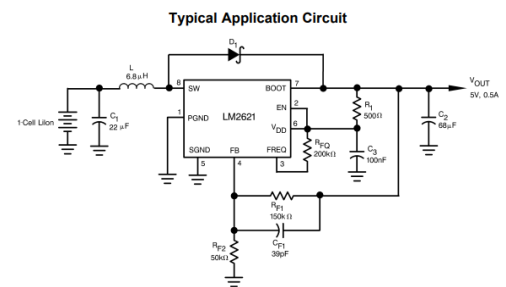


[12]: Benene på en LM7805 (LM7805 Voltage Regulator, 2019)

### 3.9. LM2621

LM2621 er en step-up DC-DC switching regulator. Denne kan motta en spenning på mellom 1.2V og 14V. På utgangen får man en regulert spenning som man selv kan justere mellom 1.24V og 14V. LM2621 har en høy effektivitet som kan nå 90%. Dette gjør den godt egnet til batteridrevne enheter som trenger lang batterilevetid (Texas Instruments, 2015). I figur 13 ser man en typisk krets bygget rundt LM2621. Denne kretsen passer bra til å regulere et ujevnt signal som f.eks. fra et solcellepanel hvor spenningen endrer seg gjennom dagen.

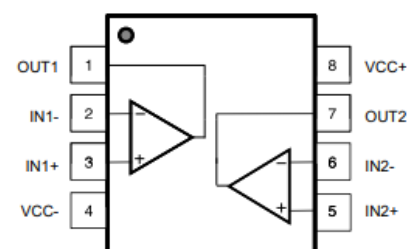
(Texas Instruments, 2015)



[13]: En typisk krets bygget rundt LM2621 (Texas Instruments, 2015)

### 3.10. LM2904DT

LM2904DT blir brukt i mange forskjellige applikasjoner, inkludert signalforsterkning, filtrering, komparatorer, oscillatorer og strømstyring. Denne inneholder to uavhengige operasjonsforsterkere. En operasjonsforsterker har fem tilkoblinger, to innganger, en utgang og to spenningsforsyning tilkoblinger. LM2904DT har felles spenningsforsyning tilkobling. I figur 14 ser

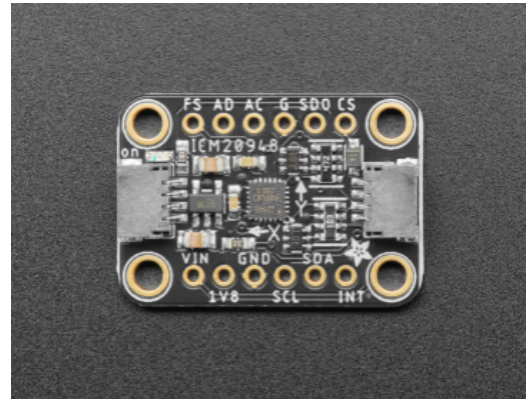


[14]: To operasjonsforsterkere i LM2904DT (STMicroelectronics, 2019)

man de to operasjonsforsterkerne i samme integrerte krets ovenfra. LM2904DT har et veldig lavt strømforbruk som passer godt i batteridrevne applikasjoner. LM2904DT kan operere i temperaturer mellom  $-40^{\circ}\text{C}$  og  $125^{\circ}\text{C}$ . (STMicroelectronics, 2019) (*Operasjonsforsterker — Dataelektroniker*, u.d.) (*Operasjonsforsterker*, 2022)

### 3.11. ICM-20948

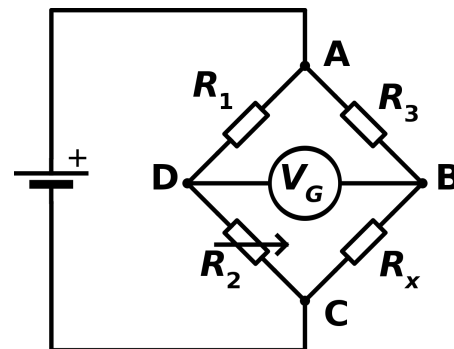
ICM-20948 er den bevegelsessensoren med ni akser som har det laveste strømforbruket. Sensoren er illustrert i figur 15. Den har et akselerometer som måler akselerasjonen i tre akser (x,y,z), et gyroskop som måler rotasjonshastigheten i tre akser (x,y,z) og et magnetometer som måler magnetfeltets styrke og retning i de tre aksene (x,y,z). I tillegg til dette har den også en innebygd temperatursensor. Denne sensoren kan kommunisere over I2C og SPI noe som gjør den kompatibel med mange forskjellige mikrokontrollere (Ewald, 2021) (TDK InvenSense, 2017).



[15]: Bevegelsessensoren innebygget med akselerometer, gyroskop og magnetometer. (Sjepert, u.d.)

### 3.12. Wheatstone målebro

En Wheatstone målebro er bestående av fire resistanser. Målebroen er koblet opp slik som illustrert i figur 16. Denne målebroen fungerer ved at man kobler en spenning på begge endene av målebroene. Her vil  $V_G$  spenningsforskjellen mellom punkt D og B. Hvis  $R_1$  og  $R_3$  er like og  $R_2$  og  $R_x$  er like, vil denne broen være i balanse. Da er forskjellen mellom D og B lik null. (Ametherm, u.d.)



[16]: Oppkoblingen av en målebro (Blumlein, u.d.)

## 4. Metode

### 4.1. Planlegging

Prosjektets art krever en nøye og gjennomtenkt planlegging, med mye forberedelser og prosjektering for et suksessfullt resultat. Perioden innvies med et forberedelsesmøte med arbeidsgiver, hvor forskjellige parametere skal klargjøres og bestemmelser skal foretas. Arbeidsgiver hadde et bestemt produkt i tankene idet oppgaven ble sendt ut til arbeidstaker. De ønsket et produkt som oppfylte oppgavens kriterier, var bærekraftig og ressurseffektivt, jamfør vedlegget Oppstartsmøte med kommunen, som er et referat fra innledende møte med arbeidsgiver til prosjektet. Arbeidsgiver hadde ingen spesielle kriterier til utseende, fremgangsmåte og teknologi, men ønsket en viss funksjonalitet som er beskrevet i de innledende kapitlene i

Løsning	230V forsyning	Solcellepanel og batteri	Hybridløsning
Fordeler	Stabil, kontinuerlig forsyning	Mulig med forflytning i større grad. Enkelt å koble opp	Stabil forsyning ved oppladning.
Ulemper	Lite tilgjengelig, krever at produktet står sentralt Krever mer innviklet oppkobling	Litt mindre stabil forsyning, væravhengig i større grad.	Krever lav nok strømtrekk til at batteriet ikke utlades på dagen. Igjen mer innviklet oppkobling.

[17]: De forskjellige forsyningsalternativene sammenlignet. Med fordeler og ulemper.

rapporten. I tillegg til selve produktet, trengtes det en løsning for et grensesnitt tilgjengelig også i brøytemaskinene. Videre ble det diskutert med arbeidsgiver, hvilken løsning for batteri og strømforsyning som var ønskelig. I utgangspunktet sto det mellom en løsning med direkte oppkobling på 230V, en løsning med uavhengig oppladning i form av solcellepanel og batteri, samt en hybridløsning som bruker forsyningen til lyktestolper og et batteri. Lyktestolpene har kun forsyning på natten, da de er aktive. Dermed vil batteriet evt. kunne opplades på natten, og utlades på dagen. Løsningen som ble bestemt i dette prosjektet ble solcellepanel og batteri, da dette var mest relevant for veinettet i området. Arbeidsgiver krevde at produktet skulle kunne monteres opp på strekninger hvor det ikke er like tilgjengelig med lyktestolper eller andre løsninger for strømforsyning.

Ved et videre eventuelt samarbeid kan det være ønskelig for arbeidsgiver å kunne få produkter med både 230 volts oppkobling og solcellepanel, avhengig av hvor i vegnettet de skal kobles opp.

Fortløpende etter oppgaven ble utgitt og kriteriene var tydelige, ønsket gruppen å planlegge produkt og begynte å gjøre research. Tidlig i oppgaven ble det tegnet opp en skisse for hvordan innfatningen på produktet burde se ut. En prosjektering av ideen ble designet som viser bruk, som vist i figuren på høyre side. Produktet vil ideelt monteres loddrett på en lyktstolpe, rettet mot veien. I denne situasjonen vil sensoren kunne få nok, men ikke for mye, distanse fra veien som skal måles. Optimal høyde ble i møtet anslått til rundt 3 meter. Dermed kan sensoren stå trygt fra eventuell snøsprut ved brøyting og høy snømengde. Ved bruk av trigonometriske beregninger, kan riktige verdier måles og sensoren kalibreres.



*[18]: Tidlig skisse av montering for produkt. Her skissert i Tinkercad. (Tinkercad, u.d.)*



## 4.2. Fremdriftsplan

I planleggingsfasen av prosjektet ble det fremstilt en fremdriftsplan for prosjektet.

Fremdriftsplanen viser den fremgangen gruppen ønsket å ha. Fremdriftsplanen til oppgaven ble innlevert i vedlegget, prosjektplanen for perioden. Planen ble endret i løpet av perioden, pga. uforutsette hendelser og endringer i planleggingen. Disse endringene diskuteres senere i rapporten. Fremdriftsplanen tar for seg hovedaktiviteter og delaktiviteter i emnet.




Fremdriftsplanen, lagt i vedlegget *prosjektplan*, er den opprinnelige fremdriftsplanen vi ble enige om i begynnelsen av oppgaven. Endelig revisjon av fremdriftsplan ligger i figur 19.

Endringer er markert i mørkeblått. Som man kan se i planen, er de fleste endringene et produkt av forsinkede komponenter, noe som har ført til at enkelte prosesser har blitt forlenget i noen uker. I tillegg er det lagt til et ekstra veiledningsmøte for oppfølging med veileder.

