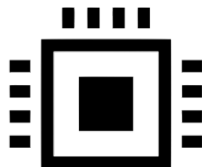


Arduinoprogrammering

Zumo32U4-Øving 7

IELET1002

Linjefølger



DATATEK.

DIGITAL KOPI

Praktisk info

Hva trenger jeg å ha med for å gjennomføre laben?

For gjennomføring av denne laboratorieoppgaven behøver ikke studenten å ta med noe. Det vil bli mulig å låne følgende utstyr:

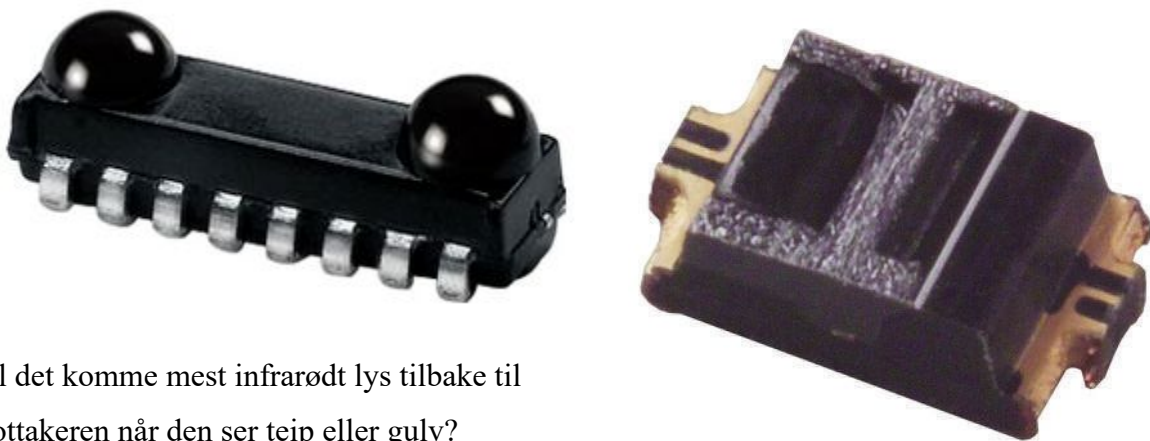
- Zumo32U4
- PC
- USB-kabel for å koble Arduino til PC



Bakgrunn for lab

Bakgrunnsinfo

I denne laboratorieoppgaven skal studenten bli bedre kjent med Zumo32U4. Både knapper, lysdioder, sensorer og motorer skal brukes. Sensorene som sitter i bunn av bilen kalles «IR transceiver» på engelsk. IR betyr infrarødt, lys som ikke er synlig for oss mennesker. Transceiver er sammenslått av ordene transmitter og receiver, sender og mottaker. Dette komponentet kan både sende ut IR, og lese av hvor mye som kommer tilbake.



Vil det komme mest infrarødt lys tilbake til mottakeren når den ser teip eller gulv?

Laboppgaven

I denne oppgaven skal studenten utforske hvordan linjesensorene på Zumo32U4 fungerer i praksis. Dette skal gjøres ved hjelp av biblioteket Zumo32U4.h. I tillegg skal motorene brukes for å kunne følge en teiplinje på bakken.

Husk at internett er en kjemperessurs for programmering og Arduino. Ikke glem å se på www.arduino.cc for masse fine guider og prosjekt. I tillegg er det en fin vane å bruke søkemotorer når man er usikker på noe. Spør oss om hjelp, dersom noe er uklart! ☺

LYKKE TIL!

Oppgaver

Biblioteker og oppstart

Gjør deg kjent med ressursene på <https://www.pololu.com/docs/0J63 for Zumo32U4>. Her finner du full brukermanual, bibliotek og oppstartsguide.

Bruk guiden for å installere biblioteket.

<https://pololu.github.io/zumo-32u4-arduino-library/>

Eksempelkode

Last opp en valgfri eksempelkode fra biblioteket for å verifisere at du har kontakt med bilen, og får til å laste opp kode.

Oppgave 1)

Gjør deg kjent med hvordan knappene og lysdiodene brukes, og skap følgende funksjonalitet:

- Bilen gjør ingenting frem til buttonA trykkes
- Når buttonA trykkes skal lysdiodene blinke, og buzzeren gjøre et utslag (valgfritt)

Oppgave 2)

Utvid lysshowet med å kjøre bilen i et forhåndsbestemt, tredelt mønster:

- Mønster 1: Bilen skal kjøre i et åtte-tall
- Mønster 2: Bilen skal gå over til å kjøre i sirkel (samme vei som den sluttet åttetallet)
- Mønster 3: Bilen skal endre retning på svingen (ikke rygge)

Oppgave 3) Linjefølger – Kalibrering av sensorer

I void setup:

