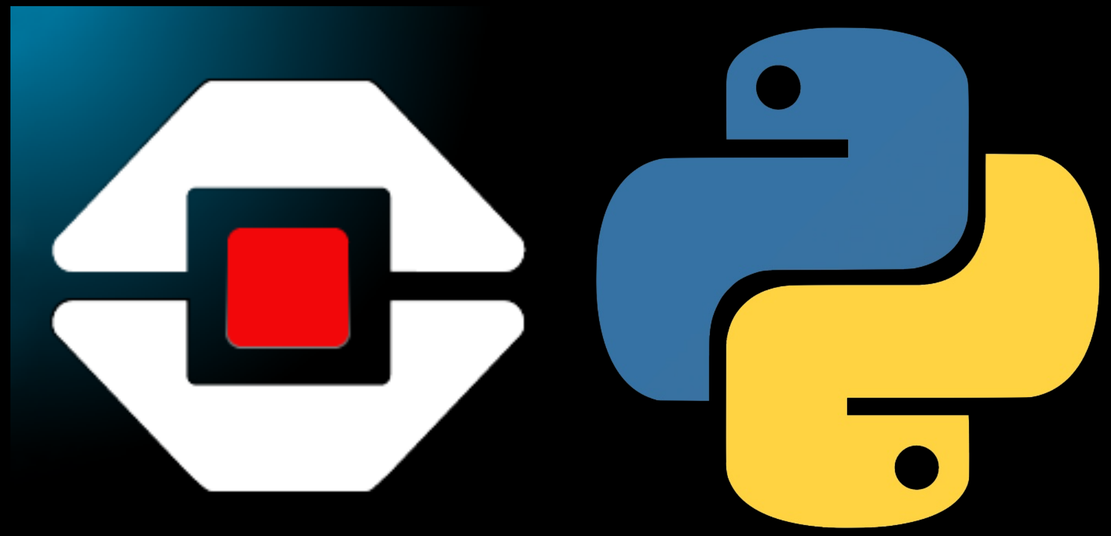


Lecția 2 - Senzorul de Culoare EV3

M. NICHITIU

Curriculum

Bazat pe:

- „Introduction to Programming EV3 Curriculum” dezvoltat de Carnegie-Mellon Robotics Academy https://www.cmu.edu/roboticsacademy/roboticscurriculum/index.html

- FLL City Shaper Challenge
  https://www.firstinspires.org/resource-library/fll/challenge/challenge-and-resources

Obiectivele

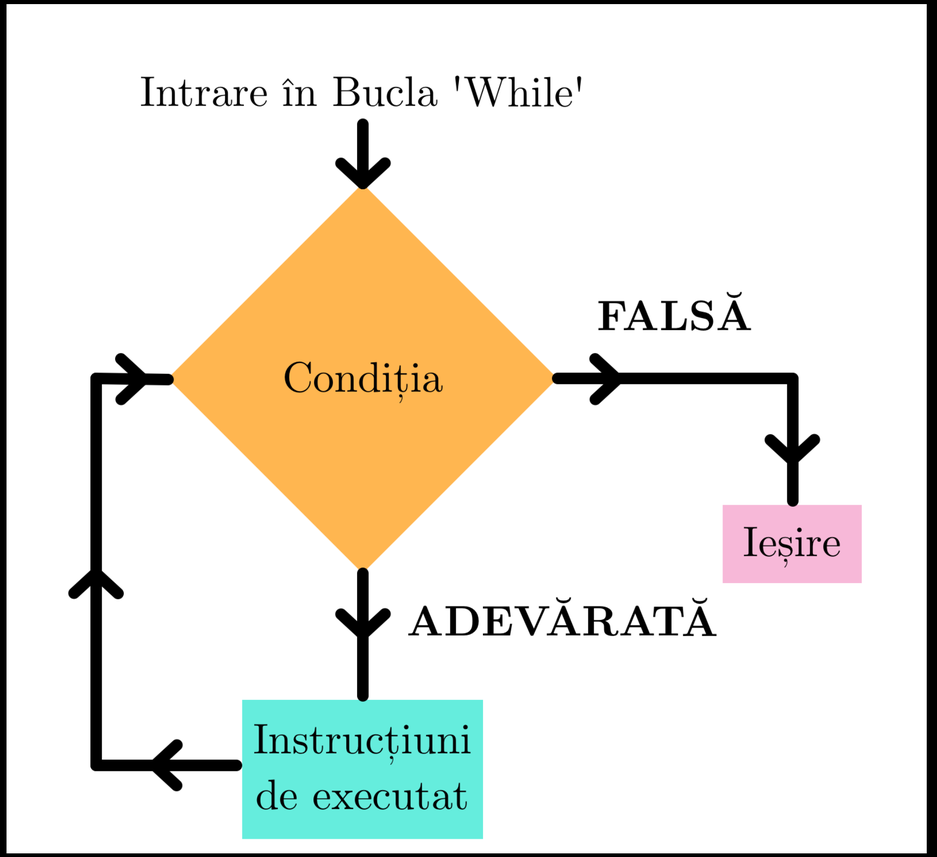
- Învațăm/Recapitulăm cateva elemente de programare in Python
- Învațăm despre Senzorul de Culoare EV3
- Învațăm despre culori și compoziția fizica a luminii.
- Învațăm cum să folosești senzorul de culoare ca să oprim robotul la o linie neagră

