Øving 4

INGT1001 - Ingeniørfaglig innføringsemne

Formål med øvingen

Benytte det dere har lært fra de forrige tre øvingene til å lage en racer-robot. Målet er å bli kjent med programmering av hele Legosettet, samt få utløp for litt kreativitet og ha det litt gøy[1].

Oppgavebeskrivelse

Det er på tide å bruke det dere har lært hittil i faget til å bevise at deres lego-robot er Gløshaugens beste! Roboten deres skal nå følge en oppmerket bane av samme type som i forrige øving, men denne gangen er banen mer kompleks. I tillegg skal dere konkurrere mot andre grupper sin robot. Siden dere ikke har fullført Exphil enda har dere lov til å programmere roboten deres slik at den kan sabotere for motstandernes robot under konkurransen. Dette betyr selvsagt at deres robot også kan bli utsatt for sabotasje. Dette, kombinert med den mer krevende banen, gjør at det er viktig å fokusere på robusthet og testing i denne øvingen.

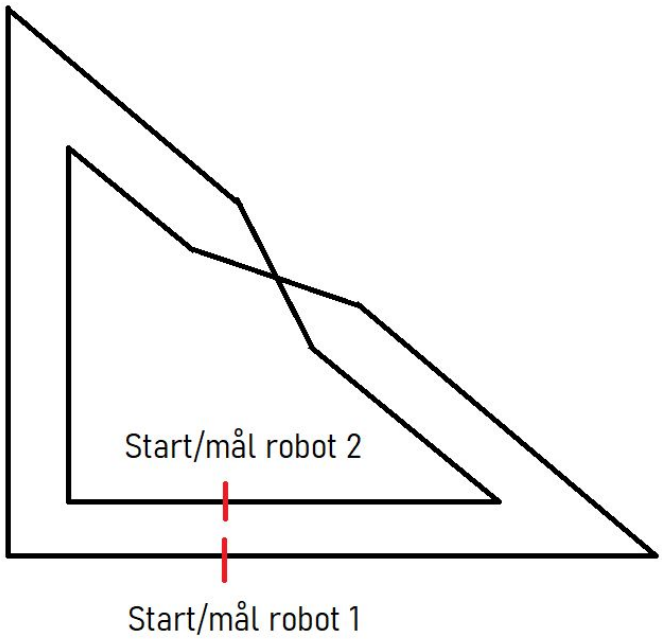
Informasjon om banen

Banen vil, som i forrige øving, bestå av en svart strek (15 mm bred) på hvit bakgrunn. Det vil være minst 10 cm fra ytterste del av banen til kanten av papiret banen er printet på. Banen kommer til å bestå av to filer, og det vil være et krysningspunkt mellom filene slik at de to robotene som kjører mot hverandre har samme kjørebane. Altså vil begge robotene kjøre både innerste og ytterste bane én gang, og kjøre krysset to ganger. Det vil være 20 cm mellom de to filene, unntaket er selvsagt i krysningspunkt.

Hardkoding av kjørebanen er ikke tillatt, og uansett en dårlig idé da vi i læringsstaben velger bane på konkurransedagen ut i fra været, månens posisjon, smaken på morgen-kaffen og Bernts favoritthøyde på gress.

Et eksempel på en bane som oppfyller kravene er vist i illustrasjonen nedenfor.

[1] https://i.ibb.co/xDzxvst/fun.jpg



Konkurranse

Når øvingen er avsluttet vil det arrangeres en konkurranse. Gruppenes roboter skal da kjøre parvis etter følgende regler:

- Hele banen skal kjøres (altså både innerste og ytterste bane, med skift i krysset)
    - Det er lov med avstikkere, men totalt sett må hele banen gjennomføres
- Robotene deres kan sabotere for motstanderens robot
- Robotene skal ikke berøres under konkurransen, men det er lov å holde dem helt frem til start.
    - Dette betyr at python-programmet kan være kjørende, men dere unngår at roboten kjører avgårde før start dersom dere holder den i lufta
- Roboten skal heller ikke styres underveis hverken vha. Bluetooth eller ved å stimulere sensorer på noe vis.
- Dersom roboten ved start er bredere enn at det er plass til to roboter ved siden av hverandre (det vil i praksis si ca. 20 cm), vil den måtte starte bak konkurrenten.
- Roboten skal bygges utelukkende av delene som er tilgjengelige i settet.

Leveranse

Deltagelse i konkurransen og fremvisning av kode til en av læringsassistentene.

Tips og kode-eksempler

Husk å kode slik at det er enkelt å kalibrere fargesensor, se kode-eksempel fra øving 3.

Sørg for at roboten deres håndterer rette strekninger, svinger og kryss, samt å bli sabotert av motstandernes robot.

Kode-eksempel: objektorientert robot

Hittil har vi ikke lagt opp til at dere må kode objektorientert, men vi ønsker å oppmuntre til det i denne øvingen.
I python trenger enhver klasse en konstruktør, som man definerer med `__init__(self)`-metoden. Man kan legge til flere parametere etter `self` dersom man ønsker at klassen skal ta inn parametere ved instansiering.
Metoder defineres nesten som funksjoner i python, men med `self` som (første) parameter. Man kan kalle metoder både inne i selve klassen (`self.follow_line_and_sabotage()`) og på et instansiert objekt av klassen (`rally_robot.run()`).

```
class RallyRobot():
    def __init__(self):
        """
        Initialize the RallyRobot with a DriveBase and two color-sensors
        """
        # We don't need the motors for anything else than the DriveBase,
        # so these are not defined as class-variables
        leftMotor = Motor(Port.B)
        rightMotor = Motor(Port.C)
        # We need the robot-variable and the color-sensor variables for
        # our program to execute how we want, so we define these as
        # class-variables by writing 'self.' before them.
        # This way, we can access them in class-methods later on
        self.robot = DriveBase(leftMotor, rightMotor,
                               wheel_diameter=56, axle_track=114)
        self.right_color_sensor = ColorSensor(Port.S4)
        self.left_color_sensor = ColorSensor(Port.S3)
        # If you want to use more variables or sensors, define them in
        # this __init__-method

    def run(self):
        """
        Runs indefinitely
        """
        while True:
            self.follow_line_and_sabotage() #Example of how to use methods
```

```
    def follow_line_and_sabotage(self):
        """
        Follows the rally-path
        """
        # You can access all variables you defined as class-variables
        self.robot.drive(200, 0) #Example of how to use class-variables

# Instantiate an object of your class
rally_robot = RallyRobot()
# Call the 'run'-method to start your robot
rally_robot.run() #Another example of how to use methods
```