

Erlend Gustafson Johansen

Vetle Hemstad Ås

Martin Lien

Deteksjon av limmengde i dispenser

Bacheloroppgave i Elektroingeniør - Elektronikk og sensorsystemer

Mai 2023



Erlend Gustafson Johansen
Vetle Hemstad Ås
Martin Lien

Deteksjon av limmengde i dispenser

Bacheloroppgave i Elektroingeniør - Elektronikk og sensorsystemer
Mai 2023

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



NTNU

Kunnskap for en bedre verden

Sammendrag

Hapro Electronics AS ønsket et system for å måle limnivå i limdispensere i sin produksjonslinje. Systemet varsler operatørene slik at de får byttet limpatron før det går utover kvaliteten på produktet.

Det er forsket på flere sensorer til dette formålet, deriblant hall sensor, metalldetektor, kapasitiv sensing og IR sensor. IR sensoren viste seg best egnet til dette formålet etter gitte kriterier, og det er videreutviklet et system basert på denne sensoren. Et slikt system baserer seg på å bruke IR-LED og fototransistor til å måle avstand mellom toppen av limpatronen, og ned til limnivået. Det er utviklet to kretskort, hvor et vil utføre målinger, og sende data trådløst til det andre kortet. Det andre kretskortet vil prosessere dataene, og vise dem på en skjerm. I tillegg til dette er det lagt inn en varslingsfunksjon når limnivået når et ønsket bunnivå. Modifisering av den opprinnelige limdispenseren inngår også i løsningen, og en 3D printet innkapsling til elektronikken.

Systemet fungerer til sin hensikt, og etter riktig kalibrering, skal det kunne settes i drift. Dette er allikevel en prototyp med forbedringspotensial. Ved test av prototyp har resultatene vært gode, men også pekt på forbedringer som kan gjøres med produktet. Et mer kompakt design av krets og innkapsling, ville kunne gitt fordeler under bruk i produksjonslinjen.

Abstract

Hapro Electronics AS wants a system for measuring glue level within dispensers used in their production line. The system alert an operator so they can change the glue cartridge before the quality of the product weakens.

Several types of sensors were researched. Hall sensor, metal detector, capacitive sensor, and infrared sensor were possible solutions for this type of system. The IR sensor had the best characteristics following given criteria, and a system was developed based on this sensor. The physical principle used in this project, is measuring the distance between the top of the dispenser and the current glue level, with IR radiation and a phototransistor. Two PCBs were developed during this project, were the first one will capture the measurements with the sensor and send them wirelessly to the other PCB. Then the other PCB will process this data and display it on a screen with useful animation. Also, there's a notification in the system for when the glue dispenser reaches the desired glue level. There's been some modification of the original glue dispenser in the system, as well as designed a 3D printed enclosure for the electronics.

Testing shows this system will solve Hapro's problem and has potential to be put into operation. Still, there's some potential for improvement, and some of the improvements which should be implemented in a finished product. More compact design of the circuit and the enclosure will make the product more practical in use within the production line.

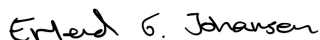
Forord

Bacheloroppgaven er utarbeidet i løpet av våren 2023 under samarbeid med Hapro Electronics AS. Prosjektet er et svar på problemstillingen til Hapro om et nivåmålingssystem, og varslingsystem til deres limdispensere. Dette vil bli det avsluttende arbeidet vårt som ingeniørstudenter ved NTNU Gjøvik og by på noen siste læringsmuligheter.

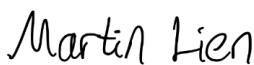
Prosjektet gir et grunnlag for videre utvikling og implementasjon i produksjonslinjen til Hapro. I tillegg har den gitt mulighet for forskning på forskjellige sensorer og utvikling av vår kompetanse og evner som elektronikingeniører innenfor felter som elektronikk konstruksjon, sensorer og 3D modellering og printing.

Vi ønsker å rette en spesiell takk til oppdragsgiveren vår fra Hapro, Jon Eilert Liane, som har bidratt med spennende oppgave. Han har svart på alt av spørsmål gruppen har kommet med og vært behjelpelig med testing og statusmøter.

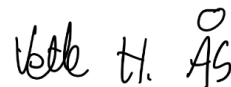
Videre ønsker vi å rette en takk til veilederen vår fra NTNU, Knut Wold, for hjelp med å vinkle oppgaven vår i riktig retning, rådgivning rundt prosjektfremgangen og rapportskrivningen.



Erlend Gustafson Johansen



Martin Lien



Vetle Hemstad Ås

Gjøvik, 19.05.2023

Innhold

Figurer	xiii
Tabeller	xiv
Kodelister	xvi
Akronymer	xviii
1 Innledning	1
1.1 Bakgrunn	1
1.2 Problemstilling	1
1.3 Omfang og begrensninger	1
1.4 Rapportinnhold	2
2 Teori	3
2.1 ESP32	3
2.2 Kommunikasjonsmetoder	3
2.2.1 I2C	3
2.2.2 UART	4
2.2.2.1 Protokoll for seriell kommunikasjon	6
2.2.3 ESP-NOW	6
2.3 Grensesnitt fra UART til USB	6
2.4 Elektromagnetisk stråling	7
2.5 Fototransistor	7
2.6 Lineær spenningsregulator	7
3 Metode	9
3.1 Ulike sensortyper	9
3.1.1 Hall effekt sensor	9
3.1.2 Kapasitiv sensing	10
3.1.3 Metalldetektor	12
3.1.4 IR sensor/lys sensor	12
3.1.5 Oppsummering og valg av sensor	13
3.2 Visualisering av systemet	14
3.3 Komponentvalg	14
3.3.1 ESP32	15
3.3.2 OLED skjerm	15

3.3.3	Schottky diode	15
3.3.4	Spenningsregulator	16
3.3.5	DC jack	17
3.3.6	UART-USB bridge	17
3.3.7	Fototransistor, IR-LED og tilhørende motstander	17
3.4	Design av kretskort	19
3.5	Sammenstilling av kretskort	20
3.6	Kodeforklaring	21
3.7	Modifisering av limdispenseren	25
4	Målinger og resultat	27
4.1	Testing av tidlig prototyp	27
4.2	Testing og kalibrering	27
4.3	Resultat av målingene på skolen	28
4.4	Resultat av kalibreringen hos Hapro	29
5	Diskusjon	31
5.1	Diskusjon av resultatene	31
5.2	Forbedringer/videreutvikling	32
5.3	Bærekraftige aspekter	35
5.4	Økonomiske aspekter	35
5.5	Etiske sider av prosjektet	35
6	Konklusjon	37
	Bibliografi	39
A	Kretsskjemaet til sendernoden	43
B	Kretsskjemaet til mottakernoden	45
C	Koden til sendernoden	47
D	Koden til mottakernoden	49
E	Koden for å finne MAC adresse	57
F	Kretsskjemaet til Arduino IR sensor	59
G	Testing av LED og fototransistor	61
H	Testing av IR-LED og fototransistor	63

Figurer

2.1	I2C ramme	4
2.2	Oppkobling av UART	5
2.3	Dataramme for UART sending	5
2.4	Oppbygning av spenningsregulator	8
3.1	Hall effekt sensor	10
3.2	Kapazitiv sensor	11
3.3	Metalldetektor	12
3.4	Skisse av systemet	13
3.5	3D modell av det ferdige produktet.	14
3.6	Visualiserer nivåmålingen.	14
3.7	Krets rundt schottky diode	16
3.8	Krets rundt spenningsregulator	16
3.9	Karakteristikk for sensoren	18
3.10	PCB utleggene til begge kretskortene.	19
3.11	Blokkdiagrammet til sendernoden	21
3.12	Blokkdiagrammet til mottakernoden	22
3.13	Modifisert 3D lokk	25
4.1	Kalibreringsgraf fra skolen	28
4.2	Kalibreringsgraf fra Hapro	29
5.1	Ferdige produktet hos Hapro	34
A.1	Kretsskjemaet til sendernoden	43
B.1	Kretsskjemaet til mottakernoden	45
F.1	Kretsskjemaet Arduino IR sensor	59

Tabeller

G.1	Testing av LED og fototransistor	61
H.1	Testing av IR-LED og fototransistor	63

Kodelister

3.1	Case setning	23
3.2	Bruk av millis funksjon	24
C.1	Koden til sendernoden	47
D.1	Koden til mottakernoden	49
E.1	Finne MAC-adressen til ESP32	57

Akronymer

- ε** permittiviteten. 10
- ACK** Acknowledge. 4
- ADC** Analog til digital konverter. 3
- DAC** Digital til analog konverter. 3
- EMI** Electromagnetic interference. 20
- GPIO** General-Purpose Input/Output. 3
- I2C** Inter-Integrated Circuit. 3
- IR** Infrarød. 7
- OLED** Organisk lysemitterende diode. 15
- P2P** Peer-to-Peer. 6
- ROM** Read Only Memory. 3
- SCL** Serial Clock. 4
- SDA** Serial Data. 4
- SMD** Surface Mount Device. 14
- SPI** Serial Peripheral Interface. 3
- SRAM** Static Random-Access Memory. 3
- UART** Universal Asynchronous Receiver-Transmitter. 3

Kapittel 1

Innledning

1.1 Bakgrunn

Hapro Electronics AS, heretter forkortet til Hapro, holder til på Jaren Hadeland, og har utviklet seg til å bli en av de største elektronikkprodusentene i Norge. Hapro sammenstiller og leverer alt fra kretskort til større komplette enheter, både med elektronikk og mekatronikk [1].

Hapro bruker limdispensere i produksjonen til flere av produktene de lager. Denne dispenseren er per dags dato, avhengig av en erfaren operatør til å bytte limpatronen før den er gått tom. Når den nærmer seg tom, vil kvaliteten på limingen svekkes. De er avhengige av å få byttet limpatronene før de nærmer seg denne grensen. For å sikre god kvalitet på produktene, ønsker de at denne prosessen skal mer eller mindre automatiseres.

Hapro ønsker en løsning som er enklest mulig, slik at den lett kan implementeres i andre liknende situasjoner. En enkel, men god løsning vil gjøre at Hapro lett kan videreutvikle produktet, og tilpasse sine andre produksjonslinjer.

1.2 Problemstilling

Ut fra denne bakgrunnen og etter diskusjon med Hapro har følgende problemstilling blitt formulert:

«Hvordan lage et sensorsystem som måler limmengde i en dispenser, og gi et varsel-signal slik at patronen kan bli byttet før kvaliteten på limingen blir svekket?»

1.3 Omfang og begrensninger

Prosjektet avgrenses ved at det skal konstrueres et system innen gitte rammer avtalt med Hapro. Systemet skal bestå av enkel elektronikk som kan gi en indikasjon på limnivå i limdispenser. Det skal også gi ut et signal på 24 V ved tom dispenser. Denne elektronikken

skal kunne drives på innspenninger mellom 24-30 V. Det er på forhånd blitt enighet mellom partene, at en løsning med to elektroniske noder som styrer systemet, vil være en god fremgangsmåte. Disse nodene vil kommunisere ved hjelp av trådløs kommunikasjon. I tillegg vil festemekanisme og innkapsling til elektronikken være en del av systemet. En del av oppgaven vil være å forske på og finne den, etter kravene, mest hensiktsmessige løsningen på et slikt system.

1.4 Rapportinnhold

Denne rapporten er delt inn i ulike kapitler for å tydeliggjøre innholdet.

Kapittel 2 - Teori: Kapitlet tar for seg den nødvendige teorien som skal til for å forstå de ulike aspektene senere i rapporten. Dette skal bygge grunnlaget for resten av innholdet, slik at leseren skal forstå valgene som er tatt.

Kapittel 3 - Metode: Her blir de ulike sensorene som er vurderte å bruke introdusert, samt komponentvalg, design av både kretskort og 3D modellene. Kapitlet inneholder også kodeforklaring av de mest relevante delene.

Kapittel 4 - Resultat: I resultatkapitlet blir rammene rundt testing forklart. Deretter blir resultatene presentert. Dette er både målinger fra testing og kalibrering som er gjort på skolen, i tillegg til testingen som ble gjennomført hos oppdragsgiver, Hapro.

Kapittel 5 - Diskusjon: Diskusjonskapitlet omhandler drøfting av de oppnådde resultatene. Her blir også aspektene rundt videreutvikling, bærekraft, økonomi og etiske sider av prosjektet tatt opp.

Kapittel 6 - Konklusjon: I dette avsluttende kapitlet oppsummeres prosjektet. Her blir oppgaven konkludert med å sammenlikne problemstillingen opp mot de oppnådde resultatene.

Kapittel 2

Teori

For å forstå mange av valgene som er tatt underveis i prosjektet, er det viktig å kjenne til teorigrunnet som ligger bak. Her presenteres teorien som er lagt til grunn for sensorvalg og kretsdesign til det endelige produktet. Det finnes også teori rundt noen av komponentene som er valgt, og som beskriver den enkelte komponentens gode egenskaper.

2.1 ESP32

ESP32-Wroom-32 er en liten, men kraftfull mikrokontroller som er utviklet av Espressif Systems. Denne er svært nyttig ettersom den er billig, har lavt effektforbruk, samt at den støtter både WiFi og Bluetooth kommunikasjon. ESP32 er bygget rundt en chip som heter ESP32-D0WDQ6. Det interne minnet består av 448 KB Read Only Memory (ROM) og 520 KB Static Random-Access Memory (SRAM). Mikrokontrolleren er oppgitt i databladet til å trekke 240 mA under aktiv bruk og bruk av WiFi. I tillegg støtter ESP32 andre nyttige grensesnitt som: [2, 3]

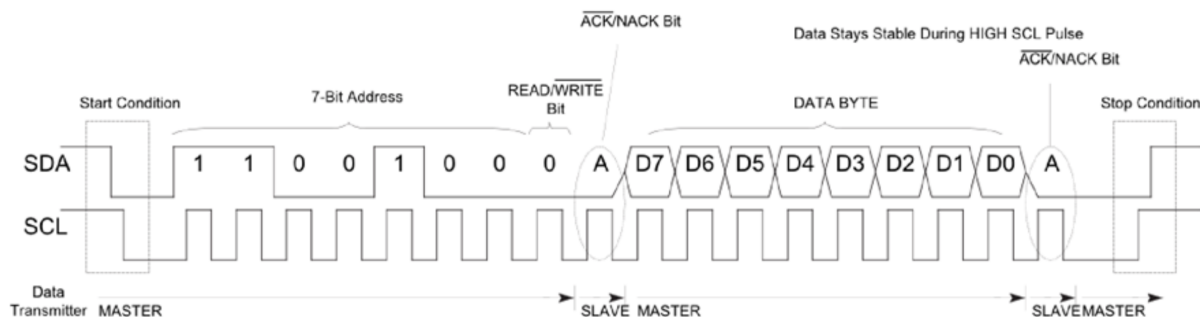
- 34 programmerbare General-Purpose Input/Output (GPIO)
- 18 kanaler med 12-bit Analog til digital konverter (ADC)
- 2 × 8-bit Digital til analog konverter (DAC)
- 4 × Serial Peripheral Interface (SPI)
- 2 × Inter-Integrated Circuit (I2C)
- 3 × Universal Asynchronous Receiver-Transmitter (UART)

2.2 Kommunikasjonsmetoder

2.2.1 I2C

I2C er en mye brukt kommunikasjonsmetode. Den bruker en master/slave konfigurasjon der klokken blir brukt til å synkronisere avsender og mottaker. Metoden er godt egnet ved

tilkobling mellom for eksempel mikrokontrollere og sensorer, da det enkelt kan kobles flere enheter til på samme databuss. Her skjer kommunikasjonen på halv-dupleks form. Dette vil si at sending av data kan kun skje i en retning om gangen. Denne metoden bruker kun to tilkoblinger for kommunikasjon, Serial Clock (SCL) og Serial Data (SDA).

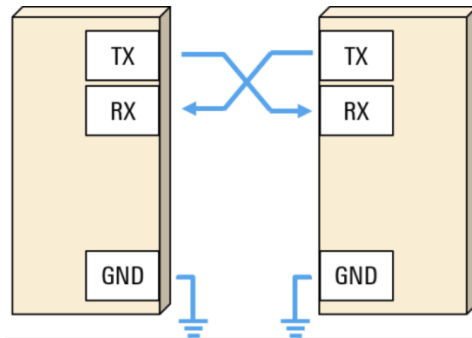


Figur 2.1: Oppsett for en I2C melding [4].

