

ÎNTO AR CERILE CU GYRO

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

LESSON OBJECTIVES

- Învățăm cum să întoarcem utilizând senzorul giroscopic

METODELE DE CARE AI NEVOIE ÎN ACEASTĂ LECȚIE

- Motion Sensor methods – Utilizată pentru citirea și resetarea valorilor senzorului gyro `motion_sensor.tilt_angles()`

- Această metodă returnează un tuple care conține 3 valori. Poți afla mai multe despre Python Tuples în lecția Lists and Tuples pe w3Schools [Python Tuples](#)

- Fiecare valoare din tuple este deci - grade (zeci de grade). Așa că verifică pentru > 90 de grade, va trebui să verifici valori > 900 .

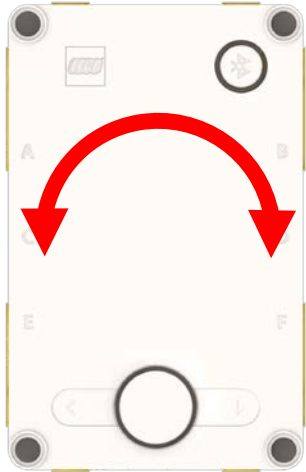
- `motion_sensor.reset_yaw()`

- `motion_sensor.stable()`

- Această metodă returnează valoarea de adevăr când senzorul va avea valoare 0.

ORIENTAREA ROBOTULUI: YAW, PITCH ȘI ROLL

Yaw înregistrează rotația
Hub-ului la dreapta sau la
stânga



Pitch înregistrează
mișcarea Hub-ului
sus și jos



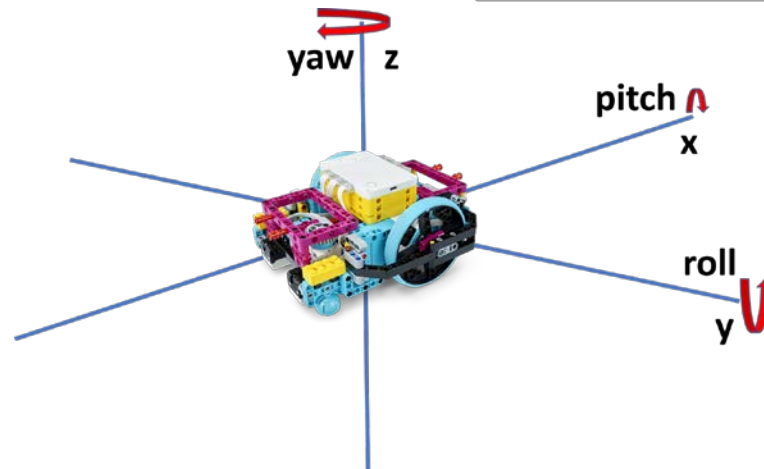
Roll înregistrează rotația
Hub-ului de pe o parte pe
alta



La fel ca și pe axele x, y și z
coordonatele sunt utilizate
pentru a descrie poziția
robotului,
yaw, pitch and roll sunt termeni
utilizați pentru a descrie
orientarea robotului.

Yaw este rotația în jurul axei z.
Pitch este rotația în jurul axei y.
Roll este rotația în jurul axei x.

Senzorul giroscopic măsoară
poziția și orientarea robotului,



UTILIZAREA SENZORULUI GYRO PENTRU ÎNTOARCERI

- Senzorul giroscopic poate fi programat pentru a măsura valorile hub-ul pentru yaw, pitch and roll
- Aceste valori pot fi utilizate pentru ca robotul să simtă dacă s-a mișcat pe una din axele x,y, sau z
- În această lecție, ne vom concentra pe „yaw” ce poate fi folosit pentru a determina dacă robotul s-a mișcat stânga sau dreapta
- Pentru „pitch” and „roll”, robotul utilizează gravitația pentru a determina ce e citirea 0. Robotul pus pe podea va duce la o citire 0 „pitch” și 0 „roll”.
- Pentru „yaw”, robotul nu are un compas pentru a spune dacă se află la nord sau la sud. De aceea trebuie să spui robotului ce ar trebui să considere 0. Acest lucru se realizează cu ajutorul block-ului “set yaw angle to 0”.
- Aceasta se realizează cu metoda `reset_yaw()`.

```
motion_sensor.tilt_angles()  
motion_sensor.reset_yaw()
```

