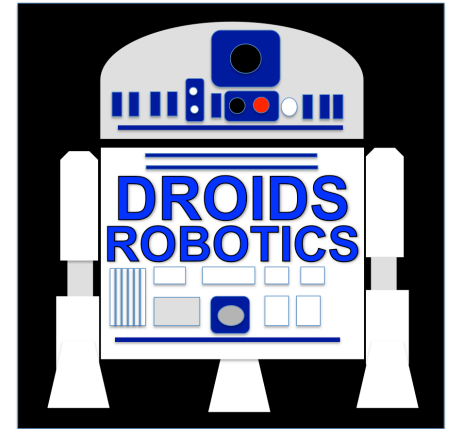


ADVANCED EV3 PROGRAMMING LESSON



ALIGNING ON A LINE/ SQUARING OFF

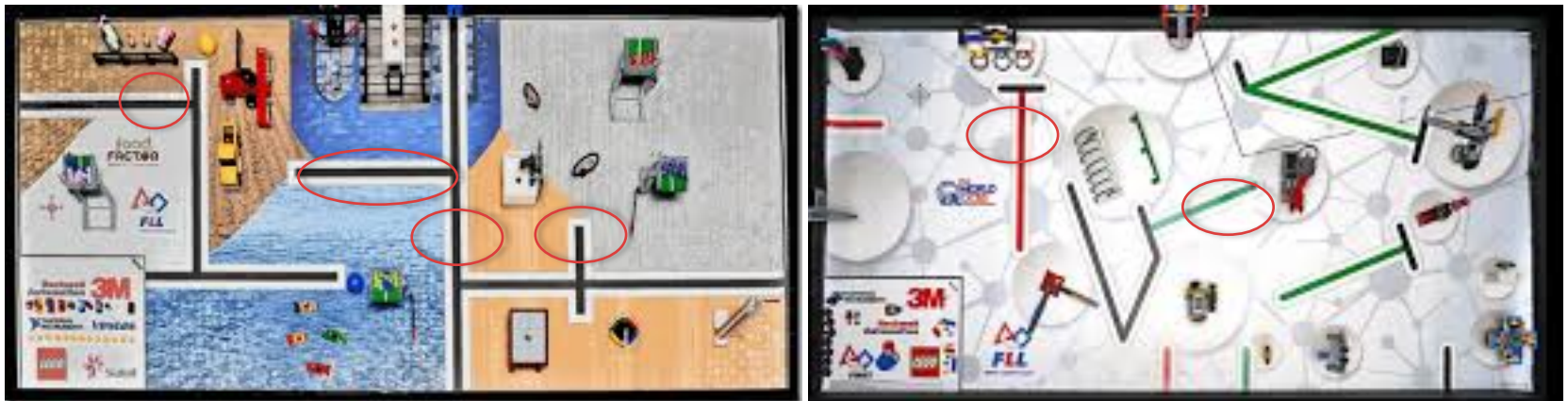
BY DROIDS ROBOTICS

ALIGNING ON A LINE

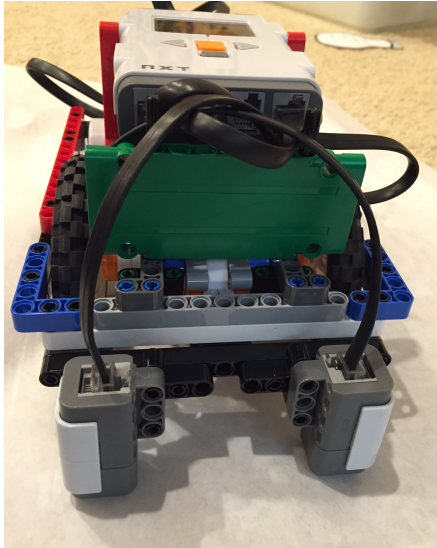
Another team asked us if we could do a lesson on aligning on a line, aka Squaring on a Line. This lesson is dedicated to your team.

FLL mats have lines that you can follow but also useful lines that you can use to square on. We highlight some examples below.

Start at Step 1 in the corresponding EV3 Program.



