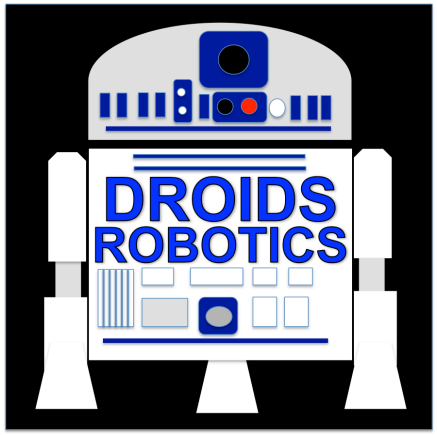
ADVANCED EV3 PROGRAMMING LESSON



LINE FOLLOWING: BASIC TO PROPORTIONAL

BY DROIDS ROBOTICS

UPDATED 10/18/2014

LINE FOLLOWING IS VALUABLE

Many FLL Mats are covered with lines

You can use these lines to navigate to mission models

Having a good line follower program can really help your team

We present 4 line followers in this lesson that would work for beginner through advanced teams.

WHICH PROGRAM WORKS BEST FOR WHICH LINE?

Simple Line Follower

- Most basic line follower
- Wiggles a lot due to sharp turns
- Good for rookie teams → need to know loops and switches

Smooth Line Follower

- Almost the same as simple
- Turns are less sharp
- Has trouble on sharp curves
- Good for rookie teams → need to know loops and switches

3-Stage Follower

- Best for straight lines
- Droids do not recommend this. Just learn the proportional line follower.
- Need to know nested switches

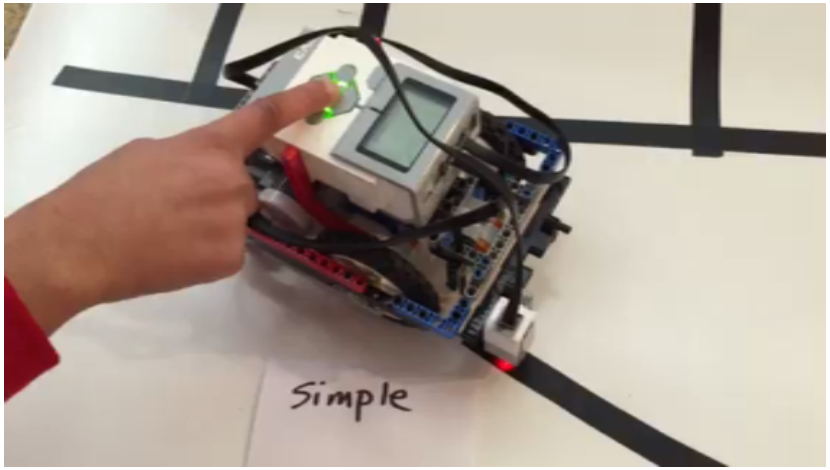
Proportional Follower

- Uses the “P” in PID
- Makes proportional turns
- Works well on both straight and curved lines
- Good for intermediate to advanced teams → need to know math blocks and data wires

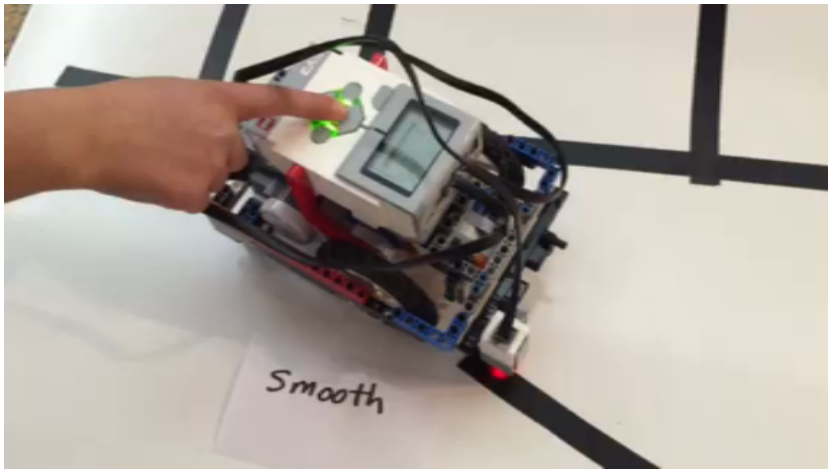
Watch the videos on the next 2 slides to see the programs in action.

