

LINE FOLLOWER

BY SANJAY AND ARVIND SESHAN

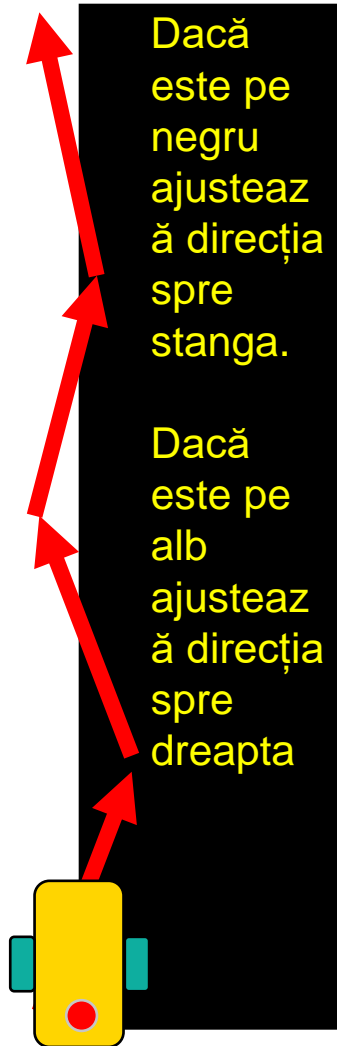
This lesson uses SPIKE 3 software

OBIECTIVELE LECȚIEI

- Învățăm cum putem face robotul să urmărească o linie utilizând Color Mode sau Reflected Light Mode a senzorului de culoare de pe SPIKE Prime
- Învățăm cum să combinăm senzorii, loop și condiționările

ROBOTUL URMĂREȘTE MARGINEA LINIEI

Drive
Base 1

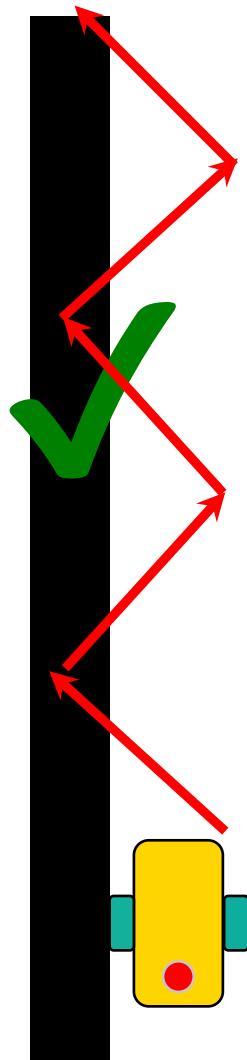
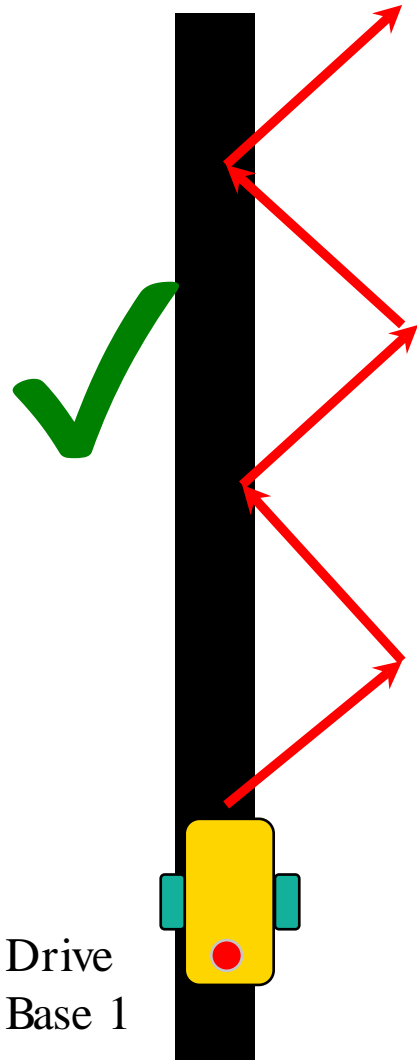


Robotul trebuie să aleagă în ce parte să ajusteze direcția atunci când senzorul de culoare vede o culoare diferită.

