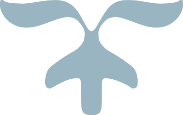


HOVEDPROSJEKT

3. ROBDIG 2025



Thomas E. Ortynski

Karl Magnus Myrvold

Erik Magnussen

Odd Levi Skar

**Forord**

Denne rapporten presenterer vår oppgave med å realisere en autonom minirobot kalt «lidarino», som er tildelt oss av vår oppdragsgiver Gunnar Storebø (Faglærer ved Fagskulen Vestland).

Gunnar har gitt oss i oppgave å sette sammen en minirobot av et ferdig chassis og deretter sette inn komponenter med selvlagde holdere og plassering av disse. Vi har også fått i oppgave å skrive programmene til disse slik at roboten skal bli selvkjørende.

Målet til roboten er å navigere nedover en korridor eller i et rom og deretter parkere seg selv nøyaktig 1cm fra et hjørne i en gitt retning i rommet (nord, sør, øst eller vest) og deretter snu seg med «fronten» utover.

Denne prosjektgruppen er satt sammen av 4 avgangsstudenter i en deltidsklasse ved Fagskulen Vestland fra Robotteknologi og digital produksjon i industrien.

Vi i prosjektgruppen ønsker å uttrykke vår takknemlighet til vår oppdragsgiver, Gunnar Storebø, for å ha gitt oss muligheten til å arbeide med et spennende og utfordrende prosjekt. Din veiledning har vært avgjørende for vår faglige utvikling og forståelse av prosjektets kompleksitet. Vi setter pris på din tillit og faglige ekspertise du har delt underveis. Denne erfaringen har gitt oss verdifulle innsikter, kunnskaper og faglig utvikling som vi vil ta med oss videre. Takker for din støtte og engasjement gjennom hele prosessen.

Videre vil vi takke våre venner og familie for tålmodighet, bistand og forståelse gjennom prosessen med hovedprosjektet og at vi ikke alltid har kunnet være til stede når det trengtes.

**Sammendrag**

I dette hovedprosjektet har gruppen sett på bruken av mikrokontrollere, motorer, sensorer og omnidrive samt robotikk og mulighetene for å bruke disse på en minirobot som til slutt skal bli fullt autonom med disse verktøyene.

Rapporten beskriver hvordan en kobler opp sensorer, motorer og omnidrive ved bruk av tegninger (cirkit designer) og utdrag av koden som blir skrevet med noe beskrivelse av hvordan koden fungerer opp mot de forskjellige komponentene og formålet med disse.

**Grovt utdrag av sammendraget. Vi får skrive dette ferdig mot slutten av hovedprosjektet.**

Innholdsfortegnelse

[Innledning: 6](#_Toc189413728)

[Problemstilling: 7](#_Toc189413729)

[Målsetting: 7](#_Toc189413730)

[Valg av komponenter. 7](#_Toc189413731)

[Omnidrive system. 8](#_Toc189413732)

[Mikrokontroller 9](#_Toc189413733)

[ESP Wroom 32 9](#_Toc189413734)

[Raspberry Pi 10](#_Toc189413735)

[Gyroskop 11](#_Toc189413736)

[Akselerometre 13](#_Toc189413737)

[Gyroskop og akselerometer??? Trenger vi dette? 13](#_Toc189413738)

[Lidar 14](#_Toc189413739)

[Motordriver L298N Mini 16](#_Toc189413740)

[DC-Motor med enkoder 19](#_Toc189413741)

[Batteripakke 20](#_Toc189413742)

[Ultralydsensor 21](#_Toc189413743)

[Kompass 24](#_Toc189413744)

[Hvorfor disse komponentene 25](#_Toc189413745)

[Metode 26](#_Toc189413746)

Presentasjon av medlemmer

Thomas Eiksund Ortynski: Prosjektleder:

Utdannet elektriker og ansatt hos FLM Elektro. Jobber som montør på alt fra små til store prosjekter både innen industri, bolig og næringsbygg i privat og offentlig sektor.

Karl Magnus Myrvold Teknisk Ansvarlig:

Utdannet sveiser med fagbrev og ansatt som platearbeider, bygger og sveiser alt fra små trapper og leidere til helikopterdekk og bevegelige gangveier med teleskopisk funksjon. Har ansvar for en ny CNC-maskin og reiser ofte offshore for serviceoppdrag på plattformer og fartøy.

Erik Magnussen: Prosjektmedarbeider.

Utdannet serviceelektroniker og svakstrømsmontør. Har jobbet med innbrudd- og brannalarm, kameraovervåking, nettverk og radioinstallasjoner. Jobber i dag som selvstendig næringsdrivende i firma kalt Smart Lys AS.

Odd Levi Skare: Prosjektmedarbeider  
  
 Utdannet serviceelektroniker og ansatt i Forsvarets   
 logistikkorganisasjon som systemtekniker med ansvar  
 for ulike systemer, deriblant radarsystemer. Det meste  
 av arbeidet foregår ombord fartøy som korvetter og  
 fregatter.

# Innledning:

Robotteknologi har de siste årene sett en kraftig vekst både innen forskning og kommersielle bruk. Fra industrielle roboter som håndterer ulike oppgaver, til autonome kjøretøy som navigerer i komplekse bymiljøer, har roboter blitt en viktig del av det moderne samfunnet. Denne raske utviklingen har ført til økt etterspørsel etter kompetanse innen robotdesign, programmering og systemintegrasjon.