CURVED LINE: WATCH VIDEOS

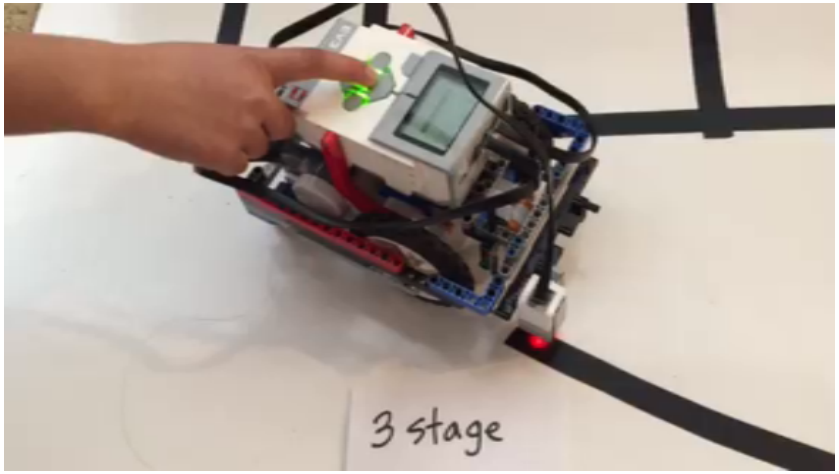
Simple Line Follower



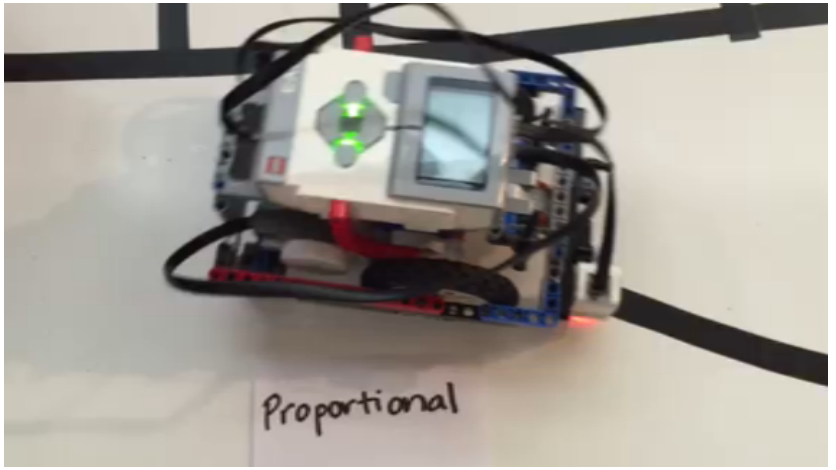
Smooth Line Follower



3-Stage Follower

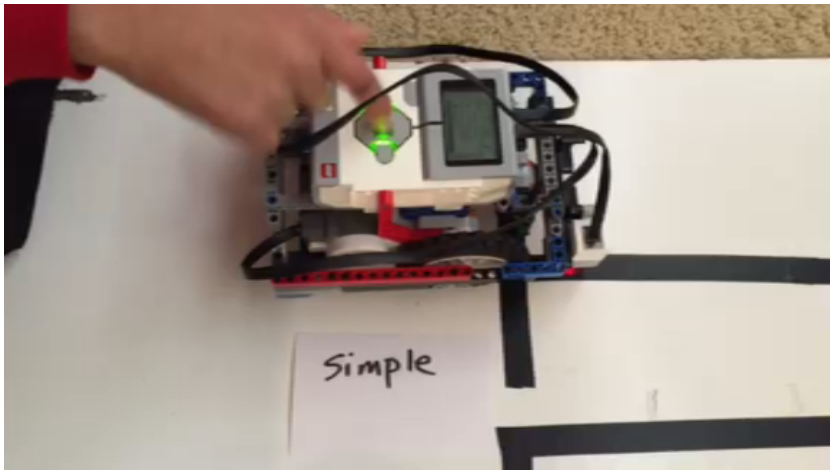


Proportional Follower

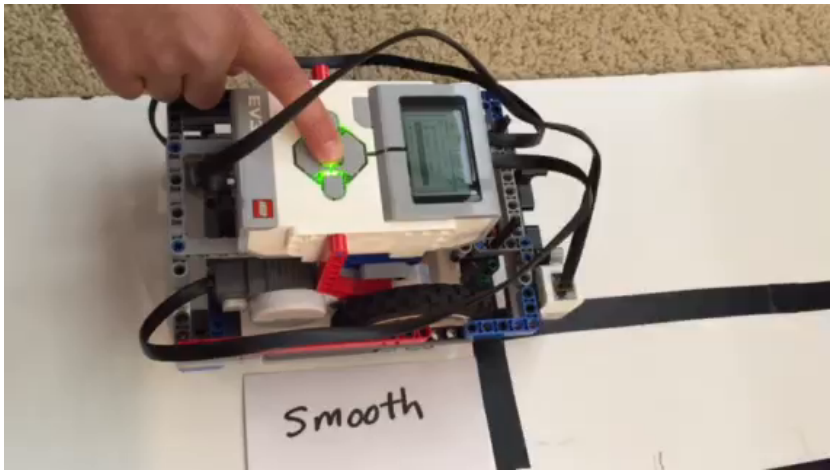