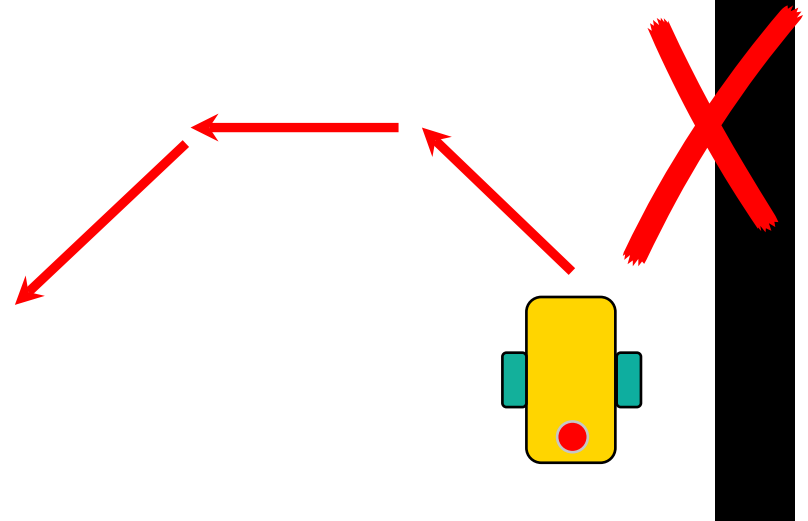
Răspunsul depinde de pe ce parte a liniei robotul urmărește linia!



PE CE PARTE A LINIEI ESTE INDICAT SĂ ÎNCEPI



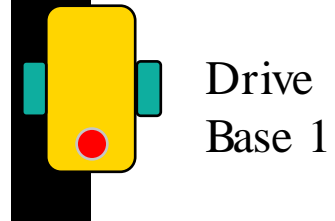
Dacă scrii un urmăritor de linie care să urmărească linia pe partea dreaptă, robotul trebuie să înceapă pe partea dreaptă a liniei.



PROVOCARE: URMĂREȘTE O LINIE

- Scrie un program care să urmărească marginea dreaptă a liniei
- Dacă senzorul vede negru, întoarce dreapta.
- Dacă senzorul vede alb, întoarce stânga.
- Utilizează block-ul IF/ELSE pentru ca robotul să execute aceste acțiuni
- Repetă următorul de linie la infinit
- Utilizează modul Color Mode sau Reflected Light Mode

Notă: Pentru a urmări linia cu Advanced Driving Base (ADB) în modul Color Mode, va trebui să faci modificări la design-ul robotului deoarece senzorul de culoare nu recunoaște negru de la înălțimea din construcția inițială. Vezi lecția cu Senzorul de culoare



DOUĂ MODURI DE ÎNTOARCERE

În lecția anterioară, “Turning with the Gyro” explică două funcții motor pair pentru a face robotul să se întoarcă. Parcurge lecția respectivă pentru alte detalii.

1. Poți folosi `motor_pair.move` și ajustează valoare de mișcare. Această lecție folosește `steering` pentru a merge înainte.

Schimbă valorile aici. Valoarea
0 face ca robotul să meargă
înainte

```
motor_pair.move(pair, steering)
```

1. Poți folosi `motor_pair.move_tank` și poți introduce valori diferite pentru motorul din dreapta și motorul din stânga. Poți încerca, tu însuși.

Schimbă valorile de viteză
aici. Aceeași viteză face ca
robotul să meargă drept
înainte

```
motor_pair.move_tank(pair, left_velocity, right_velocity)
```

LINE FOLLOWER – MODUL CULOARE ȘI INTENSITATEA LUMINII REFLECTATE

```
from hub import port
import motor_pair, color_sensor, runloop
```

```
# Constante pentru Drive Base 1
```

```
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
```

```
# Urmărește partea dreaptă a liniei negre (Negru-Alb margine). NOTĂ: Testul este ca robotul să meargă pe  
planșă pe marginea negru-alb.
```

```
# Dacă planșa are mai multe culori, va trebui să cobori valoarea de comparație pentru a evita confuzia cu alte  
culori.
```

```
# Pentru a urmări marginea Alb-Negru, schimbă condiția de la < 50 la > 50
```

```
# Pentru a utiliza color mode, importă color, și utilizează condiția:
```

```
# dacă (color_sensor.color(port.A) == color.BLACK)
```

```
async def line_follow_forever():
```

```
    while (True):
```

```
        if (color_sensor.reflection(port.A) < 50): # sensor is on Black. Lower threshold as needed for your case.
```

```
            # Întoarce dreapta, i.e. departe de negru
```

```
            motor_pair.move(motor_pair.PAIR_1, 30, velocity = 300)
```

```
        else: # senzorul este pe alb
```

```
            # Întoarce stânga, i.e. spre Negru
```

```
            motor_pair.move(motor_pair.PAIR_1, -30, velocity = 300)
```

```
async def main():
```

```
    await line_follow_forever()
```

```
runloop.run(main())
```

