

# ADVANCED EV3 PROGRAMMING LESSON



## Mișcarea înainte cu Gyro

---

By Sanjay and Arvind Seshan



# Obiectivele lecției

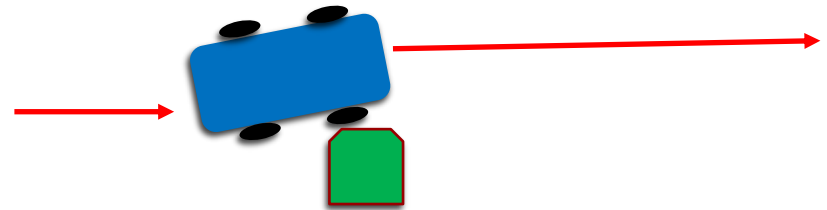
- Învățăm ce reprezintă controlul proporțional și cum îl folosim
- Învățăm cum să aplicăm controlul proporțional pentru a face robotul să meargă drept înainte
- Învățăm să aplicăm controlul proporțional senzorului Gyro la mișcarea la un anumit unghi.
- **Cerințe:** block-uri matematice, fire de date, control proporțional, lecțiile Gyro Sensor.

# Sfaturi pentru succes

- Trebuie să parcurgi lecția Proportional Control Lesson și lecția Proportional Line Follower înainte de a urmări această lecție
- Trebuie de asemenea să parcurgi și cele 3 lecții despre senzorul Gyro (Introduction to Gyro Sensor, Gyro Turns and Gyro Revisited)

# Ce este mișcarea în linie dreaptă înainte cu senzorul Gyro?

- Imaginează-ți că vrei ca roborul să meargă în linie dreaptă 200 cm.
- Pe măsură ce robotul parcurge distanța, acesta este lovit de ceva.
- Un program de mers în linie dreaptă pe senzorul Gyro ajută robotul să se corecteze înapoi la linia dreaptă, dar acesta rămâne decalat cu distanța cu care a fost împins.



# Cum merge asta?

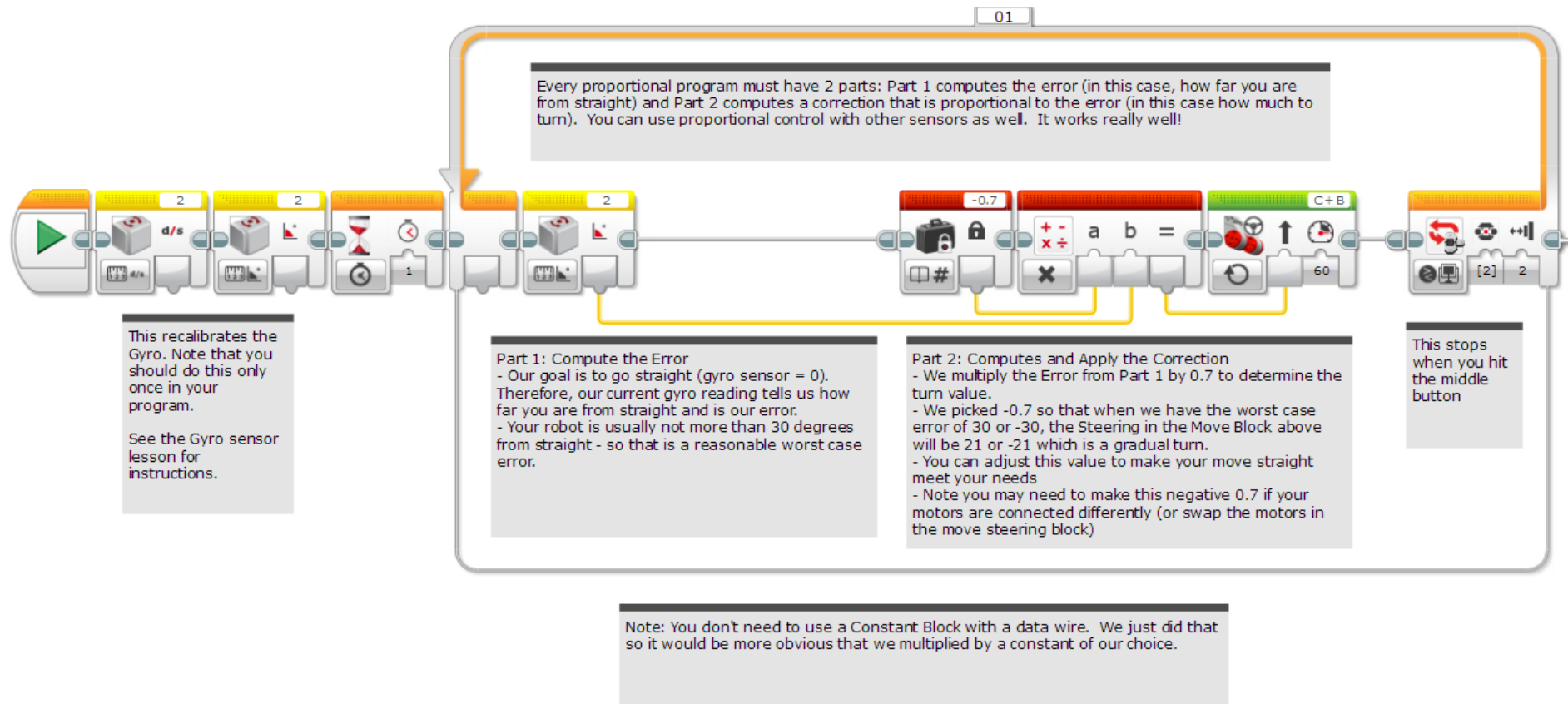
- Un cod Proportional line follower și un cod de mers în linie dreaptă cu gyro au aceleași proprietăți
- Pentru a scrie un program de mers în linie dreaptă cu gyro, trebuie mai întâi să te gândești care este eroarea și cum poți s-o corectezi.

