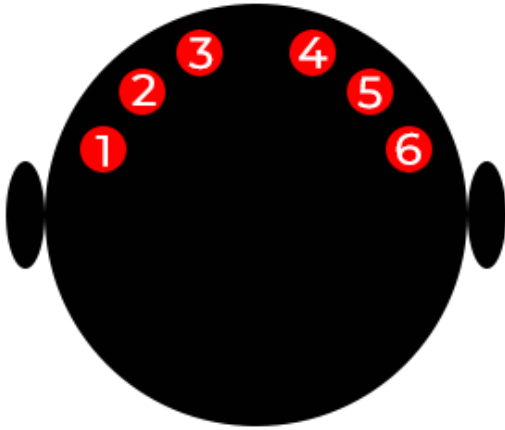


Line follower

Selles ülesandes äratate te ellu ühe vahva kuue anduriga roboti.



Roboti liigutamiseks on kolm funktsiooni:

1. Vasaku ratta kiiruse määramine:
 - `set_left_wheel_speed(kiiruse protsent)`
 - Näiteks `set_left_wheel_speed(50)`
2. Parema ratta kiiruse määramine:
 - `set_right_wheel_speed(kiiruse protsent)`
3. Mõlema ratta kiiruse määramine
 - `set_wheels_speed(kiiruse protsent)`

Andurite kasutamine:

Andurid tuvastavad taustal asuvat värvi, mille väärtus on vahemikus 0 (must) kuni 1024 (valge)

Ühe anduri väärtuse saamiseks on meetodid sellised:

1. `get_third_line_sensor_from_left()` (andur 1)
2. `get_second_line_sensor_from_left()` (andur 2)
3. `get_left_line_sensor()` (andur 3)
4. `get_right_line_sensor()` (andur 4)
5. `get_second_line_sensor_from_right()` (andur 5)
6. `get_third_line_sensor_from_right()` (andur 6)

Mitme anduri väärtuste korruga saamiseks on samuti võimalusi:

1. `get_left_line_sensors()` (andurid 1, 2 ja 3)
2. `get_right_line_sensors()` (andurid 4, 5 ja 6)
3. `get_line_sensors()` (kõik andurid)

Asukoht

Kui tahad teada, kus robot asub, saad selle teada meetodiga

get_position()

Vastuse saad pikslites, üks piksel = 1cm (selles ülesandes)

Sleep

Kui oled paika pannud kõik liikumise ja andurite tegevused, peab roboti tegutsema panekuks kutsuma välja meetodi **sleep(aeg)**

Aeg tuleb sisestada kas täisarvu või ujukomaarvuna nt 2 või 2.5.

Tegevuse lõpp

Kui kõik liikumine, andurid ja sleep on paigas, tuleb kutsuda välja funktsioon **done()**. Seejärel salvestatakse pilt roboti teekonnast ("robot_path.png"), mida saad pärast ise vaadata, et näha, mida robot tegi.

Näidis koos seletustega:

Roboti importimine ja roboti loomine on teie eest õigete koordinaatidega juba tehtud!

<code>from FollowerBot import FollowerBot</code>	Bot importitakse, et teda saaks kasutada.
<code>robot = FollowerBot()</code>	Luuakse robot, muutujanimi pole oluline, aga mõistlik on siia 'robot' panna :)
<code>robot.set_wheels_speed(50)</code>	Roboti mõlemale rattale määratakse kiiruseks 50%.
<code>robot.sleep(2.5)</code>	Sleep ajal robot sõidab.
<code>robot.set_wheels_speed(0)</code>	Roboti rattad peatatakse (kiirus = 0%)
<code>robot.done()</code>	Lõpetamise käsuga salvestatakse pilt, kus saad näha roboti teekonda.

Koodis on suured vahed, sest nii sai selgemalt seletusi kõrvale kirjutada, parem paigutus oleks:

```
from FollowerBot import FollowerBot

robot = FollowerBot()

robot.set_wheels_speed(50)
robot.sleep(2.5)
robot.set_wheels_speed(0)
robot.done()
```

Enne ülesannete juurde minekut on soovituslik näidiskoodi ja juhiste abil robotiga pisut tutvuda. Katsetada saad meetodis "katsetus".

Roboti importimine ja roboti loomine on teie eest õigete koordinaatidega juba tehtud!

ESIMENE ÜLESANNE

Funktsioon koodis: "jooneni"

1. Robot sõidab musta jooneni.
2. Jooneni jõudes jääb robot seisma.
3. Programm salvestab pildi roboti teekonnast.

Vihjed:

Mõttele, milliseid andureid oleks mõistlik kasutada joone tuvastamiseks. Näidiskoodis on näha, kuidas robot liikuma ja seisma panna.

Joone tuvastamisel pea meeles, et musta värvi väärtus on siin 0 ja valge väärtus 1024.

Mõistlik oleks roboti liikumist ja anduri kontrollimist teostada while loopi sees.

If-lausega kontrollida, kas anduri tulemus võrdub mustaga / ei võrdu valgega.

TEINE ÜLESANNE

Teine, kolmas ja neljas ülesanne tuleks teha funktsioonis: "joonejalgija"

1. Robot sõidab musta joone alguspunktini.
2. Robot sõidab mööda musta sirget joont.
3. Joone lõpus jääb robot seisma.
4. Programm salvestab pildi roboti teekonnast.

Vihjed:

Esmalt tuleks sensorite abil sõita joone alguseni (siin saab rakendada esimest ülesannet).

Robot asetseb kohe joone alguspunktiga kohakuti, kuid pead sõitma veel edasi, et selleni jõuda.

Roboti liikumine ja andurite kasutamine tuleks panna taaskord while loopi sisse.

Mõttele, kuidas anduritega kogu aeg jälgida, kas robot asub mustal joonel ning kui ta enam mustal joonel ei asu, siis kuhu poole tuleks liikuda.

Kui must joon kaob parempoolsete andurite vaateväljast, tuleks liikuda vasakule ja vastupidi.

KOLMAS ÜLESANNE

Teine, kolmas ja neljas ülesanne tuleks teha funktsioonis: joonejalgija

1. Robot sõidab musta joone alguspunktini.
2. Robot sõidab mööda musta kaart.
3. Joone lõpus jääb robot seisma.
4. Programm salvestab pildi roboti teekonnast.

See ülesanne on enamjaolt sama, mis teine ülesanne. Roboti andurid ja liikumine tuleks teha täpsemaks, et ta kumerusest üle ei sõidaks ning ära ei eksiks.

Siin ja ka järgmises ülesandes tuleb seega palju katsetada, et leida õiged kiirused raja edukaks läbimiseks.

NELJAS ÜLESANNE

Teine, kolmas ja neljas ülesanne tuleks teha funktsioonis: joonejalgija

1. Robot sõidab musta joone alguspunktini.
2. Robot sõidab mööda 90-kraadise nurgaga joont.
3. Joone lõpus jääb robot seisma.
4. Programm salvestab pildi roboti teekonnast.

Sisuliselt sama, mis teine ja kolmas ülesanne. Kui robot seni kirjutatud koodiga seda rada edukalt ei läbi, peab programmi veel pisut putitama.

NB! Uut ülesannet tehes jälgi, et ka eelmise ülesande rada saab edukalt läbitud.

Kui soovid ise uusi radu robotile teha, siis pea silmas, et joone paksus peab olema 5px.