Innebygd i biblioteket for linjefølgersensorene ligger en funksjon kalt `calibrate`. Denne gjør en avlesning av sensorene og lagrer responsen. Målet med kalibreringsrutinen vi skal skrive er at bilen får kjørt `calibrate`-funksjonen både på og av tape. På denne måten får bilen informasjon om forventet største og minste verdi fra sensorene.

```
linesensor.calibrate();
```

Den enkleste måten vi kan gjøre dette på er å låse fast bilen i en `while`-løkke, og kontinuerlig kjøre `calibrate`-funksjonen. Nå må vi få flyttet bilen fra teip til ikke-teip, mens denne `while`-løkken kjører. Dette bruker vi motorene på bilen til. Sett bilen i til å snurre om sin egen akse så lenge `while`-løkken kjører.

Det er mange veier å gå for å komme oss ut av `while`-løkken. En knapp, en teller som når en verdi, et tidsstempel som blir nådd. Dette er opp til dere å implementere. Merk at det å bruke knapp her vil bli litt kontraintuitivt, da det ikke er så lett å trykke på en knapp når bilen snurrer.

I void loop:

For å se om dette fungerer behøver vi ikke gjøre mye i `loop`. Vi trenger å lese linjesensor, lagre i variabel, plassere pekeren på lcd-skjermen på punkt (0, 0) og skrive ut variabelen til lcd. Dette gjør vi i `loop`.

```
int position = linesensor.readLine(linesensorValues)
```

Oppgave 4) Linjefølger – Motorstyring

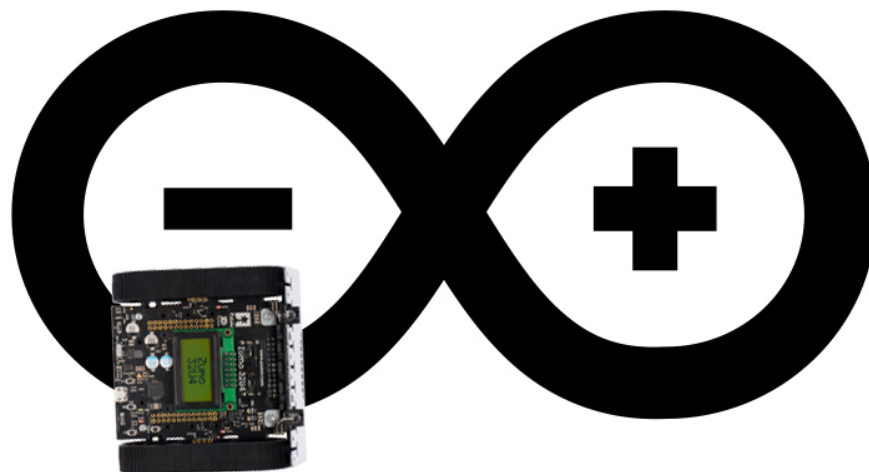
Nå sitter vi igjen med en variabel som forteller oss veldig mye om hvor bilen er i forhold til teipen. Husk at verdien skal være 2000 dersom bilen er midt på, 0 dersom den er alt for langt til høyre og 4000 når den er alt for langt til venstre. Dette kan vi definitivt bruke til å styre motorene.

Vi skal nå programmere inn et par spørsmål til bilen.

- Hvis sensorverdien er litt mindre enn 2000, hva skal motorene gjøre?
- Hvis sensorverdien er litt større enn 2000, hva skal motorene da gjøre?
- Hvis sensorverdien er veldig nært 2000, hva skal motorene gjøre?

Her er det opp til deg å gå tilbake til tidligere oppgaver, eller bruke Cheatsheet for å programmere motorene. Eller spørre en labassistent. Komplette pseudokode for bilen blir noe som dette.

- Sette opp variabler, og elementer vi bruker fra Zumo-biblioteket
- Kalibrere sensorene
- Still spørsmålene ovenfor
- Gjør handling på motorene avhengig av svaret vi fikk



Finpuss – Ferdigstilling av kode

Nå som delkomponentene har blitt utprøvd og implementert gjenstår å ferdigstille koden. Vi ønsker at LCD-skjermen skal være så informativ at en person uten kjennskap til koden skal kunne sette i gang bilen. Dette innebærer å oppdatere LCD med nødvendig informasjon om hva som forventes av brukerinntut og hva koden gjør akkurat nå.

Følgende informasjon skal implementeres:

- Bilen skal si fra at den er klar for å plasseres på tape og kalibreres. Bruker får beskjed om å trykke knapp A for å starte kalibrering.
- Etter kalibrering skal bilen stoppe opp, og informere bruker om at prosessen er utført.
- LCD-skjerm sier at bilen er ferdigkalibrert, og at bruker kan trykke knapp A for å starte linjefølgeren.
- Når bilen er i linjefølger-modus skal posisjonstallet skrives ut på LCD