Gjennom dette prosjektet har vi ønsket å bidra til denne spennende utviklingen ved å designe, bygge og programmere en autonom minirobot. Med utgangspunkt i kunnskapen vi har opparbeidet oss i fag som f.eks. elektronikk, mekanikk og programmering, har vi utviklet en robot som selvstendig kan navigere innenfor et definert miljø. Prosjektet har gitt oss verdifull erfaring i å anvende teoretiske prinsipper i praksis, samtidig som vi har styrket våre praktiske ferdigheter innen programmering, sensorteknologi og robotikk

Denne rapporten beskriver hele prosessen, fra design og implementering til testing av vår autonome minirobot. Vi redegjør for de tekniske løsningene vi har valgt, og analyserer robotens ytelse.

Prosjektet har vært både utfordrende og lærerikt, og har krevd tett samarbeid og kreativ problemløsning fra alle medlemmer av teamet.

# Problemstilling:

Målet med dette prosjektet er å bygge og programmere en autonom robot basert på en forhåndsdefinert plattform. Roboten skal være i stand til å navigere selvstendig i et kontrollert miljø, identifisere og unngå hindringer, samt parkere med en nøyaktighet på ±1 cm fra enten det østre eller vestre hjørnet av et angitt område.

For å oppnå dette må vi implementere en rekke tekniske komponenter, inkludert mikrokontrollere, motorer med enkoder, lidar, kompass og sensorer. Samtidig må vi utvikle programvare som kan koordinere dataflyt og sikre nøyaktig styring og effektiv samhandling mellom komponentene.

Dersom tiden tillater det, vil vi utforske flere avanserte funksjoner, som for eksempel implementering av dynamisk rutenavigasjon, sanntids kartlegging av miljøet (SLAM), eller optimalisering av parkeringsevnen for å håndtere ulike scenarier, som skråstilte eller smale parkeringsområder.

# Målsetting:

Målet med dette prosjektet er å se på mulighetene for å utvikle et karosseri av en robot til å kunne bli en selvkjørende minirobot som skal kunne navigere selvstendig og unngå hindringer ved hjelp av integrerte sensorer og systemer som lidar og ev kompass.

Roboten skal kunne parkere med en nøyaktighet med 1cm fra valgt hjørne av rommet eller korridoren den kjører i og demonstrere pålitelig samhandling mellom maskinvare, programvare inkludert mikrokontrollere, motorer med enkoder og sensorteknologi.

I løpet av dette prosjektet er det forventet at vi vil bruke og bygge på utviklingen om kunnskapen vi har både i programmering, sensorteknologi og robotikk. Dersom tiden strekker til, vil vi utforske muligheten til å implementere andre teknologier for å gjøre roboten enda mer sømløs med for eksempel computer vision og sanntids kartlegging.

# Valg av komponenter.

## Omnidrive system.

Omnihjul, eller polyhjul, er hjul med små «ruller» rundt omkretsen som står vinkelrett på dreieretningen. Effekten er at hjulet kan drives forover, men også gli sidelengs med stor letthet. Disse hjulene brukes ofte i holonomiske kjøresystemer.A black wheel with yellow arrows

Description automatically generated

Det første omnidireksjonelle hjulet som ble patentert, er designet av J. Grabowiecki i 1919. Hjuldesignet blir fortsatt ofte brukt i dag, og hjulet består av et hovedhjul med tverrgående ruller rundt hovedhjulet.

En plattform som bruker tre omnihjul i en triangulær konfigurasjon kalles generelt Kiwi Drive. Killough-plattformen er lik; oppkalt etter Stephen Killoughs arbeid med omnidireksjonale plattformen ved Oak Ridge National Laboratory. Killoughs design fra 1994 brukte par med hjul montert i bur i rette vinkler til hverandre og oppnådde dermed holonomisk bevegelse uten å bruke omnihjul (<https://en.wikipedia.org/wiki/Omni_wheel#:~:text=Omni%20wheels%20or%20poly%20wheels,slide%20laterally%20with%20great%20ease>.) Kilde til wikipediaside om omni wheel.

(<https://en.wikipedia.org/wiki/File:Triple_Rotacaster_commercial_industrial_omni_wheel.jpg>) Link til bilde. Gjør om til hyperlink og få under bildet.

## Mikrokontroller

### ESP Wroom 32

ESP-WROOM-32 er en brikke utviklet av Espressif System, og er basert på ESP32 brikken, lansert rundt 2016, som er omtalt som system på en brikke, eller “System on Chip(SoC)” på engelsk.

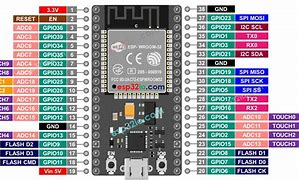
Denne har både WiFi og Bluetooth innebygget som gjør det meget attraktiv for utvikling av forskjellige IoT løsninger. WROOM modellen har i tillegg til brikken, flashminne og RF elektronikk.

**Maskinvare**

* Dobbeltkjerne Xtensa® 32-bit LX6 CPU (opptil 240 MHz)
* 520 KB SRAM som standard

**Tilkoblinger**

GIPO’ene (General Purpose Input/Output) har forskjellige funksjoner som vist i illustrasjonen under. Ikke alle utgangene kan programmeres fritt da noen av de har dedikerte oppgaver.