STRAIGHT LINE: WATCH VIDEOS

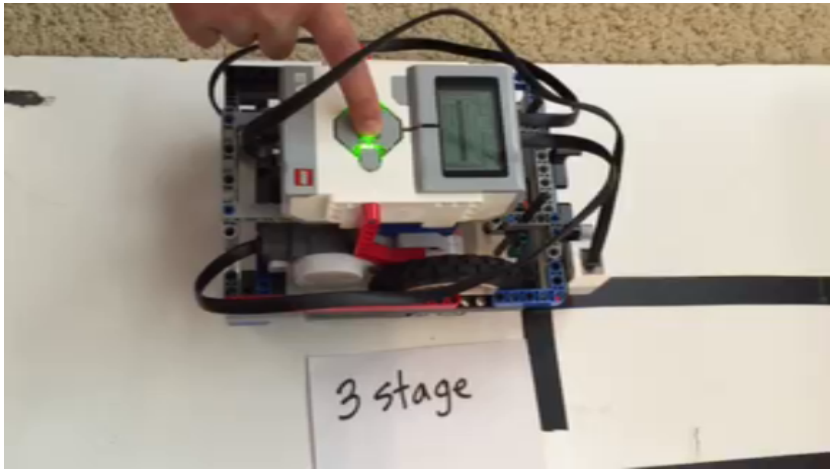
Simple Line Follower



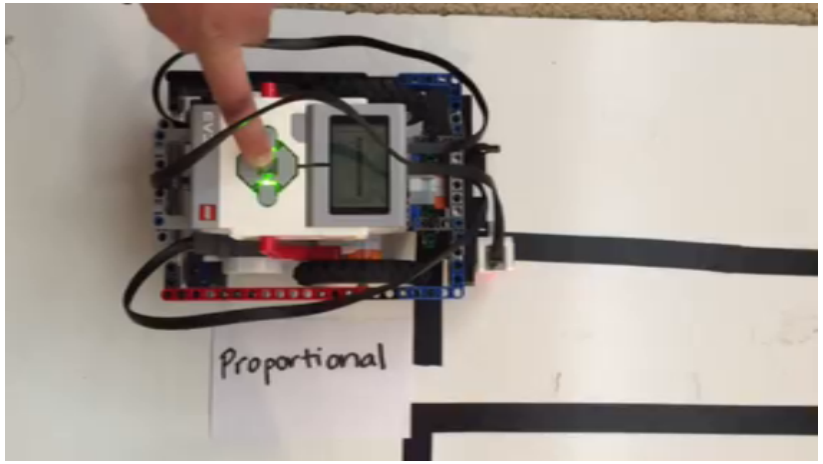
Smooth Line Follower



3-Stage Follower



Proportional Follower



BEFORE YOU RUN THE CODE

CALIBRATE:

The programs use the EV3 Color Sensor in Light Sensor mode