Python - Bucla While

- Dacă vrem să așteptăm ca un eveniment să se întâmple, folosim o buclă ‘while’.

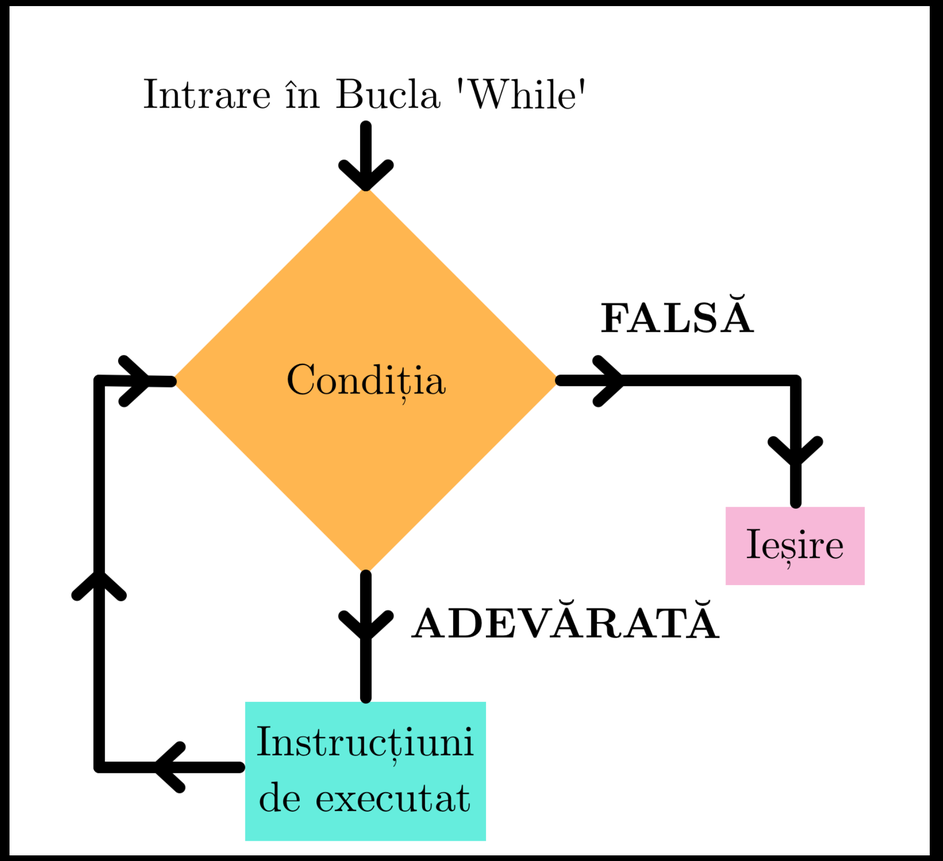
```
>>> while(conditie):
...     actiune()
...
>>> # sfârșit de buclă
```

- Bucla se termină când condiția devine falsă.