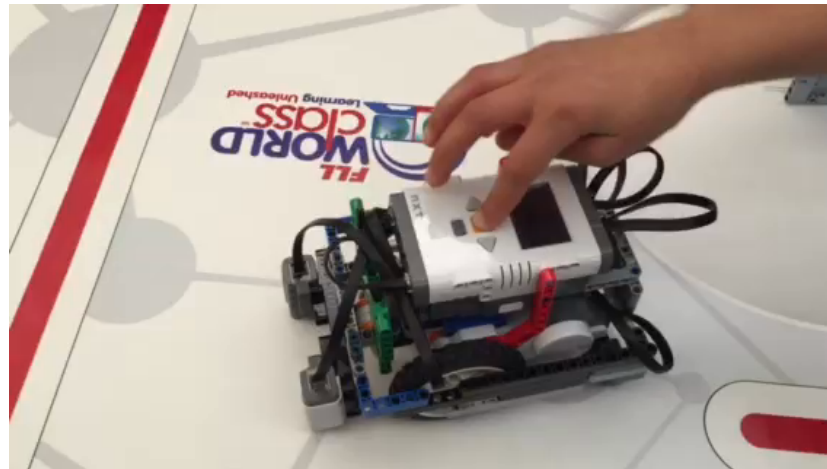
GOAL: ROBOT THAT STRAIGHTENS ON A LINE



You will need 2 Color Sensors (connected in Ports 1 and 4 in this program). Works on EV3 and NXT.

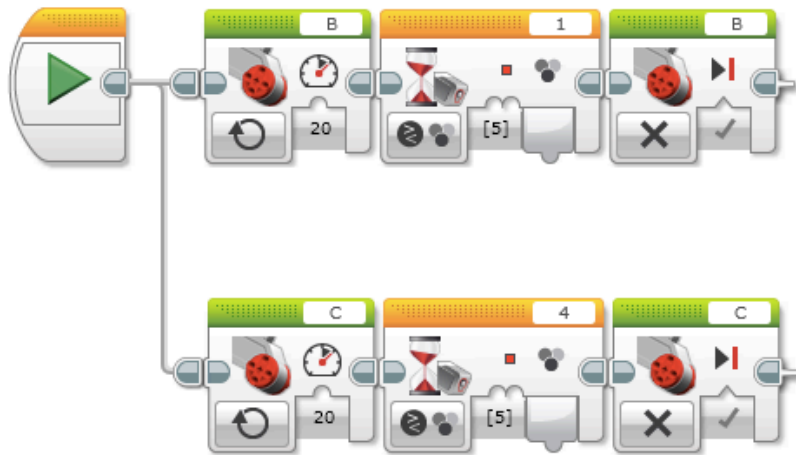
You should adjust the ports as needed - e.g. this assumes that a color sensor on port 1 is next to the wheel on motor port B and color sensor on port 4 is next to the wheel on motor port C.

Click on video



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.