### Raspberry Pi

Raspberry Pi er i førlge Wikipedia en ettkortdatamaskin som ble utviklet i 2012 av det britiske selskapet Rasbery Pi Foundation, som et støtte i undervisning i informatikk. Kortet har gjennomgått en utvikling og er nå kommet til Pi 5 som vi bruker er ett av det siste tilskuddet fra selskapet. Selv om det er kommet nye utgaver av kortet har de fysiske målene og plassering av GPIO- kontaktene vært de samme, noe som har muligjort enkel oppgradering etter hvert som nye kort har blitt lansert.

Vårt valg av Paspberry Pi er grunnet at denne har mulighet for mer kompleksitet enn vi vil klare med kun en ESP 32.

**Maskinvare**

* Broadcom BCM2712 quad-core Arm Cortex A76 processor @ 2.4GHz
* 8GB RAM (Random Access Memory)

**Tilkoblinger**

* 40x GPIO – kontakter, som kan konfigrueres og programmeres
* 4x USB A porter
* 1 Gb Ethernet port
* USB C
* 2 HDMI mini utganger.

**Operativsystem**

Operativsystemet ligger på et SD-kort, eller NVMe disk koblet til PCI porten via et M.2 grensesnitt.

Raspberry Pi OS er det offisielle operativsystemet, også kjent som Raspbian, og er basert på Debian Linux. Men det er andre alternative operativsystemer og distribusjoner.

**Utvidelsesmuligheter**

Raspberry’en kan utvides med utvidelsesmoduler HAT (Hardware Attached on Top), disse kobles til GIPO’ene eller via PCI express kontakten.

## Gyroskop

Det finnes seks eller flere typer gyroskoper, men vi fokuserer bare på den gamle, mekaniske typen og MEMS-gyroskoper, som vi tar i bruk. MEMS-gyroskoper er en av de mest anvendte typene ettersom de er billige å masseprodusere og krever lite energi.

Et bilde som inneholder sirkel, kunst, design

Automatisk generert beskrivelseMekaniske gyroskoper bruker et hjul eller en rotor som roterer i høy hastighet. Rotoren er montert i et oppheng som lar den rotere fritt i flere akser. Når rotoren roterer raskt nok, skapes det et vinkelmoment som opprettholder rotorens orientering i rommet. Denne gyroskopeffekten gjør at rotoren motstår endringer i orientering, unntatt langs sin egen rotasjonsakse. Selv om mekaniske gyroskoper er svært nøyaktige og robuste over tid, har de noen ulemper: De er store og tunge, noe som gjør dem upraktiske for moderne, kompakte enheter, og de bevegelige delene er utsatt for slitasje. De krever også regelmessig vedlikehold og kalibrering for å opprettholde nøyaktighet. Likevel er mekaniske gyroskoper fortsatt mye brukt for å holde orienteringen til blant annet raketter og satellitter over lengre tid, fremfor presis måling og instrumentering.

MEMS-gyroskoper er derimot svært små, mikroskopiske enheter. De fungerer ved at interne strukturer vibrerer ved en gitt frekvens. Når gyroskopet roteres rundt en akse, oppstår Coriolis-effekten, som skaper en liten forskyvning i vibrasjonsmønsteret. Denne forskyvningen registreres for å måle rotasjon. Takket være sin lave kostnad, lille størrelse og energieffektivitet er MEMS-gyroskoper vanlige i alt fra smarttelefoner, droner og biler til spillkontrollere og VR-utstyr. Til tross for fordelene har de også noen begrensninger. De er mindre presise enn mekaniske gyroskoper og kan være utsatt for drift over tid, noe som påvirker nøyaktigheten. Eksterne faktorer som temperaturendringer og vibrasjoner kan dessuten påvirke ytelsen negativt, noe som gjør dem mindre egnet for applikasjoner som krever ekstrem nøyaktighet eller langvarig stabilitet.

Selv om mekaniske gyroskoper er mer robuste og presise over lengre tid, har MEMS-gyroskoper revolusjonert hvordan vi bruker gyroskoper i hverdagslige applikasjoner. Teknologien utvikler seg fortsatt, med stadig mer presise og effektive varianter som kan dekke flere behov i fremtiden.

<https://snl.no/gyroskop>

<https://science.howstuffworks.com/gyroscope.htm>

<https://en.wikipedia.org/wiki/Gyroscope>

<https://www.analog.com/en/resources/technical-articles/mems-gyroscope-provides-precision-inertial-sensing.html>

## Akselerometre

Akselerometre er sensorer som reagerer på eller registrerer endringer i fart eller retning, og de brukes ofte sammen med gyroskoper for navigasjon og styring av fly, raketter og ubåter. I tillegg anvendes de i vibrasjonsmåling i bygninger, som utløsere for kollisjonsputer i biler, og i spesialiserte formål som jordskjelvmåling via seismometre.

Et akselerometer består av en bevegelig masse, som er suspendert i en ramme og koblet til fjærer. Når enheten utsettes for akselerasjon, beveger massen seg i forhold til rammen, og denne bevegelsen registreres. MEMS-akselerometre, som er blant de mest brukte typene, benytter ofte endringer i kapasitans for å måle denne forskyvningen. Kapasitansen endres når massen beveger seg, og dette signalet brukes til å beregne akselerasjonen.