Aplicație	Obiective	Eroare	Corecție
<b>Gyro Straight</b>	Menține robotul la un anumit unghi constant	Cât de departe este față de unghiul target	Întoarce mai strâns în funcție de cât de departe de unghi este.
<b>Line Follower</b>	Stă pe marginea liniei	Cât de departe sunt citirile senzorului de lumină față de valorile considerate target.(cele de la marginea liniei) (citirea curentă– valoare target)	Întoarce mai strâns în funcție de distanța față de marginea liniei.

# Pseudocode-ul

- Recalibrează senzorul gyro (dacă nu ai facut-o altundeva în cod) sau resetează senzorul tău gyro (block-ul galben gyro setat pe modul reset) în așa fel încât valoarea să înceapă de la “0” și să nu existe drift
- Într-un loop, calculează eroarea și aplică corecția
  - Partea 1: Calculează eroarea (cât de departe față de unghiul target)
    - Pentru a merge înainte → Target gyro angle=0
    - Distanța față de unghiul target este de fapt citirea curentă a senzorului gyro.
  - Partea 2: calculează corecția care este proporțională cu eroarea
    - Multiplicați eroarea din partea 1 cu o constantă (aceasta trebuie să experimentezi și să descoperi pentru robotul tău)
  - Legați valoarea de la Partea 2 într-un block Move Steering
- ieșirea din LOOP – condiție

# Soluția: Mișcarea în linie dreaptă cu Gyro



# Ghid de discuții

1. Comparați codul „proportional line follower” cu codul „proportional move straight”. Ce similarități și diferențe observați?

**Răspuns.** Codul este aproape la fel. Singura diferență este cum se calculează eroarea. Eroarea este calculată utilizând senzorul Gyro. Corecția este identică.

2. Cum procedăm dacă dorim ca robotul să parcurgă distanța la un anumit unghi (nu neapărat drept)? Cum va arăta codul diferit?

**Răspuns.** În Partea 1 codul soluție, nu există block de diferență din moment ce scădem “0” pentru că ținta noastră se mișcă drept. Trebuie să scazi unghiul curent din unghiul pe care l-ai stabilit ca target dacă vrei ca robotul să se miște la un anumit unghi.



Target angle = -5 degrees



# Credits

- Această lecție de Mindstorms a fost realizată de Sanjay Seshan și Arvind Seshan.
- Mai multe lecții sunt disponibile pe [ev3lessons.com](http://ev3lessons.com)
- 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/).