MĂSURĂTORILE YAW ANGLE (UNGHIULUI DE ROTĂȚIE)

Valorile Yaw așa cum se văd în app:

- Măsurarea unghiului de rotație începe de la 0 și utilizează valori pozitive dacă robotul este întors spre dreapta, și mai mici decât 0 dacă robotul întoarce spre stânga.

- La semnul de 180 de grade, semnele se schimbă! Citirile unghiului de rotație arătat în aplicație vor merge de la 179 la -180 dacă se întoarce în sensul acelor de ceasornic, și -180 la 179 dacă se întoarce în sensul opus acelor de ceasornic.

- Dacă dorești să întorci mai mult de 180 de grade într-o anumită direcție, va trebui să faci câteva verificări suplimentare, pe care le vom explica mai târziu în lecție.

Valorile unghiului de rotație este citi de `tilt_angles` tuple:

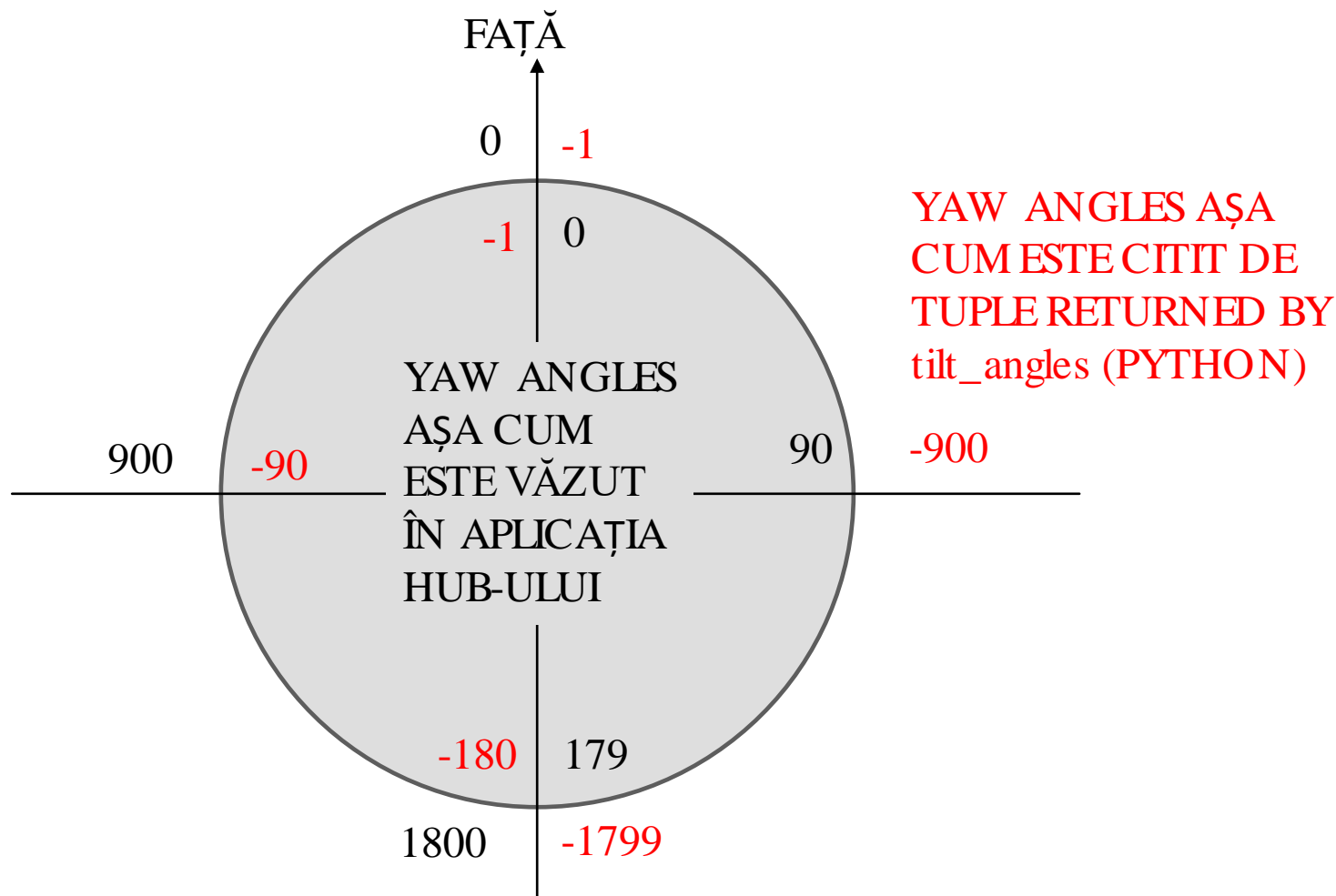
- Citirile senzorului mișcare sunt **de semn contrar** cu valorile arătate de citirile unghiului de rotație din aplicație sau din block-uri, în deci-grade (zeci de grade).