Moderne MEMS-akselerometre er svært små og energieffektive. De finnes i alt fra smarttelefoner og spillkontrollere til droner og biler, der de brukes til å registrere bevegelser og rotasjon eller skifte mellom portrett- og landskapsmodus. MEMS-kapasitansbaserte akselerometre er også egnet for lavfrekvente vibrasjonsmålinger, noe som gjør dem verdifulle i strukturell helseovervåking og andre ingeniørapplikasjoner.

Begrensninger med MEMS-akselerometre inkluderer følsomhet for støy og miljøfaktorer som temperaturendringer og vibrasjoner. Selv om de er svært nøyaktige for mange formål, kan ekstreme påkjenninger påvirke ytelsen negativt, og de er mindre presise enn avanserte optiske eller laserbaserte systemer.

<https://www.kistler.com/INT/en/mems-accelerometer/C00000759>

<https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2564175>

<https://faculty.uml.edu/xwang/16.541/2010/MEMS_accelerometers.pdf>

<https://valerionew.github.io/triennale-elettronica-polimi/4/MEMS/MEMS_and_Microsensors.pdf>

Gyroskop og akselerometer??? Trenger vi dette?  
  
Kompass

## Lidar

Lidar (Light Detection and Ranging) er en teknologi som bruker laserlys til å måle avstander og lage detaljerte 3D-modeller av omgivelsene. Dette oppnås ved å sende ut laserimpulser og måle tiden det tar før lyset reflekteres tilbake til sensoren. Avstanden beregnes ved hjelp av lysets hastighet og tiden det bruker på å reise frem og tilbake. Lidar kan operere i faste retninger eller rotere for å skanne et bredt område, noe som gir et komplett bilde av miljøet rundt sensoren.

Et bilde som inneholder sketch, tegning, sirkel, strektegning

Automatisk generert beskrivelse

Lidarens evne til å levere nøyaktige målinger gjør den ideell for autonome systemer. Den brukes i alt fra kartlegging og terrengmodellering til navigasjon i selvkjørende biler og roboter. For roboten vi skal produsere, vil lidar spille en avgjørende rolle i sanntidsregistrering av miljøet. Den vil identifisere hindringer, kartlegge omgivelsene, og bidra til presis navigasjon i dynamiske og komplekse omgivelser.

Et bilde som inneholder sketch, diagram, tegning, design

Automatisk generert beskrivelse

Ved å bruke lidar kan roboten oppnå høy grad av autonomi. Teknologien gir roboten mulighet til å oppfatte både små og store objekter med høy nøyaktighet, noe som er essensielt for å sikre effektiv drift og unngå kollisjoner. Lidarens evne til å fungere uavhengig av lysforhold er også en stor fordel i varierende miljøer.

(<https://en.wikipedia.org/wiki/Lidar>) Kildehenvisning tekst.  
https://bucket-(download.slamtec.com/d1e428e7efbdcd65a8ea111061794fb8d4ccd3a0/LD108\_SLAMTEC\_rplidar\_datasheet\_A1M8\_v3.0\_en.pdf) Kildehenvisning bilder.

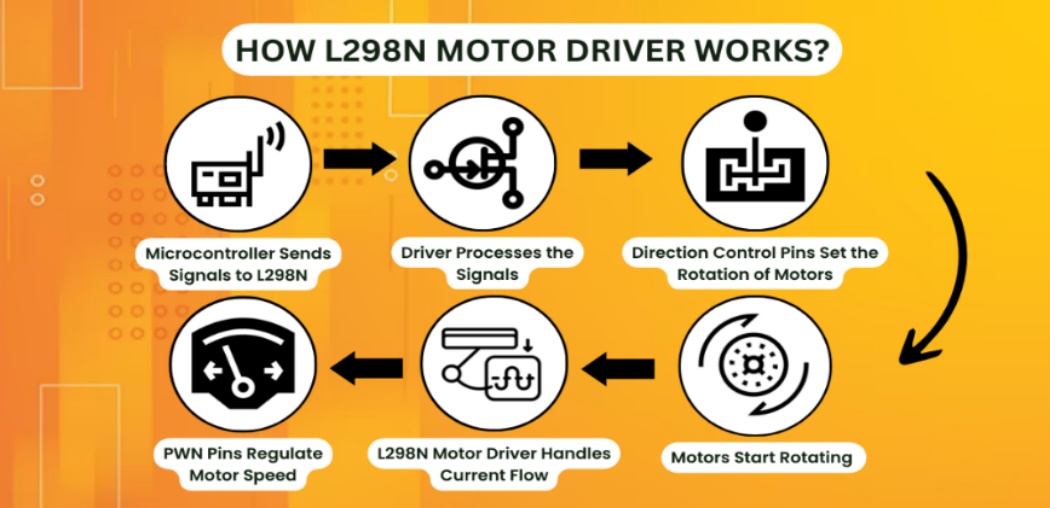
## Motordriver L298N Mini

L298N Motordriver er en modul som er designet for å kontrollere hastigheten og retningen for opptil to DC-motorer. Modulen benytter en H-bro for å styre strømretningen gjennom motoren og gjør det mulig å kontrollere både fremover- og bakoverrotasjon.

