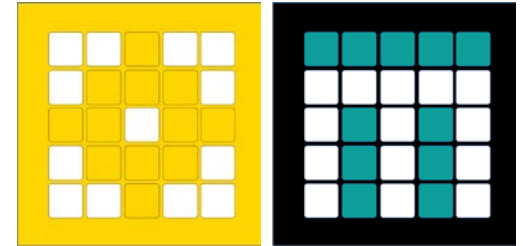


PRIME LESSONS

By the Makers of EV3Lessons



MIȘCAREA OBIECTELOR & DETECȚIA DE CALARE

BY SANJAY AND ARVIND SESHAN

This lesson uses SPIKE 3 software

OBIECTIVELE LECȚIEI

- Învăță cum să miști motoarele non-drive.
- Învăță despre calarea motoarelor.

FUNȚIILE SINGLE MOTORS (ACȚIUNI)

- Pentru a utiliza funcțiile unui singur motor, importă modulul:

```
import motor
```

- Funcția fiecărui motor, portul motorului este parametru.
- Pentru a rula codul o anumită perioadă de timp, utilizează metodele(vezi lecția Knowledge Base pentru mai multe informații). Aceste funcții sunt **asynchronous** așa că utilizează **await** dacă vrei să aștepti pentru ca task-ul să fie complet.

```
run_for_degrees(port, degrees, velocity)
run_for_time(seconds, duration, velocity)
```

- Pentru a începe funcționarea motoarelor, până la oprirea într-un alt loc.

```
run(port, velocity)
stop()
```

- Pentru a rula motorul până la o poziție specifică.

```
run_to_relative_position(port, position, velocity)
run_to_absolute_position(port, position, velocity, direction=motor.SHORTEST_PATH)
```

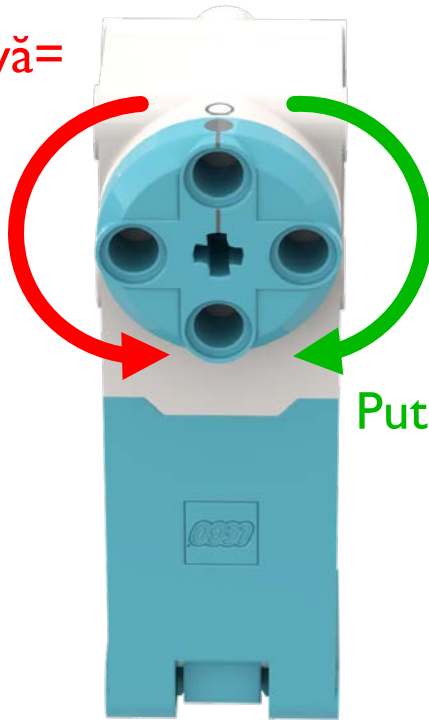
- Opțiunile de mișcare sunt în sensul CLOCKWISE, COUNTERCLOCKWISE, LONGEST_PATH
- Aceste funcții sunt **asynchronous** astfel utilizează **await** dacă vrei să aștepti ca task-ul să fie complet.

SINGLE MOTOR FUNCTIONS (MEASUREMENTS/SETTINGS)

- Senzorul de rotație din motor poate fi utilizată pentru a spune câte grade s-a rotit motorul.
- Pentru a face asta, utilizează `reset_relative_position(port, position)` și `relative_position(port)`
- La fel ca funcția `motor pairs`, poți folosi parametri pentru a schimba comportamentul motorului.
 - Setează acțiunea stop (BRAKE, COAST etc.)
 - Setează accelerația, viteza etc.
- Poți de asemenea citi diferite metode de măsurare asociat cu motorul.
 - `velocity(port)`
 - `relative_position(port)`
 - `absolute_position(port)`

VALORILE NEGATIVE

Putere negativă=
Înapoi



Putere pozitivă=
Înainte

- Poți folosi valori negative pentru putere și distanță.
- Acest lucru va face robotul să se miște înapoi.
- Dacă folosești două valori negative (de ex. Puterea și distanța sau distanța și direcția înapoi), robotul va merge înainte.

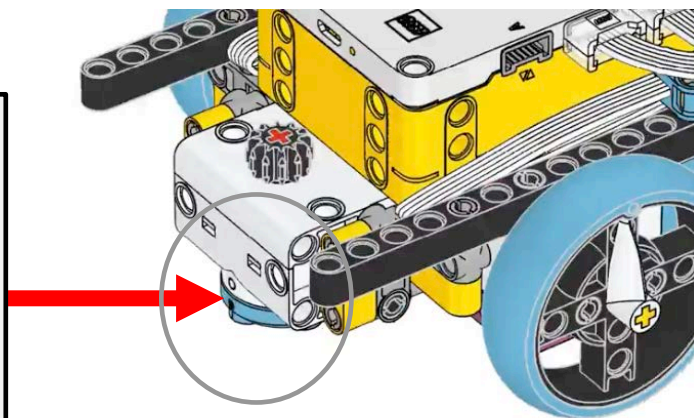
DETECTAREA CALĂRII

- Deseori, programezi motorul să se miște pentru un timp specific. Totuși, motorul devine blocat înainte să ajungă la acel moment.
- Detectarea calării presupune ca programul să se miște automat până la linia următoare din cod când comanda motorului se blochează (este imposibil să completeze mișcarea).
- SPIKE Prime are o detectarea calării implicită.
- Prin presetare, detectarea calării este **on** când utilizează funcțiile unui sinur motor
- Ca și versiune 3.4, SP3 nu permite detectarea calării să fie schimbată.