I Figur 2.1 vises oppsettet til meldingsformatet innen I2C kommunikasjon. Den følger et spesielt oppsett for hvordan sendingen foregår. I starten av meldingen settes SDA lav litt før SCL. Dette gjøres slik at mottakeren er bevisst på at det nå kommer en melding. Ettersom det kan kobles flere enheter til I2C bussen, er det nødvendig at hver slavenode har sin unike adresse slik at masternoden kan skille de ulike enhetene. Denne adressen blir lagt til etter initialiseringen av meldingen. ESP32 støtter både adresse på 7- og 10-bit [5]. Etter adressebiten kommer en indikator, R/W. Den symboliserer om master ønsker å motta data fra slaven eller om den skal sende data til slaven. Acknowledge (ACK) bit forteller avsenderen om dataen er mottatt, i dette tilfellet adressen. Her får da mottakeren mulighet til å skrive til SDA linjen. Hvis mottakeren setter nivået lavt, betyr det at dataen er mottatt. Om nivået holdes høyt til neste klokkepulss indikerer det at dataen ikke har nådd fram. Dersom adressen er mottatt, er det klart for å sende data på linjen. Dataen blir gjerne sendt med en lengde på 8 bit, 1 byte, om gangen. Det kan sendes flere bytes etter hverandre, men da blir det lagt til en ACK etter hver byte. Dette gjøres slik at det er enklere å oppdage hvilken byte som eventuelt ikke blir mottatt. Når avsender har sendt alle sine ønskede data, blir det sendt et stoppbit. SDA vil gå høy etter at SCL nivået har gått høyt. Dette gjøres for å signalisere at det ikke kommer mer innkommende data [4].

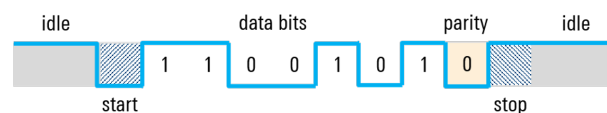
2.2.2 UART

UART er en mye brukt kommunikasjonsprotokoll mellom små elektroniske enheter. Denne kommunikasjonen er seriell, som betyr at all data sendes i rekkefølge over kun en datalinje, altså bit for bit. Dette betyr at en enhet kun trenger en linje for å sende data og en linje for å motta data, noe som er en stor fordel i enkelte applikasjoner. For at to enheter skal kommunisere må den kobles slik som vist i figur 2.2.



Figur 2.2: Oppkobling av UART mellom to enheter [6].

Baudraten kan programmeres og velges selv. UART bruker ikke et internt klokkesignal for synkronisering mellom nodene. Det vil si at sendingen skjer asynkront. For at mottakeren skal kunne lese den mottatte dataen, er det viktig med riktig timing. Senderen sender en bitstrøm mens mottakeren benytter sitt klokkesignal for å sample og time dataen.



Figur 2.3: Oppsett for en UART ramme [6].

Sending av data foregår i såkalte rammer som følger en gitt struktur. Dette er visualisert i figur 2.3. Det skilles mellom to nivåer, høy og lav, som tilsvarer binære verdiene 1 og 0. Når det ikke er sending av data, settes nivået høyt slik at det enklere kan oppdages feil eller brudd på linja. Første bitet i dataen er et startbit. Da vil nivået bli satt lav i en klokkepuls, som gjør at mottakeren er klar til å lese, siden sendingen er asynkron. Etter startbitet kommer dataen som skal sendes. Lengden på dataen kan variere, men ESP32 støtter kun lengder på 5-8 bit. Etter dataen kommer det som heter paritetbit. Dette er med for å kontrollere om det har skjedd en feil underveis i sendingen. Den kan enten være partall eller oddetall. Det vil si at antall 1'ere i dataen blir telt opp. Deretter blir paritetsbitet lagt til som 1 eller 0, slik at summen av 1'ere enten blir oddetall eller partall. Det finnes mer avanserte metoder for feilsjekking, men dette er en enkel metode som ofte kan oppdage enkle feil. På slutten av datarammen blir det lagt til et stoppbit. Dette gjøres for å signalisere til mottakeren at datarammen er ferdig. For å realisere dette, settes nivået høyt i form av en eller to bit. Datalengden og paritet blir forhåndsbestemt før sending av dataen, slik at både avsender og mottaker opererer etter de samme kriteriene. Dette er nødvendig for at de skal kunne kommunisere med denne protokollen [6].

2.2.2.1 Protokoll for seriell kommunikasjon

En standardprotokoll som brukes for seriell kommunikasjon er RS-232. Den brukes for å koble datamaskinen til eksterne eller interne enheter for å utføre seriell datautveksling mellom dem. Protokollen ble introdusert tilbake på 60-tallet, men brukes fremdeles i noen applikasjoner i dag. Dette gjelder for eksempel noen typer mikrokontrollere. Protokollen har en rekkevidde på opptil 15 meter, med maks hastighet på opptil 1 Mbps. Mange bruker fremdeles denne protokollen på grunn av den lave kostnaden ved implementering. Det brukes både positive og negative spenninger. Et 0-bit tilsvarer en positiv spenning mellom 3 og 15 V. 1-bit, tilsvarer en spenning på mellom -3 og -15 V. Det betyr at det ligger inne en stor margin på støy, og gjør denne protokollen robust mot støyspenninger. Oppkoblingen av denne typen krets baserer seg på Tx og Rx pinnene slik som UART koblingen, figur 2.2 [7, 8].

2.2.3 ESP-NOW

Det er mange måter å benytte seg av trådløs kommunikasjon. En av dem er Peer-to-Peer (P2P). Espressif Systems, utviklerne av ESP32, lager også protokoller og funksjoner tilpasset denne mikrokontrolleren. ESP-NOW er en slik protokoll, som muliggjør P2P kommunikasjon mellom to eller flere mikrokontrollere. Denne går ut på å etablere et privat trådløst nettverk mellom mikrokontrollerene. WiFi tilgang er derfor ikke nødvendig. Protokollen benytter seg av det ulisensierte 2.4 GHz båndet. Dette er det samme som brukes i mange andre trådløse tilkoblinger, som for eksempel mellom en trådløs datamus og en PC. Dataen som sendes på nettverket er begrenset til 250 bytes. Dette er ikke nok om det skal sendes video eller liknende. Det vil derimot fungere bra i en applikasjon hvor det kun skal sendes et lite antall data, i form av float verdier. Et eksempel på dette kan være i IoT sammenhenger [9].

2.3 Grensesnitt fra UART til USB

Silicon Labs lager en integrert krets, CP2102N, med formålet å kunne implementere USB grensesnittet til ulike enheter. Denne kretsen muliggjør en overgang mellom UART RS-232 design og USB. Den integrerte kretsen gjør mye av jobben selv, og trenger få eksterne komponenter for å drive et fungerende USB grensesnitt. Den har blant annet en innebygd klokke, slik at det ikke trengs en ekstern krystall, som er typisk for USB design. Den krever likevel noen få enkle komponenter rundt for å fungere optimalt. IC-en er fysisk liten, og selv med enkelte komponenter rundt vil den ikke ta mye plass i et kretsdesign. Silicon Labs har også drivere til mange forskjellige operativsystem for denne chipen, som ligger ute gratis på nettsiden deres. Det er derfor overkommelig å implementere denne komponenten for de aller fleste, som gjør den svært allsidig og populær i mange design [10].

2.4 Elektromagnetisk stråling

Elektromagnetisk stråling er energi i bevegelse i form av bølger. Det beveger seg i lysets hastighet, hvor frekvensen og bølgelengden til strålingen bestemmer hvilken type stråling det er. Siden det alltid beveger seg i samme fart, vil kortere bølgelengde gi høyere frekvens og høyere energimengde. Det elektromagnetiske spekteret strekker seg fra radiobølger, med lavest frekvens, til gamma stråling, med høyest frekvens. Omtrent midt på det elektromagnetiske spekteret finnes synlig lys, som har bølgelengde mellom 400 og 700 nanometer. Stråling med bølgelengde fra synlig lys og opp til omtrent 1 millimeter kalles Infrarød (IR) stråling. IR stråling er stråling som sendes ut på grunn av temperaturen til et objekt. Desto varmere et objekt er, desto sterkere vil strålingen være. Dette vil si at bølgelengden synker og frekvensen øker. To viktige egenskaper til elektromagnetisk stråling er refleksjon og absorpsjon. Det vil si hvor mye av energien som blir sendt tilbake fra gjenstanden som blir utsatt for strålingen. Både type materialet og fargen vil ha mye å si. Svarte overflater absorberer mye av strålingen, mens hvite overflater reflekterer mer av strålingen tilbake [11–13].

2.5 Fototransistor

En fototransistor er ganske lik en vanlig transistor, med base, kollektor og emitter. Forskjellen er at den ikke har en fysisk kobling til basen. Den er i stedet bygget opp av et halvleder materiale, slik at det dannes en strøm når den blir utsatt for elektromagnetisk stråling. Denne strømmen vil variere ut ifra lysintensiteten, som også kan sees på som energi, slik at strømmen ut fra emitter vil variere. En fototransistor er godt egnet til å bruke for å måle lysintensiteten, på grunn av egenskapene en transistor innehar. Dette er spesielt med tanke på at den fungerer som en forsterker for strømmen som kommer av lys som treffer basen. For en ideell transistor vil:

$$I_e = I_b + I_c. \quad (2.1)$$

Kollektor strømmen kan uttrykkes som:

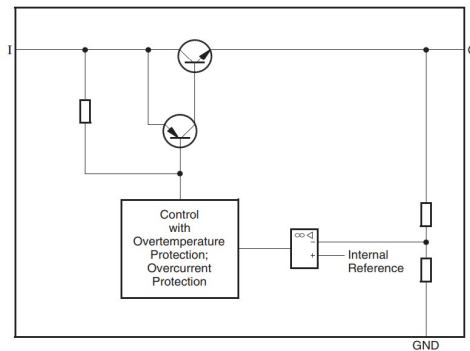
$$I_c = I_b * \beta. \quad (2.2)$$

For en ideell transistor, vil ligning (2.2) gi et direkte proporsjonalt forhold mellom kollektor og base strømmen [14, 15].

2.6 Lineær spenningsregulator

En krets med mikrokontrollere og nøye kalibrerte sensorer, er avhengig av en stabil spenningskilde. Til dette formålet er det vanlig å bruke en spenningsregulator. Den enkleste formen for spenningsregulator er lineær. Denne typen kalles lineær selv om den består av

ikke lineære komponenter, som transistorer. Innvendig består den også av en tilbakekobling og en spenningsdeling for å finne referansespenningen. En slik pakke har ofte tre pinner, hvorav en av de er jord, og de to andre er inn- og utgang. Det er noen fordeler og ulemper med denne typen spenningsregulator kontra den andre mest vanlige regulatoren, switchet regulator. Lineære regulatorer produserer lite støy på utgangen, er relativt billige og krever få komponenter rundt for å fungere. De fleste slike regulatorer krever kun et par kondensatorer eksternt for å fungere som tiltenkt. Ulempen er at de er forholdsvis ineffektive i forhold til switchede regulatorer. Noe effekttap vil forekomme i form av varme. [16, 17].



Figur 2.4: Innsiden av en lineær spenningsregulator [17].

Kapittel 3

Metode

Prosjektet ble delt opp i flere forskjellige faser. Den første fasen gikk ut på å teste flere forskjellige sensorer og ideer som kunne brukes i et slikt system. Herunder har de forskjellige sensorene blitt bygget, og testet mot en prototyp av limdispenseren. Nedenfor begrunnes fordelene og ulempene for de ulike systemene som ble testet. Deretter ble sensortype valgt, og det ble undersøkt mer i dybden på å implementere sensoren på best mulig måte. Det blir i tillegg beskrevet flere valg rundt komponenter og kretsdesign. Blokkskjema for systemet, vil forklare hvordan det fungerer, og det er lagt inn kodeforklaring på viktige kodeblokker. Til slutt blir det beskrevet hvordan limdispenseren ble modifisert for å tilpasses den valgte sensoren, som var avgjørende for at resultatet skulle bli bra.

3.1 Ulike sensortyper

For å måle limnivået i tuben, var det flere mulige sensorer som kunne oppfylle målet. Derfor trengtes det nøye undersøkelser og testing for å finne sensoren som oppfylte kravene på best mulig måte. De neste underkapitlene vil gå nærmere inn på de forskjellige sensorene som ble undersøkt. De ulike sensortypene som ble undersøkt er:

- Hall effekt sensor
- Kapasitiv sensing
- Metalldetektor
- IR sensor/lys sensor

3.1.1 Hall effekt sensor

Hall effekt sensor er en sensor som brukes for å kunne måle styrken til et magnetfelt. Den kan brukes på to måter, enten analogt eller digitalt. Ved analog bruk måles styrken på magnetfeltet og kan avgjøre hvor nær en magnet er. Hvis metoden blir brukt digitalt, fungerer det mer som en bryter når styrken til magnetfeltet krysser et gitt nivå.

Måten dette kunne ha blitt brukt i prosjektet var å plassere en magnet inne i limpatronen, mens sensoren blir plassert på utsiden. Deretter kunne magnetfeltet ha blitt målt. Problemet med denne metoden var at magnetfeltet ble relativt svakt på utsiden. Dette gjorde at en analog hall effekt sensor ville fungert dårlig. Ved bruk av digital hall sensor, ville det vært nødvendig å implementere et større antall sensorer langs tuben. Grunnen til dette er for å få en nøyaktig nivåindikasjon. I tillegg krever denne måten at en operatør husker å flytte magneten over i den nye tuben når den skal byttes. Etter nærmere diskusjon med Hapro, kom det fram at de ønsket en mer eller mindre service-fri sensor, hvor en operatør slipper å flytte magneten hver gang limpatronen byttes [18].



Figur 3.1: Hall effekt sensor.

3.1.2 Kapasitiv sensing

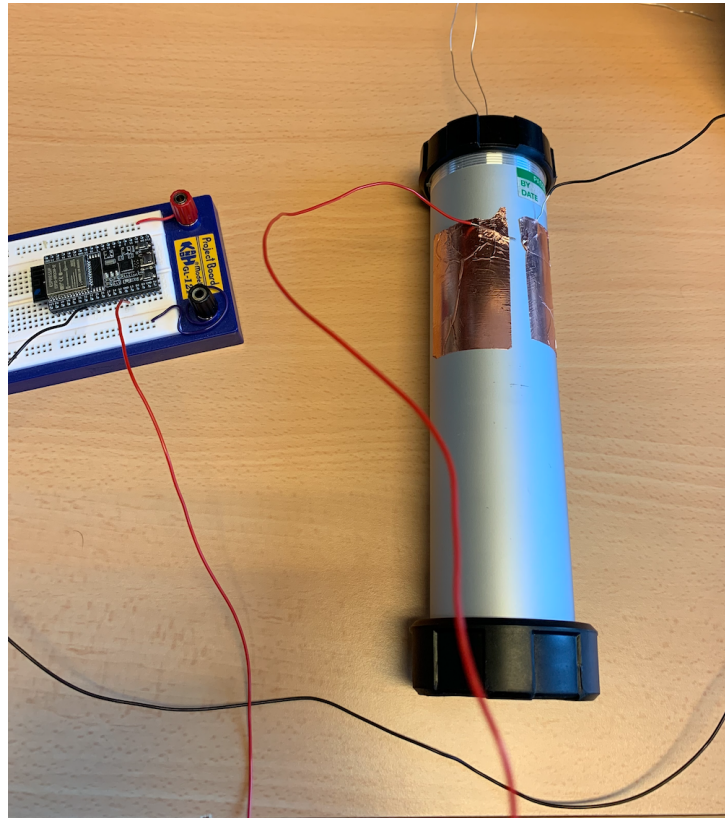
Tanken med denne typen sensor var å konstruere en stor kondensator rundt limdispenseren. Dette betyr et ledene materiale på hver side av limdispenseren. Kapasitansen er en elektrisk egenskap ved kondensatorer som er målbar. Den vil endre seg ut ifra de fysiske størrelsene til kondensatoren. Kapasitans måles ut ifra formelen

$$C = \epsilon \cdot \frac{areal}{d} \quad (3.1)$$

Arealet på lederne som brukes vil være fast, og det samme vil distansen mellom dem være. Det eneste som vil endre seg i systemet er permittiviteten (ϵ) til materialet mellom lederne. Dette er en verdi som beskriver evnen ett stoff har til å bli polarisert [19]. I teorien vil permittiviteten mellom lederne endre seg med mengden lim som er til stede mellom dem.