- Când întoarcem în sensul acelor de ceasornic, citirile vom merge de la 0 la -1799, și apoi de la 1800 în jos la 0

- Multiplicarea citirilor tuple cu -0.1 convertește în aceeași sistem de valori ca în App/Blocks.

- Dacă întoarcerile sunt limitate la mai puțin de 180 de grade, utilizați valori absolute pentru a evita bug-urile.

MĂSURĂTORILE YAW ANGLE (GRAPHIC)



AȘTEAPTĂ PÂNĂ CÂND SENZORUL GYRO ATINGE VALOAREA

Sunt 2 opțiuni pentru a măsura dacă robotul ajunge la unghiul dorit

Opțiunea I: LEGO-specific API

Utilizează funcția `runloop.until`. Se va aștepta ca funcția să returneze valoarea de adevăr: ADEVĂRAT.

Funcția nu poate utiliza parametri. Utilizează variabile globale așa cum se intenționează în locul parametrilor.

```
import runloop
```

```
await runloop.until(<function that returns true or false based on conditions/sensor readings>)
```

Această opțiune este mai ușor de utilizat și detaliile pot fi găsite în Spike Knowledge Base

Opțiunea II: General Python API

Utilizează WHILE LOOP

```
start moving....
```

```
while (motion_sensor.get_yaw_angle() < ANGLE):
```

```
    <code>
```

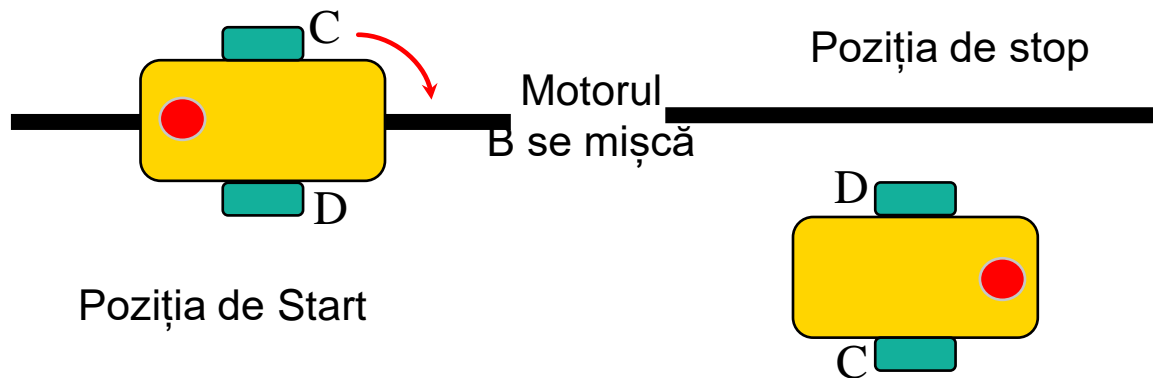
```
stop moving....
```

Mai ușor de rulat codul în timp ce funcția așteaptă. Poți de asemenea să definești `operator_function` în `wait_until()` – dar while loop face codul mai clar.