EXTENSI – SCHIMBAREA CONDIȚIILOR DE IEȘIRE

■ În FLL, tipic nu vrei să urmărești linia la infinit. E posibil să-ți dorești să te oprești la anumite condiții, unele din ele pot fi:

1. Senzorul ultrasonic detectează ceva
2. Senzorul de atingere este apăsat.
3. Ai un senzor suplimentar de culoare pe robot care poate sesiza un marker de pe planșă. Asta este extrem de util în FLL.
4. Trebuie să urmărești linia pentru o anumită distanță.

■ SFAT: Poți reseta poziția individuală relativă a motorului și apoi atunci când trece peste valoarea considerată ca reper. Ia în considerare rotația în sensul acelor de ceasornic și opus acelor de ceasornic.

■ Combină această lecție cu lecția despre LOOP pentru a rezolva aceste probleme.

LINE FOLLOW UNTIL SECOND SENSOR SEES BLACK

```
from hub import port
import motor_pair, color_sensor, runloop, sys

motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)

# urmărește partea dreaptă a liniei negre (marginea Negru-Alb) până când al doilea senzor vede negru
# Testează pe planșa doar cu culorile negru și alb. Ajustează valoarea de comparație la o valoare mai mică.
async def line_follow_until_line():
    # Drive Base 1 este modificat cu un al doilea senzor de culoare pe portul B
    # Urmărește linia până când senzorul B vede negru.
    while (color_sensor.reflection(port.B) > 50): # Ajustează valoarea de comparație în funcție de necesități.
        if (color_sensor.reflection(port.A) < 50): # sensorul este pe negru. Ajustează valoarea de comparație în
            funcție de necesități
                # Întoarce dreapta, i.e. mai departe de negru
                motor_pair.move(motor_pair.PAIR_1, 30, velocity = 300)
        else: # sensorul este alb
            # Întoarce stânga, i.e. spre negru
            motor_pair.move(motor_pair.PAIR_1, -30, velocity = 300)

async def main():
    await line_follow_until_line()
    sys.exit("Stopping")

runloop.run(main())
```

LINE FOLLOW FOR APPROXIMATE DISTANCE

```
from hub import port
import motor, motor_pair, color_sensor, runloop, sys

motor_pair.pair(motor_pair.PAIR_1, port.C, port.D) # Drive base 1 (DB1)
WHEEL_CIRCUMFERENCE = 17.5 # cm – mărimea roții pentru DB1

# urmărește partea dreaptă a liniei negre (marginea Negru-Alb) până când e acoperită distanță.
async def line_follow_for_distance_cm(distance_cm):
    # Calculează numărul de grade pentru a acoperi distanța de care ai nevoie
    # Vezi lecția More Accurate Turns pentru explicații.
    motor_degrees = int((distance_cm/WHEEL_CIRCUMFERENCE) * 360)
    # Utilizează motorul D pentru DB1 deoarece robotul se mișcă în sensul acelor de ceasornic și gradele de
    rotație se adună.
    motor.reset_relative_position(port.D, 0)
    while (motor.relative_position(port.D) < motor_degrees):
        if (color_sensor.reflection(port.A) < 50): # sensor is on Black. Adjust threshold as needed if this is too high
            motor_pair.move(motor_pair.PAIR_1, 30, velocity = 300) # Întoarce dreapta
        else: # sensorul este pe alb
            motor_pair.move(motor_pair.PAIR_1, -30, velocity = 300) # Întoarce stânga

async def main():
    await line_follow_for_distance_cm(70)
    sys.exit("Stopping")
```

```
runloop.run(main())
```

CREDITS

Această lecție a fost creată de Sanjay Seshan și Arvind Seshan for SPIKE Prime Lessons

La această lecție au contribuit membrii comunității FLL Share & Learn.

Mai multe lecții sunt disponibile pe www.primelessons.org

Această lecție a fost tradusă în limba română de echipa de robotică FTC – ROSOPHIA
#21455 RO20



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).