Python - Bucla While - Exemplu

```
>>> a = 1
>>> print(a)
>>> while(a < 3):
...         a = a+1 # ‘a’ crește cu unu
...         print(a)
...
>>> print("Done; a = " + str(a))
```



- Care va fi rezultatul ?

Python - Bucla While - Exemplu

- Program:

```
>>> a = 1
>>> print(a)
>>> while(a < 3):
...     a = a+1
...     # ‘a’ a fost incrementat
...     print(a)
...
>>> print("Done; a = " + str(a))
```

- Output:

```
>>> 1
>>> 2
>>> 3
>>> Done; a = 3
```

Python - Functii Utilizator

```
# Intai definim functia
# Important: Faptul ca definim functia nu inseamna ca o si executam
# Asa ca variabila r nu e inca definita. O sa o definim inainte de a apela functia

def Aria_Cercului(r):
    pi = 3.14
    resultat = pi * r * r
    return resultat

#definim toti parametrii de care are functia nevoie
Raza = 5

#apoi apelam functia
Aria = Aria_Cercului(Raza)
print(Aria)
# 78.5
```

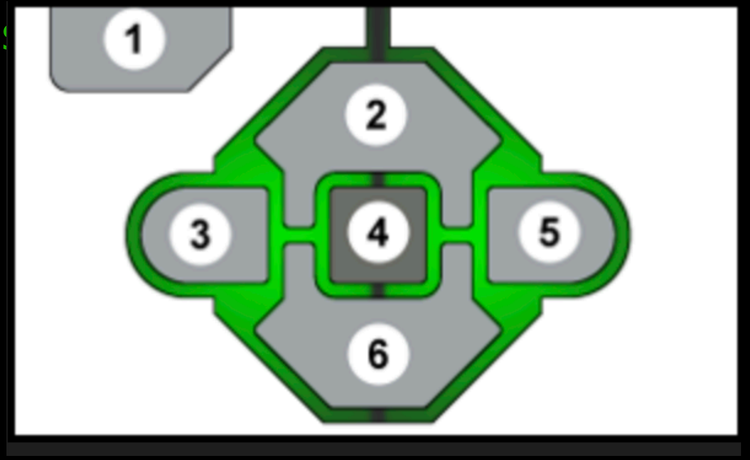
Python - Funcții Predefinite

```
# Butoanele EV3

class ev3dev2.button.Button
```

- buttons_pressed () intoarce o listă cu numele butoanelor apăsate în acel moment.
- Cuvinte reservate: 'UP', 'DOWN', 'LEFT', 'RIGHT', 'ENTER', 'BACKSPACE'

```
# # Putem apela direct functiile/metodele clasei.Cum o folo
```

