

ADVANCED EV3 PROGRAMMING LESSON



EV3 Classroom: Squaring or Aligning on a Line

By Sanjay and Arvind Seshan



EV3 CLASSROOM LESSON
BY EV3LESSONS.COM

Lesson Objectives

- Learn how to get your robot to square up (straighten out) when it comes to a line
- Learn how squaring (also known as aligning on a line) can help the robot navigate
- Learn how to improve initial code for aligning by repeating a technique
- Practice creating a useful My Block

- Prerequisites: My Blocks with Inputs & Outputs, Parallel Beams (Events), Parallel Beams Synchronization

Review: Motor Movements

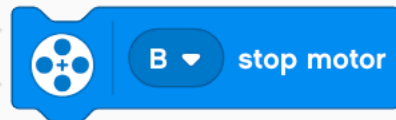
- Move Steering lets you control both motors at the same time
- What if you want to move or stop one motor at a time?
 - Use the Motor Blocks



Run Motor For Duration at Speed Block



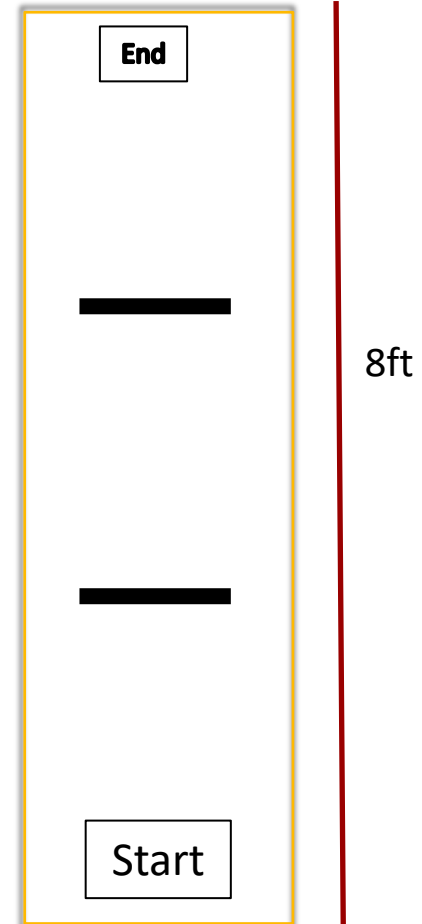
Start Motor at Speed Block



Stop Motor Block

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 feet and continue to be straight?



Three Easy Steps to Align

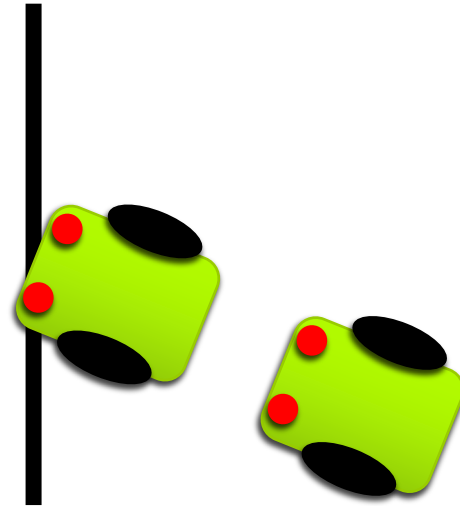
Challenge: Make the robot straighten out (align/square up)

