NTNU

Kunnskap for en bedre verden

EV3 LEGO-prosjekt 4

Rapport

Gruppe 4

IDATG1004 - Teambasert Samhandling

22. september - 10. oktober

Innhold

1 Innledning

I denne øvingen skulle vi bygge og programmere en LEGO-racerrobot som kunne følge en forhåndsbestemt bane og konkurrere mot andre grupper. Målet var å bruke kunnskapen vi hadde tilegnet oss fra tidligere øvinger til å utvikle en rask og robust robot. I tillegg ble vi oppfordret til å benytte objektorientert programmering for å strukturere koden mer effektivt og gjøre den enklere å videreutvikle. Øvingen skulle også være en kreativ utfordring hvor roboten både kunne følge linjen presist og eventuelt sabotere motstanderen på en kontrollert måte.

2 Problemstilling

Hvordan kan vi programmere en LEGO EV3-robot ved hjelp av objektorientert programmering slik at den følger en linje raskt og stabilt, samtidig som den håndterer svinger, kryss og variasjoner i lysforhold på banen?

3 Planlegging

- Sven og Jørgen: Ansvar for ombygging av robot.
- Oliver: Ansvarlig for opprettelse og vedlikehold av wiki-siden og issue-boardet i GitLab.
- Helle, Sadaf og Jørgen: Ansvar for programmeringen, samt skrive rapport.

4 Ombygging av robot

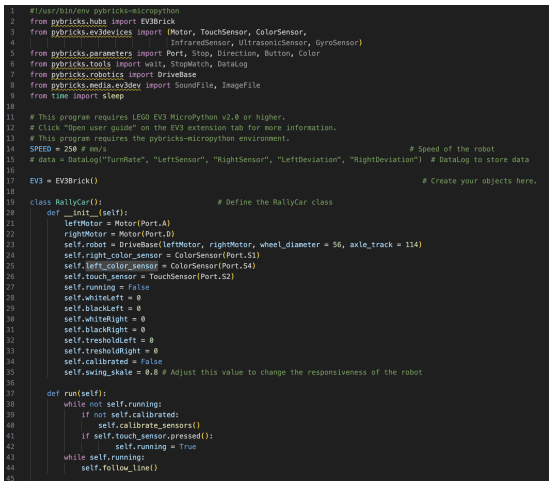
Vi benytter tidligere LEGO EV3 robot. På den tidligere roboten, legger vi til en ekstra Color Sensor. Grunnen til dette er fordi, vi ønsker å ha en sikkerhet dersom en av sensorene kjører av den svarte linjen. Da er det enklere at den kommer seg tilbake i sporet igjen.

5 Programmering

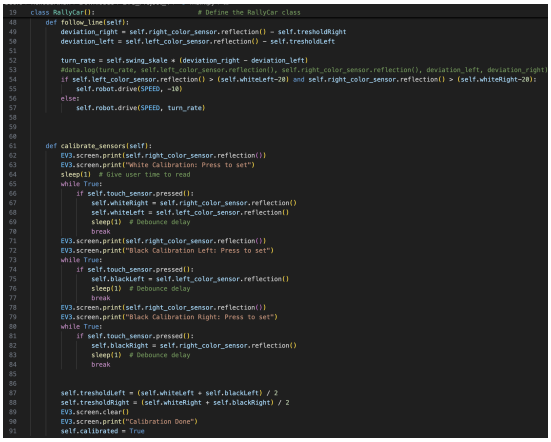
I programmet starter vi med å hente ulike bibliotek som vi trenger for at koden vår skal fungere. Deretter definerer vi en RallyCarklasse som setter opp motorer, sensorer og variabler når programmet blir startet. Før roboten begynner å kjøre, må den kalibreres:

- Først plasseres begge sensorene over hvit bakgrunn og trykkes på knappen.
- Deretter plasseres venstre sensor over svart linje og trykkes igjen.
- Til slutt det samme for høyre sensor.

Etter kalibreringen vil programmet regne ut terskelverdier mellom svart og hvit for begge sensorene. Når touchsensoren blit trykket på igjen, vil roboten starte å følge den svarte linjen. I hele perioden vil den måle lysrefleksjon fra begge sensorene og sammenligner de terskelverdien som kommer fra hver. I svingene vil roboten bruke begge colorsensorene til å styre svingen slik at roboten holder seg på linjen.



Figur 1: Utsnitt av programmet 1



Figur 2: Utsnitt av programmet 2

```
95    rally_robot = RallyCar()
96    rally_robot.run()
```

Figur 3: Utsnitt av programmet 3

6 Testing

Etter at vi bygget roboten og skrev programmet i Python ved bruk av pybricks-micropython, gjennomførte vi flere tester for å justere verdiene til lyssensorene. Vi startet med kalibrering av sensorene for å finne gjennomsnittet mellom refleksjon fra svart og hvitt underlag. Dette gjorde roboten i stand til å beregne en terskelverdi (threshold) for linjefølging.

Etter kalibreringen testet vi roboten på:

- Rett linje – for å sikre stabil kjøring uten svinging.
- Svinger – for å justere svingeskala swing-skale = 0.84 slik at roboten ikke kjørte ut av banen.
- Kryss – for å kontrollere at roboten reagerte riktig når begge sensorer registrerte hvitt felt.

7 Diskusjon

- Sensorene kan være unøyaktige hvis lys- eller lydforhold varierer.
- Ultralydsensoren må være riktig plassert for å gi pålitelige målinger.
- Fargesensoren kunne noen ganger ikke oppfatte riktig farge, og gikk derfor ut av linjen.
- Arkene med den svarte linjen kunne være litt bøyd, og derfor fikk ikke sensorene til å oppfatte alt.

8 Konklusjon

Vi klarte å utvikle en LEGO EV3-racerrobot som kunne følge banen stabilt ved hjelp av to fargesensorer og objektorientert programmering. Robotens ytelse ble betydelig forbedret etter grundig kalibrering og justering av svingparametere. Prosjektet ga oss verdifull erfaring i både programmering, testing og feilsøking av roboter. Vi lærte viktigheten av strukturert kode, gode tester og små justeringer for å oppnå høy presisjon.

9 Kilder

- `https://docs.pybricks.com/en/latest/pupdevices/ultrasonicsensor.html`
- `https://docs.pybricks.com/en/latest/pupdevices/colorsensor.html`