Activity	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	
Report																								
Start-up meeting																								
Planning																								
Guidance																								
Milestone 1: Project plan																								
Ordering components																								
Sensor testing																								
Milestone 2: English presentation																								
Circuit design																								
Code for testing																								
Exam INGG2300																								
Code for product																								
Soldering																								
Assembly																								
Completion																								
Milestone 3: Final title																								
Milestone 4: Submit report																								
Milestone 5: Submit poster																								
Milestone 6: Present Bachelor																								

[19]: Endelig revisjon av fremdriftsplan. Enkelte deler forskjøvet pga. sen komponentlevering.

### 4.3. Valg av type sensor

Type sensor	Ultralydsensor (eksempel: MB7374-100)	Infrarød sensor (eksempel: GP2Y0A710K0F)	Lidar sensor (eksempel: TIM 100-3010200)
			
Temperaturområde	-40°C til +65°C	-10°C til +60°C	-10°C til +50°C
Nøyaktighet	+/- 1 mm ved 1 meter måling	Lite dokumentert, avhenger av egenskapene til materialet. Men generelt mye error	+/- 20 mm
Måleområde	0 - 5 meter	1 - 5.5 meter	0 - 3 meter
Nominelt strømtrekk	ca. 2-3mA	ca. 30mA	Lite oppgitt, men totalt forbruk på 2.2W
Kostnad	ca. 2200 kr	ca. 3-400 kr	ca. 11500 kr

[20]: Sammenligning mellom de ulike typene avstandssensorer som ble vurdert. (MaxBotix, u.d.) (Elfa Distrelec, u.d.a) (RS Components, 2018)

I planleggingen av prosjektet ble det vesentlig å velge hvilken type avstandssensor vi skulle basere oss på. I dette tilfellet ble det naturlig å sammenligne de forskjellige egenskapene til sensorene og ta et valg ut ifra de egenskapene som er mest relevante for formålet til sensoren. I figur 19 er de sensorene som gruppen vurderte sammenlignet opp mot hverandre, med forskjellige egenskaper. Spesielt hvilke temperaturer sensoren kan prestere på er viktig, og her viste ultralydsensoren seg å være velegnet. Sensoren vil være nødt til å prestere ved temperaturer ned til minst -30°C. Her vil de andre sensorene ikke kunne garantere funksjonalitet. Nøyaktigheten på ultralydsensoren viste seg også å være høyere rundt den avstanden som er ønskelig i prosjektet. Med et lavt strømtrekk og relativt lav kostnad, ble dette det naturlige valget for sensor. Måleområdet er også tilstrekkelig for funksjonaliteten i prosjektet.

(Shawn, 2019)

#### 4.4. Gjennomføring og bestilling av komponenter

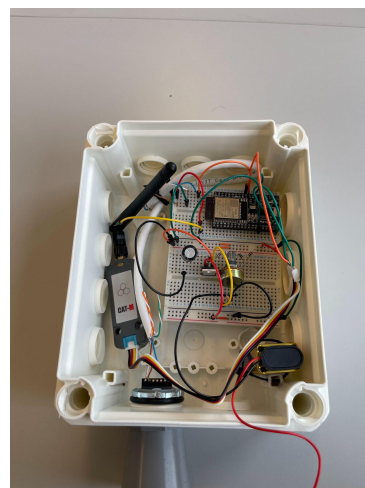
Ved inngangen av uke 6, var planleggingsfasen i prosjektet i utgangspunktet ferdig. Deretter fulgte bestilling av komponenter og konstruksjon av produktet. Som vist i vedlegget med komponentene, ble sensoren først bestilt da dette var den mest kritiske komponenten av produktet. Av erfaring med leveringstid ble denne bestilt godt i forveien da den var inkludert i produktet uansett hvilke endringer som ble gjort. Sensoren ble testet på egen hånd for å forsikre at avstand og nøyaktighet stemte mot det som var planlagt. Databladet oppga at den skulle levere en feilmargin på 0,1% på 1 meter. Sensoren ble plassert mot en gjenstand og avstanden ble målt med målebånd og en annen avstandssensor, og ultralydsensoren leverte innenfor akseptable marginer. Sensoren ble enkelt koblet opp mot en nettside, med telefonen til den ene kandidaten som en nettverks-hotspot. Etter at nøyaktighet og avstand ble forsikret, ble de andre nødvendige komponentene bestilt.



[21]: Første testing av ultralydsensoren MB7374-100.  
Foto: Jørgen Johannessen

#### 4.5. Oppkobling av løsning

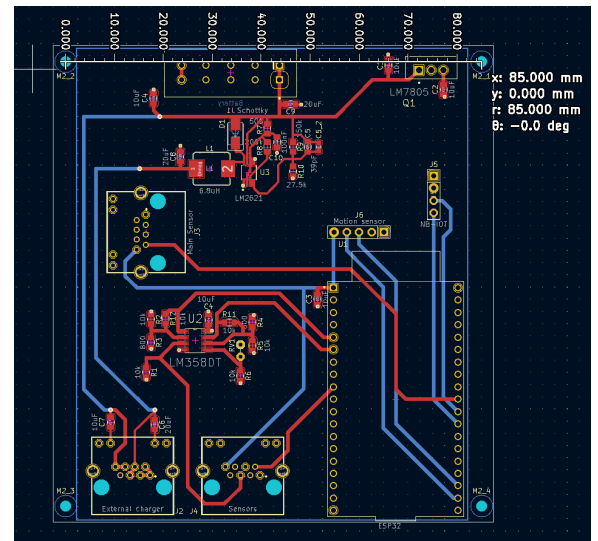
Til høyre kan man se endelig oppkobling før PCB ble bestilt for å bytte ut breadboard oppkoblingen. Kretsen består av ultralydsensoren koblet til breadboardet og en ESP32 koblet sammen med NB-IoT senderen som sørger for trådløs kommunikasjon til serveren som leverer det grafiske grensesnittet. Selve innkapslingen er en midlertidig løsning for testing og for å få innsikt i hvordan den endelige løsningen skal være.



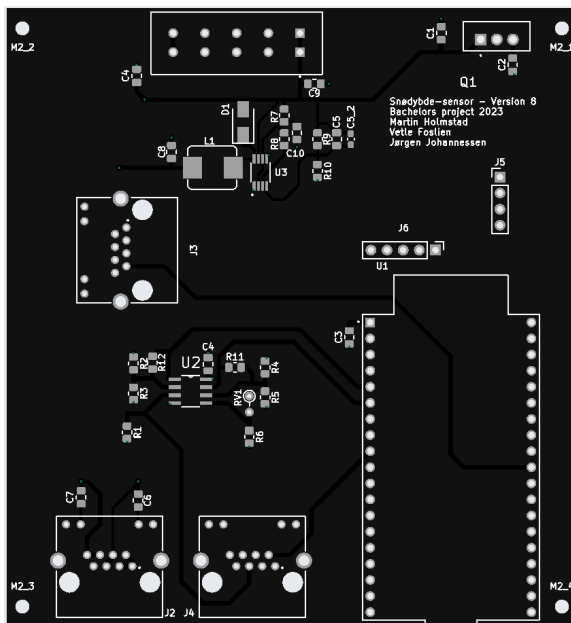
[22]: Hvordan kretsen ser ut ved endelig revisjon, kun med prosjektert oppkobling her da PCB ikke var ferdig produsert.  
Foto: Jørgen Johannessen

#### 4.6. Kretskort

Kretskortet er designet for å kunne ha et mest mulig plasseffektivt design med tanke på minst mulig åpne baner, for bedre plassering av flest mulig komponenter. Derfor valgte vi å bruke et 4-lags design med et felles jordplan koblet til alle jord-punktene, og et 5 volt power-plane som tilførsel for alle komponentene som behøver 5 volt inn-spenning. Vi har også lagt opp kretskortet med 3 andre spennings-områder, 7.5 volt, 3.3 volt og en udefinert spenning. 7.5 volt er spenningen batteriets utspenning ligger rundt, og er også en akseptert innspenning for lading av batteriet. Å bruke fellesspenningen tillater oss å bruke en rekke forskjellige metoder for oppladning av batteriet, og å tilføre spenning til systemet ved å steppe spenningen ned til 5 volt i 5 volt step-down regulatoren LM7805(Q1). Inn på 7.5 volt området, har vi to måter å tilføre kretsen spenning for drift og oppladning. Disse to metodene er begge lokalisert i RJ45 tilkoblingen J2 for



[23]: Viser hvordan ledningsbaner og komponenter plasseres på kortet. designet i KiCad (KiCad, u.d.)



[24]: Hvordan selve kretskortet ser ut, uten komponenter påloddet og montert. Platene ved hvert komponent viser loddeplatene som komponentene festes til.

ekstern tilkobling av spenningskilder. Her har vi både direkte tilførsel av forhåndsregulert 7.5 volt fra en ekstern kilde, og en mulighet for en uregulert spenning mellom 3 og 15 volt inn på pin 4 og pin 5 på J2. Denne uregulerte spenningen reguleres så til en spenning på 7.5 volt av regulatoren LM2621(U3) uavhengig av innspenningen. Dette er designet slik for å kunne akseptere et bredt spektrum av innkilder, for og så regulere disse enten opp eller ned slik at spenningen er på fast 7.5 volt og da i et område vi får utnyttet tilstrekkelig. Eksempler på slike innkilder kan være enten en USB lader med typisk 5 volt utgang, en lader med 3-15 volt utspenning, eller et solcellepanel med varierende utspenning

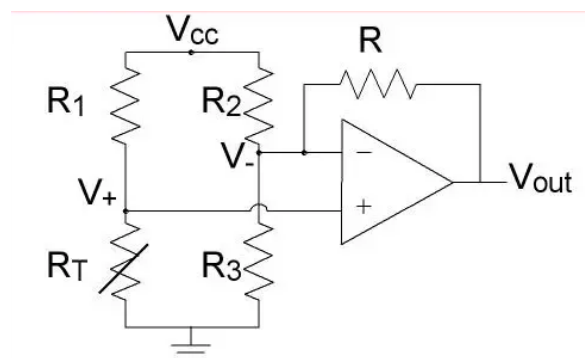
innenfor spenningsområdet 3-15 volt. I vårt tilfelle er hovedhensikten å gi kretsen tilførsel via solcellepanelet vi har tatt i bruk, men også via disse andre spenningskildene som potensielt kan være aktuelle i fremtiden, avhengig av bruksområdet.