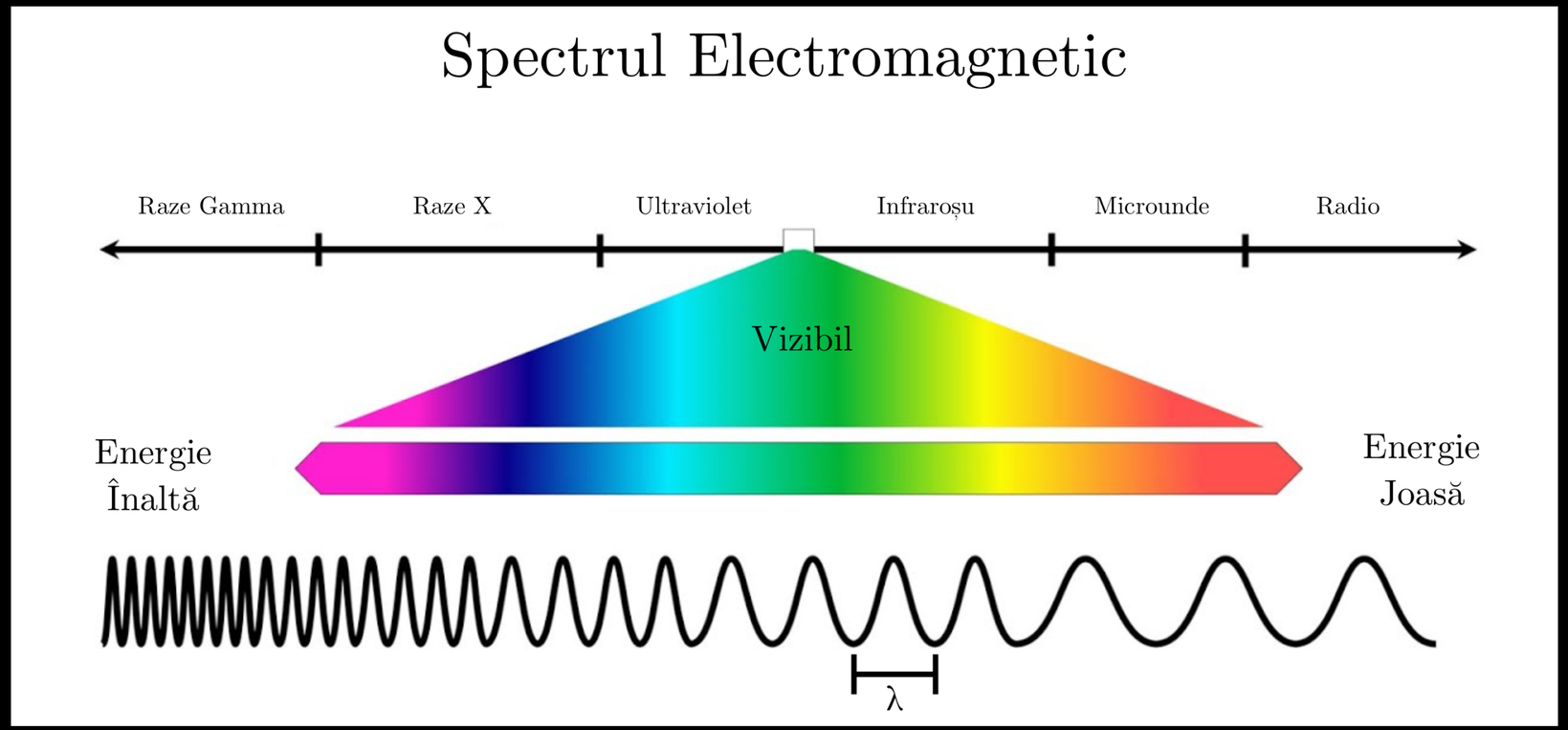



```
#daca butonul din centru este printre butoanele apasate/pressed
    if Button.CENTER in ev3.buttons.pressed():
        ev3.screen.clear()
        ev3.screen.print("Centrul e apasat")
        ev3.speaker.beep()
```

Senzorul de Culoare

Lumina

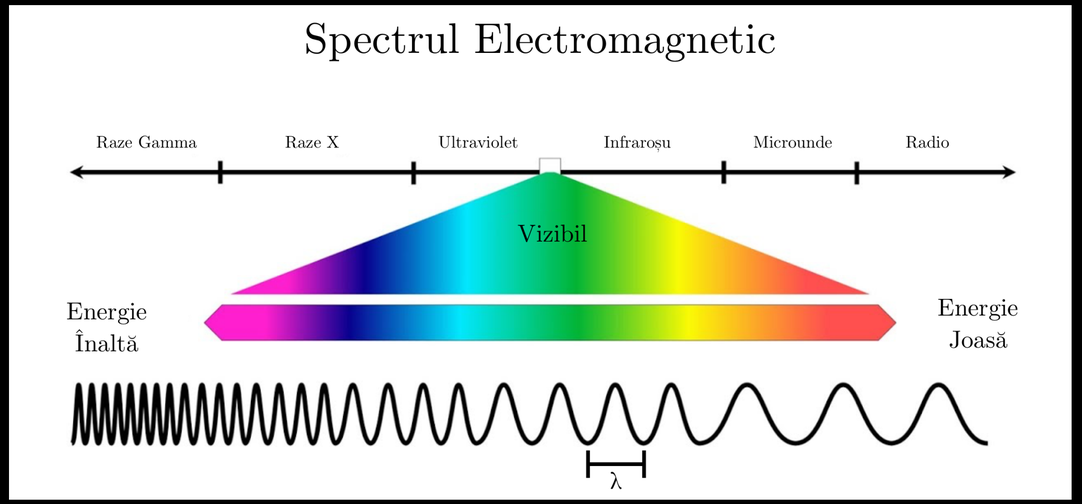
- Lumina este compusă din unde electromagnetice fiecare cu lungimea ei de undă.