Når limmengden synker mellom lederne vil kapasitansen nærme seg verdien den ville hatt med bare luft imellom.

Kapasitiv sensing er en vanlig metode for å måle væsknivå i en beholder. Dette prinsippet viste seg å være vanskelig å bruke, spesielt på grunn av at denne kondensatoren ville blitt montert på utsiden av dispenserens. Plasseringen medførte at det var vanskelig å få den sensitiv nok til å merke forskjell på innholdet i dispenserens. Størrelsen på dispenserens gjorde også at en kondensator måtte bygges ganske stor for å kunne bruke innholdet i limpatronen som dielektrikum.



Figur 3.2: Kapasitiv sensor.

3.1.3 Metalldetektor

Metalldetektor brukes for å kunne detektere metallgjenstander som for eksempel er gjemt under bakken. I dette prosjektet var tanken å kunne legge et metallobjekt inne i limdispenseren slik at det ga utslag når den kom nære nok. En metalldetektor fungerer ved at den setter opp ett elektromagnetisk felt. Hvis en gjenstand laget av metall kommer inn i dette feltet, vil det sendes et elektromagnetisk felt tilbake. Dette er i teorien en god løsning, men siden øvre og nedre del av dispenseren er laget av metall, ble det mye støy på målingene. Systemet hadde også samme ulempe som hall effekt sensoren, at det var nødvendig å legge en metallgjenstand nedi dispenseren, hver gang limpatronen byttes.

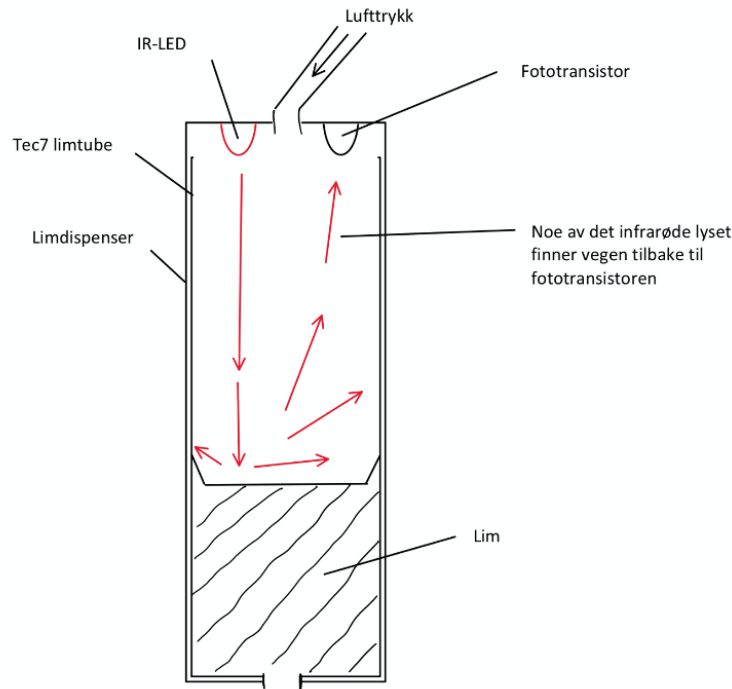


Figur 3.3: Metalldetektor.

3.1.4 IR sensor/lys sensor

Prinsippet for å måle avstand ved hjelp av stråling og refleksjoner er brukt i flere sammenhenger profesjonelt. Grunnen til at denne ideen dukket opp, var ønsket om et produkt som krever minst mulig av operatøren under drift. Denne løsningen går ut på å feste en LED og fototransistor på innsiden av toppen til sylindere. LEDen vil da sende stråling ned mot bunnen av limpatronen, og avhengig av nivået og høyden, vil mengden stråling som reflekteres tilbake variere. Dette er illustrert i figur 3.4.

Endringen i nivå for bunnen på limpatronen, var tilnærmet lineær i store deler av arbeidsområdet med mengden stråling som reflekteres tilbake til sensoren. Dette ga også en tilnærmet lineær endring i strømmen som går gjennom fototransistoren. Denne egenskapen



Figur 3.4: Skisse av ideen.

kan lett måles, for eksempel ved å måle spenningen over en motstand som ligger i serie med emitter på fototransistoren. Spenningen måles ved hjelp av en mikrokontroller, med innebygd ADC. Deretter vil dataene kunne behandles, og brukes videre i systemet. En viktig forutsetning som må være til stede for at systemet skal fungere, er at det er helt lystett. Lys som kommer inn i dispenserene fra utsiden, vil kunne gi store forstyrrelser på sensoren. For eksempel en taklampe, vil også gi ut små mengder IR-stråling, som vil kunne påvirke resultatene fra sensoren.

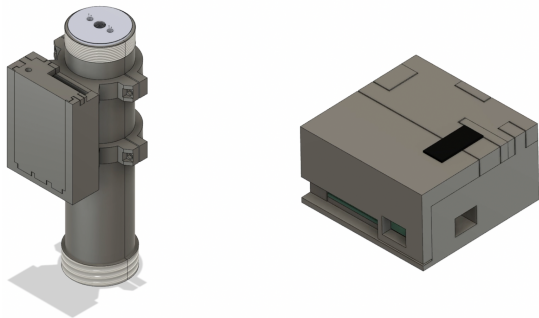
Et slikt system er avhengig av en god kalibrering. Grunnen til dette er at teksturen på veggene inne i sylindren vil ha mye å si for refleksjonen tilbake til sensoren. Dette gjelder også spesielt fargen på veggene. Ved test viste sensoren mye større utslag, og ga respons over et større område ved hvitfargede vegger, i forhold til mørke vegger. Om Tec7 endrer designet på limpatronen sin, og for eksempel lager den av et annet materiale enn hvit plast, vil det være nødvendig å recalibrere sensorsystemet.

3.1.5 Oppsummering og valg av sensor

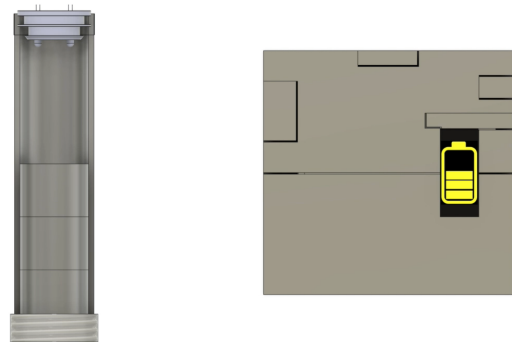
Det ble lagt ned mye tid på å bygge og teste de forskjellige sensorene. Dette ga varierende resultater, men førte til en klar konklusjon om hvilken sensor som var interessant for videre forskning. Både hall sensoren, og metalldektoren hadde betydelige minus, ved at en fysisk gjenstand måtte legges ned i limdispenseren under drift. Kapasitiv sensing, ble også strøket raskt etter at testingen hadde gitt dårlige, og utydelige resultater. IR sensoren viste umiddelbart stort potensiale, ved at det var mulig å få en tilnærmet lineær respons fra systemet ved

nivåmåling. Et annet pluss var at det krever få komponenter og sensorer, samtidig som en operatør slipper å plassere noe i dispenserens. Etter mer omfattende testing, og møter med Hapro, ble denne løsningen valgt ut til å forske videre på. Det ble bygget en testtrigg, som vist i vedlegg F, som er brukt til senere forsøk, beskrevet i avsnitt 3.3.7.

3.2 Visualisering av systemet



Figur 3.5: 3D modell av det ferdige produktet.



Figur 3.6: Visualiserer nivåmålingen.

Figur 3.5 og figur 3.6 viser et overblikk over de to nodene i sin helhet. Til venstre på figur 3.5 vises sendernoden montert på dispenserens. Det er denne noden som tar målingene. Til høyre vises mottakernoden. Denne noden prosesserer den mottatte dataen og viser det på skjermen. Her er innkapslingene lukket slik de vil være under drift. Mottakernoden er tiltenkt å kunne plasseres etter ønske. Eneste faktoren å ta hensyn til, er rekkevidden til den trådløse kommunikasjonen. Figur 3.6 viser hvordan mottakernoden visualiserer limnivået. Til venstre vises et tverrsnitt av limdispenserens med sensoren i toppen. Til høyre er mottakernoden som visualiserer det respektive limnivået. Målingene deles inn i fem nivåer. Her viser systemet nivå tre. Begge nodene vist her vil være innkapslingen til hvert sitt respektive kretskort. Disse ble designet i Fusion 360, etter dimensjonene på de allerede designede kretskortene. Deretter ble de 3D printet ved hjelp av tradisjonell 3D printing i en type plastikk kalt PLA.

3.3 Komponentvalg

Det var flere hensyn som skulle tas når det gjaldt valg av komponenter. En beslutning som ble tatt før bestilling av komponentene, var størrelsene på dem. Det ble standardisert på 0805 størrelse på Surface Mount Device (SMD) komponenter. Dette ble gjort for å sikre et ryddigere kretskort. Denne størrelsen var ikke for liten, og var mulig å lodde ved hjelp av utstyret tilgjengelig på skolens areal. Ved å velge SMD kunne arealet minskes sammenliknet med hullmonterte komponenter. Det var i tillegg vektbesparende, som også var en fordel

ved montering. Enheten vil monteres inne i en produksjonslinje, og det vil være fordelaktig om den ikke tar for mye plass. SMD komponenter er også billigere i forhold til hullmonterte, som gjorde at denne typen komponenter var den foretrukne løsningen.

Bestillingstid var et hensyn som ble diskutert ved oppstart av prosjektet. Dette var også spesifikt tatt hensyn til i prosjektplanen som ble laget. Lang leveringstid ville kunne forsinke prosjektet betydelig. De mest kritiske komponentene ble derfor bestilt tidlig i prosessen for å sikre at de ankom i tide. Deretter ble resterende komponenter, som ikke var like viktige før produktet skulle ferdigstilles, bestilt litt senere. De mest kritiske komponentene ble bestilt fra Mouser Electronics, en distributør som holder til i USA.

3.3.1 ESP32

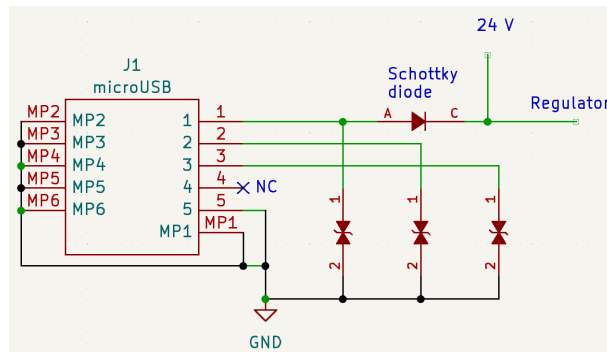
Mikrokontrolleren ESP32 ble valgt til dette prosjektet på grunn av dens allsidige egenskaper, lave pris og tilgjengelighet. Den har mulighet for innebygd WiFi, som var en viktig egenskap. I tillegg er denne mye brukt i IoT sammenheng på grunn av de gode innebygde funksjonene, som for eksempel muligheter til å redusere effektforbruket. Det har blitt opparbeidet mye kompetanse om denne typen mikrokontrollere tidligere. På grunn av dens utstrakte bruk innen IoT på verdensbasis, fantes det også mye dokumentasjon og ressurser på nettet om hvordan dens mange funksjoner og egenskaper kunne benyttes.

3.3.2 OLED skjerm

For å kunne vise fram de ulike nivåene som blir målt, ble en Organisk lysemitterende diode (OLED) skjerm benyttet. Denne var av type I2C, som var et bevisst valg for å unngå mange ledninger. Dette gjorde design av kretskort enklere ettersom det kun krevde fire tilkoblinger. Slike skjermer er også allsidige. Det kan enkelt lages ulike ikoner og animasjoner som kan gjøre visualiseringene av målingene bedre, slik det er gjort på figur 3.6. Her er det laget et batteriikon med ulike nivåer. Dette er gjort ved hjelp av en konverter som gjør bilder om til bitkart, som vist i vedlegg D [20].

3.3.3 Schottky diode

Det vil være to innganger for strømforsyning. Den ene er fra en mikro USB, som også brukes til å programmere kretsen. Den andre inngangen brukes til å drive kretsen, under normal drift, ved 24 V. Kun en av inngangene vil være i bruk om gangen. Grunnen til at en schottky diode blir brukt, er for å skjerme mikro USB-en for 24 V signalet.

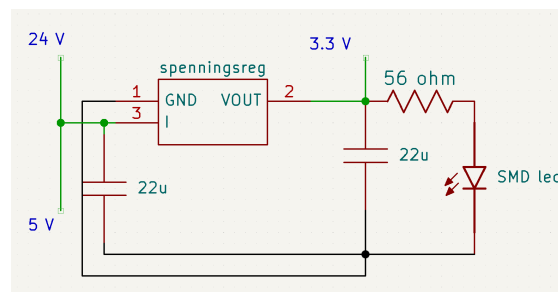


Figur 3.7: Krets rundt schottky diode.

Som vises i figur 3.7 er schottky dioden plassert rett etter utgangen av mikro USB-en. Hapro har spenninger på opptil 30 V på strømforsyningene, og schottky dioden må derfor dimensjoneres deretter. Derfor ble schottky dioden valgt for å tåle 40 V i sperreretning, for å ha god margin. Schottky dioder har også et lavt spenningsfall i lederetning, som gjør det til et godt valg av diode til en slik applikasjon.

3.3.4 Spenningsregulator

En spesifikasjon ved oppgaven som ble gitt fra Hapro, var at produktet skulle kunne operere med en inngangsspenning på opp imot 30 V. Mye av utstyret i deres produksjon opererer mellom 24-30 V, og sensorsystemet ble konstruert slik at det ikke var et unntak. Samtidig var strømtrekket en viktig faktor når det skulle avgjøres hvilken regulator som skulle brukes. Som skrevet i avsnitt 2.1 så vil mikrokontrolleren typisk trekke rundt 240 mA ved høy aktivitet. I realiteten ble det erfart at den kan trekke godt over 300 mA under tyngre bruk. Dette var fordi en del andre komponenter, som sensorsystemet med fototransistorer og IR-LEDen, også skulle ha forbruk. For å oppfylle disse kravene, og samtidig legge inn en liten buffer, ble det valgt en regulator med oppgitt maks strømtrekk på 1 A, og 40 V som høyeste inngangsspenning. Et annet krav var at den skulle gi ut 3.3 V, ettersom dette er spenningsnivået mikrokontrolleren og sensorene drives på.



Figur 3.8: Krets rundt spenningsregulator.

Kretsen rundt spenningsregulatoren, samt inngangs- og utgangssignalene, kan sees på figur 3.8. Her vises det at to forskjellige innganger er brukt, som nevnt i avsnitt 3.3.3. Rundt

spenningsregulatoren brukes to avkoblingskondensatorer. I tillegg til dette brukes et statuslys for å indikere at kretskortet er koblet til en strømforsyning. En annen ønsket egenskap ved denne spenningsregulatoren, er dens innebygde beskyttelse mot overspenning, kortslutning og temperatur. Dette gjør at kretsen har et vern så nære inngangen som mulig. Den kan derfor være med på å redde resten av komponentene om et tilfelle med for eksempel overspenning skulle oppstå.

3.3.5 DC jack

Av hensyn til brukervennlighet ble det lagt ned en del tid i hvordan strømforsyningen skulle løses best mulig. Produktet vil kobles av og på mange ganger under operasjon. Tall oppgitt fra Hapro er at limpatronen må byttes opptil hver dag. For å sikre at det går raskt å koble av og på, ble det bestilt opp runde jack innganger som da er bidireksjonale. Dette gjelder både inngangen og utgangen. Disse ble også plassert på venstre side av kortet, slik at de vil komme samme retning som slangen for trykkluften brukt i dispensereren.