Spenningen, enten fra en innkilde via det uregulerte avgrensningen, fra en 7.5 volt innkilde, eller batteriet, blir deretter regulert ned til 5 volt av LM7805(Q1), som igjen blir nedregulert av den integrerte spenningsregulatoren i ESP32 modulen, til en spenning på 3.3 volt. Denne 3.3 volt spenningen blir brukt av tre spesifikke komponenter som er mer sensitive på spenning enn de andre. Disse mer sensitive komponentene er bevegelsessensoren (ICM20948), vindsensoren (Cotech FT0203) og dybdesensoren (MaxBotix MB7374-100). Disse tre komponentene forbruker 3.3 volt spenning av forskjellige grunner.

Bevegelsessensoren har denne spenningen oppgitt som en optimal spenning, men kan også operere på 5 volt. 3.3 volt ble valgt noe tilfeldig da det i tidlig stadiet av designet ikke hadde noe å si, før 5 volt planet ble introdusert i plantegningen. Vindsensoren er en sensor vi har tilpasset til vårt forbruk fra et eksisterende produkt, hvor vi etter rekonstruksjon og måling ikke klarte å fastslå spenning. Det ble derfor valgt å legge opp 3.3 volt som en mulig relativt sikker spenning. Etter litt mer intensiv testing viste det seg at denne sensoren fungerte ved hjelp av en magnetstyrt reed-kontakt. Dermed endret dette hvordan vi har implementert sensoren i systemet vårt, og istedenfor å anta at den krever et spenningspåtrykk, bruker vi denne som en normal knapp/bryter og registrerer digitalt antall endringer i status på sensoren ved bruk av en intern pull-up resistor på ESP32 modulen.

For å måle temperaturene både internt og eksternt, har vi brukt PT1000 temperaturfølere beskrevet i 3.4 sammen med LM2904DT beskrevet i 3.10. Disse blir koblet sammen i en komparatorkrets. Denne komparatorkretsen er bestående av en Wheatstone målebro beskrevet i 3.12, en op amp fra LM2904DT og en PT1000 i målebroen. Illustrert i figur 24 er oppkoblingen av en slik løsning. Denne fungerer ved at  $R_1$  og  $R_2$  i målebroen er like motstander,  $R_T$  er en PT1000 sensor og  $R_3$  er en fast motstand.  $R$  er satt til samme verdi som  $R_2$  slik at forsterkningen er lik en. Siden

PT1000 er 0 ohm ved  $0^{\circ}\text{C}$  og vi ønsket å måle i et område fra  $-40^{\circ}\text{C}$  til  $40^{\circ}\text{C}$ , valgte vi at  $R_3$  skulle være litt over 800 ohm (Sterling Sensors, 2019). Dette var for at broen skulle være i balanse ved  $-40^{\circ}\text{C}$ . Alle temperaturer over  $-40^{\circ}\text{C}$  vil gi en høyere



[25]: Oppkobling av Wheatstone målebro og komparator (Chien, u.d.)

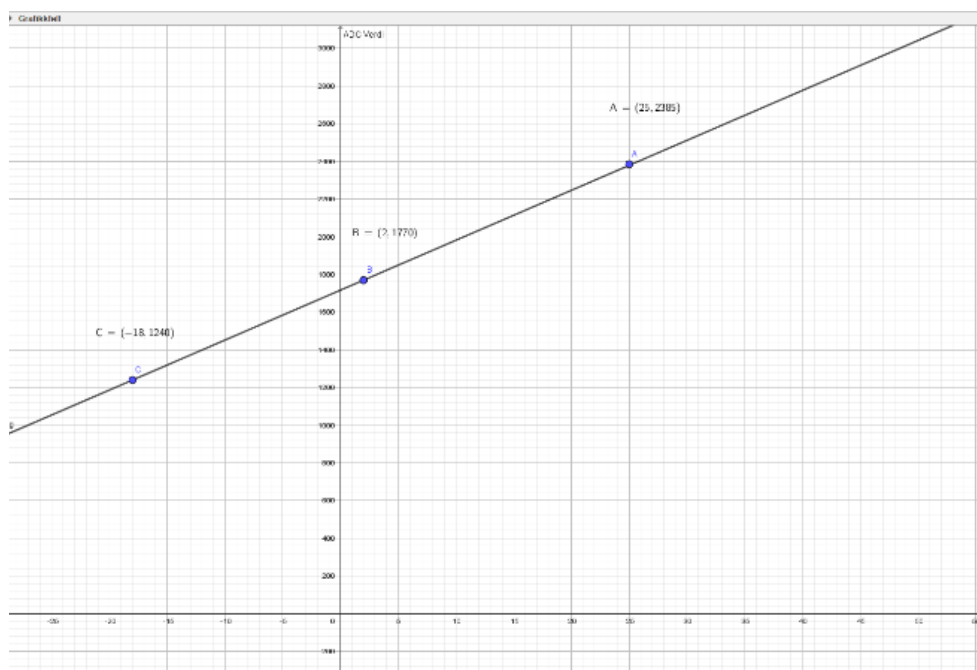
spenning på grenen PT1000 sensoren sitter og komparatoren vil forsterke forskjellen før signalet går inn på mikrokontrolleren (Ametherm, u.d.) (Electronics Tutorials, u.d.).

#### 4.7. Kalibrering av temperatursensorer

I dette systemet er det viktig at temperatursensorene er nøyaktige og pålitelige. Dette er fordi oppdragsgiver trenger nøyaktige målinger for å bestemme hva de skal strø med og om det er fare for is på veien. Kalibreringen spiller en viktig rolle i å sørge for at målingene er nøyaktige og pålitelige.

I dette systemet er det temperatursensoren som er den eneste sensoren som trengs å kalibreres. Denne sensoren gir et analogt signal som den integrerte ADC-en på ESP32 modulen leser av. Her får vi en verdi mellom 0 og 4095 da dette er en 12 bit ADC.

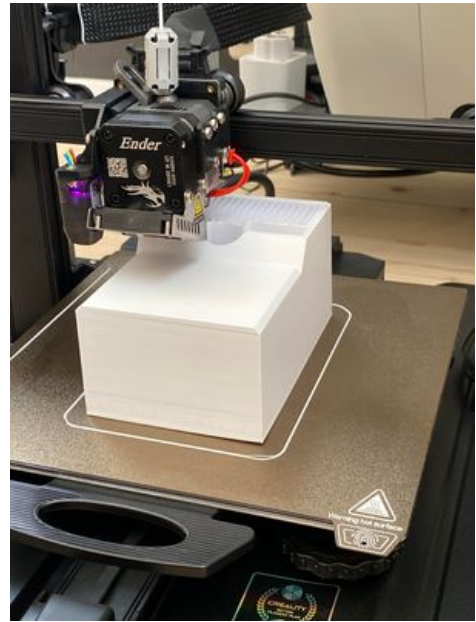
Metoden vi har valgt å bruke for kalibrering er å sammenligne mot en referanse som vi allerede kjente til. Vi valgte ut 3 forskjellige temperaturer å måle på, -18 (fryser), 2 (kjøleskap) og 24 (midt i et rom). Da sensoren hadde fått stå en stund leste vi av verdien og plottet dette i et koordinatsystem. Ved hjelp av disse tre punktene lagde vi en graf hvor vi fant funksjonsuttrykket  $-53x + 2y = 3434$  hvor x er temperatur og y er verdien ESP32 leste av. Med funksjonsuttrykket lagde vi en ligning som er blitt tatt i bruk i koden for at ESP32 modulen skal sende riktig temperatur.



[26]: Kalibrering av temperatursensor opp mot referansepunkter med tydelige temperaturer. plottet i Geogebra (Geogebra, u.d.)

#### 4.8. Innkapsling

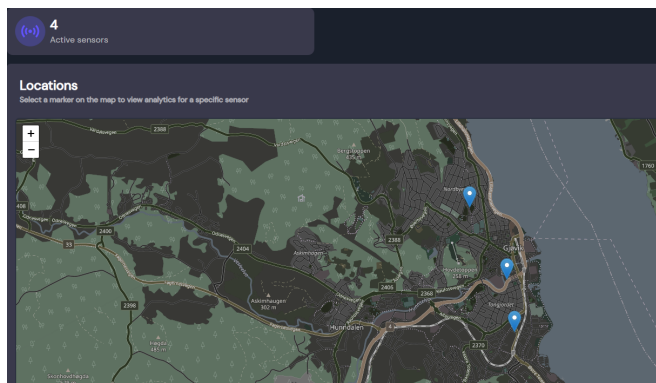
Innkapslingen valgt i oppgaven måtte oppfylle visse krav i forhold til klimaet produktet skulle operere i. Som vist til i planleggingen er det gunstig at sensoren har en viss vinkling på veien for optimal måling. Dette for å forenkle beregninger med vinkling til måleområdet. Innkapslingen er 3D-printet av Steffan Kjenslie, som viste interesse for prosjektet og bidro med sin kompetanse på 3D-printing. Størrelsen og utformingen på innkapslingen er konstruert etter funksjonaliteten til produktet. Kretskortet festes på innsiden, inn mot festeordningen. Materialet som er brukt, er en værfast kompositt, anvendelig for det klimaet og de utfordringene som kunne oppstå under bruk.