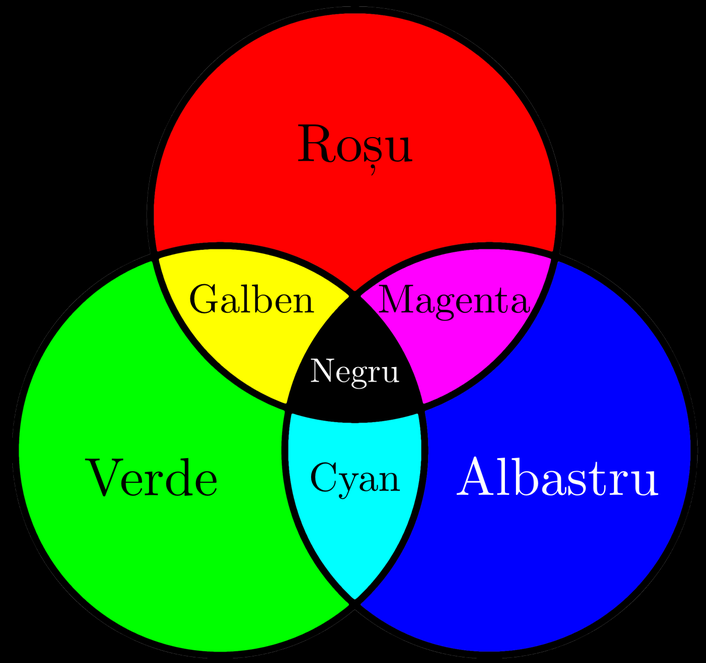


Lumina reflectată

- Lumina poate fi descompusă în culorile elementare: roșu, verde, și albastru.

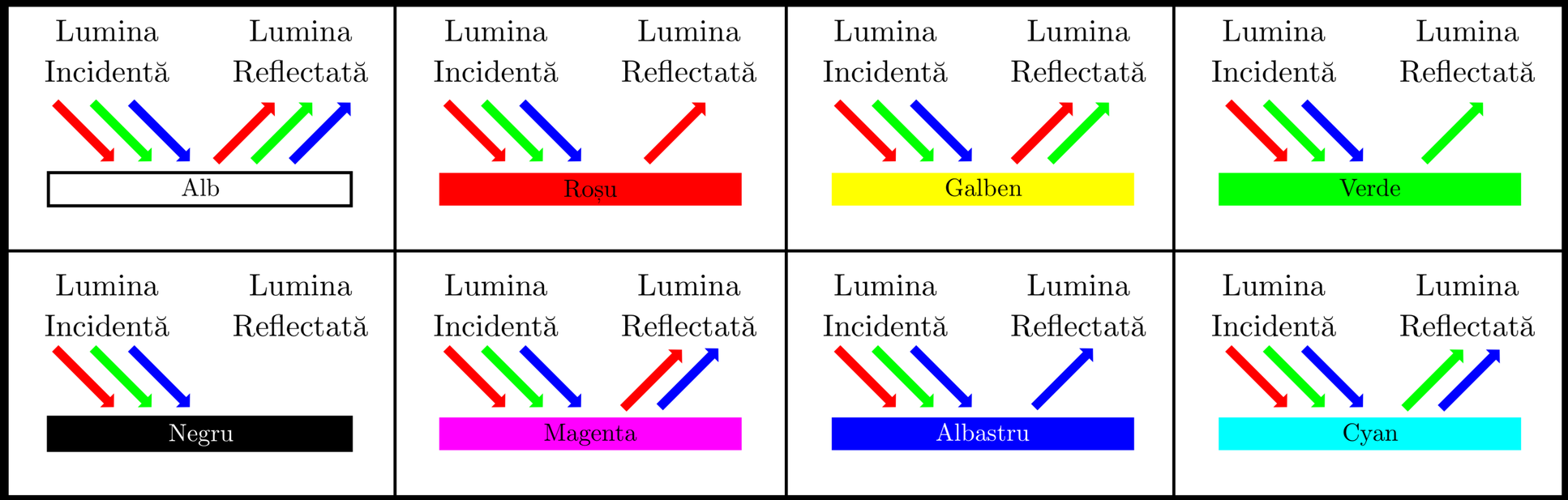
- Ceea ce vedem este lumina reflectată de (sau emisă de) obiecte.
- Un creion este roșu dacă reflectă numai lumină roșie.
- Frunzele sunt verzi pentru că verde este singura culoare pe care ele o reflectă.