3.3.6 UART-USB bridge

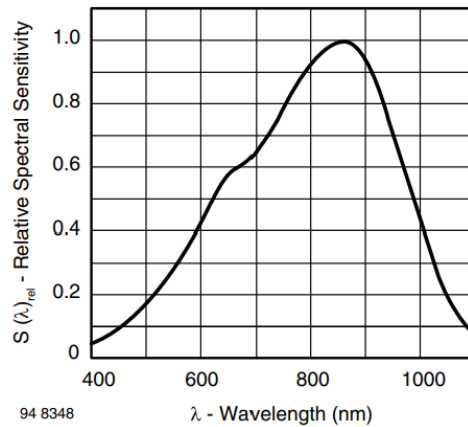
Mikrokontrolleren som ble valgt, ESP32, bruker UART som grensesnitt. Dette er et tungvint grensesnitt å bruke, om ikke en UART-USB overgang mellom PC-en og kretskortet benyttes. For å forenkle opplastinger og oppdateringer av koden til mikrokontrolleren, har det blitt valgt å gi kretskortet tilkoblingsmulighet via mikro USB. For å gjøre denne typen krets mulig, er det vanlig å bruke en UART-USB bridge. Dette er en IC med de nødvendige funksjonene til å endre grensesnittet. Komponenter CP2102N, som er skrevet om i avsnitt 2.3, ble valgt blant annet på grunn av at den har innebygd klokke. Dette gjorde det mulig å realisere en UART-USB krets med færre komponenter, og på et litt mindre område. I tillegg var denne komponenten lett tilgjengelig fra mange leverandører til en overkommelig pris.

3.3.7 Fototransistor, IR-LED og tilhørende motstander

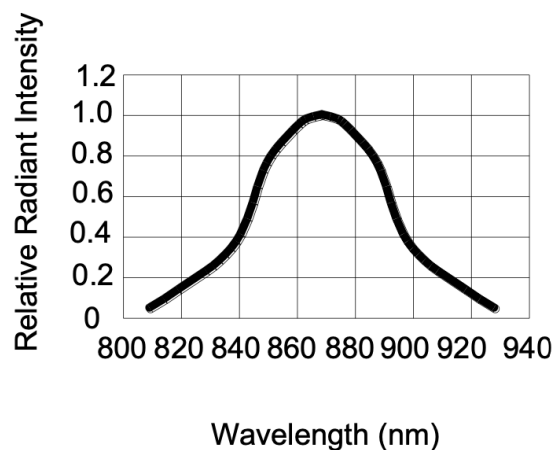
Fototransistoren som ble valgt, var viktig at stod i samsvar med IR-LEDen. Det ble da funnet en transistor med karakteristikk som tilsvarer de samme bølgelengdene som IR-LEDen utstråler mest av. Dette vises i figur 3.9a og i figur 3.9b.

Grunnen til at IR-stråling ble valgt foran synlig lys var både på grunn av testing, men også på grunn av at IR-stråling krever mindre energi, og med det mindre strøm. Testingen som ble gjort var å teste både LED og fototransistorer, for synlig lys og IR-stråling, med forskjellige motstander. Det ble testet med limpatron inne i den lukkede limdispensereren, for å få et virkelighetsnært resultat som mulig. Disse testene ga gode målinger for begge typene transistorer, og viste at et system for nivåmåling var realiserbart. IR-stråling hadde allikevel større spenn i målingene, og ble derfor valgt som best egnet. Dette betyr større oppløsning

i målingene fra toppen av limpatronen til bunnen, noe som var ønskelig for å oppnå bedre nøyaktighet fra systemet.



(a) Sensitiviteten til fototransistoren ved forskjellige bølgelengder [21].

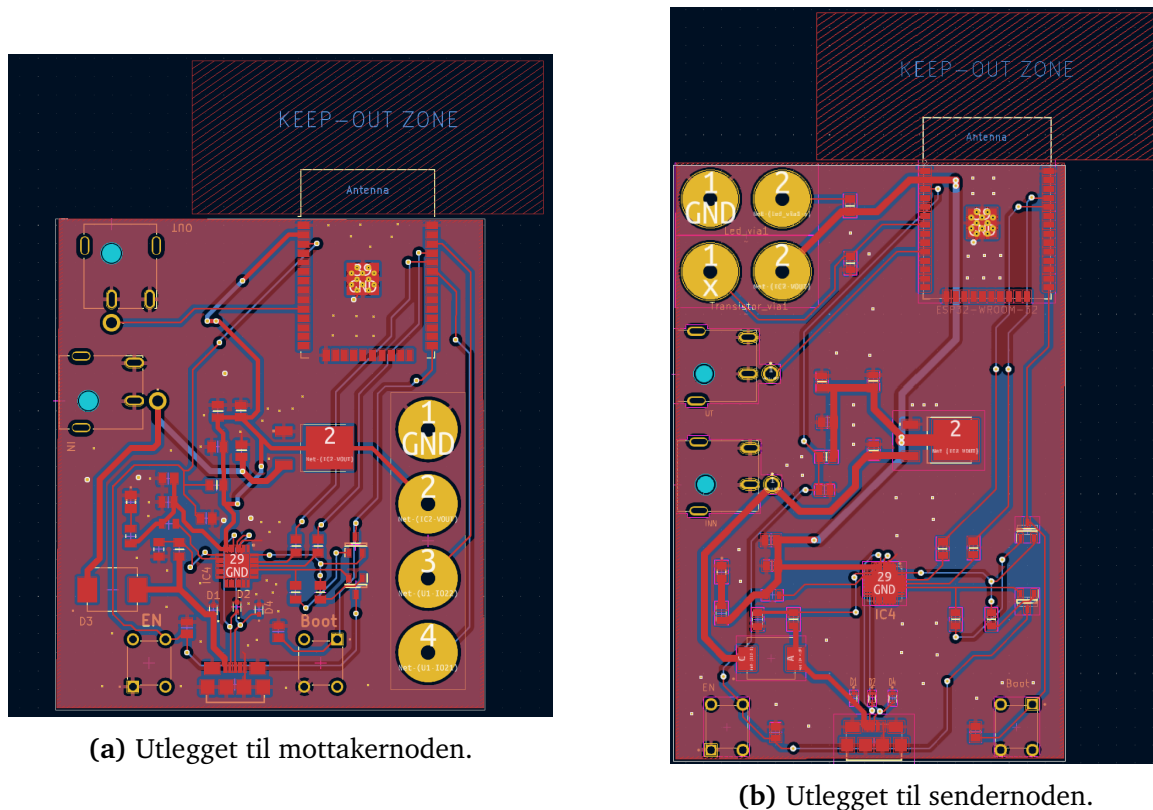


(b) Intensiteten til IR-LEDen ved forskjellige bølgelengder [22].

Figur 3.9: Graf for stråling i forhold til bølgelengde for sensorene som er brukt.

Motstandene til både LEDen og fototransistoren ble valgt ut gjennom utprøvningsmetoden. Ved å endre disse to verdiene, var det mulig å justere både lysintensiteten til LEDen, og samtidig sensitiviteten på utgangssignalet. For å komme fram til de beste verdiene ble det kjørt en rekke forsøk, i tilsvarende arbeidsomgivelser. Disse forsøkene ble kjørt innenfor de typiske verdiene i de forskjellige databladene. Det finnes cutoff punkter i begge ender av skalaen, og de beste verdiene ble testet fram mellom disse punktene. Tabell som viser data fra forsøk med synlig lys finnes i vedlegg G. Tabellen som viser data fra forsøk med IR-LEDen og fototransistoren finnes i vedlegg H. Disse testene ble utført ved bruk av en Arduino, og koblet slik som i vedlegg F.

3.4 Design av kretskort



Figur 3.10: PCB utleggene til begge kretskortene.

I fasen som gikk ut på å designe kretsen og lage et utlegg, ble programmet KiCad tatt i bruk. Dette gikk raskt å lære og det var mulig å legge inn både symbol og footprint av komponentene som skulle bli brukt. For å være sikker på at komponentene ville passe på kretskortet, var dette viktig. Når kretsen skulle designes, var det mange hensyn å tenke på. Prosjektet er bygget på ESP32-Wroom-32 chipen. Denne har en antenne som er svært viktig å ta hensyn til. Der måtte det være en klaring på minimum 15 mm på hver side av antennen. Det vil si det ikke kunne være gjenstander av metall, komponenter eller ledningsbaner innenfor dette område. Derfor var plassering av denne chipen svært viktig. For å redusere forstyrrelser rundt antennen, ble chipen plassert på kanten av kretskortet slik at antennen akkurat ligger utenfor kretskortet [23]. Dette vises på figur 3.10. Kretskortene ble relativt like, men den store forskjellen er de fire gule viahullene. I figur 3.10a, går alle viahullene direkte til I2C skjermen. De to øverste i figur 3.10b er beina til IR-LEDen, mens de to nederste er beina til fototransistoren.

Et annet tiltak var å samle alle komponentene på en side for å gjøre det enklere å sette de på. Dette gjorde det enklere og billigere. For at kortet ikke skulle komme i veien for roboten, kunne ikke kretskortet være for stort. Faren med å komprimere kortet for mye, kunne gjøre jobben med lodding og feilsøking vanskeligere. Derfor har det vært viktig å tenke på plassering og ikke sette komponentene for nærme hverandre.

Det ble sett nærmere på både bredden og lengden til ledningsbanene. Lengden bør minimeres slik at det ikke blir for mye tap i linjene. Når det gjelder bredden til ledningsbanene er det viktig å tenke over hva slags signaler som skal gå gjennom de ulike banene. De banene det skal gå mye strøm i, er det viktig å øke bredden på. Dette gjelder rundt mikro USB-en og DC jacken. Dette kan ses på figur 3.10, på venstre side av begge kortene. Hvis banene er for smale, medfører det ofte overoppheting. Når bredden skulle bestemmes, ble det sjekket opp mot IPC standarder. IPC er en organisasjon som prøver å standardisere monterings- og produksjonskrav av elektronisk utstyr, blant annet kretskort [24]. IPC 2152 standard tar for seg hvordan størrelse på ledningsbaner og viahull bør velges. Der er det anbefalt at ved baner der det skal gå 1 A bør bredden på banene være 0.25 mm, mens baner der det skal gå opp mot 2 A, bør ha en bredde på 0.76 mm [25].

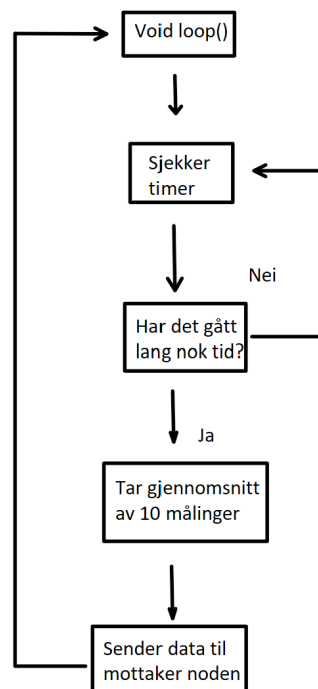
Jordingsstrategi er også et viktig hensyn. I tilfeller der kretskortet har flere lag, er det gjerne ønskelig å ha egne dedikerte jord-, signal- og spenningsplan. Dette går ikke ved et tosidig kretskort. En måte å optimalisere jording på var å bruke åpne og ubrukte områder på begge sider til jord. Dette kan være med å redusere støy, i form av Electromagnetic interference (EMI) [26]. Støy er gjerne noe som kan oppstå og kan forårsake problemer ved utvikling av elektronikk.

3.5 Sammenstilling av kretskort

Det endelige designet ble bestilt opp fra JLCPCB, som er en fabrikant fra Kina som produserer kretskort. Denne bedriften har godt rykte på seg for å være raske til å produsere og levere kort, men samtidig lage kort av høy kvalitet. De tilbyr også loddeservice, hvor de kan lodde komponenter på kortet. På grunn av lite erfaring og begrenset med utstyr på universitetets lokaler ble det tatt et bevisst valg om å bestille enkelte komponenter ferdig loddet. Disse komponentene var ESP32, UART-USB bridge, schottky dioden og de tre minste diodene ved USB porten. Dette var også komponenter JLCPCB hadde inne på sitt lokale lager, slik at det ikke gikk ut over bestillingstiden. Resten av komponentene ble bestilt eksternt, og loddet fast på kortet ved hjelp av varmluftspistol og loddepasta på universitetets lokaler.

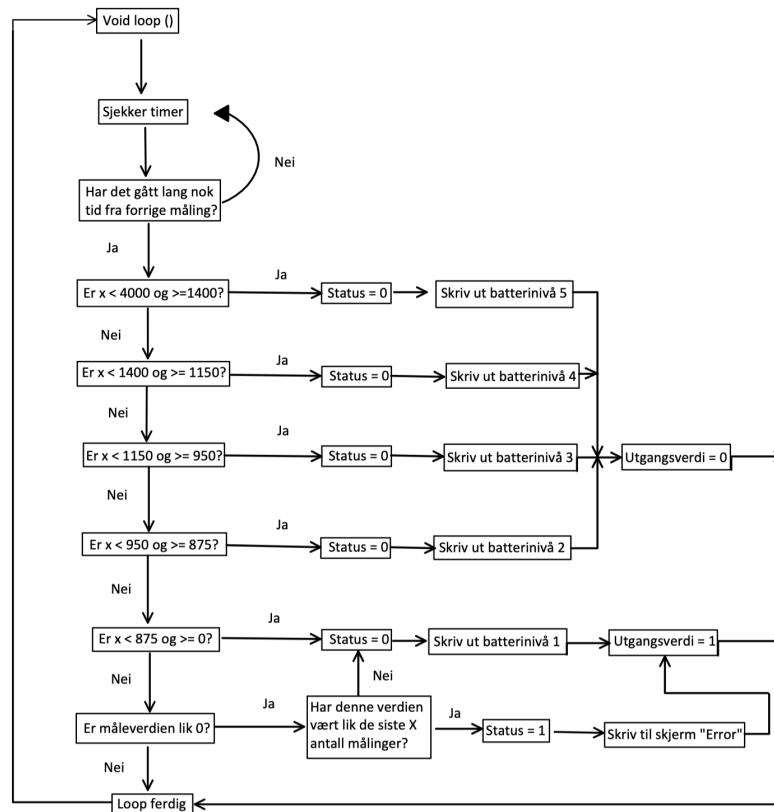
3.6 Kodeforklaring

Programmering av mikrokontrolleren, står sentralt i prosjektet. Det er den som sørger for når og hvordan målingene blir gjort, samt hvordan denne dataen blir prosessert. I figur 3.11 er det laget et blokkdiagram som beskriver loop-funksjonen til sendernoden. Det er altså den delen av koden som kjøres kontinuerlig.



Figur 3.11: Blokkdiagrammet til sendernoden.

Det vises her at sendernoden ikke har noen kompliserte grener. Hovedoppgaven til sendernoden er å ta målinger, for så å sende dataen til mottakernoden. Før det sendes, er det både lagt inn en timer og en utregning av gjennomsnittet til de ti siste målingene. Limnivået vil ikke endre seg veldig fort, det er derfor hensiktsmessig å ha en timer som sørger for at målingene og sending av data ikke blir gjort hele tiden. Den sjekker om det har gått lang nok tid ut ifra ønsket intervall. Hvis det ikke har gått lang nok tid, kjøres koden og sjekker timeren helt til det har gått lang nok tid. Kodesnutten som utfører dette, er kodeliste 3.2. Gjennomsnittet blir tatt hovedsakelig for å utelukke enkelte feilmålinger. Deretter sendes dataen i form av denne gjennomsnittsmålingen via kommunikasjonsprotokollen ESP-NOW over til mottakernoden. Denne kommunikasjonsprotokollen krever et sett med funksjoner for å fungere. Oppsettet er vist under overskriften ESP-NOW fra linje 11 til 42 av koden til sendernoden i vedlegg C. Utførelsen av sendingen ligger mellom linje 62 og 71.



Figur 3.12: Blokkdiagrammet til mottakernoden.