[27]: Innkapslingen for produktet under produksjon. Foto: Steffan Kjenslie

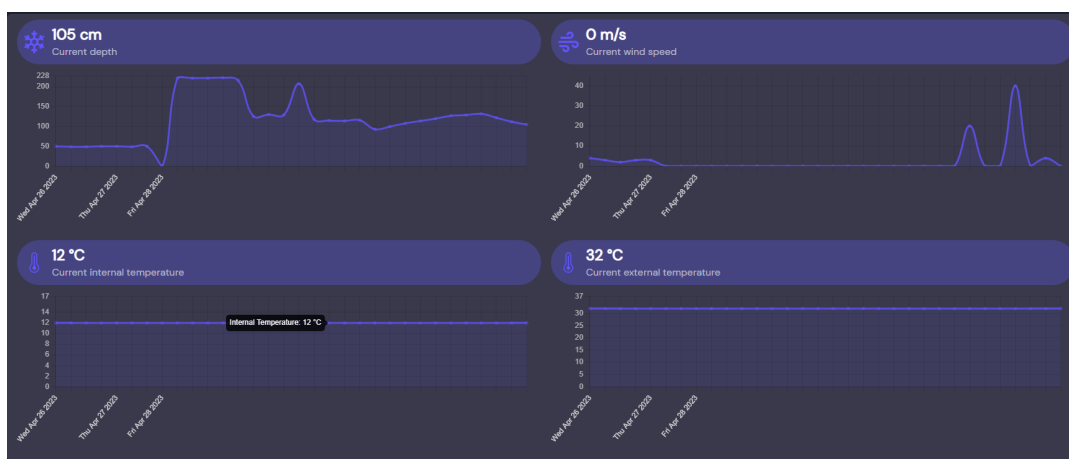
## 4.9. Grensesnitt

Fremvisningen av informasjon på et forståelig medium er viktig for at produktets funksjonalitet skal komme frem på en forståelig måte til brukerne. I denne oppgaven har vi tatt i bruk en nettside, designet og programmert av en ekstern aktør, nevnt i forord. Nettsiden er basert på HTML-kode. Nettsiden bruker grafer for å vise endring i både snømengde, vindstyrke, innvendig og utvendig temperatur over tid. Dette for å kunne spore tilbake hvor stort snøfall man har hatt over en periode og man kan forberede seg på hvor mye nyfallen snø som finnes. Her kan man også føre statistikk på hvor jevnt eller ujevnt klimaet har opptrådt over en periode.



[28]: Oversikt over alle sensorer utplassert over et visst område. Gir mulighet for å endre fokus fra sensor til sensor. Foto hentet fra selvdesignet grensesnitt

Under en annen mappe på nettsiden har man en oversikt over alle innretningene som er plassert ut, og hvor disse er lokalisert. Dette vil være mer relevant hvis man i ettertid bestemmer seg for å utvide og ekspandere løsningen. På nettsiden kan man legge til og fjerne sensorer og bestemme hvilken sensor som skal fremvise informasjonen sin.



[29]: Informasjonen fra sensoren fremvist under ene mappen på nettsiden. F.v. Total snødybde, vindstyrke, innvendig temperatur og utvendig temperatur. Foto hentet fra selvdesignet grafisk grensesnitt



## 4.10. Kode

```
// Velg hvilket modem som blir brukt
#define SIMPLE_NB_MODEM_SIM7080

// Enable DBG debug output to Serial Monitor for debug prints, if needed
#define SIMPLE_NB_DEBUG Serial

// Velg Serial port for AT interface
#define SerialAT Serial2

// inkluder SimpleNB bibliotek
#include <SimpleNBClient.h>
#include <WiFi.h>

#define BAUD_RATE 115200 // Baud rate til bruk for kommunikasjon med modemet

#define USE_SSL // Kommenter bort denne linjen hvis det ikke blir brukt SSL
#ifdef USE_SSL
const int port = 443;
#else
const int port = 80;
#endif

// Server konfigurasjon
const char host[] = "snodybde-sensor.vercel.app";
const char resourceInit[] = "/api/import/sensor/*****";
const char resourceRef[] = "/api/import/referance/*****";
const char resourceData[] = "/api/import/sensor/data/*****";

int initialiseringen = 1;

// Pin for sensor
#define SNO_PIN 5
#define VIND_PIN 25
#define TEMP_EXT_PIN 34
#define TEMP_INT_PIN 35

// Definer tidsintervall for avlesning i millisekunder
#define WRITE_INTERVAL 600000 // 10 minutter
#define READ_INTERVALSNO 30000 // 30 sekunder

// Variabler for snøsensoren
float pulsLengde = 0;
int filtrertVerdi = 0;
unsigned long forrigeAvlesningSno = 0;

// Definer filterkonstanter
const float alpha = 0.1;
const float beta = 1 - alpha;

// Definer variabler for rolling median filter
const int numMalinger = 20; // Antall målinger som skal inkluderes i filteret
int malinger[numMalinger]; // Array for lagring av målingene
int sorted[numMalinger]; // Array for sorterte målinger
int malingerIndex = 0; // Index for å legge til nye målinger i arrayet

// Definer konstant for maksimal avstand mellom avlesninger
const int maxDeviation = 600;
```

```

// Variabler for antall rotasjoner og tid siden siste avlesning
int rotasjoner = 0;
int hastighet = 0;
unsigned long lastReadTime = 0;
int rotasjonsTilstand = HIGH;
int forrigeRotasjonsTilstand = HIGH;

// Variabler for debounce-funksjonen
unsigned long debounceDelay = 50; // Debounce-tiden i millisekunder
unsigned long lastDebounceTime = 0;

// Variabel som holder styr på om vindsensoren er høy
bool roterer = false;

//variabler for temperatur sensorene
float TempExt = 0;
float TempInt = 0;

SimpleNB modem(SerialAT);

```

Koden starter med å definere hvilket modem som blir brukt.

`SIMPLE\_NB\_MODEM\_SIM7080`. Denne brukes til å kommunisere med serveren. Kommunikasjonen skjer via AT-grensesnittet på Serial2-porten. Deretter inkluderes det to biblioteker, `SimpleNB` som blir brukt til kommunikasjon med NB-IoT-modulen (Tech Studio Design, u.d.) og biblioteket `WiFi` som er et standard bibliotek som følger med Arduino IDE (*WiFi*, u.d.). Det er hentet kode fra `SimpleNB` biblioteket vi har brukt i funksjonene `initialisering()`, `referanse()` og `sendData()`. Koden er endret litt på for at det skal passe til vårt formål. Videre defineres serverkonfigurasjonen med servernavn og portnummer ut ifra om det blir brukt SSL. I tillegg defineres også ressursplasseringen. Har her valgt å sensurere nøkkelen man trenger for å sende dataene til riktig plassering av sikkerhetshensyn.

Videre defineres variabler for å håndtere avlesningen fra de forskjellige sensorene. Dette inkluderer tidsintervall for avlesning og sending av data, rotasjon, hastighet og debounce-tid.

Variablene `debounceDelay` kan endres slik at programmet ikke godtar avlesninger innenfor et større eller mindre tidsområde.

```

void vindMaling(){
  // Les av knappens tilstand
  rotasjonsTilstand = digitalRead(VIND_PIN);

  // Sjekk om det har gått lang nok tid siden forrige debounce
  if ((millis() - lastDebounceTime) > debounceDelay) {
    // Sjekk om vindsensorens tilstand har endret seg fra forrige avlesning
    if (rotasjonsTilstand != forrigeRotasjonsTilstand) {
      // Oppdater siste debounce-tidspunkt
      lastDebounceTime = millis();
    } else {
      // Sjekk om knappen er trykket ned
      if (rotasjonsTilstand == HIGH && !roterer) {
        rotasjoner++;
        DBG("Rotasjon registrent!");
        roterer = true;
      }
    }
  }
}

```

```

    // Sjekk om knappen er sluppet opp
    if (rotasjonsTilstand == LOW && roterer) {
        roterer = false;
    }
}
forrigeRotasjonsTilstand = rotasjonsTilstand;
}

```

Videre i koden er funksjonen `vindMaling()`. Denne funksjonen starter med å lese av tilstanden til vindsensoren ved hjelp av `digitalRead()` funksjonen.

Deretter sjekkes det om det har gått lang nok tid siden forrige debounce slik at vi kun får en måling per rotasjon av vindsensoren. Hvis det har gått lang nok tid sjekkes det om vindsensorens tilstand har endret seg fra forrige avlesning. Hvis denne tilstanden har endret seg oppdateres `lastDebounceTime` til nåværende tidspunkt. Hvis tilstanden ikke er endret sjekker den om sensoren er i en posisjon som er enten høy eller lav.

Hvis tilstanden til sensoren er høy vil antall rotasjoner økes og `roterer` endres til `true`. Hvis tilstanden til sensoren er lav vil `roterer` endres til `false`. Til slutt endres `forrigeRotasjonstilstand` til nåværende `rotasjonstilstand`.

```

void snoMaling(){
    // Les lengden på PWM-signalet
    unsigned long pulsLengde = pulseIn(SNO_PIN, HIGH);

    // Legg til målingen i arrayet hvis den ligger innenfor en definert avstand fra gjennomsnittet
    float avg = filtrertVerdi;
    if (abs(pulsLengde - avg) <= maxDeviation) {
        malinger[malingerIndex] = pulsLengde;
        malingerIndex = (malingerIndex + 1) % numMalinger;

        // Beregn gjennomsnittet av avlesningene
        int sum = 0;
        int count = 0;
        for (int i = 0; i < numMalinger; i++) {
            if (malinger[i] != 0) {
                sum += malinger[i];
                count++;
            }
        }
        avg = (float)sum / (float)count;
        Serial.print("gjennomsnittet:");
        Serial.println(avg);

        // Oppdater filtrertVerdi
        filtrertVerdi = alpha * avg + beta * filtrertVerdi;
        Serial.print("filtrertVerdi: ");
        Serial.println(filtrertVerdi);
    } else {
        // Hvis avlesningen er utenfor avstanden, kast den og ta en ny avlesning
        Serial.println("Ugyldig avlesning: " + String(pulsLengde));
        pulsLengde = pulseIn(SNO_PIN, HIGH);
    }

    // Skriv ut avlesningene for debugging
}

```

```

for (int i = 0; i < numMalinger; i++) {
    Serial.print("Reading ");
    Serial.print(i);
    Serial.print(": ");
    Serial.println(malinger[i]);
}
DBG(pulsLengde);
forrigeAvlesningSno = millis();
}