Dacă nu dorești să rulezi codul, poți plasa `pass` în locul `<code>` pentru a trece peste iterația din LOOP

SUNT 2 MODALITĂȚI DE ÎNTOARCERI PE CARE LE POȚI FACE

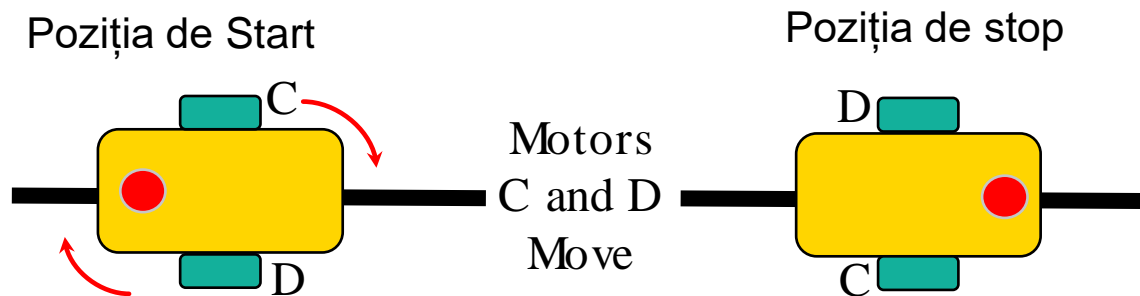
180 grade Întoarcere pe Pivot



Remarcă unde s-a oprit robotul în ambele imagini la o întoarcere de 180 de grade.

În întoarcerea de tip Spin, robotul s-a mișcat mult mai puțin, ceea ce face ca acest tip de întoarceri să fie ideale pentru spațiile strâmte. Întoarcerile Spin par a fi mai rapide dar și mai puțin precise.

180 grade Întoarcere Spin pe poziție



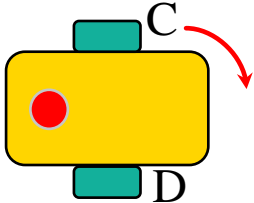
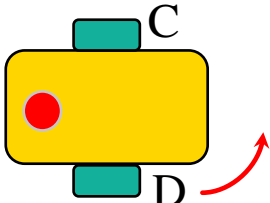
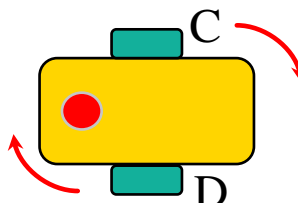
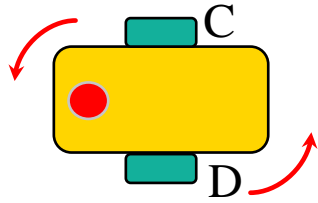
Așa că, atunci când ai o întoarcere de făcut, tu trebuie să decizi care e cea mai potrivită pentru tine!

HOW TO MAKE PIVOT AND SPIN TURNS- 1

Using move:

Change steering
value here. A value
of 0 moves straight

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

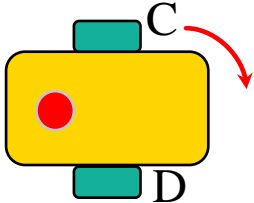
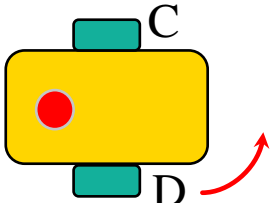
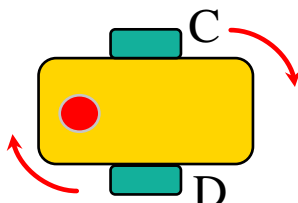
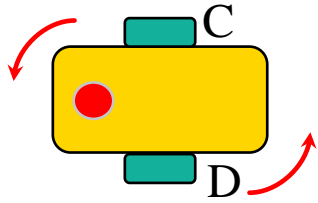
Steering Values			
50	-50	100	-100
			