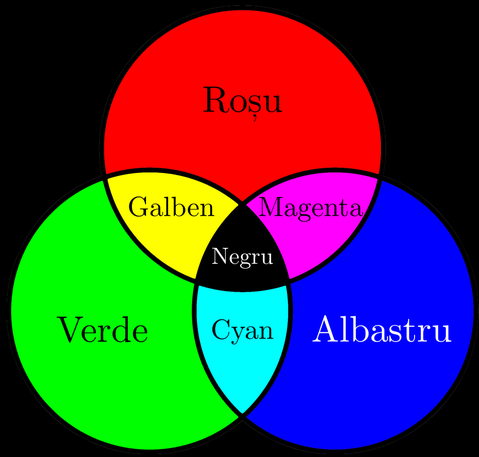




Cum Funcționează Senzorul de Culoare

- Pentru a măsura culoarea, senzorul măsoară cantitățile reflectate din roșu, verde, și respectiv albastru.
- Fiecare culoare are o combinație unică de roșu, verde, și albastru.

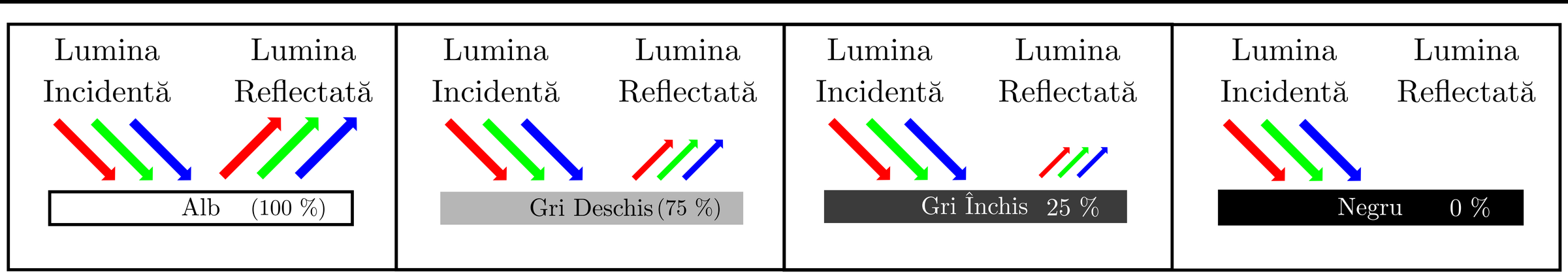




Cum Funcționează Senzorul de Culoare

Intensitatea Luminii Reflectate

- Senzorul de asemenea măsura cât de luminoasă este lumina reflectată.
- Acest proces măsoară cât de închisă (concentrația alb/negru) este culoarea din fața senzorului.



Senzor de Culoare - Folosire

Clasa ColorSensor(*port*)

Parametru : **port** (*Port*) – port-ul/intrarea la care se conectează senzorul

Inițializează un obiect de tip senzor de culoare:

```
colorS = ColorSensor(Port.S3)
```

Senzor de Culoare - Metoda predefinită

Metoda a clasei ColorSensor:

`.reflection()` măsoara intensitatea luminii reflectate de o suprafață

```
colorS = ColorSensor(Port.S3)
ReflexieBruta = colorS.reflection()
print(ReflexieBruta)
```

Calibrarea senzorului de culoare

Idee de structura generala

white_max = 74
black_min = 7
IntervalNegru_Alb= white_max - black_min # Range=Interval
#sensorT nu e inca definit. Va trebuie sa-l definim inainte de a apela functia
def Functia_de_Calibrare(sensorT):