```

Etter dette kommer funksjonen `snoMaling()`. Denne funksjonen brukes for å måle avstanden fra sensoren til snøen. Her blir et PWM signal lest av på `SNO\_PIN`. Ved hjelp av funksjonen `pulseIn()` kan vi lese av hvor lenge dette PWM signalet er høyt. Deretter blir det sjekket om avlesningen er innenfor en definert avstand fra gjennomsnittet. Dette gjøres ved å sammenligne differansen mellom `pulsLengde` og `avg` med `maxDeviation`. Dette er for å ikke få med feilmalinger i tilfelle det kjører en bil eller det er andre gjenstander under som gjør at avstanden plutselig blir mye kortere. Hvis avlesningen er innenfor blir den lagt til i arrayet `malinger` på den gjeldende plasseringen `malingeIndex`, deretter oppdateres `malingeIndex` for å holde styr på plasseringen i arrayet. Videre blir gjennomsnittet av `malinger` arrayet regnet ut og blir lagret i variabelen `avg`. Deretter blir signalet filtrert ved hjelp av `alpha`, `avg`, `beta` og den tidligere `filtrertVerdi`. Dette filtrerte signalet lagres i variabelen `filtrertVerdi`. Hvis en av avlesningene ikke er innenfor den definerte maks avstanden fra gjennomsnittet blir den kastet og det blir tatt en ny avlesning. Deretter endres `forrigeAvlesningSno` til nåværende tidspunkt som blir brukt i `loop()`.

```

void vind(){
    // Skriv ut antall rotasjoner siden forrige avlesning
    DBG("Antall rotasjoner i siste periode: ");
    DBG(rotasjoner);
    float rotasjon = (rotasjoner/2);
    DBG(rotasjon);
    hastighet = ((rotasjon/(WRITE_INTERVAL))*1000)*40.84;
    DBG(hastighet);

    // Nullstill telleren
    rotasjoner = 0;
}

```

Deretter kommer funksjonen `vind()`. Her blir antall rotasjoner som er målt over en periode håndtert. Først blir antall rotasjoner dividert på to for å få faktisk antall rotasjoner. Deretter beregnes hastigheten ved å dividere antall faktiske rotasjoner med tidsintervallet for perioden. Videre multipliseres dette med 1000 for å få antall rotasjoner per sekund. Dette multipliseres så med omkretsen av vindskovlen for å beregne vindhastigheten, i m/s. Deretter nullstilles telleren for å starte en ny periode.

```

void tempMaling(){
    TempExt = -((3600-2*(analogRead(TEMP_EXT_PIN)))/53);
    TempInt = -((3600-2*(analogRead(TEMP_INT_PIN)))/53);
}

```

Etter dette kommer funksjonen `tempmaling()`. Denne funksjonen leser av den analoge verdien fra den eksterne og interne temperatur sensoren og justert med et uttrykk vi fikk etter kalibrering. Disse verdiene blir lagret i variablene `TempExt` og `TempInt`.

```
void initialisering(){
  SimpleNBClientSecure client(modem, 0);
  DBG("Connecting to", host, "port", port);
  if (!client.connect(host, port)) {
    DBG("... failed");
  } else {
    // Lag en http post request med TCP:
    client.println("POST " + String(resourceInit) + " HTTP/1.1");
    client.println("Host: " + String(host));
    client.println("Content-Type: application/json");
    client.println("Connection: close");
    client.print("Content-Length: ");
    String payload = "{\"id\":\"SENSOR_FERDIG\", \"name\": \"Olaf Kynsveens veg\"}";
    client.println(payload.length());
    client.println();
    client.println(payload);

    // Vent på respons fra serveren
    uint32_t start = millis();
    while (client.connected() && !client.available() && millis() - start < 30000L) {
      delay(100);
    };

    // Les av responsen fra serveren
    start = millis();
    char received[680] = {'\0'}; // adjust the length accordingly as per your payload size
    int read_chars = 0;
    while (client.connected() && millis() - start < 10000L) {
      while (client.available()) {
        received[read_chars] = client.read();
        received[read_chars + 1] = '\0';
        read_chars++;
        start = millis();
      }
    }
    DBG(received);
    DBG("Data received:", strlen(received), "characters");
    client.stop();
    initialisering = 0;
  }
}
```

Etter dette kommer funksjonen `initialisering()`. Her starter funksjonen ved å opprette en sikker NB-IoT klient som kan kommunisere med en server over HTTPS-protokollen. Her prøver den å koble seg til `host` og `port` som er definert tidligere i koden. Hvis denne tilkoblingen er vellykket vil den sende en HTTP POST-forespørsel som inneholder data i JSON-format. Dataen som sendes er `id` på sensoren som blir brukt på serversiden for å skille data fra hverandre hvis man utvider systemet slik at det blir utplassert flere løsninger.

Det sendes også `name` som serveren viser frem på grensesnittet for at det blir lettere for brukeren å vite hvor den er utplassert.

Når dataen er sendt venter den på en respons fra serveren og hvis den får et svar tilbake vil programmet lese dataen og skrive dette til serieporten. Til slutt stoppes klienten ved hjelp av `client.stop()` funksjonen og variabelen `initialiseringen` settes til 0 for å indikere at initialiseringen er fullført.

```
void referanse(){
  SimpleNBClientSecure client(modem, 0);
  DBG("Connecting to", host, "port", port);
  if (!client.connect(host, port)) {
    DBG("... failed");
  } else {
    // Lag en http post request med TCP:
    client.println("POST " + String(resourceRef) + " HTTP/1.1");
    client.println("Host: " + String(host));
    client.println("Content-Type: application/json");
    client.println("Connection: close");
    client.print("Content-Length: ");
    String payload = "{\"sensor_id\": \"SENSOR_FERDIG\", \"referance_point\": \" + String(filtrertVerdi) + "\"";
    client.println(payload.length());
    client.println();
    client.println(payload);

    // Vent på respons fra serveren
    uint32_t start = millis();
    while (client.connected() && !client.available() && millis() - start < 3000L) {
      delay(100);
    };

    // Les av responsen fra serveren
    start = millis();
    char received[680] = {'\0'}; // adjust the length accordingly as per your payload size
    int read_chars = 0;
    while (client.connected() && millis() - start < 1000L) {
      while (client.available()) {
        received[read_chars] = client.read();
        received[read_chars + 1] = '\0';
        read_chars++;
        start = millis();
      }
    }
    DBG(received);
    DBG("Data received:", strlen(received), "characters");
    client.stop();
    initialiseringen = 0;
  }
}
```

Videre kommer funksjonen `referanse()`. Denne funksjonen gjør det samme som initialisering, bare at her er dataene som sendes blitt endret. Her sendes `sensor\_id` som er ideen vi registrerte i initialiseringen og `referance\_point` som er referansepunktet som blir brukt for måling av snødybde.

```
void sendData(){
```

```

SimpleNBClientSecure client(modem, 0);
if (!client.connect(host, port)) {
    DBG("... failed");
} else {
    // Lag en http post request med TCP:
    client.println("POST " + String(resourceData) + " HTTP/1.1");
    client.println("Host: " + String(host));
    client.println("Content-Type: application/json");
    client.println("Connection: close");
    client.print("Content-Length: ");
    String payload = "{\"sensor_id\": \"SENSOR_FERDIG\", \"
                    \"current_depth\": " + String(filtrertVerdi) + \", \"
                    \"internal_temperature\": " + String(TempInt) + \", \"
                    \"external_temperature\": " + String(TempExt) + \", \"
                    \"wind_speed\": " + String(hastighet) +
                    \"}\";

    client.println(payload.length());
    client.println();
    client.println(payload);

    // Vent på respons fra serveren
    uint32_t start = millis();
    while (client.connected() && !client.available() && millis() - start < 30000L) {
        delay(100);
    };

    // Les av responsen fra serveren
    start = millis();
    char received[680] = {'\0'};
    int read_chars = 0;
    while (client.connected() && millis() - start < 10000L) {
        while (client.available()) {
            received[read_chars] = client.read();
            received[read_chars + 1] = '\0';
            read_chars++;
            start = millis();
        }
    }
    DBG(received);
    DBG("Data received:", strlen(received), "characters");

    client.stop();
}

DBG("Deactivate Data Network... ");
if (!modem.deactivateDataNetwork()) {
    delay(1000);
    return;
}

DBG("Powering Down...");
//skru av modulen
modem.powerOff();
DBG("Done!\n\n");

lastReadTime = millis();
}

```

Deretter kommer funksjonen `senddata()`. Denne funksjonen fungerer ganske likt som `initialisering()` og `referanse()`. Her har dataen som sendes blitt endret. Her sendes `sensor\_id` som er ideen vi registrerte i initialiseringen. `current\_depth` som er snødybden vi

målte i funksjonen `snoMaling()`. `internal\_temperature` som er den interne temperaturen i kapslingen som vi målte i funksjonen `tempMaling()`. `external\_temperature` som er temperaturen utenfor kapslingen målt i funksjonen `tempMaling()`. Det siste som sendes er `wind\_speed` som er hastigheten vi beregnet i funksjonen `vind()`. Denne funksjonen venter også på respons og stopper klienten når den er ferdig.

