



**NTNU**

Kunnskap for en bedre verden

# EV3 LEGO-prosjekt 3

## Rapport

**Gruppe 4** .....

**IDATG1004 - Teambasert Samhandling** .....

**22. september - 10. oktober** .....

# Innhold

1 Innledning	2
2 Problemstilling	2
3 Planlegging	2
4 Bygging av robot	2
5 Programmering	2
6 Testing	3
7 Diskusjon	3
8 Konklusjon	3
9 Kilder	4

# 1 Innledning

I dette prosjektet skal vi programmere en LEGO-robot som kan kombinere sensor teknologi og underholdning. Roboten skal følge en svart linje ved hjelp av fargesensoren, stoppe hvert tiende sekund for å utføre et tilfeldig valgt underholdningsnummer, og til slutt registrere en hindring med ultralydsensoren. Målet er å utforske hvordan Pybricks kan brukes til å styre robotens bevegelser og handlinger på en kreativ og teknisk måte.

## 2 Problemstilling

Problemstillingen i dette prosjektet er hvordan vi kan programmere en LEGO-robot ved hjelp av Pybricks til å følge en svart linje med fargesensor, stoppe med jevne mellomrom for å utføre tilfeldige underholdningsnummer, og deretter registrere en hindring med ultralydsensoren. Vi ønsker å undersøke hvordan kombinasjonen av sensorer, tidsstyring og tilfeldigheter kan brukes til å skape en funksjonell og underholdende robot.

## 3 Planlegging

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

## 4 Bygging av robot

Vi benyttet robot fra prosjekt 1 og 2, men gjorde noen få endringer. I prosjekt 1 og 2 benyttet vi følgende oppskrifter: <https://assets.education.lego.com/v3/assets/blt293eea581807678a/bltc8481dd2666822ff/5f8801e3f4f4cf0fa39d2fef/ev3-rem-driving-base.pdf> og <https://assets.education.lego.com/v3/assets/blt293eea581807678a/blte04fb7bf4f7165f8801e3bf5ab07ee90076c9/ev3-ultrasonic-sensor-driving-base.pdf?locale=en-us>.  
I prosjekt 3 la vi til en "color sensor", slik at den følger den svarte linjen.

## 5 Programmering

Koden ble skrevet i Pybricks MicroPython v2.0. Hovedkomponentene i programmet er: En terskelverdi ble beregnet fra antatte svart- og hvitverdier (BLACK=5, WHITE=85). Avviket mellom måling og terskel ble multiplisert med en P-gain på 1.2. Dette ga en styringsvinkel som holdt roboten på linjen.

Tidsstyrte stopp: Ved hjelp av ‘time.time()’ sjekket programmet hvert 10. sekund om roboten skulle stoppe og utføre underholdning.

Underholdning: Funksjonen ‘entertain()’ valgte tilfeldig mellom fire alternativer:

- 1. Spille av lyden FANFARE.
- 2. Spille av lyden HELLO.