PROVOCAREA 1: ÎNVĂȚĂM DESPRE DETECTAREA CALĂRII CU DBI

- Scrie un program să întorci motorul atașamentului în portul E cu 1000 de grade.
- Aduă un sunet de beep după comanda de motor.
- Ține motorul cu mâna pentru a preveni ca acesta să efectueze 1000 de grade. Menține câteva secunde.
- Se va auzi sunetul?

Cauzează calarea prin menținerea motorului de partea blue pentru a preveni ca acesta să întoarcă.



PROVOCAREA I SOLUȚIA

- Detectarea calării permite codului să se miște la următoarea linie, chiar și dacă motorul se blochează. Beep-ul se audea chiar dacă motorul nu a terminat comanda.

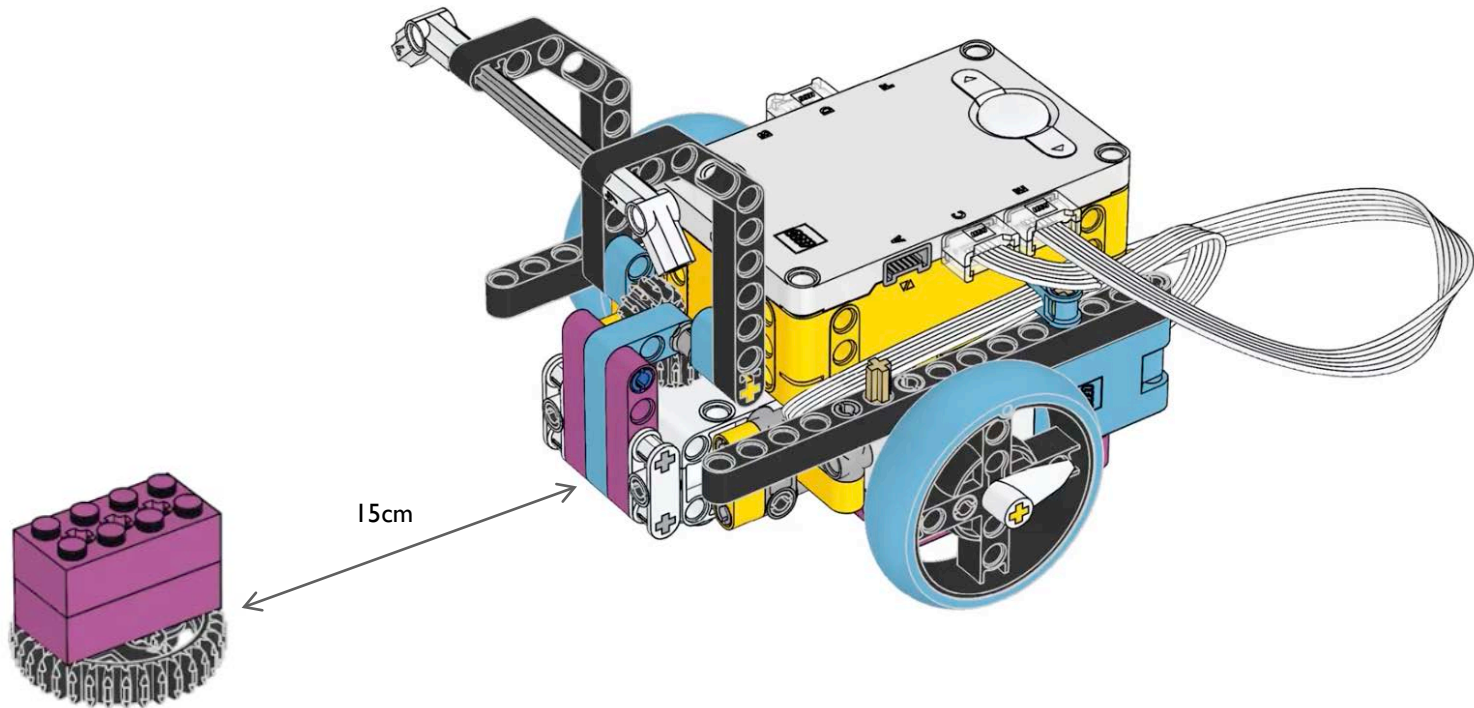
```
from hub import port, sound
import motor, runloop, sys

async def main():
    # Pornește motorul
    await motor.run_for_degrees(port.E, 1000, 100)
    await sound.beep()
    sys.exit("Stopping")

runloop.run(main())
```


PROVOCAREA 2: APUCĂ UN OBIECT (DB2 PROVOCARE)

- Construiește un atașament al DB2
- Mergi înainte, ia un marker și adu-l înapoi la start și eliberează-l.
- Marker-ul se află aproximativ la 15cm de la panel-ul magenta-bleu a robotului.



PROVOCAREA 2 SOLUȚIA

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

# Constante pentru Drive Base 1, ajustează robotul
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
WHEEL_CIRCUMFERENCE = 17.5 # cm

# Vezi lecția More Accurate Turns pentru explicații matematice
def degrees_for_distance(distance_cm):
    return int((distance_cm/WHEEL_CIRCUMFERENCE) * 360)

async def main():
    # Mergi înainte 14.5cm
    await motor_pair.move_for_degrees(motor_pair.PAIR_1, degrees_for_distance(14.5),0)
    # coboară brațul
    await motor.run_for_degrees(port.E, -90, 100)
    # mergi înapoi
    await motor_pair.move_for_degrees(motor_pair.PAIR_1, degrees_for_distance(-14.5),0)
    # ridică brațul
    await motor.run_for_degrees(port.E, 90, 100)
    sys.exit("Done")

runloop.run(main())
```

EXTENSII

- Gândește-te la anumite situații FIRST LEGO League când detectarea calăriei va fi de ajutor
Când robotul se poate bloca?

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/).