I figur 3.12 vises blokkdiagrammet for mottakernoden. Den tar også for seg loop-funksjonen i koden. Timeren er den samme som blir benyttet i loop-funksjonen til mottakernoden. Da det har gått lang nok tid vil mottakeren sjekke om målingene, definert med bokstaven «x», er mellom ulike nivåer. Da riktig intervall er målt oppdateres en statusvariabel, lik «0», og en utgangsvariabel. I tillegg skrives det tilsvarende batteriikonet til skjermen. Hvis det oppstår målinger lik «0», tyder all testing på at det har skjedd en feil. For å være sikker på at det ikke er en enkel feilmåling, er det lagt inn en teller som holder styr på hvor mange feilmålinger som skjer. Hvis det oppstår ti feilmålinger etter hverandre blir statusvariabelen satt lik «1». I dette tilfelle vil mikrokontrolleren skrive ut «Error!». Da vet operatøren at det har skjedd noe og kan undersøke hva som har gått galt. For å se nærmere på det som skrives til skjermen, se vedlegg D.

Utgangsvariabelen vil i utgangspunktet være «0», bortsett fra da det oppstår gjentatte feilmålinger og ved det laveste batterinivået. Et av kravene var at systemet kunne gi ut et signal på 24 V til en robot. Da kan denne variabelen bli brukt. Da en av disse to tilfellene inntreffer blir utgangen satt høy, slik at det kan brukes på et relé. Dette kan da gi ut det ønskede 24 V signalet som kan brukes til å stoppe roboten. Dette her utføres i kodeliste 3.1.

I kodeliste 3.1 vises oppbygningen av switch-case setningen for status ut ifra målingene som blir gjort av fototransistoren. Det denne gjør er å printe enten «Error!» eller de faktiske målingene som blir gjort. Error brukes når det har blitt målt «0» et gitt antall ganger. Denne målingen skal som nevnt aldri skje, og det vil derfor bety at noe har gått galt. Telleren som benyttes vises på linje 12 til 16 i kodeliste 3.1. På linje 20 til 25 blir status satt til enten «0», eller «1». «0» betyr at målingene som blir gjort skal printes. «1» betyr at det har blitt målt «0» et gitt antall ganger, og en error beskjed blir gitt.

Kodeliste 3.1: Case setning

```
1 // Blir deklartert over loop og setup
2 int count1 = 0;
3 int status = 0;
4
5 void loop() {
6     .
7     .
8     .
9     // Teller
10    if (x == 0) {
11        count1 += 1;
12    } else {
13        count1 = 0;
14    }
15
16    // Statussjekk
17    if (count1 > 10) {
18        status = 1;
19    } else {
20        status = 0;
21    }
22
23    switch (status) {
24        case 1:
25            // Displayer error på skjermen
26            break;
27        default:
28            // Viser de forskjellige nivåene på skjermen
29            .
30            .
31            .
32    }
```

I kodeliste 3.2 vises bruken av `millis` funksjonen. Dette er en funksjon som returnerer tiden i millisekunder, som sørger for en presis måte for å beregne tid. Denne er gunstig å bruke når det ønskes at noe i koden skal gjøres i et gitt tidsintervall, uten å hindre resten av koden fra å kjøre. Den kan minne om `delay` funksjonen, men den store forskjellen er at `delay` vil stoppe hele koden i en gitt tid, og `millis` er derfor å foretrekke i de fleste tilfeller [27].

Denne metoden fungerer ved å definere et ønsket intervall som er fast, «`intervalSending`». Hver gang koden kjøres oppdateres en variabel, kalt «`naaverendeMillis`», som angir hvor lang tid det har gått siden oppstarten av koden. Denne tiden blir videre sjekket opp mot en annen variabel, «`forrigeMillis`». Det «`forrigeMillis`» lagrer er tiden for forrige hendelse i forhold til oppstart. Hvis differansen mellom de siste to variablene er større eller lik det ønskede intervallet så gjør koden sine målinger. I tillegg blir variabelen «`forrigeMillis`» oppdatert ettersom det har skjedd en ny hendelse. Dette vises i linje 10-12 i kodeliste 3.2. Slik fortsetter koden om og om igjen for å sikre at målinger og andre hendelser skjer med ønsket intervall.

Noe å legge merke til er at det blir benyttet `unsigned long` på variablene som brukes til å lagre tiden. Dette blir gjort fordi `long` består av 32 bit, mot 16 bit for `integer`. `Unsigned` betyr at variablene kun har positive verdier, som gjør den hensiktsmessig å bruke når det kommer til tid. 32 bit vil da si at den kan telle i rundt 49 dager før den må startes på nytt, som er nok ettersom den vil restarteres daglig. Skulle det derimot ha blitt brukt `integer`, med 16 bit, vil dette si rundt ett minutt [28].

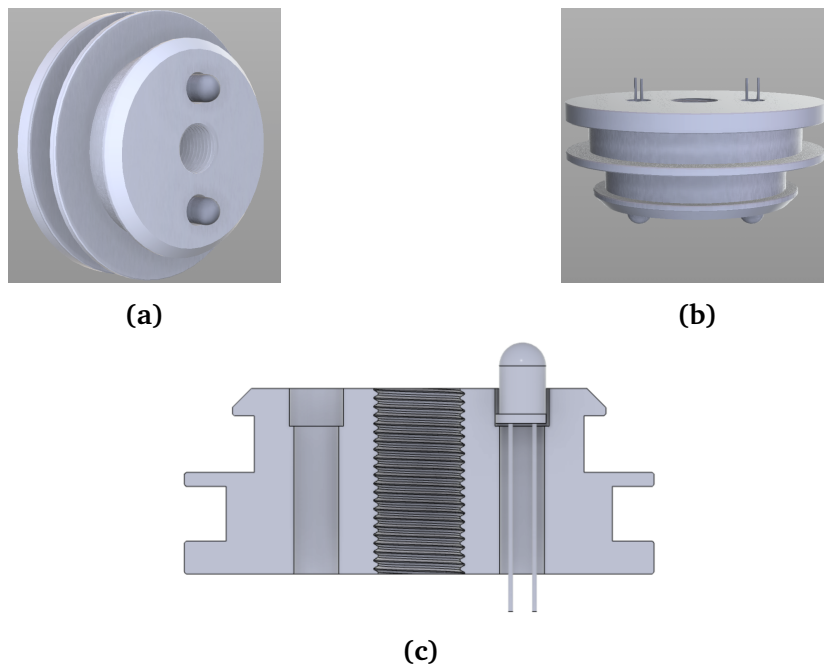
Hvis det skulle være ønskelig å endre intervallet mellom sendingen senere så er det enkelt å gjøre. Det eneste som trengs å endres på er verdien til variabelen, «`intervalSending`». Den skal være ganske allsidig ettersom intervallet kan være ganske stort, slik som det er forklart i avsnittet over.

Kodeliste 3.2: Bruk av `millis` funksjon

```
1 // forrigeMillis og intervalSending blir deklartert utenfor setup og loop
2 unsigned long forrigeMillis = 0;
3 const long intervalSending = 2000;
4
5 void loop() {
6   unsigned long naaverendeMillis = millis();
7   if (naaverendeMillis - forrigeMillis >= intervalSending) {
8     // Oppdater forrigeMillis når betingelsen stemmer
9     forrigeMillis = naaverendeMillis;
10    .
11    .
12    .
13  }
14 }
```

3.7 Modifisering av limdispenseren

Toppen på limdispenseren ble demontert for å kunne modifiseres. For å kunne lage en nøyaktig modell, og forsikre om at den passer i dispenseren, ble 3D printing med resin benyttet. Resin er et mer solid materiale enn det som benyttes i tradisjonelle 3D printere, og gir høyere nøyaktighet. Hapro hadde ikke tegningene til denne delen. 3D skanning ble derfor testet for at toppen enkelt kunne modifiseres på PC-en, og deretter printes ut. Dette fungerte ikke, mye på grunn av den glatte overflaten til delen. Derfor ble hele delen rekonstruert i CAD-programvare, og tilpasset de forbedringene som trengtes. Fototransistor, og IR-LEDen ble senket delvis ned i toppen, slik at de skulle sitte stabilt. Her er det viktig å legge merke til at hullene til sensorene har større diameter de første millimeterne fra innsiden, enn diameteren på utgangen. Det er derfor ikke fysisk mulig at komponentene vil presses gjennom hullet og ut på den andre siden, som vist på figur 3.13c. Deretter ble det fylt igjen med Tec7, for å lime dem fast i hullet. Dette sørget for at det ble lufttett, og var avgjørende for at systemet skulle fungere. På figur 3.13 er det rekonstruerte lokket, med plass til fototransistoren og IR-LEDen, vist.



Figur 3.13: Lokket som ble 3D printet med resin printer.

Kapittel 4

Målinger og resultat

Her presenteres de endelige testene som er utført. Første del av testingen ble utført ved hjelp av en Arduino, for å få et grunnlag for testingen med det ferdige systemet. Testingen på selve systemet er delt inn i to, hvorav den første kalibreringen er utført med en prototyp på NTNU's lokaler. Den andre er kalibreringen fra bedriftsbesøket hos Hapro, utført med deres utstyr. Disse to resultatene er noe avvikende fra hverandre, og grunnen til at resultatene ser ut som de gjør, er beskrevet nedenfor i avsnitt 4.3 og avsnitt 4.4. Videre diskusjon rundt bakgrunnen for disse resultatene finnes i diskusjonsdelen avsnitt 5.1.

4.1 Testing av tidlig prototyp

Testmålinger før montering av de endelige kretskortene, er utført med en prototyp bestående av en Arduino Uno mikrokontroller. Denne har en annen ADC enn ESP32. Arduinoen har en innebygd 10 bit ADC. Kretsen er bygd rundt utviklingskortet på et koblingsbrett, og testet. IR-LEDen og fototransistoren er de samme komponentene som er brukt i det endelige produktet. Koblingsskjema finnes i vedlegg F. Det ferdige kretskortet har derimot en ESP32 mikrokontroller, og denne er utstyrt med en innebygd 12 bit ADC. Disse ADC-ene fungerer på samme måte, og bruker den samme teknikken for å konvertere. Den eneste forskjellen er oppløsningen. Derfor vil verdiene som er skrevet ned og testet i vedlegg G og vedlegg H være forskjellig fra verdiene som er testet fram på det endelige kretskortet.

4.2 Testing og kalibrering

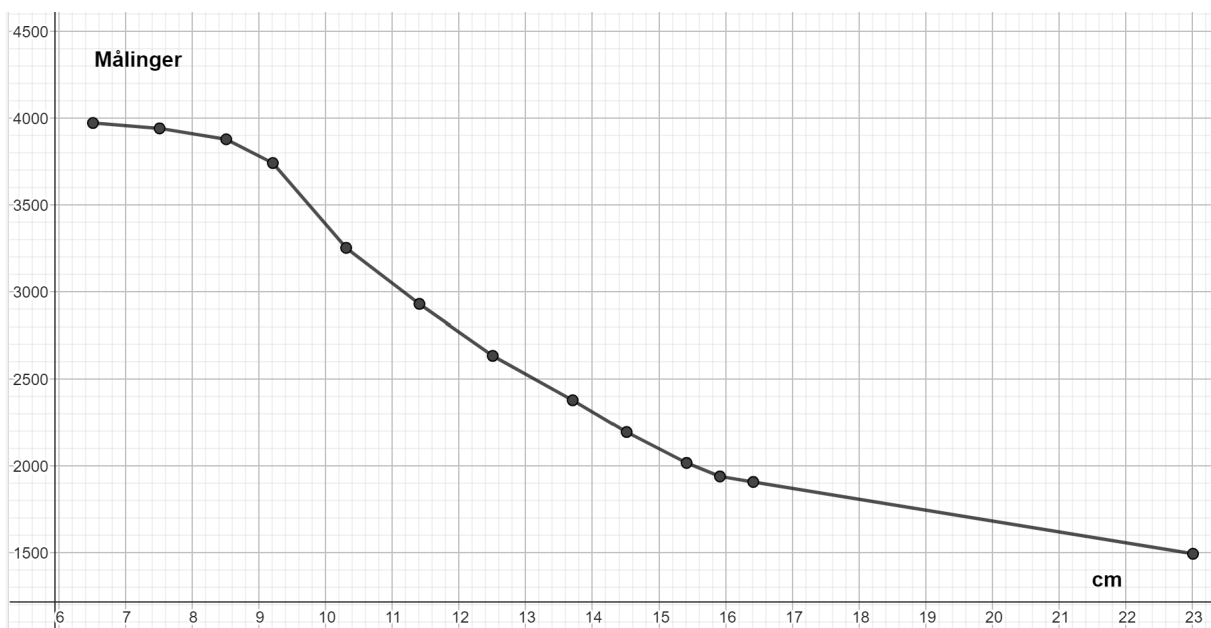
På skolen ble det benyttet en luftkompressor for å simulere trykket som benyttes for å presse ut lim hos Hapro. For å være forsiktige og ikke ødelegge tuben, og for at limet ikke skulle komme ut i for stor hastighet, ble det brukt et trykk på rundt 3 bar gjennom nesten hele testen. Målingene ble tatt ved å måle nivået på limet i tuben, målt i cm fra toppen, deretter målingen under trykk med sensoren. Dette ble gjentatt gjennom hele tuben, med målinger

rundt hver cm.

Testen hos Hapro ble gjennomført på en liknende måte som den utført på skolen. Limnivå i dispenseren ble også denne gangen forsøkt sjekket med jevne mellomrom. Maskinen som blir brukt for limet fungerte ikke som den skulle, noe som vil bli diskutert nærmere i kapittel 5. Dette hadde store konsekvenser for kalibreringstesten hos Hapro.

4.3 Resultat av målingene på skolen

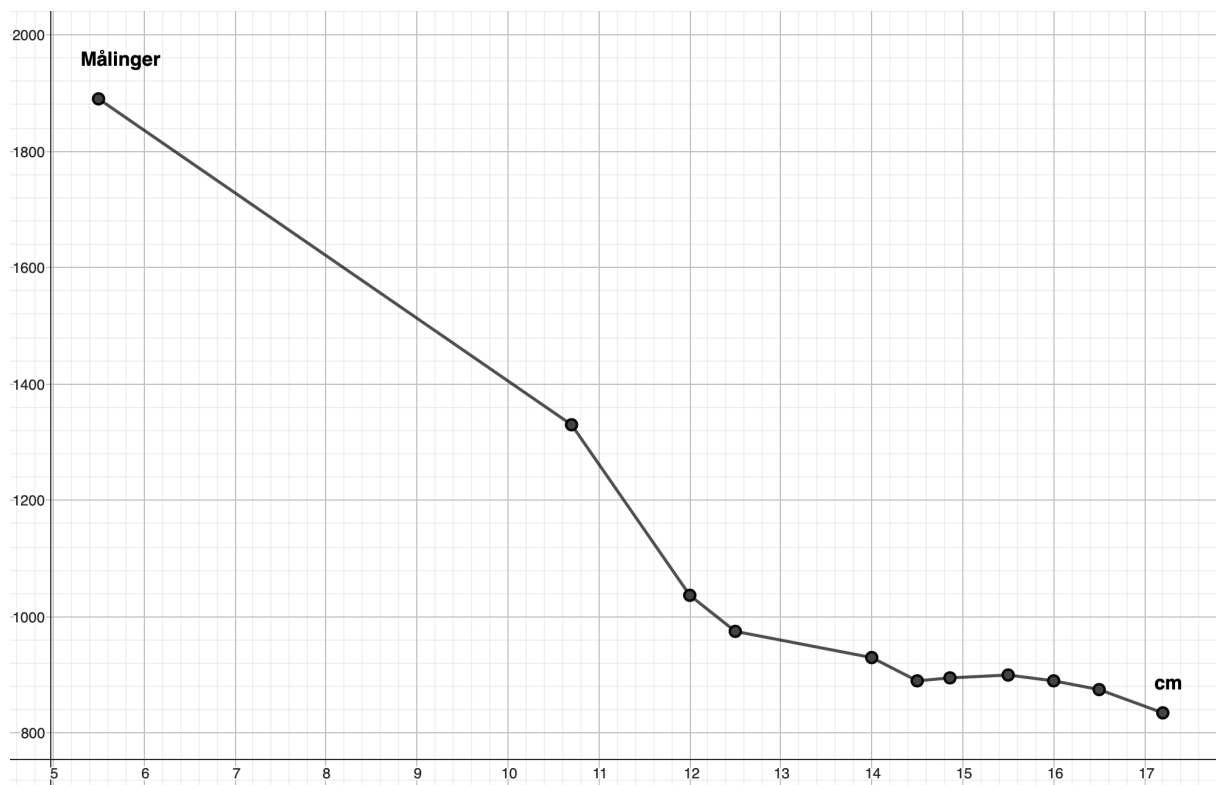
I figur 4.1 vises resultatene av de første målingene gjort under trykk på 3 bar. Figuren gir et godt overblikk over endringen ved forskjellige nivåer. Målingene er tatt med hensyn til at Hapro ønsker å bytte ut limpatronen ved rundt 16 cm avstand fra toppen, og det er derfor tettere med målinger rundt dette punktet. 16 cm tilsvarer avstanden fra toppen av limpatronen og ned til limet, ved nivået der kvaliteten på limingen begynner å svekkes. Figuren viser at endringen er minst på toppen, samtidig som den flater noe ut i bunnen. Den er mest sensitiv i det midtre området på limpatronen. Dette er et godt første resultat med forutsigbar endring i verdi når nivået i limpatronen synker, som viser at sensoren er godt egnet til dette formålet. Siste del av testen ble utført ved et lufttrykk på 6 bar, for å teste at komponentene tålte trykket. Ettersom det kun var én limpatron tilgjengelig for testen, måtte stresstesten på komponentene utføres på samme limpatron på skolen. Dette gjorde at limnivået sank raskt til bunnen. Det ble derimot bekreftet at komponentene tålte trykket. Dette er grunnen til den store avstanden mellom de siste to målingene i figur 4.1.