Pivot Turn Right	Pivot Turn Left	Spin Turn Right	Spin Turn Left

CUM SĂ FACI ÎNTOARCERI DE TIPUL PIVOT ȘI SPIN - 11

Utilizează comanda move tank:

Schimbă valorile vitezei aici.
Aceași viteză pentru a
merge drept

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

Valorile Vitezei din comanda TANK			
200, 0	0, 200	200, -200	-200, 200
			
Întoarcere Pivot Dreapta	Întoarcere Pivot Stânga	Întoarcere Spin Dreapta	Întoarcere Spin Stânga

PROVOCAREA I

- Scrie un program care să întoarcă robotul 90 de grade spre dreapta (în sensul acelor de ceasornic), utilizând întoarcerea de tip PIVOT
- Pași de bază:
 - Definește motor pair
 - Resetează unghiul de întoarcere yaw
 - Așteaptă până când senzorul de mișcare este stabil
 - Pornește motor pair utilizând move sau move_tank după cum dorești.
 - Așteaptă până când unghiul de întoarcere yaw s-a schimbat la 90 de grade 90
 - Stop
- NOTĂ: Folosește întotdeauna operatorii `<` and `>` pentru a compara valorile senzorilor! Nu utiliza niciodată `==` deoarece senzorul este posibil să nu citească exact acea valoare și LOOP-ul nu se va opri.

PROVOCAREA 1 SOLUȚIA

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

Funcția care va returna valoarea „adevărat” când valoarea absolută a yaw angle este 90 de grade

```
def turn_done():
    # convertește „tuple” deci- grade în același format ca în app sau block-uri
    return abs(motion_sensor.tilt_angles()[0] * -0.1) > 90
```

```
async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
    motion_sensor.reset_yaw(0)
    await runloop.until(motion_sensor.stable)
    # mergi înainte cu o viteză de 50
    motor_pair.move(motor_pair.PAIR_1, 50, velocity=200)
    # Alternativ utilizează tank:
    # motor_pair.move_tank(motor_pair.PAIR_1, 200, 0)
    await runloop.until(turn_done)
    motor_pair.stop(motor_pair.PAIR_1)
    sys.exit("Done")
```

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

Copyright © 2021 PrimeLessons (primellessons.org) CC-BY-NC-SA. (Last edit: 09/14/2023)

PROVOCAREA 1I (ADVANSATĂ)

■ Scrie un program care întoarce până la 355 de grade, în sensul acelor de ceasornic sau în sens opus acelor de ceasornic, utilizând întoarcerea de tip spin.

■ Dacă robotul se întoarce în sensul acelor de ceasornic, citirile „yaw angle tuple” vor fi contorizate de la 0 până la 1800, dar va deveni negativă și va continua contorizare în jos! Reversul e adevărat dacă întorci în sensul acelor de ceasornic. Vezi graficul.

■ Pași importanți:

■ Setează întoarcerea la valori pozitive dacă se întoarce în sensul acelor de ceasornic, și negative dacă se întoarce în sens opus acelor de ceasornic the steering to positive if turning clockwise.