En H-bro er en elektrisk krets som lar strømmen gå i begge retninger gjennom en motor. Dette er nødvendig for at motoren skal kunne forandre retning alt etter hvilken vei strømmen går. L298N inneholder to integrerte H-broer som hver kan styre en motor. Kretsen fungerer ved å bruke fire brytere (transistorer eller MOSFET) til å skifte mellom ulike kombinasjoner av strømretninger. PWM brukes sammen med H-broen for å kontrollere motorens hastighet ved å variere pulsbredden.

Motordriveren har to spenningsinnganger som forsyner motorene (Vs) og de logiske kretsene (Vss)

L298N har en integrert 78M05 5V regulator som kan aktiveres for å drive de logiske kretsene. Modulen støtter motorer med driftsspenning fra 5V til 35V og kan levere opptil 2A strøm per kanal. Skruterminaler og enkle tilkoblinger gjør modulen brukervennlig.



Fordeler:

* Fleksibilitet - Kan stryre både DC-motor og steppermotor
* Enkel å bruke – krever ikke mye ekstra komponenter
* To H-broer - Støtter uavhengig kontroll av to motorer
* Kostnadseffektiv - Rimelig løsning for mange prosjekter
* Kompabilitet - Støtter populære mikrokontrollere som Arduino og Raspberry Pi
* Integrert 5V regulator – Kan forsyne logiske kretser uten behov for ekstern spenning

Ulemper:

* Begrenset strømkapasitet – Maks 2A per kanal kan være en begrensning for kraftigere motorer.
* Varmeutvikling – Krever kjøling ved høy belastning
* Effektivitet – Bruker eldre bipolare transistorer, som gir høyere varmetap sammenlignet med nyere MOSFET-baserte drivere.

Hvordan teste L298N motor driver:

Koble motordriveren til en strømkilde og koble til denne. Mål deretter spenningen på pinnene til motor A og B for å se at de får rett spenning. Skal man se om at motordriveren virker som den skal, kan man lage en enkel kode som roterer motoren i begge retninger med noen sekunders stopp mellom retningene. Er de valgfrie PWM-pinnene (ENA og ENB) tilkoblet, kan koden også teste om hastighetskontrollen via PMW virker.

A screen shot of a computer program

Description automatically generated

([L298n Motor driver Arduino | Motors | Motor Driver | L298n | Arduino Project Hub](https://projecthub.arduino.cc/lakshyajhalani56/l298n-motor-driver-arduino-motors-motor-driver-l298n-7e1b3b)) Testkode fra Arduino siden

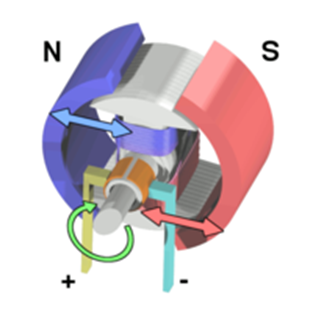
Vi har ikke den vanlige L298N driveren, men «mini» varianten av denne da den er mer kompakt og bruker mindre plass på roboten. De vesentlige forskjellene her er at den er mindre og mer kompakt, har lavere strømkapasitet, lite eller ingen dedikert kjøling m.m, men i hovedsak virker den på samme måte som L298N vanlig motor driver.

(<https://vayuyaan.com/blog/everything-you-want-to-know-about-l298n/#How_L298N_Motor_Driver_Works>) Kildehenvisning teoridel og gult bilde

(<https://www.robotpark.com/Mini-L298N-Motor-Driver>**) kildehenvisning bilde**

## DC-Motor med enkoder

En DC-motor er en elektromagnetisk maskin som konverterer elektrisk energi til mekanisk energi ved hjelp av magnetiske krefter. Hovedkomponentene i denne motoren er statoren, armaturen, kommutatoren og børstene.

[](https://en.wikipedia.org/wiki/DC_motor)Den elektromagnetiske delen av motoren fungerer etter prinsippet når det går elektrisk strøm gjennom en leder, skapes et magnetisk felt rundt lederen. I motorens armatur er spoler viklet rundt en jernkjerne, og når strømmen går gjennom disse viklingene skapes et elektromagnetisk felt. Dette magnetiske feltet samhandler med det magnetiske feltet fra statormagnetene og skaper et dreiemoment som får armaturen til å rotere.

For at rotasjonen skal opprettholdes kontinuerlig, må strømretningen i viklingene endres periodisk. Dette gjøres av kommutatoren som er koblet til armaturen. Kommutatoren er en roterende bryter som bytter retningen på strømmen på viklingene slik at motoren fortsetter å rotere i en retning. Børstene fungerer som elektriske kontakter og overfører strømmen fra den stasjonære delen til den roterende kommutatoren.

(<https://en.wikipedia.org/wiki/DC_motor>) Kilde til DC motor + bilde

En DC-motor med enkoder kombinerer den grunnleggende funksjonaliteten til en DC-motor med enkoder som gir posisjonstilbakemelding. Denne typen motor brukes ofte i robotikk, automatiseringssystemer og andre områder der nøyaktig posisjonering er nødvendig.

Motoren består av to hoveddeler: selve motoren og koderen. Motoren drives ved å tilføre spenning til terminalene, noe som får rotoren til å rotere. Koderen registrerer derimot rotasjonen av motorakselen og gir tilbakemelding om posisjonen.