- #Functia va reajusta valorile white_max, black_min cu ceea ce va citi senzorul cand este plasat pe alb sau negru
- #trebuie sa initieze un dialog cu copiii

while True:
....# decidam daca vrem sa calibram sau pornim programul cu valorile date

while True
....# cerem utilizatorului sa puna senzorul pe alb

while True
....# cerem utilizatorului sa puna senzorul pe negru

Calibrarea senzorului de culoare

```
def Functia_de_Calibrare(sensorT):
    ev3.screen.print("LEFT: Calibram")
    ev3.screen.print("CENTER: Pornim”)

    while True:

       #daca butonul din centru este printre butoanele apasate/pressed
        if Button.CENTER in ev3.buttons.pressed():
                    return

        #daca butonul din stanga este printre butoanele apasate/pressed
        if Button.LEFT in ev3.buttons.pressed():
            ev3.screen.clear()
            ev3.screen.print("Ok, calibram")
            ev3.speaker.beep()
            break
        time.sleep(0.05)
```

Calibrarea senzorului de culoare

```
ev3.screen.print(“Pune pe alb ")
ev3.screen.print(“Apasa pe centru”)

while True:
    if Button.CENTER in ev3.buttons.pressed():
            white_max = colorS.reflection()
            ev3.speaker.beep()
            break
    time.sleep(0.05)
```

Calibrarea senzorului de culoare

```
ev3.screen.clear()
ev3.screen.print(“Pune pe negru si apasa”)
ev3.screen.print("press right button”)

while True:
    if Button.RIGHT in ev3.buttons.pressed():
        black_min = colorS.reflection()
        ev3.speaker.beep()
        break
    time.sleep(0.05)
ev3.screen.clear()
IntervalNegru_Alb= white_max - black_min # calculam noul interval
```

Calibrarea senzorului de culoare

```
def readCalibratedColor():
    ReflexieBruta = colorS.reflection()
    CuloareCalibrata = int(100 * float(ReflexieBruta - black_min)/
float(IntervalNegru_Alb))
    return CuloareCalibrata
```

Calibrarea senzorului de culoare

```
# Main program

# Initializarea senzorului de culoare pe portul 3
colorS = ColorSensor(Port.S3)
Functia_de_Calibrare(colorS)
```

Testam calibrarea si apoi o folosim

Device Browser prietenul tău !!

Device Browser arată:

- ce motoare sau senzori sunt conectați, și ce informații furnizează ei
- la care porturi sunt conectate aceste motoare și acești senzori

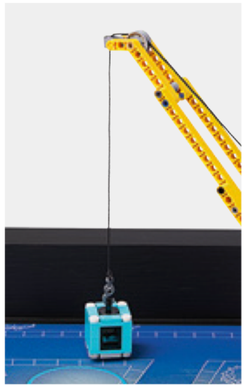
Cum găsesc toate acestea??

- Pe modulul EV3, se găsește „Device Browser” > „Sensors”
  - Pe ecran va apărea o listă de senzori conectați la robot.
  - Apăsând pe fiecare, putem vedea valorile transmise in Watch Values

