ADVANCED EV3 PROGRAMMING LESSON

Squaring or Aligning on a Line

By Droids Robotics

Lesson Objectives

1) Learn how to get your robot to Square (straighten out) when it comes to a line

2) Learn how Squaring (also known as Aligning on a Line) can help the robot navigate

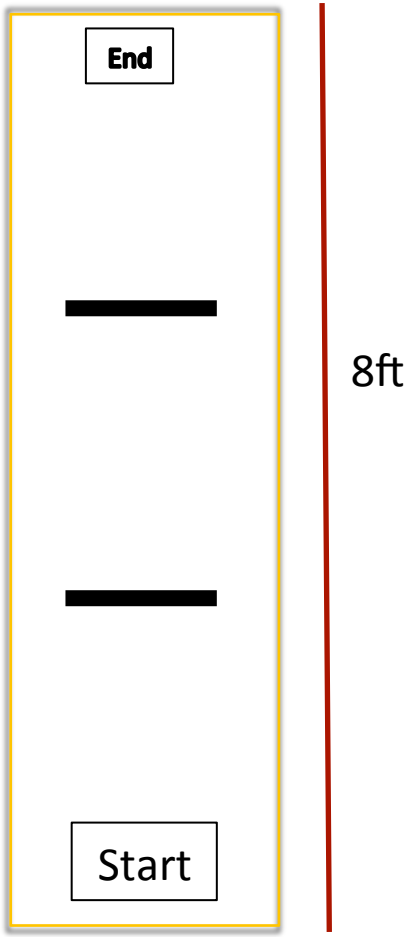
3) Learn how to improve initial code for aligning by repeating a technique

4) Practice creating a useful My Block

Prerequisites: My Blocks with Inputs & Outputs, Data Wires, Parallel Beams, Parallel Beams Synchronization

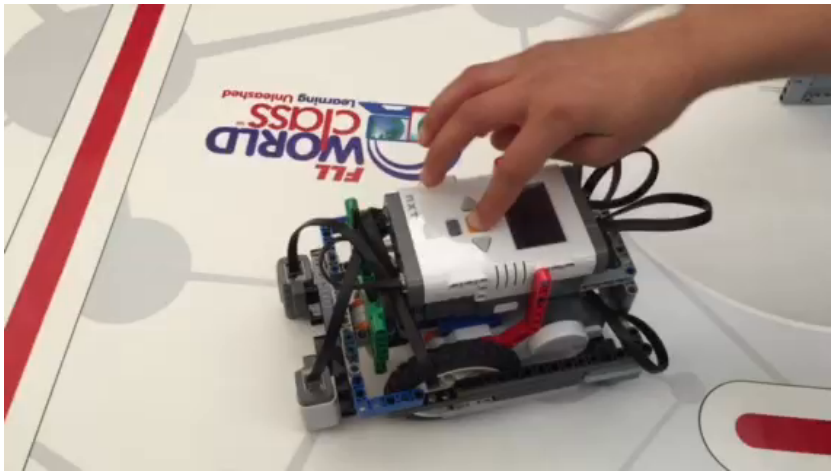
Why Align on a Line?

- Aligning on a line helps the robot navigate
  - Robots get angled as they travel farther or turn (the error accumulates)
  - Aligning on a line can straighten out a robot.
  - Aligning can tell a robot where it is when it has to travel far
- Example Goal: Your robot must deliver an object only inside a small END area. The distance between start and end is 8 feet
  - Do you think your robot can travel 8 ft and continue to be straight?



Challenge

Challenge: Create a My Block to make the robot straighten out (align/square off) on a red line (see video)

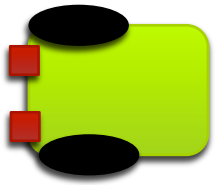


Pseudocode:

1. Start both motors
2. Stop one motor when the sensor on the corresponding side sees the red line (use parallel beams)
3. Stop moving the second motor when the sensor on that side sees the red line (use parallel beams)
4. Add some inputs (constants) to the code for power and color so you can change what color you want to align to
5. Use data wires to connect the color constant to the wait for block and the power to the Move Block
6. Create an Align My Block