En typisk koder består av en disk med et mønster eller spor og en sensor som registrerer bevegelsen når disken roterer. Under rotasjon danner sensoren elektriske signaler basert på diskens mønster, som angir motorakselens posisjon. Oppløsningen til koderen bestemmes av antallet spor på disken. Signalene fra koderen behandles av et kontrollsystem som bruker denne tilbakemeldingen til å justere spenningen som påføres motoren. Dette lukkede sløyfekontrollsystemet gjør det mulig å kontrollere motorens posisjon, hastighet og akselerasjon med høy presisjon.

DC-motorer med kodere har flere fordeler sammenlignet med andre motorer. Den største fordelen er deres evne til å oppnå nøyaktig posisjonering, som er avgjørende i mange bruksområder. I tillegg er de svært effektive, har lang levetid og krever lite vedlikehold.

(<https://no.vsdmotor.com/news/what-is-a-dc-motor-with-encoder-and-how-does-67899919.html>) Kildehenvisning til enkoder tekst.

(<https://www.digikey.no/no/products/detail/dfrobot/FIT0403/6588578>) bildehenvisning

## Batteripakke

Strømkilden til «lidarino» består av Li-ion 18650 celler som er satt sammen til en «batteripakke» for å få nok tilført strøm til både komponenter og motorer for fremdrift

## Ultralydsensor

HC-SR04 er en ultralydsensor som benytter lydpulser for å måle avstanden til objekter uten fysisk kontakt. Den opererer ved å sende ut en ultralydbølge med en frekvens på 40 kHz og måle tiden det tar før ekkoet returnerer etter å ha blitt reflektert fra et objekt. Basert på denne tidsmålingen kan avstanden beregnes, ettersom lydens hastighet i luft er kjent.

Sensoren har fire tilkoblingspinner: VCC, GND, Trig og Echo. For å initiere en måling sendes en kort puls på 10 mikrosekunder til Trig-pinnen. Dette får sensoren til å sende ut en serie på åtte ultralydpulser. Når disse pulsene treffer et objekt, reflekteres de tilbake, og Echo-pinnen registrerer tiden det tar før ekkoet returnerer. Denne tidsforsinkelsen brukes deretter til å beregne avstanden til objektet ved hjelp av formelen:

**Avstand** = (Tidsforsinkelse × Lydens hastighet i luft) / 2

Lydens hastighet i luft er omtrent 343 meter per sekund, eller 0,0343 cm/µs. Ved å måle tidsforsinkelsen i mikrosekunder kan man dermed beregne avstanden i centimeter. For eksempel, hvis tidsforsinkelsen er 500 mikrosekunder, blir avstanden:

Avstand = (500 µs × 0,0343 cm/µs) / 2 = 8,575 cm

HC-SR04 har et målområde fra 2 cm til 400 cm med en nøyaktighet på ±3 mm. Den effektive målevinkelen er omtrent 15 grader, noe som betyr at sensoren er mest nøyaktig innenfor denne vinkelen. Den drives med en spenning på mellom 3,3V og 5V DC og har en hvilestrøm på mindre enn 2 mA, med en arbeidsstrøm på 15 mA.

I praktiske anvendelser brukes HC-SR04 ofte i robotikk for hindringsunngåelse, avstandsmåling og nivådeteksjon, for eksempel i parkeringssensorer eller væskenivåmåling. Dens enkle grensesnitt og pålitelige ytelse gjør den til et populært valg for mange elektronikkprosjekter.

<https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf>

Et bilde som inneholder sirkel, skjermbilde, tekst, design

Automatisk generert beskrivelseEt bilde som inneholder sylinder, sirkel, skjermbilde, design

Automatisk generert beskrivelse

**Fordeler**:

* Er rimelig og tilgjengelig, noe som gjør den ideell for budsjettvennlige prosjekter.
* Nøyaktighet ved måling av avstander opptil +- 3mm under best mulige forhold.
* Fungerer godt med vanlige mikrokontroller som arduino og raspberry Pi.
* Fungerer godt i variert lysforhold, ikke avhengig av synlig lys som IR sensorer.
* Godt egnet til batteridrevet systemer grunnet lavt energibehov.
* Kan detektere objekter uavhengig av overflate og refleksivitet.

**Ulemper**:

* Begrenset synsvinkel på ca 15 prosent, som gjør den mindre egnet for bredde overvåkning.
* Kan være følsom for støy, som igjen kan forstyrre målingene.
* Er ikke vanntett eller designet for ekstreme miljører uten modifikasjon.
* Kan ha en begrenset oppdateringshastighet, som begrenser bruk ved raske applikasjoner.
* Under 2 cm kan sensoren gi upålitelige målinger.

For å teste kan man koble opp slik:

Et bilde som inneholder tekst, skjermbilde, Elektronteknikk, elektronikk

Automatisk generert beskrivelse

Test kode i arduino:

Et bilde som inneholder tekst, skjermbilde, Font, dokument

Automatisk generert beskrivelse

Når vi kjører denne koden på en Arduino og observerer resultatet i «serial monitor», ser vi at avstanden endres når vi beveger hånden foran sensoren. Verdiene som vises, representerer den målte avstanden.

# Kompass

HMC5883L er en 3-akset magnetisk sensormodul utviklet av Honeywell, primært designet for bruk i digitale kompasser og andre applikasjoner som krever nøyaktig måling av magnetfelt. Modulen benytter anisotrop magnetoresistiv (AMR) teknologi for å oppnå høy presisjon og følsomhet ved måling av jordens magnetfelt. Dette gjør den velegnet for navigasjonssystemer, robotikk og posisjoneringsapplikasjoner.