STEP 1: Start both motors

STEP 2: Stop one motor when the sensor on the corresponding side sees the line

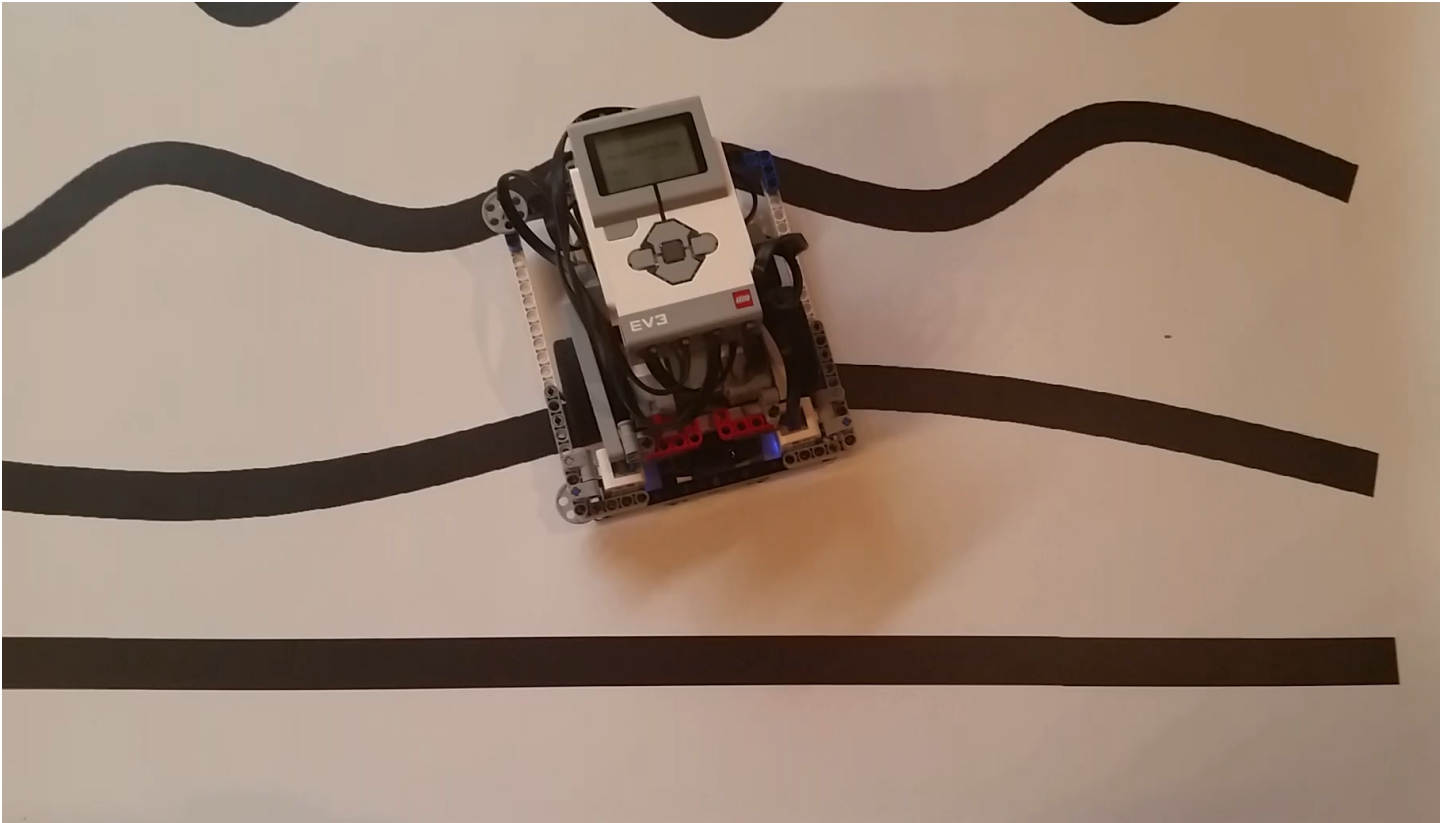
STEP 3: Stop moving the second motor when the sensor on that side sees the line

Hints: Use a Motor Block, Use Parallel Beams (Separate Events)



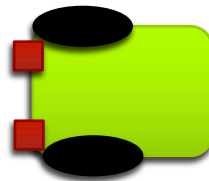
(This slide is animated)

What Aligning Should Look Like



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 boxes below in robot image. These are the color sensors.)



Basic Solution: Moving Until Line

when program starts

broadcast message1

B start motor at 20 % speed

1 wait until color is black

B stop motor

Initiates a second event when
“message1” is broadcasted

Starts motors

Wait for color sensor
to detect black

Stop motor

when I receive message1

C start motor at 20 % speed

4 wait until color is black

C stop motor

Note: Synchronization & Parallel Beams (Events)

- When you have two or more events you do not know when each beam will finish.
- If you wanted to move after the align finishes you might try to add a move block at the end of one of the events.
 - Note: This will not work because the code will play your move block without waiting for the other event to finish.
 - Solution: You need to synchronize your events. To learn more about synchronization and solutions go to the [Advanced EV3Lessons.com](https://www.ev3lessons.com) Lesson on Sync Beams.
- The problem of synchronization can be solved by using Wait Until Blocks and Variables. The second event will set a variable to a specific value at its end and the first event will wait for that value to be set.

Improving Your Align Code

- 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**

Tips for Success

- You will get better results
 -if your color sensors are about 4mm-12mm from the ground
 -if you don't come at the line at steep angles
 -if you keep your color sensors spread apart

Credits

- This tutorial was created by Sanjay Seshan and Arvind Seshan
- More lessons at www.ev3lessons.com



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).