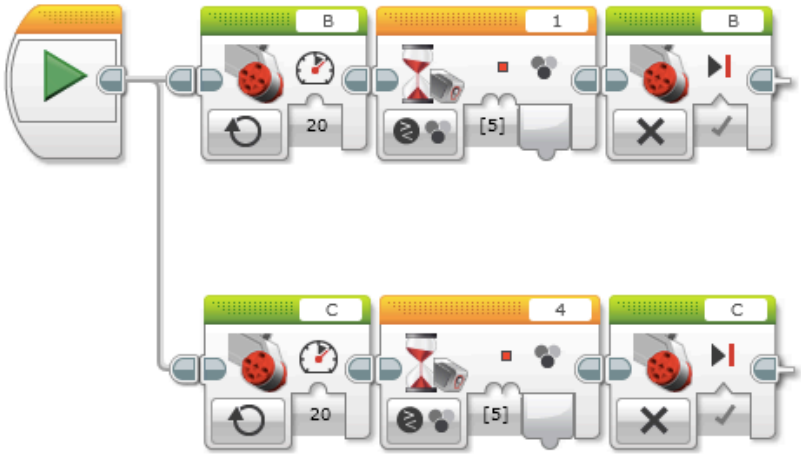
Notes About Our Solution:

- Our solution uses 2 Color Sensors (connected in Ports 1 and 4).
- Our solution assumes that the color sensor on port 1 is next to the wheel on motor port B and color sensor on port 4 is next to the wheel on motor port C.
- You should adjust the ports as needed
- Your color sensors should NOT be placed right next to each other (see red box below in robot image)



Solution Step 1: Moving Until Line

We use a parallel beam here so that we can do 2 simultaneous actions.
In the top beam, Motor B on our robot keeps moving until the Color Sensor on Port 1 sees Red. Then it stops.
In the bottom beam, Motor C on our robot keeps moving until the Color Sensor on Port 4 sees Red. Then it stops.



Step 1 Goal: Create a SIMPLE way to square up on the line

Note 1: You will need 2 EV3 Color Sensors (connected in Ports 1 and 4 in this program)

Note 2: This program squares onto a Red Line (you can change this to whatever color is available - colors found on the FLL mat are there)

Note 3: This program uses the color sensor in COLOR MODE. You can write a program that uses LIGHT MODE, but you will have to calibrate your sensors. We will show you that in another lesson.

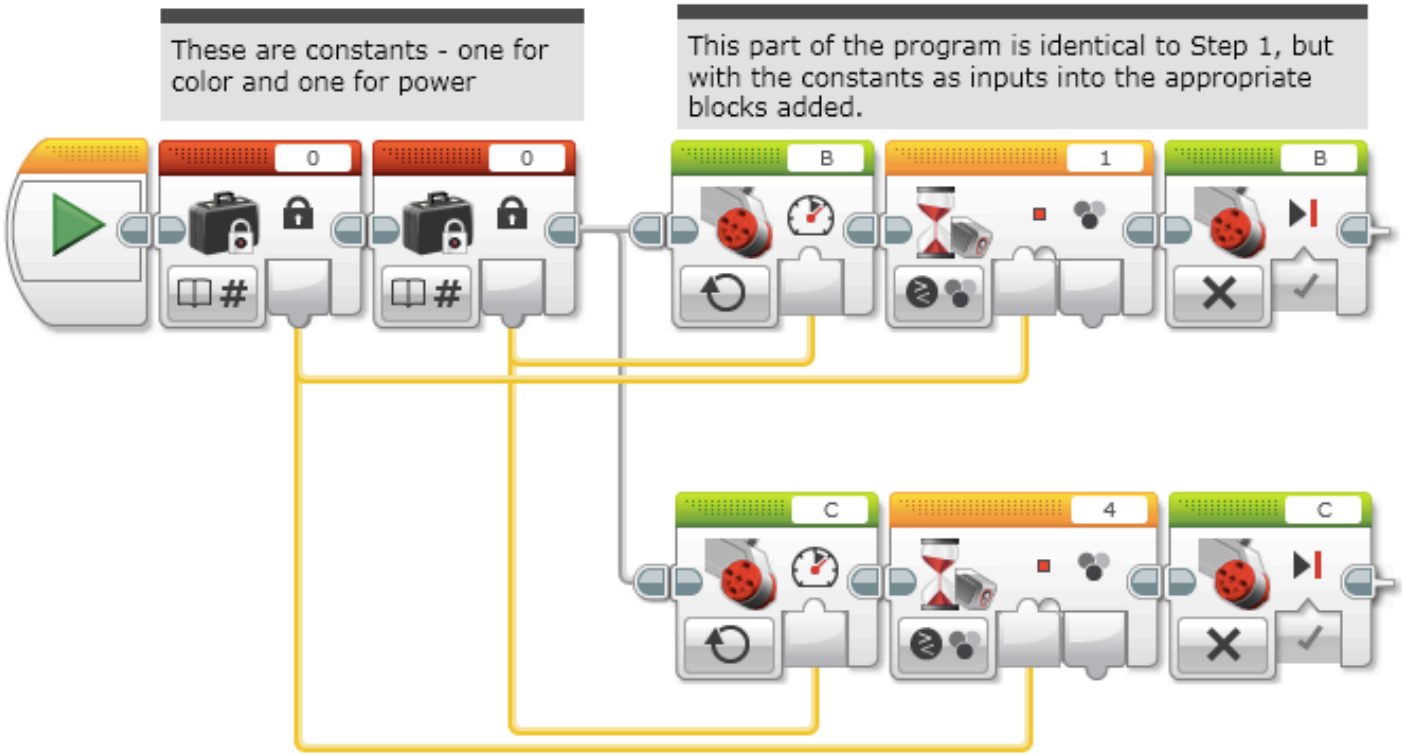
Note 4: Your robot design will make a difference - whether you have your color sensors in the rear or front of your robot, and how far apart the sensors are (the further apart, the better).