Videre brukes en funksjon kalt `deactivateDataNetwork()` fra `modem` objektet. Denne funksjonen sender en AT-kommando til modemmet for å avslutte mobildataforbindelsen. Når dette er gjort bruker den funksjonen `powerOff()` fra modem-objektet for å slå av selve modulen for å spare strøm når denne ikke er i bruk. Deretter oppdateres `LastReadTime` for å starte en ny periode for vindmåleren.

```
void setup() {
  Serial.begin(115200);
  pinMode(VIND_PIN, INPUT_PULLUP);
  WiFi.mode(WIFI_OFF);

  // Fyll opp malinger-arrayet med de første 30 verdiene
  for (int i = 0; i < numMalinger; i++) {
    malinger[i] = pulseIn(SNO_PIN, HIGH);
    filtrertVerdi += malinger[i];
    delay(10);
  }
  filtrertVerdi /= numMalinger;
}
```

Videre kommer funksjonen `setup()` her settes porten som brukes til vind sensoren som en intern pullup ved hjelp av `pinMode` funksjonen. Deretter skrus WiFi modulen av ved hjelp av `WiFi.mode(WIFI\_OFF)`. Dette er for at ESP32 skal bruke mindre strøm. Etter dette fylles `malinger` arrayet opp med 20 målinger

```
void loop() {
  vindMaling();

  // Sjekk om det har gått minst 1 minutter siden sist avlesning
  unsigned long currentTimesno = millis();
  if (currentTimesno - forrigeAvlesningSno >= READ_INTERVALSNO){
    snoMaling();
  }

  // Sjekk om det har gått minst 5 minutter siden sist avlesning
  unsigned long currentTime = millis();
  if (currentTime - lastReadTime >= WRITE_INTERVAL) {
    vind();
    tempMaling();

    // Setter modemets baudrate
    SimpleNBBegin(SerialAT, BAUD_RATE);

    DBG("Initializing modem...");
    modem.init();
  }
}
```



```

DBG("Waiting for network registration...");
if (!modem.waitForRegistration(60000L, true)) {
    delay(1000);
    return;
}

DBG("Activate Data Network...");
if (!modem.activateDataNetwork()) {
    delay(1000);
    return;
}

if (initialiseringen == 1) {
    initialisering();
    referanse();
}
sendData();
}
}

```

I `loop()` funksjonen kaller den de andre funksjonene som allerede er blitt forklart. Den starter med å kalle på `vindmaling()` funksjonen. Deretter sjekker den om det har gått en viss tid som er definert med `READ\_INTERVALSNO`. Hvis denne tiden har gått vil den kalle på funksjonen `snoMaling()`. Deretter sjekker den om det har gått en viss tid som er definert med `READ\_INTERVAL`. Hvis denne tiden har gått vil den kalle på funksjonen `vind()`, den vil deretter koble til mobilnettverket ved å kalle på funksjonene `modem.init()`, `modem.waitForRegistration()` og `modem.activateDataNetwork()`. Etter dette sjekker den om en initialisering er fullført. Er den ikke fullført kaller den på funksjonen `initialisering()`. Til slutt kaller den på funksjonen `senddata()` som sender dataen de forskjellige sensorene har samlet inn til serveren via mobilnettverket.

## 5. Resultater

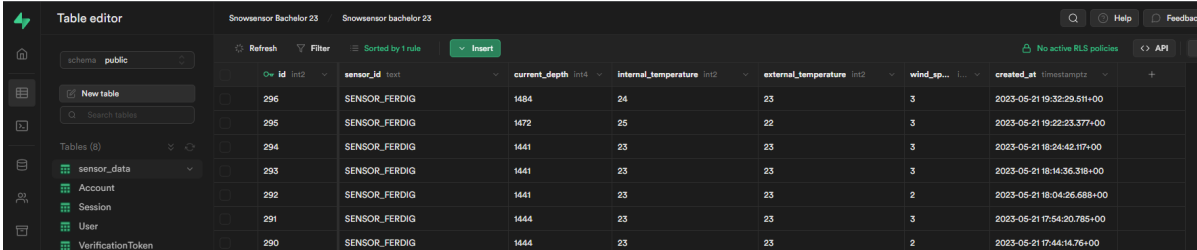
### 5.1. Kostnader og Budsjett for løsningen

Arbeidsgiver hadde ønske om lave kostnader og kostnadseffektive løsninger for produktet, men en løsning med de funksjonene i oppgaven. For å kunne konkurrere på markedet, må produktet være robust og stabilt nok til å kunne stoles på, ved krevende omgivelser.

Kostnadene for utvikling og produksjon av dette produktet, endte totalt på ca. 9600 kroner, noe som holder seg til 400 kroner under det budsjettet satt av arbeidsgiver. Dette er en førstegangskostnad for utvikling og design av produkt, som vil si at ved en eventuell senere linjeproduksjon vil produktet ende på en lavere kostnad. Totale kostnader og kostnader spesielt for design kan ses i vedlegg, budsjett for prosjektoppgaven.

### 5.2. Funksjonalitet og Effektivitet

Produktet har oppfylt de kravene som arbeidsgiver kom med i starten av perioden, og funksjonaliteten som vi ble enig om. Produktet kan måle snødybde, måle temperatur eksternt og internt og måle vindhastighet på en kostnads- og strømeffektiv måte. Løsningen forsynes av et solcellepanel, med et tilkoblet batteri som holder på overskuddsstrøm for å tilføre



id	sensor_id	current_depth	internal_temperature	external_temperature	wind_sp...	created_at
296	SENSOR_FERDIG	1484	24	23	3	2023-05-21 19:32:29.511+00
295	SENSOR_FERDIG	1472	25	22	3	2023-05-21 19:22:23.377+00
294	SENSOR_FERDIG	1441	23	23	3	2023-05-21 18:24:42.117+00
293	SENSOR_FERDIG	1441	23	23	3	2023-05-21 18:14:36.318+00
292	SENSOR_FERDIG	1441	23	23	2	2023-05-21 18:04:26.688+00
291	SENSOR_FERDIG	1444	23	23	3	2023-05-21 17:54:20.785+00
290	SENSOR_FERDIG	1444	23	23	2	2023-05-21 17:44:14.76+00

[30]: Tabell oversikt over målinger sendt til server. Utklipp hentet fra supabase server (Supabase, u.d.)

kretsen strøm på natten og ved mangel på tilstrekkelig sollys. I figur 30 kan man se informasjonen som sendes direkte fra sensoren, videre til grensesnittet hvor det blir behandlet. Her ser man en direkte oversikt over informasjon som sendes kontinuerlig.

Sensoren viser hvilken dybde sensoren måler for øyeblikket, den interne og den eksterne temperaturen, og vindhastigheten målt av vindsensoren. Dette er informasjonen som fremvises på grensesnittet.

Temperaturmålingene som er sendt er nøyaktige i forhold til målinger vi selv tar med andre pålitelige temperatur sensorer. Målingene er derfor trygge nok til det formålet disse har.

Temperatur sensoren bruker litt tid på å reagere på temperaturendringer, men denne skal være utplassert ute hvor temperaturendringene ikke er alt for raske.

Avstandssensoren er testet opp mot en pålitelig avstandsmåler hvor sensoren vår ga veldig like målinger. Denne sensoren er ganske treg på endringer, men da høyden på snøen ikke endrer seg så raskt vil denne sensoren passe bra.

Vindsensoren er testet opp mot værmeldingen på yr (Meteorologisk institutt & NRK, u.d.)

sensoren ble plassert ut og målte vindhastigheten.

Denne sensoren målte samme hastighet som yr hadde på sine nettsider. Anerkjenner at yr ikke er helt nøyaktig på alle områder i landet, så dette kan gjøre at vår kalibrering av vindsensoren heller ikke er helt nøyaktig.

I figur 31 kan man se det endelige kretskortet,

plassert i innkapslingen. ESP32 modulen er

loddet på kretskortet, og indikerer her at den er i

aktiv modus med det røde lyset. Ultralydsensoren er påkoblet innkapslingen i høyre hjørne og kretskortet med en CAT6 kabel ca. midt på kortet. Kabelen koblet til RJ45-pluggen, nærmest

mikrokontrolleren, fører til vind- og

temperatursensorene. På figur 32 og 33 kan man se den

lukkede kapslingen med kretsen inni. Figur 33 viser

også at kretsen fungerer pga. det røde lyset som lyser ut fra inngangen bak.



[31]: Endelig kretskort plassert i innkapslingen. Foto: Jørgen Johannessen



[32]: Lukket kapsling med snødydesensor foran. Foto: Jørgen Johannessen



[33]: Lukket kapsling bakfra. Strømforsyning innkoblet. Foto: Jørgen Johannessen

## 6. Diskusjon

### 6.1. Problemer underveis

Et problem som har vært reelt i tidligere prosjekter, som også viste seg ved gjennomføring av dette prosjektet, har vært leveringstid på komponenter. Dette på grunn av stor etterspørsel. Vi har sett oss nødt til å se etter alternative løsninger, samtidig som vi stadig har ventet på komponenter og jobbet med andre deler av prosjektet. Man kan også spekulere i om eksterne faktorer som uenighet og konflikter mellom nasjoner i denne perioden har påvirket leveringstid og ressurstilgang for leverandører av de komponentene som ble nødvendig for produktet. Forsinkelser av komponentene har, som diskuteres i kapittel 4.2. *Fremdriftsplan*, vært bidragsgivende for forsinkelser, også i andre arbeidsoppgaver. Ved en eventuell senere involvering med den leverandøren det angår, vil det være vesentlig at vi tar større hensyn til denne leveringstiden og kanskje også ser etter andre aktører som kan levere tidligere. Ved levering av ultralydsensoren i denne oppgaven, var det ingen andre alternativer på leverandør, da ingen hadde denne inne til et tidspunkt aktuelt for prosjektperioden.

Ved avgjørelse av hvilken trådløs kommunikasjon som skal brukes mellom produkt og server, ble det i utgangspunktet avgjort at produktet skulle basere seg på en 3G modul. Da dette var kostnadseffektivt og lite vedlikehold var nødvendig. Etter nærmere undersøkelser fant vi ut at 3G nettet ikke ble aktuelt, og vi gikk heller inn for en NB-IoT modul som baserte seg på 4G og 5G mobilnettet. Dette fordi vi fant ut at 3G nettet ikke ville være anvendelig i en periode frem i tid hvor Gjøvik kommune fortsatt ville ta i bruk produktet. NB-IoT-modulen ville kunne tas i bruk i en lang periode fremover, da den kan fornye seg samtidig med mobilnettet. NB-IoT-modulen inneholder en sim med en lav månedlig kostnad for brukeren. Denne er også da kostnadseffektiv og viste seg mer aktuell.