Den fungerer basert på prinsippet om anisotrop magnetoresistiv (AMR) effekt, hvor materialers elektriske motstand varierer i respons til ytre magnetiske felt. Sensoren består av tre AMR-elementer, hver plassert ortogonalt for å måle magnetfelt langs X-, Y- og Z-aksene. Når magnetfelt påvirker disse elementene, endres motstanden deres, og denne endringen omformes til en målbar spenning. Spenningen konverteres deretter til digitale verdier via en 12-bits ADC, noe som gir høy presisjon i feltmålingen.

For å beregne orientering benytter HMC5883L vektorsummen av de magnetiske komponentene. Ved å sammenligne målingene med jordens magnetfelt kan enheten bestemme sin retning relativt til magnetisk nord. I praksis benyttes ofte filtreringsteknikker og kalibreringsalgoritmer for å korrigere for forstyrrelser fra nærliggende metallstrukturer og elektriske komponenter. I kombinasjon med akselerometer og gyroskop kan disse dataene gi en mer robust og stabil retningsbestemmelse

Et bilde som inneholder elektronikk, krets, Elektronteknikk, Elektronisk komponent

Automatisk generert beskrivelseHMC5883L opererer innenfor et spenningsområde på 2,16V til 3,6V og kommuniserer via en I2C-grensesnitt, noe som muliggjør enkel integrasjon med mikrokontrollere som Arduino og Raspberry Pi. Sensoren har en 12-bits ADC (Analog-Digital Converter), som gir en oppløsning på 2 milligauss per LSB. Dette gir en vinkeloppløsning på mellom 1° og 2°, avhengig av kalibrering og miljøforhold. Modulen består av tre magnetiske sensorer som måler X-, Y- og Z-komponentene av det omgivende magnetfeltet. Disse verdiene kan benyttes til å beregne en enhets orientering ved hjelp av kompassalgoritmer, ofte i kombinasjon med akselerometer og gyroskop for forbedret stabilitet og presisjon.

**Fordeler:** modulen har flere fordeler, inkludert høy   
nøyaktighet og følsomhet. Den 12-bits ADC-en gir en   
feltoppløsning på 2 milligauss, noe som resulterer i en   
nøyaktig retningsbestemmelse. Modulen har lavt strømforbruk,  
med et gjennomsnittlig forbruk på 100 μA i målemodus, noe   
som gjør den godt egnet for batteridrevne systemer. Den kompakte  
utformingen, med en fysisk størrelse på 3,0 × 3,0 × 0,9 mm, gjør den  
enkel å integrere i små enheter.

**Ulemper:** På den andre siden har modulen også noen ulemper. Den er følsom for   
eksterne magnetiske forstyrrelser, og nøyaktigheten kan påvirkes av nærliggende elektriske komponenter og metallstrukturer. I tillegg har Honeywell avsluttet produksjonen av HMC5883L, noe som kan begrense tilgjengelighet og fremtidig support.

# Hvorfor disse komponentene

# Metode

I dette kapittelet vil vi gå igjennom oppbyggingen av Lidarinoen. Hvordan vi har klart å få Raspberryen til å kontrollere roboten og navigerer ved hjelp av Lidar og kompass.

Lidar er tilkoblet til Raspberry’en via USB. Vi bruker en enkel versjon av SLAM (Simultaneous Localization and Mapping), for å lage et kart over miljøet Lidarinoen er i. Ut fra dette skal det ta seg frem uten å støte borti hindringer. Programmet kjører på Raspberry som videre gir informasjon videre til ESP ved hjelp av en I2C bus eller seriel over USB.

Rasberry Pi kommer ikke med I2C aktivert som standard så dette må aktivres. Dette gjøres i Raspbian OS ved å skrive inn raspi-config i terminalen, og aktivere dette i menyen.

I2C busen kobles til GPIO 3 (SDA) og GPIO 5 (SCL), jord (GND) kobles sammen mellom

Problemer med bilen som vi har møtt på  
  
Skrive kodene til Omnidrive og IR mottaker for å få disse til å snakke sammen og få bilen til å kjøre i valgte retninger. Bilen ville kjøre fremover, bakover, venstre og høyre, men vi ble nødt å resette mikrokontrolleren for hver retning, og stoppknappen ville ikke gjøre noe. IR mottakeren ville reagere med fjernkontrollen, men ingenting ville skje.

Koden som er laget er hentet fra en bruker på github som vi har skrevet om og fått hjelp av chatGPT og Gemini for å få løst eventuelle feilmeldinger som har dukket opp når vi ikke har fått til å fikse dem selv på en bra måte.

Etter omkobling til nye motordrivere reagere kjører ikke alle hjulene som forventet, antar det er koden som må redigeres. Skal teste motorene individuelt før redigering av koden

En motor må bytte retning, men det fikset ikke problemet da den går som forventet fremover og bakover. Høyre eller venstre er det forventet at alle hjulene skal snurre med forskjellige hastigheter kan være noe med kiwidrive koden

Rullet tilbake deler av koden med resultatt av at kun en av motorene går bege veier. Ser GPT liker å ta vekk ting fra koden som er helt esensielt for koden uten å gi noen form for beskjedd om det, etter mye frem og tilbake med koden og flytting av pinner og ledninger ser eg at kiwidrive koden ikke har fått forandret formel til noe helt ukjent.