You will have to calibrate your sensors.

Please refer to Calibration Lesson – Advanced Lesson #4

PORTS:

The Color Sensor is connected to Port 3.

Please change this for your robot.

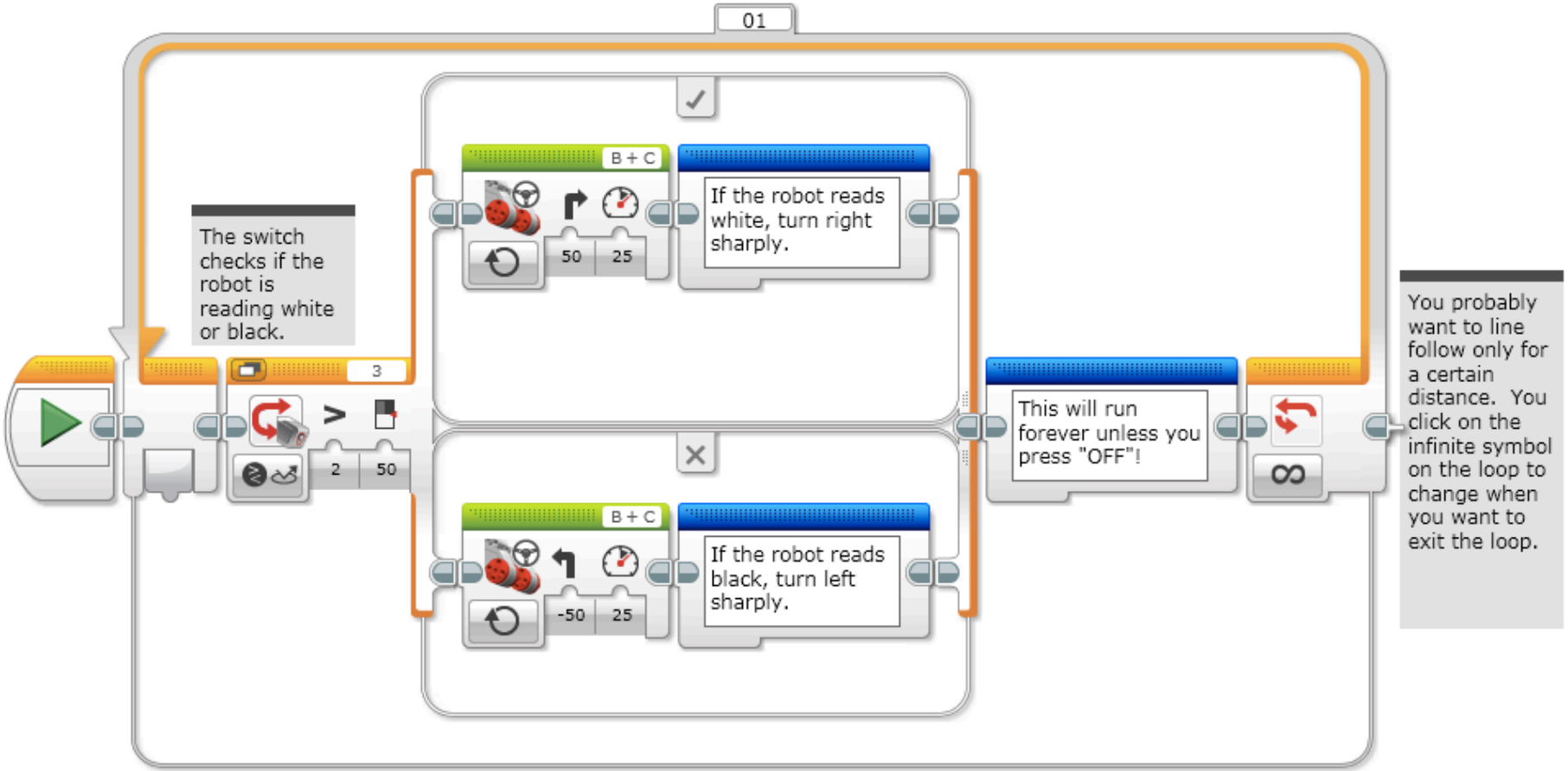
WHICH SIDE OF THE LINE:

Please take note of which side of the line the code is written for

SIMPLE LINE FOLLOWER

Simple Line Follower: The goal of this program is to create a very simple line following programming to follow the left side of a line. This is the most commonly taught program.

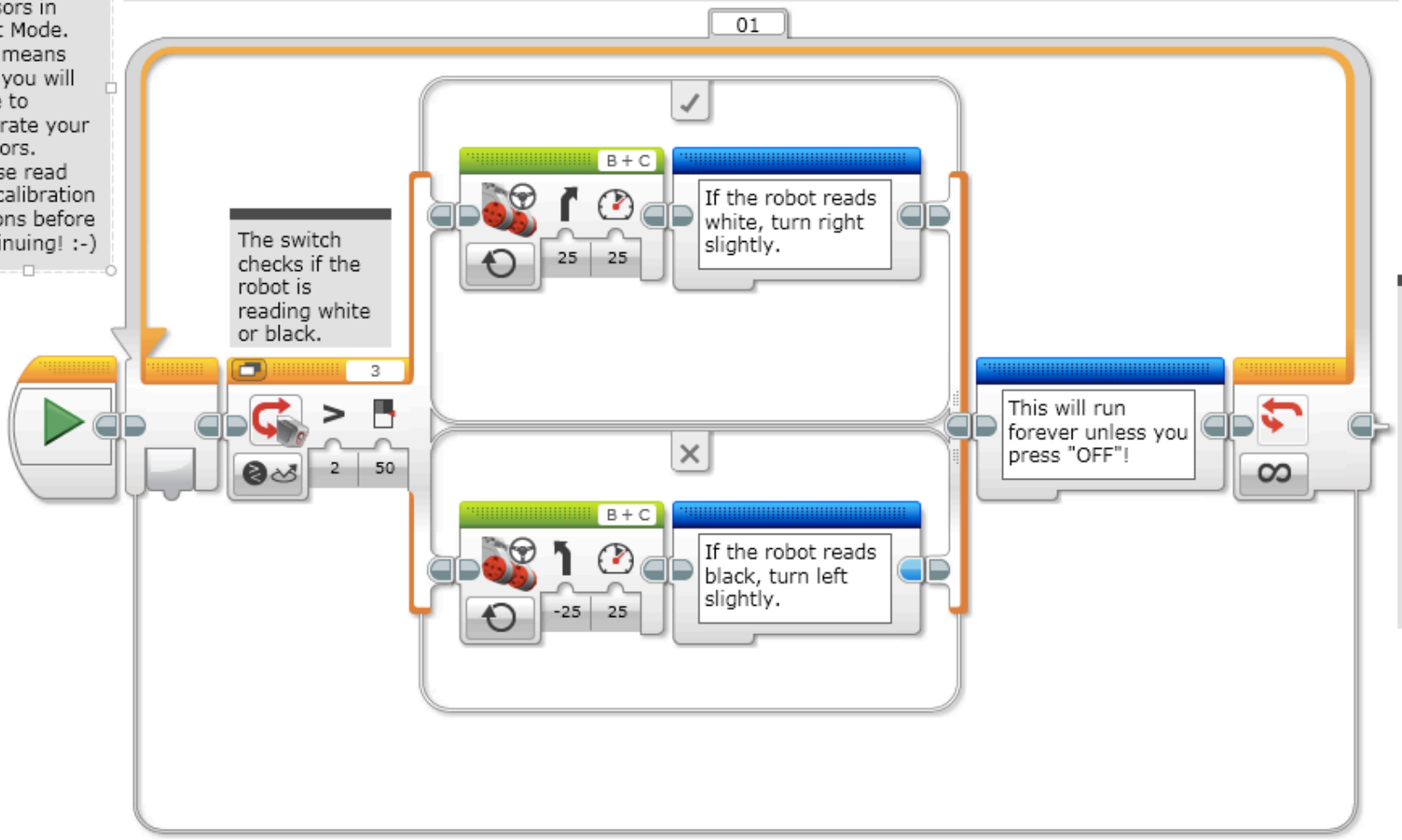
Note: This program uses the Color Sensors in Light Mode. This means that you will have to calibrate your sensors. Please read our calibration lessons before continuing! :-)