Figur 4.1: Kalibrering av sensor.

4.4 Resultat av kalibreringen hos Hapro

Grafen i figur 4.2 viser derimot at limet kom ut fra tuben i et stort tempo, at det ble vanskelig å estimere hvor mye lim som ble presset ut til enhver tid. Derfor er det større avstand enn ønskelig mellom flere av målepunktene. Resultatet blir likevel presentert ettersom det viser til en annen viktig faktor fra testen, limfarge. Grafen viser en tydelig forskjell fra målingene gjort på skolen, hvor målingene er omtrent halvparten så høye fra testen hos Hapro. En av grunnene til dette er forskjellen på limet i de forskjellige tilfellene. Under testingen på skolen ble hvitt lim brukt, mens Hapro bruker blankt lim. Dette fører til en betydelig forskjell i hvor mye stråling som vil reflekteres tilbake. I avsnitt 5.1 vil dette aspektet bli videre diskutert. Siden nivået ikke ble tatt fra en helt full limpatron, ble det ingen nøyaktig måling av toppunkt, men det kan antas at det ligger et sted rett over 2000 på grafen. Det ønskede nivået hvor limpatronen skal byttes ligger en plass mellom 800 og 900. Målingene rundt dette nivået ble noe usikre, og verdien ble ikke helt satt. Det kan være flere grunner til dette. Måten testen ble utført på, er utpekt som hovedgrunnen. Hvorfor denne testen ikke ble helt vellykket vil bli diskutert nærmere i avsnitt 5.1. Det er kritisk med en god kalibrering av systemet. Dette ble tydeliggjort da fargen på limet i limpatronen ble byttet, som ga store påvirkninger på refleksjonen til strålingen.



Figur 4.2: Målingene av nivåene hos Hapro.

Kapittel 5

Diskusjon

Resultatene fra testene utført både på skolen, og de utført hos Hapro, gir mye rom for tolkning. Det er flere sammensatte grunner til at resultatene ble som de ble. Derfor er det viktig å ta hensyn til mange av detaljene som er kommet fram til i denne diskusjonen, før en eventuell videreutvikling av prosjektet. Det er også diskutert hvilke umiddelbare forbedringer som burde bli inkludert ved en ny utførelse av prosjektet. Til slutt finnes det noen aspekter rundt hvor bærekraftig og økonomisk systemet er, samt de etiske faktorene til løsningen. Dette er faktorer som bør overveies, før en videre implementasjon av et system slik som beskrevet i denne rapporten.

5.1 Diskusjon av resultatene

Testingen og kalibreringen hos Hapro gikk ikke helt som planlagt. Planen var å teste systemet på limdispenseren den skal brukes på. Dette gikk ikke da dysen på maskinen var tett, som førte til at det ikke kom ut lim fra maskinen. En viktig faktor som ble erfart var at komponentene som er utsatt for trykk tålte trykket, som var en større usikkerhetsfaktor før testingen.

Som beskrevet i avsnitt 4.2 måtte kalibreringen hos Hapro, utføres på samme måte som testingen på skolen. I tillegg til at dysen til maskinen var tett så var det en utfordring at tuppen til limpatronen manglet. Selv om trykket ble satt ned fra 6 til 2 bar, kom det veldig mye lim ut. Dette medførte at trykket kun var på i omtrent ett til to sekunder før det måtte skrus av. Grunnen for at dette ble gjort var for å kunne måle hvor langt nivået hadde endret seg i cm. Resultatet av dette gjorde at målingene ikke ble helt realistiske grunnet to faktorer. En faktor var bruken av gjennomsnittsmålinger, som nevnt i avsnitt 3.6. Den andre faktoren var observasjonen av målingene som gikk opp, mellom 20-50 verdier, da det ble satt på trykk. Grunnet at trykket var til stede i ett til to sekunder, gjorde disse to faktorene det vanskelig å få nok data, samt at verdiene ikke stabiliserte seg i forhold til trykket. Ved testingen på skolen ble tuppen som fulgte med brukt, slik at mengden lim som kom ut var mye mindre,

selv med samme trykk. Dette gjorde det mulig å ha trykket på i en lenger periode, oppmot 15 sekunder. Her fikk målingene stabilisert seg med tanke på trykket. Målingene ble derfor mer realistisk opp mot normal drift. Ved å sammenlikne figur 4.1 og figur 4.2, er det ikke like god sammenheng mellom hvert punkt. Verdiene ved testingen hos Hapro ble noe mer ustabile. En mulig årsak er derfor at trykket kun var på i korte tidsintervaller.

Det ble diskutert med oppdragsgiver angående måleverdiene som økte da det var trykk til stede. Det at Tec7 limpatronen er laget av plast og at veggene er tynne kan være nok til at tuben vil bøyes og utvides litt når den blir utsatt for trykk. Dette vil naturlig påvirke resultatet, hvis målingene med og uten trykk sammenliknes. Derimot vil ikke dette være problematisk under drift, da det vil være konstant trykk i dispenseren.

Kapittel 4 viser målingene gjort i de to testene. Som nevnt i avsnitt 4.4 er det stor forskjell på måleverdiene til de to forsøkene. Hovedårsaken til dette er på grunn av fargen til limet. Mengden stråling som reflekteres er veldig forskjellig ut ifra både overflaten og fargen til materialet. Limpatronen som ble brukt på skolen hadde hvitt lim, mens den hos Hapro hadde blankt lim. Den med blankt lim gir en mørkere innside av limpatronen, enn hva den med hvitt lim gjør. Det hvite limet vil da føre til at mer av strålingen reflekteres tilbake, og med det gir høyere målinger. Dette gjorde at målingene fra den første testen ikke kunne brukes for å indikere når tuben er tom, derimot kan den brukes som en god indikasjon på at sensoren fungerer.

5.2 Forbedringer/videreutvikling

En videreutvikling av systemet er å koble det til WiFi. Slik vil det være mulig å vise fram nivåmålinger for limdispenserne på en nettside, eller å plote inn dataene i et system. På den måten oppnås det større rekkevidde til hvem som kan se nivåene i dispenserne. I tillegg vil det gi mulighet for automatisk kontroll på hvor raskt limpatronen brukes opp, og kan automatisere for eksempel innkjøp. En del av bakgrunnen til at mikrokontrolleren i prosjektet er valgt, er nettopp denne muligheten til WiFi. Det er en programmeringsoppgave å legge inn riktige funksjoner, som WiFi, til mikrokontrolleren. Det lar seg derfor enkelt legge til i ettertid av dette prosjektet. Samtidig er det et sikkerhetshensyn å ta for en bedrift å ha små og noe sårbare enheter tilkoblet nettet. Dette kan være en litt skummel bakvei for uvedkommende inn på deres nett.

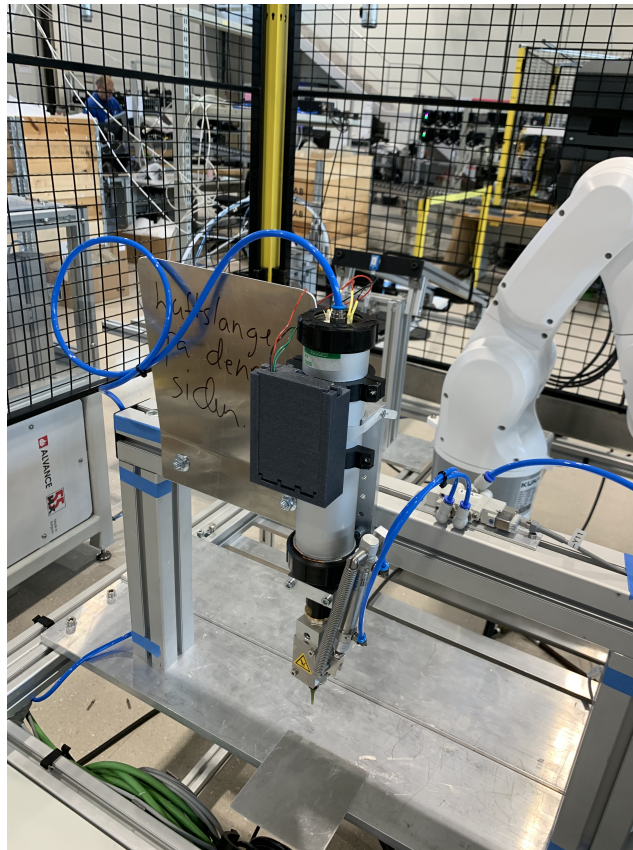
Et annet punkt som kan tas tak i til neste revisjon av prosjektet, er å fokusere på å gjøre det mer kompakt. Det er tatt enkelte valg under utvikling av produktet, som ikke er plassbesparende. For at prototypen som er laget, skulle være mulig å loddes for hånd, er det lagt inn god avstand mellom komponentene. Kretskortet kunne vært mulig å lage mindre i fysisk størrelse ved neste forsøk. Det samme gjelder da innkapslingen av kretskortet, som blir større da kretskortet er større enn nødvendig. Systemet vil da være enklere å håndtere ved mindre størrelse.

Ved gjentatte sykluser av og på med systemet, kan det henge seg opp. Dette skjer for eksempel i det strømmen blir slått på igjen etter at den har vært avslått. For å få den i gang igjen, er det nødvendig å trykke på knappen merket «EN» nede til venstre på kortet. Denne knappen restarter programvaren som kjører på mikrokontrolleren. Knappen sitter på kretskortet, og dermed også inne i innkapslingen. For å restarte denne når den henger seg opp må dekselet til kretskortet av, for å få tilgang til knappen. Denne kunne vært mer tilgjengelig i tilfelle dette er en situasjon som skulle oppstå ofte.

Når det kommer til komponentvalget til kretskortet, er det tatt enkelte valg som ville blitt revurdert ved en ny revisjon. Disse valgene ble tatt for å gjøre prosjektet enklere å gjennomføre. Selve mikrokontrolleren og IC-en ved navn CP2102N, ble loddet profesjonelt av JLCPCB, firmaet som lagde kretskortet. Dette ble gjort fordi komponentene er kritiske for produktets funksjon, i tillegg ville det vært tilnærmet umulig å lodde på egenhånd. Grunnen til dette er de små pinnene og den korte avstanden mellom dem. Denne bevisste beslutningen viste seg i ettertid å være riktig. Mikro USB-en som ble montert på kretskortet hadde de samme utfordringene som IC-ene. Til tross for dette ble det forsøkt å lodde denne komponenten for hånd. Utfordringen løste seg etter av og på lodding av komponenten, gjentatte ganger. Etter en arbeidsdag med prøving og feiling, ble denne kritiske komponenten montert på en tilfredsstillende måte og fungerte til sin hensikt. Denne komponenten kan også bli profesjonelt loddet, som vil redusere sjansen for å påføre kretskortet unødvendig skade under montering. Dette avhenger også av at JLCPCB har den riktige komponenten liggende på lager. Det kunne også ha vært hensiktsmessig å kjøpe en mikro USB som er hullmontert, slik at den får styrke i flere solide ankerpunkter på kortet. En slik komponent blir ofte utsatt for en del krefter ved inn og utkobling av ledning for programmering.

Det ble avdekket et problem angående løsningen til feste av innkapslingen til kretskortet som gjør målingene. Slik det er designet nå skal innkapslingen festes på limdispenseren. Tanken bak dette var at det ville være enklest for operatørene som skal bytte limpatron, da det kun er nødvendig å dra ut strømtilførselen. Dette fungerte dårlig grunnet måten de fester tuben til roboten. De må skru på plass hele tuben og da kommer innkapslingen i veien. Dette gjør det vanskelig å få festet limpatronen hardt nok, for å oppnå en lufttett kobling mellom limdispenser og maskin. En mulig løsning, er å feste innkapslingen på en metallplate til venstre for dispenseren, som vist på figur 5.1. Dette gjør det mulig at operatørene kan feste limdispenseren slik som før. Måten dette kan løses på, er å gjøre plass på designet slik at magneter kan tas i bruk.

Utføringen av testene hos Hapro, hadde blandet resultat. Det var problemer i starten da det ikke kom målinger fra systemet. Problemet ble oppdaget og fikset på stedet, slik at det var mulig å teste videre. Ledningene til sensoren stikker rett opp fra toppen på limdispenseren, og er derfor utsatt for en del klemming og bøyning. Ved bytting av limpatron, må dispenseren skrus løs fra maskinen, og ledningene på toppen blir da klemt og bøyd. Under test oppstod et brudd i en av ledningene, som måtte repareres ved hjelp av hurtigkoblinger. Bruk av



Figur 5.1: Det ferdige produktet plassert i produksjonslinjen til Hapro.

mykere type ledninger, eller å rute ledningene annerledes mellom dispenser og kretskort, ville kunne unngått dette problemet.

Som følge av å flytte sendernoden over til metallplaten til venstre for limdispenseren, bør designet på koblingen mellom sensoren og sendenoden endres. At dispenseren blir skrudd på mens kretskortet står stille, vil medføre at ledningene opp fra sensorene i limdispenser blir vridd. Dette øker sannsynligheten for brudd i ledningene betraktelig fra første vridning. Her bør istedet operatøren koble ut de fire ledningene. Derfor er en mulig løsning å bruke hurtigkoblinger slik at det er raskt og enkelt å ta ledningene inn og ut. Hvis kompleksiteten ved bytte av limpatron øker for mye, forsvinner noe av brukervennligheten og tidsfordelen ved et slikt system. Derfor er det viktig å komme fram til en rask og enkel måte å operere systemet. Det er kritisk at de fire ledningene kobles riktig ved bytte av limpatron. Derfor vil for eksempel fargekoding på ledningene være en nyttig løsning.

På grunn av mye usikkerhet rundt målingene som ble tatt hos Hapro, vil videre kalibrering være helt nødvendig for videre bruk. Denne kalibreringen må foretas i det systemet den skal brukes i, under samme virkelighetsnære forhold. I dette tilfellet kalibrere nivåene med målinger fra riktig maskin, under riktig lufttrykk og med riktig farge på limet. Det viktigste i starten vil være å få registrert et nøyaktig bunnivå og toppnivå. Dette vil gjøre det lettere å bestemme resten av nivåene imellom. For å få et nøyaktig mål bør det noteres ned målinger fra sensoren i det limpatronen byttes. Blir dette utført på flere limpatroner, vil det

resultere i gode og forutsigbare verdier for de ulike nivåene. Det er viktigst at systemet er nøyaktig rundt bunnivået der limpatronen skal byttes, og nivået over, hvor det skal varsles at limtuben snart skal byttes. Hvis dette blir gjort, er det enkelt å legge inn de nye nivåene i koden. Se linje 303, 310, 317, 324 og 331 i vedlegg D for å se hva som må endres for de fem ulike nivåene.