- 3. Utføre en liten «dans» ved å snu seg til høyre og venstre.
- 4. Si setningen «I am a robot».

```

1  #!/usr/bin/env pybricks-micropython
2  from pybricks.hubs import EV3Brick
3  from pybricks.ev3devices import Motor, TouchSensor, ColorSensor,
4      InfraredSensor, UltrasonicSensor, GyroSensor
5  from pybricks.parameters import Port, Stop, Direction, Color
6  from pybricks.tools import wait, Stopwatch, DataLog
7  from pybricks.robotics import DriveBase
8  from pybricks.media.ev3dev import SoundFile, ImageFile
9  import time
10 import random
11
12
13 # This program requires LEGO EV3 MicroPython v2.0 or higher.
14 # Click "Open user guide" on the EV3 extension tab for more information.
15
16 # Initialize the sensors.
17 obstacle_sensor = UltrasonicSensor(Port.S4)
18 touch_sensor = TouchSensor(Port.S3)
19 line_sensor = ColorSensor(Port.S1)
20
21 # Create your objects here.
22 ev3 = EV3Brick()
23
24 # Initialize the motors.
25 left_motor = Motor(Port.B)
26 right_motor = Motor(Port.C)
27
28 # Initialize the drive base.
29 robot = DriveBase(left_motor, right_motor, wheel_diameter=55.5, axle_track=104)
30
31 # Calculate the light threshold. Choose values based on your measurements.
32 BLACK = 5
33 WHITE = 85
34 threshold = (BLACK + WHITE) / 2 # Set the threshold to the average of black and white.
35
36
37 # Set the drive speed at 100 millimeters per second.
38 DRIVE_SPEED = 80
39
40 # Set the gain of the proportional line controller. This means that for every
41 # percentage point of light deviating from the threshold, we set the turn
42 # rate of the drivebase to 1.2 degrees per second.
43
44 # For example, if the light value deviates from the threshold by 10, the robot
45 # steers at 10*1.2 = 12 degrees per second.
46 PROPORTIONAL_GAIN = 1.2
47
48 start = time.time()

50 def entertain(): # Function to entertain the user
51     rand_nr = random.randint(1, 4) # Generates random number between 1 and 4
52     robot.stop()
53     if rand_nr == 1: # If the random number is 1, play fanfare sound
54         ev3.speaker.play_file(SoundFile.FANFARE)
55     elif rand_nr == 2: # If the random number is 2, play hello sound
56         ev3.speaker.play_file(SoundFile.HELLO)
57     elif rand_nr == 3: # If the random number is 3, do a little dance
58         robot.turn(45)
59         robot.turn(-90)
60         robot.turn(45)
61
62     else: # Else, say "I am a robot"
63         ev3.speaker.say("I am a robot")
64
65 # Start following the line endlessly.
66 while True:
67     end = time.time() # Get the current time
68     # If 10 seconds have passed, play entertainment
69     if end - start >= 10: # Check if 10 seconds have passed
70         entertain() # Call function entertain
71     start = time.time() # Reset the start time to the current time
72
73 # Calculate the deviation from the threshold.
74 deviation = line_sensor.reflection() - threshold
75
76 # Calculate the turn rate.
77 turn_rate = PROPORTIONAL_GAIN * deviation
78
79 # Set the drive base speed and turn rate.
80 robot.drive(DRIVE_SPEED, turn_rate)
81
82 while obstacle_sensor.distance() < 200: # If an obstacle is detected within 200 mm
83     robot.stop() # Stop the robot
84     ev3.speaker.play_file(SoundFile.FANFARE) # Play fanfare sound
85
86 # You can wait for a short time or do other things in this loop.
87 wait(10)
88
89

```

Figur 2: Utsnitt av programmet

Figur 1: Utsnitt av programmet

## 6 Testing

Vi testet robotten i på en svart linje som har form som en firkant. Robotten fulgte den svarte linjen, samt danset og jublet med 10 sekunders mellomrom. Vi observerte at når den skulle danse, kunne den gå ut av sporet. Da vi endret litt på programmeringskoden, var det i orden.

## 7 Diskusjon

- Sensorene kan være unøyaktige hvis lys- eller lydforhold varierer.
- Ultralydsensoren må være riktig plassert for å gi pålitelige målinger.
- Sensorene kan være defekte.
- Fargesensoren kunne noen ganger ikke oppfatte riktig farge, og gikk derfor ut av linjen.

## 8 Konklusjon

Vi bygde og programerte en LEGO EV3-robot som: følger en svart linje ved hjelp av fargesensor, stopper hvert tiende sekund og utfører tilfeldig underholdning, og stopper

foran en hindring for å spille en lyd.

Prosjektet ga oss praktisk erfaring med Pybricks-programmering, sensorer og teambasert problemløsing. Videre arbeid kan forbedre presisjonen i linjefølgeren og variasjonen i underholdningsdelen.

## 9 Kilder

<https://assets.education.lego.com/v3/assets/blt293eea581807678a/bltc8481dd2666822ff/5f8801e3f4f4cf0fa39d2fef/ev3-rem-driving-base.pdf>