SMOOTH LINE FOLLOWER

Note: This program uses the Color Sensors in Light Mode. This means that you will have to calibrate your sensors. Please read our calibration lessons before continuing! :-)

Smooth Line Follower: The goal of this program is to create a simple line follower, but smoother than the first. This line follower will be smoother because it makes less sharp turns. The only difference between the Simple and the Smooth is the angle of the turns.



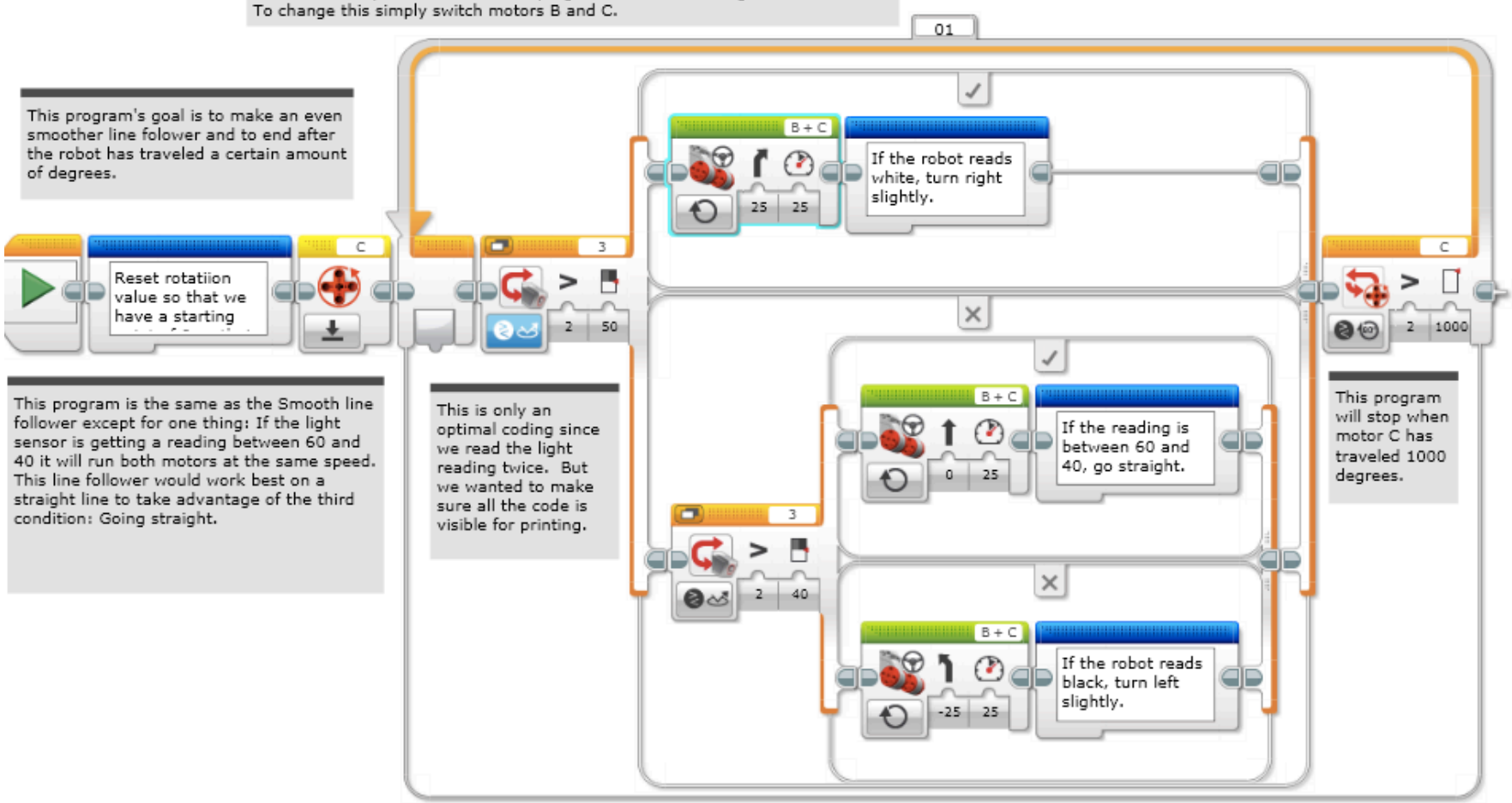
You probably want to line follow only for a certain distance. You click on the infinite symbol on the loop to change when you want to exit the loop.

3-STAGE LINE FOLLOWER

Note: We present this line follower because many teams talk about a multi-stage line follower and want to know how to write one. Our team recommends that you avoid this program and learn to make a proportional line follower!

Note: This program uses the Color Sensors in Light Mode. This means that you will have to calibrate your sensors. Please read our calibration lessons before continuing! :-)

Note 1: If B is your right motor, this program will follow the left side of the line.
Note 2: If B is your left motor, this program will follow the right side of the line.
To change this simply switch motors B and C.