5.3 Bærekraftige aspekter

Bærekraft er viktigere enn noen gang i dagens samfunn. Det handler om å ta vare på de ressursene som er tilgjengelig, og ikke kaste bort eller overforbruke dem. Dette er tydelig beskrevet i FN's bærekraftsmål nummer 12 [29]. Systemet forsøker å bidra til mer bærekraftig bruk av produksjonsavdelingens ressurser. Et system som varsler om nivå på en ressurssom blir brukt under produksjonen, vil gi full kontroll på hvor mye som brukes, og hvor fort det brukes. Tidligere var det nødvendig å se at kvaliteten på limingen ble dårligere før limpatronen ble byttet. Det er heller ønskelig å bytte for tidlig enn for sent, av kvalitetshensyn. Slik som det blir gjort nå kan Hapro risikere å måtte kaste flere produkter hvis byttet av limpatronen skjer for sent, derfor er det ønskelig å bytte limpatronen før dette oppstår. Automatisering fører til at man kan vente med bytte av limpatronen til det nøyaktige bunnivået, og får dermed utnyttet så mye som mulig av limet.

5.4 Økonomiske aspekter

Et slikt system som overvåker limnivå i dispenserens og varsler før den går tom, vil potensielt kunne spare bedriften for noen feilproduksjoner. Ved feil liming under produksjon, må det brukes verdifull tid på å rengjøre produktet, eller potensielt kaste det. I tillegg vil et system som dette standardisere bunnivået til limet, og hindre at limpatronen byttes for tidlig på grunn av usikkerhet rundt limnivå. En arbeidsoppgave mindre for operatøren vil også spare bedriften tid, som igjen alltid betyr økonomiske fordeler for bedriften i det lengre løp. Da kan ressursene brukes andre plasser. Prosjektet er også skalerbart, og vil kunne utvides i stor grad. Om ønskelig er det mulig å montere den samme sensoren på flere limdispensere, og montere flere skjermer på mottakeren. Besparelsene per enhet er ikke store, men ved et større antall sensorer, vil det være mulig å skalere de økonomiske gevinstene også.

5.5 Ethiske sider av prosjektet

Etikken rundt automatisering er mye diskutert, og mange er kanskje redd for at en del arbeidsplasser skal forsvinne om en del av arbeidsoppgavene automatiseres. Om bedriften håndterer en omstilling på en god måte, vil de kunne flytte ressursene sine, og ikke kvitte

seg med arbeidere som ikke trengs. Omskolere og omstilling blir derfor viktig for en bedrift i en slik situasjon, slik at de kan flytte arbeidstakerne til nye arbeidsoppgaver. Det større bildet av dette vil være at bedriften kan drive virksomheten sin mer lønnsomt, og at det alltid vil være behov for menneskelige egenskaper i en bedrift. De enkleste oppgavene kan derfor automatiseres, men det trengs alltid noen til å drive eller passe på denne maskinen. Selv om det i dette prosjektet automatiseres sjekking av limnivå, vil det fremdeles være behov for en operatør til å bytte limpatronen. Ved automatisering av en del prosesser, vil også kvaliteten og resultatet automatiseres. Sikring av kvalitet på produktet er vesentlig for en produserende bedrift. Ringvirkninger av at en slik bedrift går mer lønnsomt og gjør det bedre i markedet, vil kunne være behov for enda flere ansatte i framtiden.

Kapittel 6

Konklusjon

Denne oppgaven ble gitt av Hapro Electronics AS, og følgende problemstilling ble utarbeidet:

«Hvordan lage et sensorsystem som måler limmengde i en dispenser, og gi et varselingssignal slik at patronen kan bli byttet før kvaliteten på limingen blir svekket?»

Ønsket var derfor et varslingsystem som kunne varsle når limpatronen måtte byttes. Det var ønskelig med to noder, som kommuniserte trådløst med hverandre. Den ene var da tenkt å sitte i nærheten av limpatronen, for å ta målingene, mens den andre skulle ta imot dataen, og vise det på en skjerm. Et annet ønske var et signal som kunne bli sendt til roboten når limpatronen var tom. Valget falt da på en fototransistor, med sensitivitet sterkest rundt IR-stråling. Denne ble plassert i toppen av tuben, sammen med en IR-LED. Dette førte til at fototransistoren registrerte forskjellig mengde stråling i dispenseren, ut ifra nivået til bunnen, og fikk dermed presise målinger.

Med ønske om to noder, ble det også behov for to kretskort. Begge kretskortene ble bygget rundt mikrokontrolleren ESP32, som har flere gunstige funksjoner for å oppfylle kravene satt i problemstillingen. De fleste av komponentene ble loddet for hånd, men de minste og vanskeligste, som ESP32, ble bestilt ferdig loddet. Videre bestod det ferdige produktet av to 3D printede beholdere, for å gi beskyttelse til kretskortene.

Resultatene etter testingen var gode. Det ble gjennomført to tester, en på skolen og en hos Hapro. Testen på skolen ga gode resultater, og viste at sensoren fungerte som ønsket. Testen hos Hapro ble noe mer varierende ettersom maskinen ikke kunne brukes, men fikk fortsatt resultater som tilsa at sensoren fungerte.

Det ferdige produktet viser at dette konseptet fungerer og er gjennomførbart, med gode muligheter for videreutvikling. Målingene ved riktig kalibrering, var presise og repetitive, som er helt vesentlig for å kunne ta i bruk produktet. Med noen modifikasjoner vil dette være et produkt Hapro Electronics AS kan bruke og dermed få løst problemstillingen rundt presise målinger av limnivået til limpatronene sine.

Bibliografi

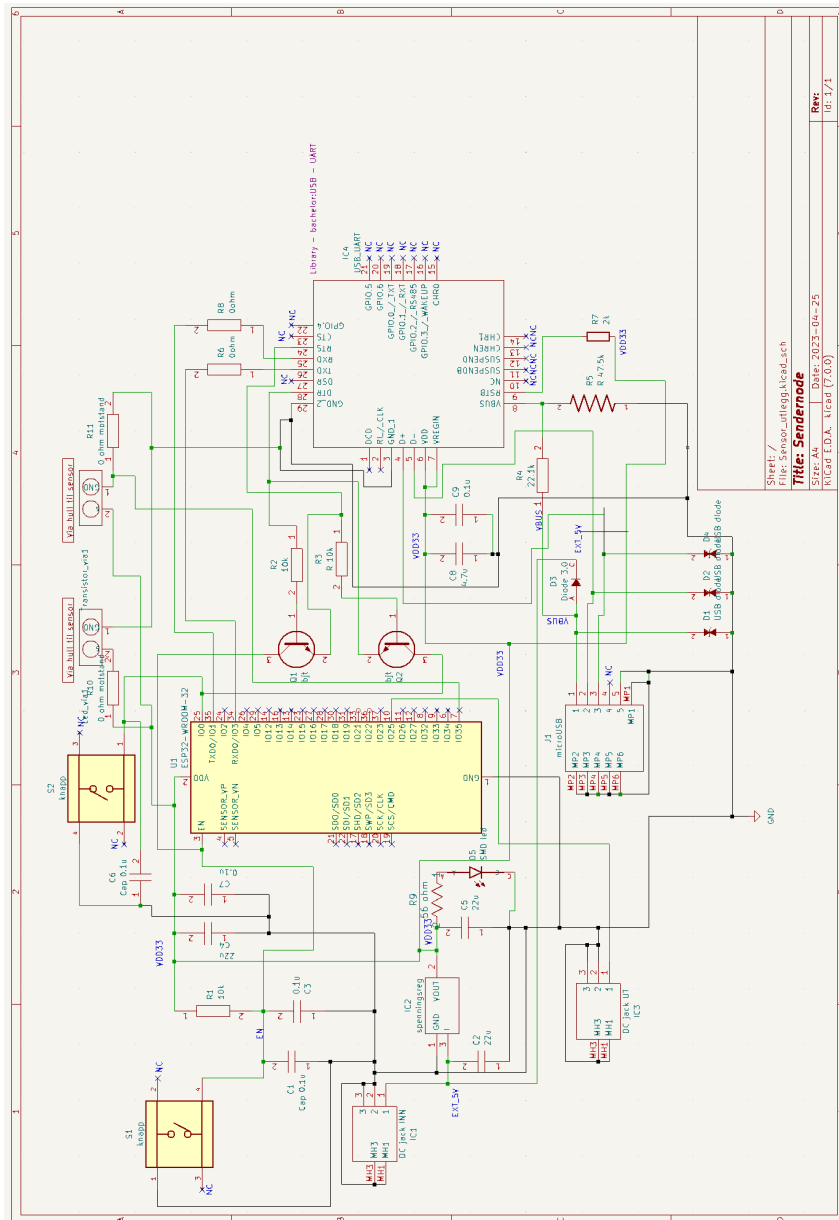
- [1] Hapro. «Førstevalget som profesjonell elektronikkprodusent og partner.» (2023), adresse: <https://haproelectronics.no/> (sjekket 19.05.2023).
- [2] Espressif. «ESP32 Series Datasheet.» (2023), adresse: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (sjekket 24.03.2023).
- [3] Espressif. «ESP32-WROOM-32 Datasheet.» (2023), adresse: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (sjekket 24.03.2023).
- [4] Sal Afzal. «I2C Primer: What is I2C? (Part 1).» (2023), adresse: <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html> (sjekket 27.03.2023).
- [5] Espressif. «ESP32 Technical Reference Manual.» (2023), adresse: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#uart (sjekket 27.03.2023).
- [6] Rohde & Schwarz. «Understanding UART.» (2023), adresse: https://www.rohde-schwarz.com/us/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html#gallery-6 (sjekket 24.03.2023).
- [7] Analog Devices. «Fundamentals of RS-232 Serial Communications.» (2023), adresse: <https://www.analog.com/en/technical-articles/fundamentals-of-rs232-serial-communications.html> (sjekket 24.03.2023).
- [8] Tamanna Sharma. «RS232 Serial Communication Protocol: Basics, Working & Specifications.» (2023), adresse: <https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications> (sjekket 24.03.2023).
- [9] Espressif. «ESP-NOW.» (2023), adresse: <https://www.espressif.com/en/solutions/low-power-solutions/esp-now> (sjekket 28.03.2023).
- [10] Silicon Labs. «USBXpress™ Family CP2102N Data Sheet.» (2023), adresse: <https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf> (sjekket 27.03.2023).

- [11] National Aeronautics and Space Administration, Science Mission Directorate. «Introduction to the Electromagnetic Spectrum.» (2010), adresse: https://science.nasa.gov/ems/01_intro (sjekket 24.03.2023).
- [12] James T. Schwiegerling. «The Physics of Light.» (2011), adresse: <https://www.healio.com/news/ophthalmology/20120331/the-physics-of-light> (sjekket 24.03.2023).
- [13] National Aeronautics and Space Administration, Science Mission Directorate. «Tour of the Electromagnetic Spectrum: Wave Behaviors.» (2023), adresse: https://science.nasa.gov/ems/03_behaviors (sjekket 28.04.2023).
- [14] Electronics Tutorials. «Bipolar Transistor.» (2023), adresse: https://www.electronicstutorials.ws/transistor/tran_1.html (sjekket 27.03.2023).
- [15] Circuit Globe. «Phototransistor.» (2023), adresse: <https://circuitglobe.com/phototransistor.html> (sjekket 27.03.2023).
- [16] Robert Keim. «What Is a Linear Voltage Regulator?» (2020), adresse: <https://www.allaboutcircuits.com/technical-articles/what-is-a-linear-voltage-regulator/> (sjekket 27.03.2023).
- [17] Infineon. «Voltage Regulator.» (2007), adresse: https://no.mouser.com/datasheet/2/196/Infineon-TLE4284-DS-v02_10-EN-1227764.pdf (sjekket 27.03.2023).
- [18] RS. «The Guide to Hall Effect Sensors.» (2023), adresse: <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/hall-effect-sensors-guide> (sjekket 29.03.2023).
- [19] Johannes Skaar. «permittivitet.» (2021), adresse: <https://snl.no/permittivitet> (sjekket 10.05.2023).
- [20] Jasper van Loenen. «image2cpp.» (2023), adresse: <https://javl.github.io/image2cpp/> (sjekket 17.04.2023).
- [21] Vishay. «Silicon NPN Phototransistor.» (2019), adresse: <https://no.mouser.com/datasheet/2/427/bpw96-1767039.pdf> (sjekket 29.03.2023).
- [22] Lite-on. «IR Emitter and detector.» (2013), adresse: <https://no.mouser.com/datasheet/2/239/HSDL-4260%20DS-1143990.pdf> (sjekket 29.03.2023).
- [23] Espressif. «ESP32 - Hardware Design Guidelines.» (2022), adresse: https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf (sjekket 29.03.2023).
- [24] IPC. «About IPC.» (2023), adresse: <https://www.ipc.org/about-ipc> (sjekket 30.03.2023).

- [25] Zachariah Peterson. «PCB Trace Width vs. Current Table for High Power Designs.» (2023), adresse: <https://resources.altium.com/p/pcb-trace-width-vs-current-table-high-voltage-design> (sjekket 30.03.2023).
- [26] Zachariah Peterson. «Understanding 2-Layer PCB Ground Planes.» (2023), adresse: <https://resources.altium.com/p/understanding-ground-planes-your-two-layer-pcb#gridding-and-pouring-ground-regions> (sjekket 16.03.2023).
- [27] Industrialshields. «Industrial Arduino millis () vs Delay ().» (2023), adresse: <https://www.industrialshields.com/blog/arduino-industrial-1/post/industrial-arduino-millis-vs-delay-248> (sjekket 20.04.2023).
- [28] Arduino. «unsigned long.» (2023), adresse: <https://www.arduino.cc/reference/en/language/variables/data-types/unsignedlong/> (sjekket 20.04.2023).
- [29] FN. «Ansvarlig forbruk og produksjon.» (2023), adresse: <https://www.fn.no/om-fn/fns-baerekraftsmaal/ansvarlig-forbruk-og-produksjon> (sjekket 19.05.2023).