■ Determinați ce valoare trebuie să aibă citirea senzorului yaw atunci când robotul trebuie să se oprească, utilizând convenția standard, așa cum se vede în aplicație:

■ Dacă se întoarce în sensul acelor de ceasornic, valorile citite de yaw angle devin negative, ȘI este MAI MARE decât unghiul la care trebuie să se oprească robotul (vezi graficul și încearcă să vă dați seama ce calcule matematice trebuie făcute)

■ Dacă se întoarce în sensul opus acelor de ceasornic, valorile citite de yaw angle devin pozitive, ȘI este mai MICĂ decât unghiul la care trebuie să oprească robotul. (vezi graficul și încearcă să vă dați seama ce calcule matematice trebuie făcute)

■ Convertește citirile „tuple” în formatul standard a senzorului yaw (grade în sensul acelor de ceasornic fiind pozitive și în sensul opus acelor de ceasornic fiind negative) prin multiplicarea cu -0.1 înainte de a realiza orice verificare.

PROVOCAREA 1I (ADVANCED) SOLUȚIA PAGINA 1 DIN 5

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

```
# VALORI GLOBALE
```

```
# Setează de la -355 la 355. Positive numbers are clockwise.
```

```
degrees_to_turn = 0
```

```
# Citirile Yaw angle indică când robotul trebuie să se oprească.
```

```
stop_angle = 0
```

Setează o serie de valori globale pentru a le utiliza în program:

1. degrees_to_turn: gradele pe care robotul trebuie să le întoarcă. Valorile pozitive – întoarcere în sensul acelor de ceasornic, Valorile negative - întoarcere în sensul opus acelor de ceasornic
2. stop_angle: citirile yaw angle sunt verificate cu valoarea de întoarcere pe care ați stabilit-o.

PROVOCAREA 1I (ADVANCED) SOLUȚIA PAGINA 2 DIN 5

Funcția care returnează „adevărat” când valoarea citită de yaw trece de condiția de stop.

```
def turn_done():
```

```
    global degrees_to_turn, stop_angle
```

```
    # convertește citirile deci-grade în același format ca în app și block-uri
```

```
    yaw_angle = motion_sensor.tilt_angles()[0] * -0.1
```

```
    # dacă e nevoie să întorci mai puțin decât 180 de grade, verifică valorile absolute.
```

```
    if (abs(degrees_to_turn) < 180):
```

```
        return abs(yaw_angle) > stop_angle
```

dacă avem nevoie să întoarcem mai mult de 180 de grade, calculează diferența până la valoarea la care robotul trebuie să se oprească.

```
    if degrees_to_turn >= 0: # începe întoarcerea în sensul acelor de ceasornic
```

```
        # Valoarea ajustată a yaw angle este pozitivă până când întoarcerea va depăși 180.
```

```
        # Apoi, valorile negative se adună.
```

```
        return yaw_angle < 0 and yaw_angle > stop_angle
```

```
    else:
```

```
        # Apoi valoarea ajustată a yaw angle este negativ până depășește 180.
```

```
        # Apoi, acum valorile positive se adună.
```

```
        return yaw_angle > 0 and yaw_angle < stop_angle
```

Adaugă o funcție și apoi lasă runloop să știe când condiția este atinsă.

Funcția are verificări diferite:

1. Unghiurile a căror valoare absolută este mai mică decât 180 i.e., ei nu trec prin punctul de tranziție.
2. Unghiurile sunt mai mari de 180 în sensul acelor de ceasornic
3. Unghiurile sunt mai mari de 180 în sensul opus acelor de ceasornic.

PROVOCAREA 1I (ADVANCED) SOLUȚIA PAGINA 3 DIN 5

```
async def setupMotors():  
    motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)  
    motion_sensor.reset_yaw(0)  
    await runloop.until(motion_sensor.stable)
```

Adaugă o funcție care:

1. Setează funcția motor pair
2. Resetează yaw angle la 0
3. Așteaptă până valorile senzorului se stabilizează.

PROVOCAREA 1I (ADVANCED) SOLUȚIA PAGINA 4 DIN 5

```
async def spinTurn(degrees):
    if abs(degrees) > 355:
        print("Out of range")
        return
    await setupMotors()
    global degrees_to_turn, stop_angle
    degrees_to_turn = degrees # setează valorile globale pentru a utiliza funcția turn_done
    # setează stop_angle global pentru a fi utilizate în funcția turn_done
    if (abs(degrees) < 180):
        stop_angle = abs(degrees_to_turn)
    else:
        stop_angle = (360 - abs(degrees)) if degrees < 0 else (abs(degrees) - 360)
    # setează mișcarea bazată pe direcția de întoarcere
    steering_val = 100 if degrees >= 0 else -100
    motor_pair.move(motor_pair.PAIR_1, steering_val, velocity=200)
    await runloop.until(turn_done)
    motor_pair.stop(motor_pair.PAIR_1)
```