Exemplu: Misiunea Macaralei - M02

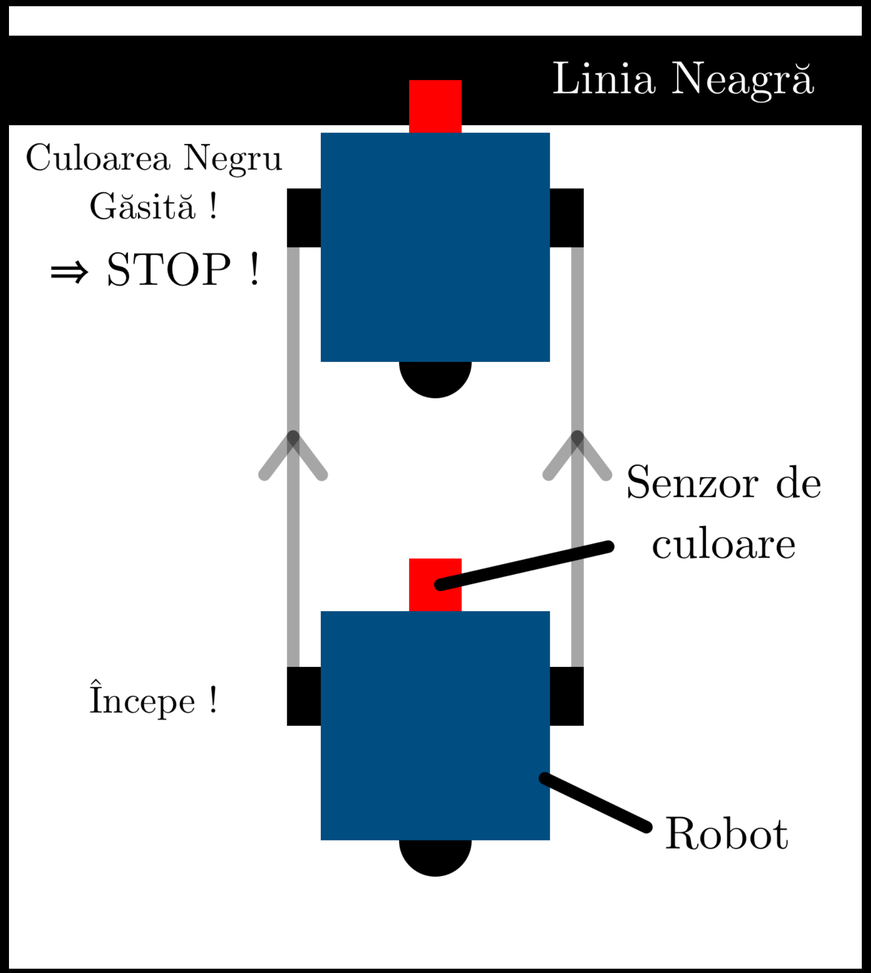
- Scop:
- Ajungem la macara

Misiunea 2: Macara
Dacă blocul albastru atârnat este:
➔ coborât indiferent de distanța parcursă verticală: 20
➔ Pus peste celelalt bloc albastru: 15
și blocul de dedesupt este interiorul cercului albastru, fără niciun ajutor al robotului: 15



Oprirea la Linia Neagră - Algoritm

- Pornește motoarele
  - Robotul avansează spre linia neagră
- **Câtă vreme (bucla while)** robotul avansează, folosește senzorul de culoare pentru a detecta prezența culorii negre.
- **Când** găsești culoarea neagră, oprește-te !



Recapitulare Inițializare șasiu

```
# Initializare motoare
left_motor = Motor(Port.B)
right_motor = Motor(Port.C)

# Initializarea șasiu (DriveBase) pentru a mișca robotului

robot = DriveBase(left_motor,right_motor,wheel_diameter=62.4,axle_track=102)
```

Utilizarea Senzorului de Culoare

```
colorS = ColorSensor(Port.S3)
#Nu uitați comanda import time înainte de bucla.

robot.straight(400)

# Cata vreme nu am gasit culoarea neagra mergem inainte
robot.drive(80,0) # 80 mm/s speed, 0 intoarcere (merge drept/straight)

while(colorS.reflection() > 7):
    time.sleep(0.01)
# Dacă culoarea neagră este găsită, ieșim din buclă. Trebuie să spunem robotului
să se oprească !
robot.drive(0,0) # Oprim robotul spunandu-i 0 mm/s speed, 0 intoarcere
ev3.speaker.beep()
robot.stop()
```

Testarea Programului

- Testați-vă programul !

Probleme

- Simptom: *Robotul s-a oprit prea devreme.*
  - Soluție: Probabil senzorul a captat altă formă neagră pe hârtie, sau liniile negre subțiri din bază. Începe prin a merge drept în față o distanță suficientă ca să depășești acest(e) obstacol(e), și apoi începe să cauți.

- Simptom: *Robotul nu s-a oprit la linia.*
  - Soluție: Merge mai încet, reduce timpul de așteptare în bucla, sau schimbă valoarea limită (inițial era 25%).