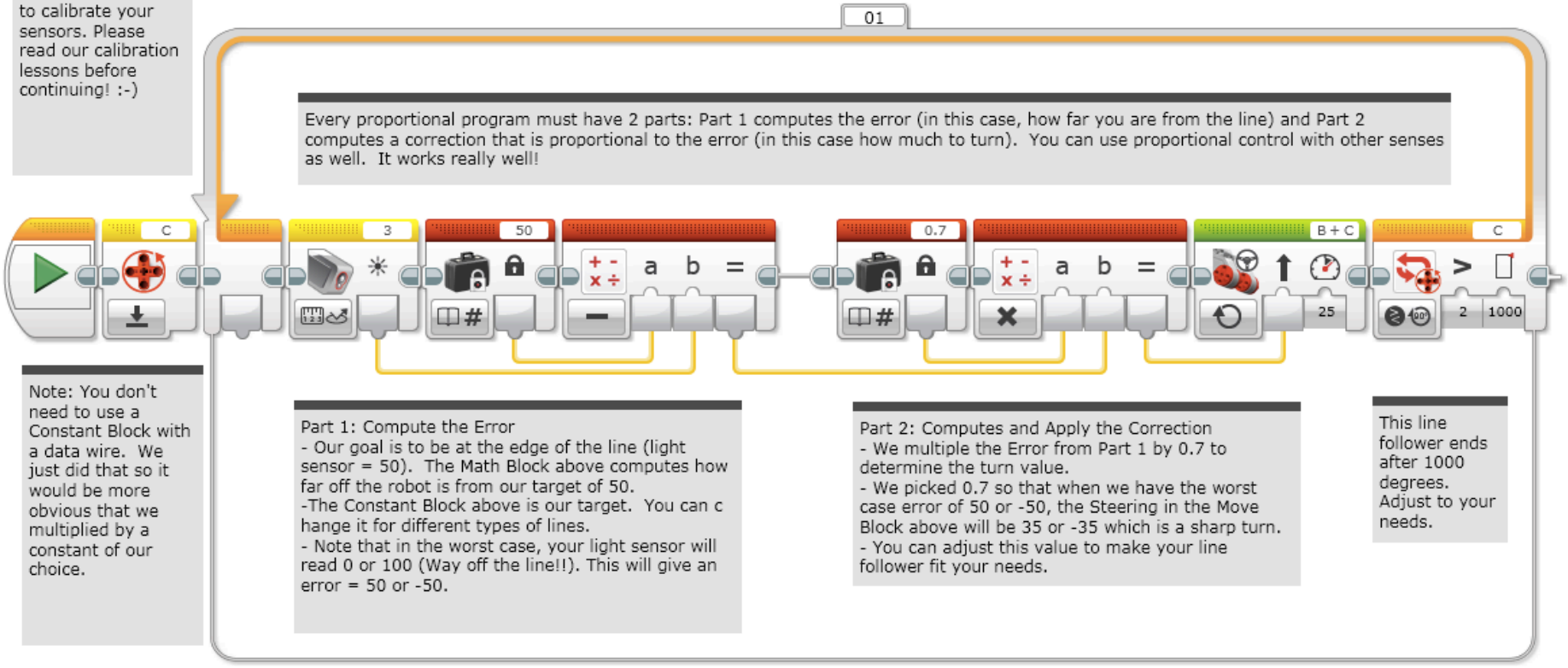


PROPORTIONAL LINE FOLLOWER

Note: This program uses the Color Sensors in Light Mode. This means that you will have to calibrate your sensors. Please read our calibration lessons before continuing! :-)

We recommend that your team uses a proportional line follower like this one. It will be smoothest of the 4 line followers in this lesson. There are even better line followers (that use PID control), but a line follower that uses the "P" is a great start.

A proportional line follower changes the angle of the turn based on how far away from the line the robot is.



Every proportional program must have 2 parts: Part 1 computes the error (in this case, how far you are from the line) and Part 2 computes a correction that is proportional to the error (in this case how much to turn). You can use proportional control with other senses as well. It works really well!

Note: You don't need to use a Constant Block with a data wire. We just did that so it would be more obvious that we multiplied by a constant of our choice.

Part 1: Compute the Error
- Our goal is to be at the edge of the line (light sensor = 50). The Math Block above computes how far off the robot is from our target of 50.
-The Constant Block above is our target. You can c hange it for different types of lines.
- Note that in the worst case, your light sensor will read 0 or 100 (Way off the line!!). This will give an error = 50 or -50.

Part 2: Computes and Apply the Correction
- We multiple the Error from Part 1 by 0.7 to determine the turn value.
- We picked 0.7 so that when we have the worst case error of 50 or -50, the Steering in the Move Block above will be 35 or -35 which is a sharp turn.
- You can adjust this value to make your line follower fit your needs.

This line follower ends after 1000 degrees. Adjust to your needs.

Updated: 10/17/2014. Code fixed to match comments. EV3 file updated.

TIPS

You will get better results

….if your color sensors are closer to the ground

….if you shield your color sensors

….remember to calibrate

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