I kapittel 2.5. *risikovurdering*, blir det diskutert risikofaktorer som kan ha en påvirkning for hvorvidt produktet blir suksessfullt og gjennomførbart. Flere av disse faktorene avhenger av egenskaper og verdier til de komponentene valgt, og hvordan de presterer i varierende klima. På grunn av velvalgte komponenter og nøye utlesning på egenskaper, viste ikke disse risikofaktorene seg å bli reelle. Eksterne faktorer som kunne påvirke tidsbruken på enkelte oppgaver derimot, ble et problem, som beskrevet før i dette kapitlet. Leveringstid på komponenter gjorde at deler av fremdriftsplanen ble forskjøvet. Gruppemedlemmene holdt seg relativt friske og dette hadde ingen vesentlig innvirkning på gjennomførbarheten til prosjektet.

Ved ende av prosjektet, viste det seg at det oppsto problemer med levering av den 3D-printede innkapslingen, som gjorde at ved innlevering av selve produktet til veileder, måtte vi bruke den prosjekterte innkapslingen. Denne innkapslingen er i seg selv like funksjonabel som den endelige løsningen, da den har en tetthetsgrad på IP67, er robust og vannsikker. Likevel hadde den ikke helt det designet som vi ønsket i utgangspunktet. Ved fremvisning av produktet, for veiledere, medstudenter og arbeidsgivere, blir produktet allikevel ferdigstilt.

Boost-konverteren, LM2621, som regulerer innsignalet fra solcellepanelet, har også vist seg å by på problemer. Vi mistenker at unøyaktigheter i komponentverdier og valg av komponenter, samt en veldig simplistisk kobling, har ført til at konverteren ikke presterer på den måten som er helt ønskelig. Vi mistenker at spolen i kretsen kan være synderen, pga. litt ugunstige egenskaper og muligens noen karakteristiske feil sammen med LM2621. Denne konverteren ble valgt over andre integrerte kretser pga. sin kostnadseffektivitet og tilgjengelighet, men det bør tas til ettertanke at denne kanskje ikke var helt optimal.

## 6.2. Krav og ønsker fra arbeidsgiver

Produktet har levert til det anslåtte budsjettet som arbeidsgiver kom med i starten av perioden. Hvis man skal sammenligne vårt produkt opp mot det andre konkurrenter har å tilby, er kostnadene relativt lavere, som det ble diskutert i 2.3. Markedsanalyse. Tatt i betraktning at vi ikke har noen utgifter på bemanning og andre lisensierte produkter per nå, er dette et rimeligere alternativ. Selvfølgelig vil ikke dette være fullstendig tilfelle ved en senere anledning utenfor bacheloroppgaven, men fra arbeidsgivers ståsted i dette øyeblikket. Vi vil også muligens kunne oppleve en stordriftsfordel ved produksjon av en større mengde produkt, enten ved avtaler med leverandører eller tilbud på bestilling av flere komponenter om gangen.

Produktet svarer til de funksjonene som arbeidsgiver ønsket i utgangspunktet, og vi har ikke måtte gi avkall på noe funksjonalitet, på grunn av økonomi eller gjennomførbarhet, foruten ønsket om et rent appbasert grensesnitt. I starten av perioden vil arbeidsgiver ha 5-6 sensorer, som de kunne plassere ut langs vegnettet i kommunen, men som det ble diskutert ble vi enige om å produsere en enhet som de kunne teste og eventuelt ta i bruk. Vi tror og håper at med den funksjonaliteten vi har kommet frem til på dette punktet, vil kommunen ta dette i bruk.

### 6.3. Forbedringer

Ved ferdigstilling av kretskortet, ble det oppdaget at det har blitt foretatt en endring i design for ESP32 modulene, som vil si at to pins som er nødvendig for prosjektet, ikke fungerte. Dette var henholdsvis pin 16 (rx) og pin 17 (tx)(ESP32.net, u.d.), som er ansvarlig for tilkoblingen mellom ESP32 og NB-IoT-modulen.

Dette ble oppdaget i ettertid av lodding, som gjorde at vi måtte ta denne av kretskortet og lodde på et eldre ESP32 modul som hadde de riktige pinsene. Som man ser i figur 28 har den nye modulen blitt fjernet og en eldre modul skal her loddes på. For å

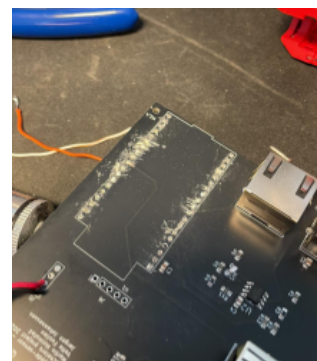
unngå dette problemet kunne vi montert ESP32 modulen med en sokkel, som ville gjort det lettere å avmontere en ESP32 ved feil, hvis modulen skulle modifiseres eller byttes ut.

(ESP32.net, u.d.)

Ledningsbanene på kretskortet fra bevegelsessensoren, beskrevet i 3.11. ICM-20948, kobler til pins på ESP32 som ikke kan brukes eller eksisterer for dette formålet. Vi måtte ta et valg om det var verdt tiden og pengene det ville ta å få den i orden, noe det ikke var.

Bevegelsessensoren var i utgangspunktet med for å lettere kunne regne ut vinkel og avstand fra sensoren til overflaten den skulle måle. Siden innkapslingen er laget med en standardvinkel, er det i utgangspunktet ikke behov for bevegelsessensoren, da sensoren sender et referansesignal en viss tid etter strømtilkobling, som viser avstand uten noe snøfall. Siden sensoren er kalibrert til en feilmargin som er akseptabel, kan vi da stole på at sensoren får riktige målinger. Til en annen gang kan man eventuelt finne en løsning på hvordan ledningsbanene kan kobles til ESP32.

Under testing og ferdigstilling, skjedde det flere ganger at vi fikk en såkalt “brownout error”. Denne oppstår kun når en av sensorene ikke er tilkoblet under oppstart, men blir plugget i etter at ESP32 modulen startet programmet. Dette kan være på grunn av dårlig mekanisk kontakt i blant annet RJ45 kontaktene og Wago klemmene brukt til tilkobling av snø sensoren, og andre mekaniske tilkoblinger, og skaper problemer for ESP32 modulen, som da skrur seg av og på. Dette, sammen med at vi gjerne skulle hatt en måte å kunne lagre modulen sammensatt, har vi konkludert med at vi burde hatt en integrert av/på bryter på kretskortet, for å kunne koble batteriet fra kretsen uten å måtte ta det ut av kontakten. Dette er antageligvis ikke nødvendig i en potensiell masseproduksjon av produktet, men for testing og utvikling har vi opplevd at dette kunne vært til stor hjelp. (StackOverflow, 2020)



[34]: Den nye ESP32 modulen fjernet fra kretskortet, i påvente av en eldre ESP32 med riktige pins  
Foto: Jørgen Johannessen

## 7. Konklusjon

Distrikt-Norge er i dag i en slik situasjon at det kreves en stor digitalisering av funksjonene som i dag utføres av manuelt arbeid. Rapportering av behov for vedlikehold og oppfølging på veinettet i Norge krever automatiserte funksjoner som kontinuerlig kan følge med og rapportere til de riktige etatene. For at kommunene skal slippe å måtte sende ut mannskap til alle strekninger, spesielt om vinteren, er det essensielt å få på plass et produkt som kan utføre denne oppgaven, og dermed avlaste behovet for å ha ansatte på jobb hele tiden. Da kan de bli varslet ved behov og kunne rykke ut.

Markedet for denne type produkt er relativt lite oppfylt. De produktene som er tilgjengelig på markedet, medfører ganske høye kostnader for brukeren, som da igjen ikke gjør investeringen like fordelaktig. Ved å kunne levere et produkt til en lavere kostnad som kan prestere på et nivå tilfredsstillende for brukeren, kan etterspørselen bli potensielt høy.

Produktet vi har konstruert, medfører en relativt lav kostnad, sammenlignet med de andre produktene tilgjengelig på markedet. Med relativt lave utgifter vil det være en potensielt høy markedsverdi. Med tanke på hvilket potensial dette produktet kan ha på et marked hvor etterspørselen er høyere enn forsyningen, er dette et produkt som gruppen har tiltro til og kommer til å fortsette å arbeide med i ettertid av prosjektet.

## 8. Litteraturliste

Arbeidstilsynet (u.d.) *Risikovurdering - HMS*, Hentet 17. April, 2023, fra

<https://www.arbeidstilsynet.no/hms/risikovurdering/>

Ametherm. (n.d.). *NTC Thermistors - Temperature Measurement With A Wheatstone Bridge*.

Ametherm. Hentet 14. Mai, 2023, fra

<https://www.ametherm.com/thermistor/ntc-thermistors-temperature-measurement-with-wheatstone-bridge>

Blumlein, A. (u.d.). *Wheatstone Bridge*. Wikipedia. Hentet 18. Mai, 2023, fra

[https://en.wikipedia.org/wiki/Wheatstone\\_bridge](https://en.wikipedia.org/wiki/Wheatstone_bridge)

Clasohlson. (2022, 23. Februar). *FT0203 sensor for vindhastighet og vindretning*.

Hentet 8. Mai, 2023, fra

<https://www.clasohlson.com/no/Cotech-sensor-for-vindhastighet-og-vindretning,-FT0203>

Chien, L. (u.d.). *We had a circuit that includes wheatstone bridge with a single strain gauge and a lm741 op-amp. How can we increase the gain of op-amp?* Quora. Hentet 10. Mai, 2023, fra

<https://www.quora.com/We-had-a-circuit-that-includes-wheatstone-bridge-with-a-single-strain-gauge-and-a-lm741-op-amp-How-can-we-increase-the-gain-of-op-amp>

DigiKey. (u.d.). *MB7374-100 MaxBotix Inc*. Hentet 25. April, 2023, fra

<https://www.digikey.com/en/products/detail/maxbotix-inc/MB7374-100/7896824>

Electronics Tutorials. (u.d.). *Op-amp Comparator and the Op-amp Comparator Circuit*.