Note 5: You should adjust the ports as needed - e.g. this assumes that a color sensor on port 1 is next to the wheel on motor port B and color sensor on port 4 is next to the wheel on motor port C.

Note 6: While the robot will be on the red line, this will not create a perfect alignment. See instruction in Step 3 for a simple fix.

Solution Step 2: Adding Inputs

Goal: This is the program from Step 1 with added constant inputs, so you can easily change your color and power.



Repeating a Technique

- What do you notice about the solution we just presented?
  - The robot isn't quite straight (aligned) at the end of it.
  - Both color sensors are on the line, but the robot stops at an angle.
- Challenge Continued: Think about how you can improve this code so that the robot ends straighter
  - Hint: Can you repeat the last process by looking for **white**?
  - This assumes that the red line we were straightening out on has white on both sides.

Step 3: My Block With Dual Stage Fix

Goal: The goal of this program is to make a reusable My Block out of our code. This is Step 2 converted into a My Block.

You can make the My Block be selecting everything in Step 2 execpt for the constants then click Tools-->My Block Builder.

Please read our My Blocks Lesson if you need help with making My Blocks.

Color Key:
0 = No Color
2 = Black
3 = Green
4 = Yellow
5 = Red
6 = White
7 = Brown

Align Block that looks for RED

Align Block backs up and looks for WHITE

WHY 2 ALIGNS BACK TO BACK?

Problem: After the first align, you might find that your robot is still a little angled but on red. This happens because even though one wheel had stopped when it found the red, the location of both sensors keep moving as a result of the movement of the second wheel.

Solution: Notice that a second align My Block was added. This will make your robot move back until the sensors see white so that it will be straighter.

Program by: Droids Robotics
www.droidsrobotics.org

Tips for Success and Next Steps

- You will get better results
  - ….if your color sensors are closer to the ground
  - ….if you shield your color sensors
  - ….if you don't come at the line at steep angles
  - ….if you keep your color sensors spread apart
- Next Steps:
  - There are alternative ways to align on lines (even with one sensor).
  - An alternative approach can be found in the Contributed Lessons tab

Discussion Guide

1. What does Aligning on a Line/Squaring mean?
   Ans. It means when your robot comes at an angle, it can use a line to straighten out

2. Why does Squaring help when you travel long distances?
   Ans. Robots do not move accurately/straight. Squaring can help you know where you are and straighten out.

3. Why might it be useful to repeat the alignment process multiple times?
   Ans. Your robot will get straighter each time you repeat

Credits

- This tutorial was created by Sanjay Seshan and Arvind Seshan from Droids Robotics.
  - Author's Email: team@droidsrobotics.org
- More lessons at www.ev3lessons.com