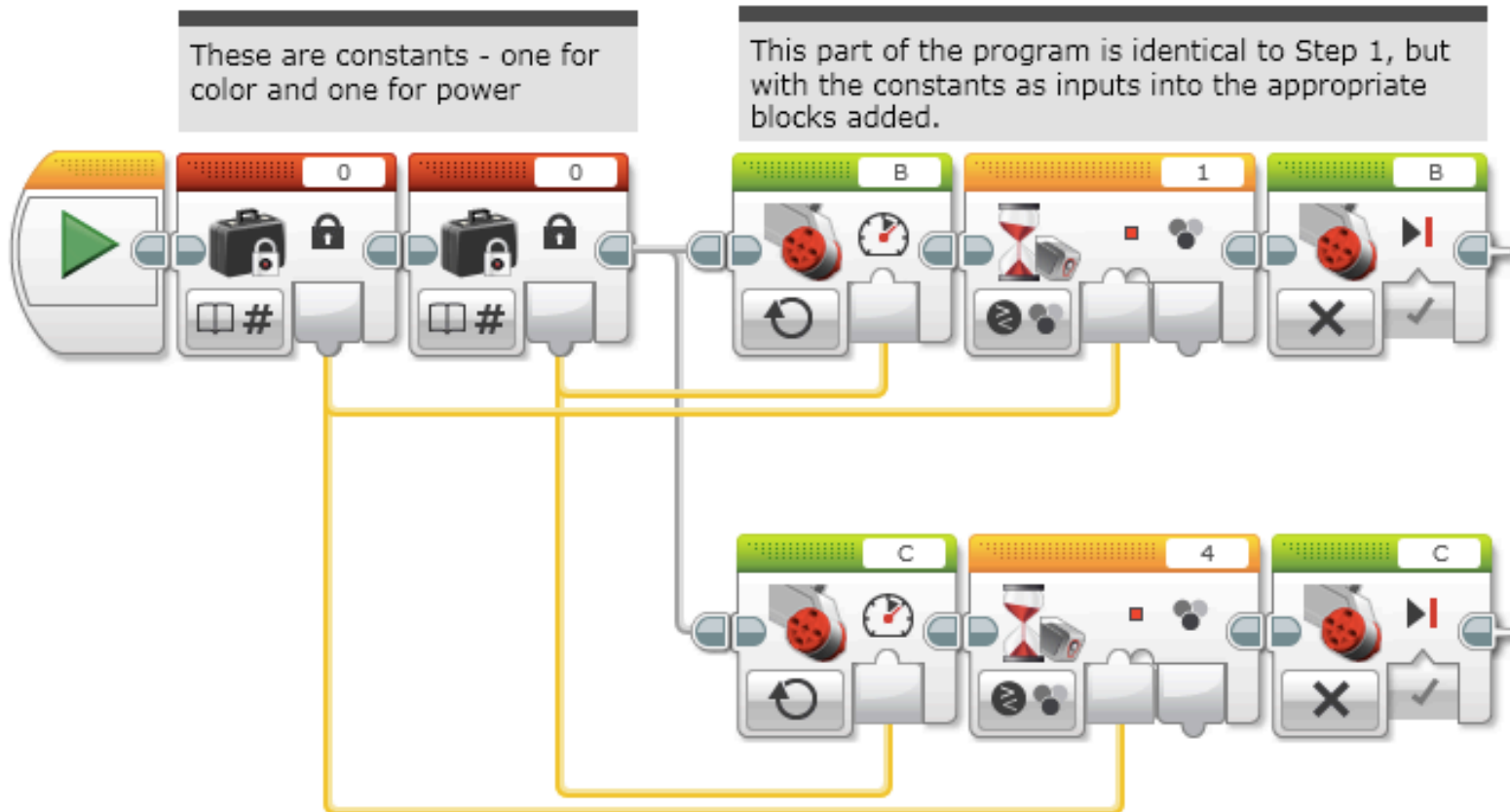
Note 4: Your robot design will make a difference - whether you have your color sensors in the rear or front of your robot, and how far apart the sensors are (the further apart, the better).

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.

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.



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.

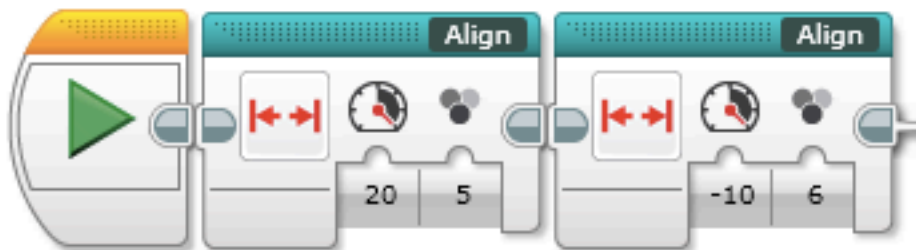
You can make the My Block by selecting everything in Step 2 except for the constants then click Tools-->My Block Builder.

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

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.



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

Program by: Droids Robotics
www.droidsrobotics.org

TIPS

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

CREDITS

- These slides and the corresponding EV3 project files were made by Sanjay Seshan and Arvind Seshan from FLL Team: Not the Droids You Are Looking For.
- They are free to use and distribute. Please just give credit to the team and send a thank you note if you can.
- You can reach the Droids at: team@droidsrobotics.org
- Calculator for converting CM/IN into degrees: www.ev3lessons.com/resources.html
- More lessons: www.ev3lessons.com