Electronics Tutorials. Hentet 13. Mai, 2023, fra

<https://www.electronics-tutorials.ws/opamp/op-amp-comparator.html>

Elfa Distrelec. (u.d.a.). *Industriell lidar sensor* hentet 9. Mai, 2023, fra

<https://www.elfadistrelec.no/no/industriell-lidar-sensor>

Elfa Distrelec. (u.d.b.). *HY-DI Oppladbar batteripakke*. Hentet 16. Mai, 2023, fra

<https://www.elfadistrelec.no/no/hy-di-oppladbar-batteripakke-sm-buss-li-ion-2v-67ah-hy-line->



[hy-di-2s2p-s1/p/30289205](https://www.elfadistrelec.no/Web/Downloads/_t/ds/U128_eng_tds.pdf)

Elfa Distrelec. (u.d.c). *SIM7080G*. Hentet 19. April, 2023, fra

[https://www.elfadistrelec.no/Web/Downloads/\\_t/ds/U128\\_eng\\_tds.pdf](https://www.elfadistrelec.no/Web/Downloads/_t/ds/U128_eng_tds.pdf)

Elfa Distrelec. (u.d.d). *M5Stack SIM7080G*. Elfa Distrelec AS. Hentet 27. April, 2023, fra

<https://www.elfadistrelec.no/no/sim7080g-cat-og-nb-iot-kommunikasjonsenhet-m5stack-u128/p/30230764>

Energy Systems. (u.d.). *2SHY-Di Smart Battery Packages* . Hentet 16. Mai, 2023, fra

[https://www.elfadistrelec.no/Web/Downloads/\\_t/ds/2S%20HY-Di\\_eng\\_tds.pdf](https://www.elfadistrelec.no/Web/Downloads/_t/ds/2S%20HY-Di_eng_tds.pdf)

ESP32.net. (u.d.). *ESP32*. Hentet 18. April, 2023, fra

<http://esp32.net/>

Ewald, W. (2021, 14. August). *ICM-20948-9-Axis Sensor Part I*. Wolles Elektronikkiste. Hentet 2. Mai, 2023, fra

<https://wolles-elektronikkiste.de/en/icm-20948-9-axis-sensor-part-i>

EzViz. (u.d.a). *EZVIZ Weatherproof Solar Charging Panel*. Hentet 26. April, 2023, fra

<https://www.ezviz.com/uk/product/solar-charging-panel/40219>

EzViz. (u.d.b). *Solar Charging Panel*. Hentet 16. Mai, 2023, fra

[https://mfs.ezvizlife.com/solar%20panel\\_datasheet.pdf](https://mfs.ezvizlife.com/solar%20panel_datasheet.pdf)

Geogebra. (u.d.). *GeoGebra*. Hentet 10. Mai, 2023, fra

<https://www.geogebra.org/>

Github, (u.d.) *WiFi bibliotek for Arduino*, Hentet 26. April, 2023, fra

<https://github.com/arduino-libraries/WiFi>

Haaland, N. H. (2020, 10. Mars). *Hvilke typer solcelle-teknologier passer for meg?* Hentet 16. Mai, 2023, fra

<https://blogg.fusen.no/alle/ulike-typer-solcelleteknologi>

Hy-Line. (u.d.). *HY-Di-2S2P-SI*. HY-LINE. Hentet 26. April, 2023, fra

<https://www.hy-line-group.com/de-en/products/energy-storage/hy-di-smart-battery-system/hy-di-2s2p-s1~p10328>

KiCad. (u.d.). *KiCad*. Hentet 5. Mai, 2023, fra

<https://www.kicad.org/>

Last\_Mile. (u.d.). *4G/5G og NB-IoT*. Hentet 19. April, 2023, fra

[https://www.lastmile.no/teknologi/teknologibeskrivelser/4g\\_5g-og-nb-iot](https://www.lastmile.no/teknologi/teknologibeskrivelser/4g_5g-og-nb-iot)

Manua.ls. (u.d.) *Cotech FT0203 Manual*, Hentet 8. Mai, 2023, fra

<https://www.manua.ls/cotech/ft0203/manual?p=35>

MaxBotix. (u.d.). *HRXL-MaxSonar-WRS Datasheet – MaxBotix*. Hentet 17. April, 2023, fra

<https://maxbotix.com/pages/hrxl-maxsonar-wrs-datasheet>

MaxSievert. (u.d.). *Hva er forskjellen mellom Pt100, Pt200 Pt500 og Pt1000?* Hentet 20. Mai, 2023,

fra <https://maxsievert.no/forskjellen-mellom-pt100-pt500-pt1000-element/>

Meteorologisk institutt & NRK. (u.d.). *Gjøvik - Langtidsvarsel*. Yr. Hentet 20. Mai, 2023, fra

<https://www.yr.no/nb/v%C3%A6rvarsel/daglig-tabell/1-130075/Norge/Innlandet/Gj%C3%B8vik/Gj%C3%B8vik>

Peak Sensors. (u.d.). *What Is A PT1000 Sensor? (Full Definition & Explanation)*. Hentet 19. April,

2023, fra <https://peaksensors.co.uk/what-is/pt1000-sensor/>

Rosencrance, L. (2020, april). *What is Narrowband IoT (NB-IoT)?* Hentet 19. April, 2023, fra

<https://www.techtarget.com/whatis/definition/narrowband-IoT-NB-IoT>

RS Components. (2018, 5. Mai). *GP2Y0A710K0F - Reflektiv sensor*, Hentet 9. Mai, 2023, fra

<https://no.rs-online.com/web/p/reflective-optical-sensors/6666577>

RS. (u.d.). *RS PRO PT1000*. RS Components. Hentet 26. April, 2023, fra

<https://no.rs-online.com/web/p/rtd-sensors/3342616>

Siepert, B. (u.d.). *Adafruit TDK InvenSense ICM-20948*. Adafruit Learning System. Hentet 28. April,

2023, fra <https://learn.adafruit.com/adafruit-tdk-invensense-icm-20948-9-dof-imu/overview>

Surplustronics. (u.d.). *ESP32 Development Board*. Surplustronics. Hentet 27. April, 2023, fra

<https://surplustronics.co.nz/products/9368-esp32-development-board-38-pin-espwroom-32s>

Shawn (2019, 23. Desember). *Types of Distance Sensors and How to Select One?* Hentet 9. Mai,

2023, fra

<https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/>

- Seedstudio, (2019, October 30) *LM7805 Voltage Regulator*. Hentet 27. April, 2023, fra <https://www.seedstudio.com/blog/2019/10/30/lm7805-voltage-regulator-features-comparison-s-lm317-and-more/>
- Smarte Mjøsbyer. (2020, 4. November). *Betelo skal utvikle løsninger for Smart vinterveg - Smartemjøsbyer - Innovasjonspartnerskap*. Hentet 20. Mai, 2023, fra <https://www.smartemjosbyer.no/aktuelt/betelo-skal-utvikle-losninger-for-smart-vinterveg>
- StackOverflow. (2020, 11. Februar). *Any solution available for for ESP32-cam 'Brownout detector was triggered' error?* Stack Overflow. Hentet 21. Mai, 2023, fra <https://stackoverflow.com/questions/60171641/any-solution-available-for-for-esp32-cam-brownout-detector-was-triggered-error>
- Operasjonsforsterker*. (2022, 17. August). Wikipedia. Hentet 2. Mai, 2023, fra <https://no.wikipedia.org/wiki/Operasjonsforsterker>
- Operasjonsforsterker — Dataelektroniker*. (u.d.). Dataelektroniker. Hentet 26. April, 2023, fra <https://dataelektroniker.fagbokforlaget.no/129-operasjonsforsterker>
- Pettersen, O. (2003, 3. Mars). *Operasjonsforsterkeren*. Hentet 10. Mai, 2023, fra <https://folk.ntnu.no/oddptr/Elkretser/Opamp-1.pdf>
- STMicroelectronics. (2019, 7. Mai). *Low-power dual operational amplifier*. Hentet 21. Mai, 2023, fra <https://docs.rs-online.com/b9d6/0900766b814b1eff.pdf>
- Supabase. (u.d.). Supabase: The Open Source Firebase Alternative. Hentet 21. Mai, 2023, fra <https://supabase.com/>
- TDK InvenSense. (2017, 06. Februar). *ICM-20948 | InvenSense*. Hentet 3. Mai, 2023, fra <https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf>
- Tech Studio Design. (u.d.). *SimpleNB: A simple Arduino library for communication with NB-IoT/Cat-M1 modules*. Hentet 12. April, 2023, fra <https://github.com/techstudio-design/SimpleNB>
- Telenor. (2019, 7. Mai). *NB-IoT*. Ulike nettverksteknologier som muliggjør IoT. Hentet 20. April, 2023, fra

<https://www.telenor.no/bedrift/iot/iot-nettverksteknologi>

Texas Instruments. (2015, November). *LM2621 Low Input Voltage, Step-Up DC-DC Converter*.

Hentet 28. April, 2023, fra

<https://www.ti.com/lit/gpn/lm2621>

Tinkercad. (u.d.). Tinkercad. Hentet 15. April, 2023, fra

<https://www.tinkercad.com/>

Valmøt, O. R. (2015, 2. Mai). *Ultralyd er en norsk spesialitet. Slik fungerer det egentlig - Tu.no*.

Hentet 17. April, 2023, fra

<https://www.tu.no/artikler/industri-ultralyd-er-en-norsk-spesialitet-slik-fungerer-det-egentlig/23463>

WatElectronics.com, (2020, 9. Oktober) *7805 Voltage Regulator*. Hentet 27. April, 2023, fra

<https://www.watelectronics.com/what-is-7805-voltage-regulator-its-working/>

What is an Ultrasonic Sensor? (2022, 10. September). Robocraze. Hentet 3. Mai, 2023, fra

<https://robocraze.com/blogs/post/what-is-ultrasonic-sensor>