Løsningnen ble å bytte til esp 32, tror arduino uno ikke er kraftig nok til å drive 6 induvidiele pwm signaler samtidig, etter bytting til esp 32 virket alle hjulene etter forventningene.

PID loop koden vi fikk fra GPT virker veldig ustabil, prøvde å justere pid verdier uten forandring. Ser etter eksempel koder på nett, men de fleste inneholder bibliotek som krever tid å sette seg inni uten å vite om de passer til vårt formål.

Karl og Thomas bestemmer seg for å jobbe med hver sin pid kode for å prøve å komme frem til en løsning. Veileder kom med forslag om å muligens drive hver motor med sin egen esp32 microcontroller for å gjøre encoder og pid koden enklere fremfor å kjøre alle motoren med en esp 32.

**RPLidar utfordringer:**

Å få RPLidar-komponenten til å fungere sammen med Raspberry Pi 5 var en utfordrende prosess. Først ønsket vi å teste selve komponenten for å bekrefte at den fungerte som den skulle. Dette innebar en enkel funksjonstest der vi koblet RPLidar til en PC for å sjekke om den roterte som forventet. Deretter var målet å koble den til en Raspberry Pi og få den til å generere et bilde. Dette viste seg å være mer komplisert enn antatt, og vi måtte ty til omfattende nettsøk, lese forumtråder, og bruke verktøy som ChatGPT og Gemini for å forstå koden i Python og bruken av Linux-terminalen.

For å teste RPLidar-enheten, ble produsentens nettside **slamtec.com** besøkt for å laste ned programvaren RobotStudio. Etter tilkobling til en Windows-PC og valg av riktig COM-port, ble programmet startet for å se om lidaren kartla rommet den var plassert i. Dette fungerte som forventet.

Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

Automatisk generert beskrivelse

Under programmeringen i spyder dukket det opp flere utfordringer. Feilmeldinger relatert til manglende eller feilaktig installerte pakker bla rplidar, meldinger som "wrong body size", samt problemer med full buffer, skapte behov for omfattende feilsøking og justeringer.

For å unngå at programmet stoppet opp på grunn av bufferproblemer, noe som dukket opp ved de første testene, ble maksimal bufferstørrelse i koden økt fra 3000 til 8000 målinger.

For å teste muligheten for å danne et sanntidsbilde, ble det laget en kode som skulle presentere et plot i sanntid. I første omgang ble kun ett enkelt plot visualisert ved kjøring av koden, uten kontinuerlig oppdatering. Dette skyldtes en innstilling i Spyder, som kun presenterte et bilde fra den første skanningen. Problemet ble løst ved å navigere til Tools > Preferences > iPython Console, og deretter endre innstillingen under Graphics Backend til Automatic. Denne endringen sørget for at et eksternt vindu åpnet seg ved kjøring av koden, som deretter viste bildet fra lidaren i sanntid.

Et bilde som inneholder tekst, diagram, skjermbilde, sirkel

Automatisk generert beskrivelse

Når Raspberry Pi skulle brukes som kontrollenhet for Lidar, ble et virtuelt miljø opprettet i operativsystemets terminal. Deretter ble alle nødvendige pakker installert for å kunne kjøre Python-koden som var laget i Spyder, og for å vise resultatene i terminalen. Utfordringene som oppstod var ...

**PID-regulatorer**

PID-regulatorer (Proportional-Integral-Derivative) er en av de mest brukte kontrollmetodene for prosesskontroll. Hovedideen bak en PID-regulator er å kombinere tre kontrollstrategier: en proporsjonal del (P), en integrerende del (I) og en deriverende del (D), for å gi et kontrollsignal som er mer presist og stabilt. Målet er å redusere feilen mellom ønsket verdi (setpoint) og den faktiske verdien (målt verdi).

1. **Proposjonal del (P)**

Den proporsjonale delen er basert på feilen mellom ønsket og faktisk verdi. Den gir en rask respons på feil, men kan føre til et steady-state-feil. Formel for proporsjonal kontroll er:

hvor:

* + ​ er kontrollsignal fra proporsjonal del,
  + ​ er proporsjonalforsterkeren,
  + er feilen, der er ønsket verdi og er målt verdi.

1. **Integrerende del (I)**

Den integrerende delen av PID-regulatoren akkumulerer feilen over tid og gir et kontrollsignal basert på summen av tidligere feil. Dette bidrar til å eliminere steady-state-feil. Formel for den integrerende kontrollen er:

hvor:

* + er kontrollsignal fra integrerende del,
  + er integrasjonsforsterkeren,
  + er den akkumulerte feilen over tid.

1. **Deriverende del (D)**  
   Den deriverende delen bruker hastigheten på endring av feilen for å forutsi fremtidige feil, og bidrar dermed til å redusere overshoot og forbedre stabiliteten. Formel for deriverende kontroll er:

hvor:

* + ​ er kontrollsignal fra deriverende del,
  + er derivasjonsforsterkeren,
  + er den tidsderiverte av feilen.

1. **Total PID-kontrollsignal**  
   Det totale PID-kontrollsignal er summen av de tre delene:

hvor u(t) er det totale kontrollsignal som sendes til systemet.

Kilde: (<https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller>)

A screenshot of a computer

Description automatically generated