Vedlegg A

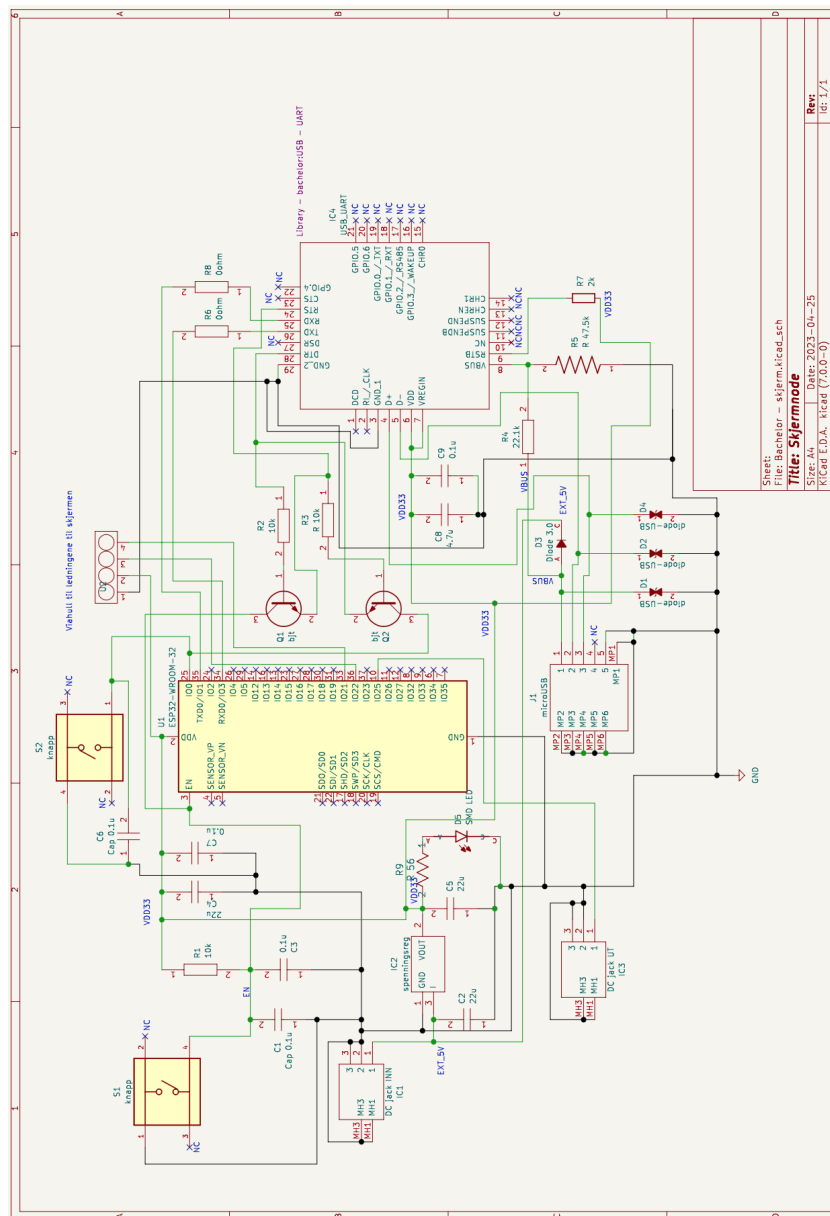
Kretsskjemaet til sendernoden



Figur A.1: Kretsskjemaet til sendernoden

Vedlegg B

Kretsskjemaet til mottakernoden



Figur B.1: Kretsskjemaet til mottakernoden

Vedlegg C

Koden til sendernoden

Kodeliste C.1: Koden til sendernoden

```
1 // ----- Nødvendig bibliotek -----
2 #include <esp_now.h>
3 #include <WiFi.h>
4
5 // ----- Deklarerer variabler -----
6 const int SensorPin = 35;
7 int SensorValue = 0;
8 unsigned long forrigeMillis = 0;
9 const long intervalSending = 2000;
10
11 // ----- ESP-NOW -----
12
13 // Mottakers MAC adresse, finnes ved hjelp av å kjøre koden i vedlegg E
14 uint8_t broadcastAddress[] = { 0xC8, 0xF0, 0x9E, 0x80, 0xF5, 0xF4 };
15
16 // Definer datatype for variabelen som blir motatt av senderen
17 // Viktig at denne datatypen samsvarer i begge kodene
18 typedef struct struct_message {
19     float a;
20 } struct_message;
21
22 // Pakker inn dataen som skal sendes i en struktur som kalles myData
23 struct_message myData;
24
25 // Henter informasjon om sammenkoblingen
26 esp_now_peer_info_t peerInfo;
27
28 void setup() {
29     // Starter serie overvåkeren
30     Serial.begin(115200);
31
32     // ----- Initialisering av ESP-NOW -----
33     // Funksjon for å sette ESP32 som WiFi stasjon
34     WiFi.mode(WIFI_STA);
35
36     esp_now_init();
37
38     // Registrer peer
```

```
39 memcopy(peerInfo.peer_addr, broadcastAddress, 6);
40 peerInfo.channel = 0;
41 peerInfo.encrypt = false;
42 esp_now_add_peer(&peerInfo);
43 }
44
45 void loop() {
46     // Oppdater timer-verdi
47     unsigned long naaverendeMillis = millis();
48
49     // Sjekker om det har gått lang nok tid mellom hver måling
50     if (naaverendeMillis - forrigeMillis >= intervalSending) {
51         // Oppdater forrigeMillis når betingelsen stemmer
52         forrigeMillis = naaverendeMillis;
53
54         // Leser av sensor
55         // Tar gjennomsnittsverdi av de 10 siste målingene til sensoren
56         long SensorValue = 0;
57         for (int i = 0; i < 10; i++) {
58             SensorValue += analogRead(SensorPin);
59         }
60         SensorValue /= 10;
61
62         // Tilordner pakken som skal sendes med gjennomsnittsmålingene
63         myData.a = SensorValue;
64         Serial.println(myData.a);
65
66         // Funksjon som sender pakken, samt henter informasjon om status av sendingen
67         esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *)&myData, sizeof(myData));
68         if (result == ESP_OK) {
69             Serial.println("Sending_vellykket");
70         } else {
71             Serial.println("Sending_mislykket");
72         }
73     }
74 }
```

Vedlegg D

Koden til mottakernoden

Kodeliste D.1: Koden til mottakernoden

```
1 // ----- Nødvendig bibliotek -----
2 #include <esp_now.h>
3 #include <WiFi.h>
4 #include <SPI.h>
5 #include <Wire.h>
6 #include <Adafruit_GFX.h>
7 #include <Adafruit_SSD1306.h>
8
9 // ----- Oppsett til skjermen -----
10 #define SCREEN_WIDTH 128 // Bredden til OLED skjermen, i piksler
11 #define SCREEN_HEIGHT 32 // Høyden til OLED skjermen, i piksler
12 #define OLED_RESET -1 // Reset pin
13 #define SCREEN_ADDRESS 0x3C // Adressen til skjermen
14 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
15
16 // 'Ikon 1', 128x32px
17 const unsigned char BatteriBitIkon_1[] PROGMEM = {
18   0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
19   0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff,
20   0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,
21   0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
22   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07,
23   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07,
24   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
25   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
26   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
27   0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
28   0xf0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
29   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
30   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
31   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
32   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
33   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
34   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
35   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
36   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
37   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
38   0xe0, 0x00, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x0e, 0x07,
```



```

151 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
152 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,
153 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xff,
154 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
155 };
156 // 'Ikon 5', 128x32px
157 const unsigned char BatteriBitIkon_5[] PROGMEM = {
158 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
159 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7f,
160 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,
161 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
162 0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07,
163 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x3c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
164 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
165 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
166 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
167 0xfc, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
168 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
169 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
170 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
171 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
172 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
173 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
174 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
175 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
176 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
177 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
178 0xe0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
179 0xf0, 0x00, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
180 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
181 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
182 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
183 0xff, 0xfc, 0x07, 0x00, 0x01, 0xe0, 0x00, 0x1c, 0x00, 0x03, 0x80, 0x00, 0x70, 0x00, 0x0e, 0x07,
184 0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07,
185 0xff, 0xfc, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07,
186 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07,
187 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f,
188 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xff,
189 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
190 };
191
192 // Array av de fem bitkartene.
193 const int BatteriBitallArray_LEN = 5;
194 const unsigned char *BatteriBitallArray[5] = {
195   BatteriBitIkon_1,
196   BatteriBitIkon_2,
197   BatteriBitIkon_3,
198   BatteriBitIkon_4,
199   BatteriBitIkon_5
200 };
201
202 // ----- Deklarerer variabler -----
203 int tellerKnapp = 0;
204 int count1 = 0;
205 int count2 = 0;
206 int status = 0;

```



```
207 unsigned long forrigeMillis = 0;
208 const long intervalSending = 2000;
209 float x;
210 float y;
211 int utgangsverdi = 0;
212
213 // ----- ESP-NOW -----
214
215 // Definer datatype for variabelen som blir motatt av senderen
216 // Viktig at denne datatypen samsvarer i begge kodene
217 typedef struct struct_message {
218     float a;
219 } struct_message;
220
221 // Pakker inn dataen som mottas i en struktur som kalles myData
222 struct_message myData;
223
224
225 // Funksjon som henter informasjon fra mottatt data
226 void OnDataRecv(const uint8_t *mac, const uint8_t *incomingData, int len) {
227     memcpy(&myData, incomingData, sizeof(myData));
228     // Lagrer mottatt data i en variabel, forenkler senere bruk av dataene
229     x = myData.a;
230 }
231
232
233 void setup() {
234     // Starter serie overvåkeren
235     Serial.begin(115200);
236     pinMode(utgangsverdi, OUTPUT);
237
238     // ----- Initialiserer skjermen -----
239     display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);
240     display.clearDisplay();
241     delay(2000);
242
243     // ----- Initialisering av ESP-NOW -----
244     // Funksjon for å sette ESP32 som WiFi stasjon
245     WiFi.mode(WIFI_STA);
246
247     // Initialiserer ESP_now protokollen
248     // Brukes for feildeteksjon
249     if (esp_now_init() != ESP_OK) {
250         Serial.println("Klarte ikke å initialisere ESP-NOW");
251         return;
252     }
253
254     // Funksjonen som mottar data
255     esp_now_register_recv_cb(OnDataRecv);
256 }
257
258 void loop() {
259     // Oppdater timer-verdi
260     unsigned long naaverendeMillis = millis();
261
262     // Sjekker om det har gått lang nok tid mellom hver måling
```

```

263 if (naaverendeMillis - forrigeMillis >= intervalSending) {
264     // Oppdater forrigeMillis når betingelsen stemmer
265     forrigeMillis = naaverendeMillis;
266
267
268     // Tellerfunksjon som sjekker gjentatte feilmålinger
269     if (x == 0) {
270         count1 += 1;
271     } else {
272         count1 = 0;
273     }
274
275     // Oppdaterer status for systemet
276     if (count1 > 10) {
277         status = 1;
278     } else {
279         status = 0;
280     }
281
282     //////////////////////////////////////
283     // Case som bestemmer hva som skal skrives ut til skjermen.
284     // Ved default, er det de ulike batterinivåene som skrives ut.
285     // Ved status = 1 skrives det ut "Error!", og utgangsverdi = 1,
286     // som kan brukes på et rele for å stoppe roboten.
287     //////////////////////////////////////
288     switch (status) {
289         case 1:
290             display.clearDisplay();
291             display.setTextColor(WHITE);
292             display.setTextSize(2);
293             display.setCursor(0, 0);
294             display.println("Error!");
295             display.display();
296             // Setter utgangen høy slik at roboten stopper da feilmåling oppstår
297             utgangsverdi = 1;
298             break;
299
300         default:
301             Serial.print("Verdi:");
302             Serial.println(x);
303             // Endre de to verdiene i linjen under for å endre intervallet til nivå 5
304             if (x < 4000 && x >= 1400) {
305                 display.clearDisplay();
306                 display.drawBitmap(0, 0, BatteriBitIkon_5, 128, 32, WHITE);
307                 display.display();
308                 utgangsverdi = 0;
309             }
310             // Endre de to verdiene i linjen under for å endre intervallet til nivå 4
311             if (x < 1400 && x >= 1150) {
312                 display.clearDisplay();
313                 display.drawBitmap(0, 0, BatteriBitIkon_4, 128, 32, WHITE);
314                 display.display();
315                 utgangsverdi = 0;
316             }
317             // Endre de to verdiene i linjen under for å endre intervallet til nivå 3
318             if (x < 1150 && x >= 950) {

```

```
319     display.clearDisplay();
320     display.drawBitmap(0, 0, BatteriBitIkon_3, 128, 32, WHITE);
321     display.display();
322     utgangsverdi = 0;
323 }
324 // Endre de to verdiene i linjen under for å endre intervallet til nivå 2
325 if (x < 950 && x >= 875) {
326     display.clearDisplay();
327     display.drawBitmap(0, 0, BatteriBitIkon_2, 128, 32, WHITE);
328     display.display();
329     utgangsverdi = 0;
330 }
331 // Endre de to verdiene i linjen under for å endre intervallet til nivå 1
332 if (x < 875 && x >= 0) {
333     display.clearDisplay();
334     display.drawBitmap(0, 0, BatteriBitIkon_1, 128, 32, WHITE);
335     display.display();
336     utgangsverdi = 1;
337 }
338 }
339 }
340 }
```


Vedlegg E

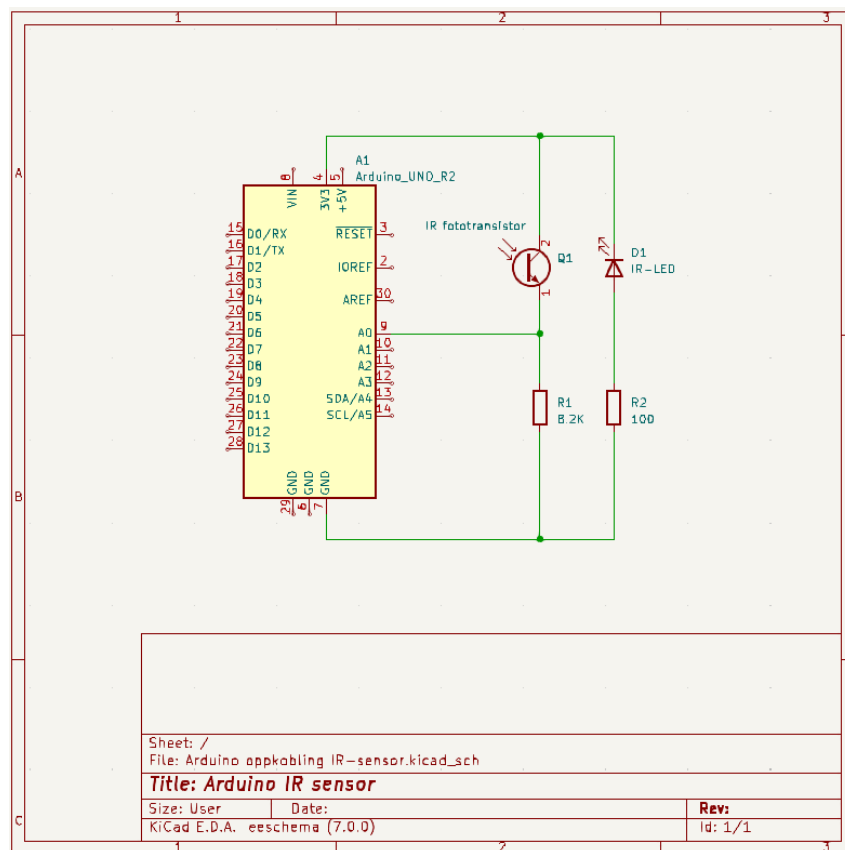
Koden for å finne MAC adresse

Kodeliste E.1: Finne MAC-adressen til ESP32

```
1  #include <WiFi.h>
2
3  void setup() {
4    Serial.begin(115200);
5
6    // Starter WiFi modulen i mikrokontrolleren, noe som tar litt tid
7    // derav påfølgende delay
8    WiFi.mode(WIFI_MODE_STA);
9    delay(1000);
10
11   Serial.println("MAC_adresse:");
12
13   //Funksjon innebygd i biblioteket for å hente MAC adresse
14   Serial.println(WiFi.macAddress());
15 }
16 void loop() {
17 }
```


Vedlegg F

Kretsskjemaet til Arduino IR sensor



Figur F.1: Kretsskjemaet til Arduino IR sensor

Vedlegg G

Testing av LED og fototransistor

Tabell G.1: Testing av LED og fototransistor

LED type	Min limnivå	Maks limnivå	Motstands- verdi transistor	Kommentar
Hvit LED + 220 Ω	289	438	10 $k\Omega$	Mer sensitiv rundt maks nivå
Hvit LED + 100 Ω	427	620	10 $k\Omega$	Ganske jevn gjennom hele tuben
Hvit LED + 100 Ω	620	640	15 $k\Omega$	Jevn gjennom hele tuben
Hvit LED + 100 Ω	520	635	12 $k\Omega$	Jevn gjennom hele tuben
Hvit LED + 100 Ω	215	320	4.7 $k\Omega$	Jevn gjennom tuben
Hvit LED + 100 Ω	370	636	8.2 $k\Omega$	Jevn gjennom tuben
Hvit LED + 47 Ω	540	630	8.2 $k\Omega$	Jevn og stabil måling

Hvit LED + 10 Ω	640	641	8.2 k Ω	Målingene endrer seg ikke. Blir for mye lys. Kommer til et punkt hvor det er lyst hele tiden inne i tuben, og refleksjonen fra bunnplata utgjør ingen forskjell
---------------------------	-----	-----	----------------	---

Vedlegg H

Testing av IR-LED og fototransistor

Tabell H.1: Testing av IR-LED og fototransistor

LED type	Min limnivå	Maks limnivå	Motstands- verdi transistor	Kommentar
IR + 100 Ω	280	616	8.2 k Ω	Veldig stabile målinger hele veien
IR + 100 Ω	315	635	10 k Ω	Veldig stabile målinger hele veien
IR + 100 Ω	350	640	12 k Ω	Veldig stabile målinger hele veien
IR + 100 Ω	140	450	4.7 k Ω	Noe stabile målinger
IR + 220 Ω	170	515	8.2 k Ω	Mer ustabil, mer sensitiv for plassering av IR-LEDen inne i sylinder