Adaugă spinTurn care :

1. Intru numărul de grade pentru întoarcere și realizează verificările pentru erori.
2. Setează motoarele și calculează valorile variabilelor globale
3. Pornește motor pair cu setările de mișcare corecte
4. Așteaptă ca, condiția de oprire să se îndeplinească și apoi oprește motoarele.

PROVOCAREA 1I (ADVANCED) SOLUȚIA PAGINA 5 DIN 5

```
async def main():  
    await spinTurn(270)  
    sys.exit("Done")
```

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

Puneți tot împreună:

1. Funcția principală realizează o întoarcere cu numărul de grade dorit (-355 la 355), așteaptă pentru încheiere și iese.
2. Runloop rulează funcția principală

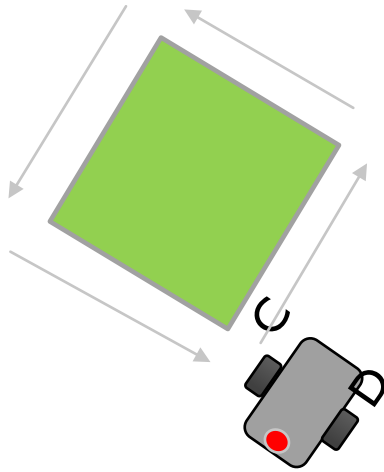
Poți extinde aceste concepte pentru a scrie funcția pivot care va întoarce robotul pe pivot între -355 la 355. Acestea sunt funcții perfecte pentru a le adăuga la o librărie.

Observați că citirile senzorului yaw readings nu sunt precise la viteză mare. Întoarceri mai încete dau rezultate mai bune. Dacă ai nevoie, poți ajusta unghiul de întoarcere dacă întoarcerea se oprește prea repede.

PROVOCAREA ÎNTOARCERE

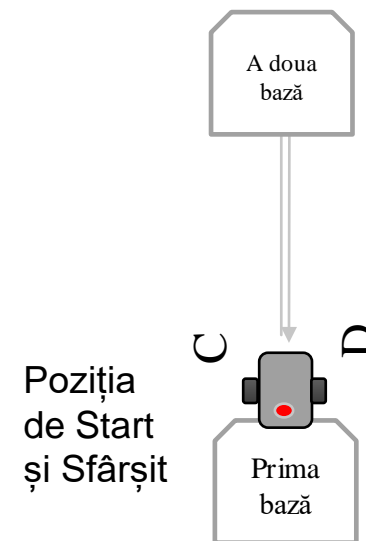
Provocarea 1

- Robotul tău este un jucător de baseball care trebuie să alerge la toate bazele și să se întoarcă înapoi pe placa de pe care a plecat.
- Poți programa robotul tău să meargă înainte și să se întoarcă stânga?
- Utilizează o cutie pătrată sau o linie trasată cu banda izoler



Provocarea 2

- Robotul tău este jucător de baseball care trebuie să alerge la baza a doua, **să se întoarcă apoi la prima.**
- Mergi înainte. Întoarce 180 de grade și se întoarce în același loc.



SOLUȚIILE PROVOCĂRII

Provocare 1

Probabil ai utilizat combinația dintre metoda `move()` pentru a merge drept și face **o întoarcere pivot turns** pentru a merge în jurul cutiei.

Provocare 2

Probabil ai folosit o întoarcere de tip spin pentru că e mai bine să folosești întoarceri strânse și ajunge mai aproape de punctul de start!

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/).