ADVANCED EV3 PROGRAMMING LESSON

Line Followers: Basic to Proportional

By Droids Robotics

Lesson Objectives

1. Evaluate and compare different line followers
2. Learn to use the concept of “proportional” to create a proportional line follower

Prerequisites: Basic Line Follower, Color Line Follower, Color Sensor Calibration, Proportional Control, Math Blocks, Data Wires

Which Program Works Best for Which Situation?

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

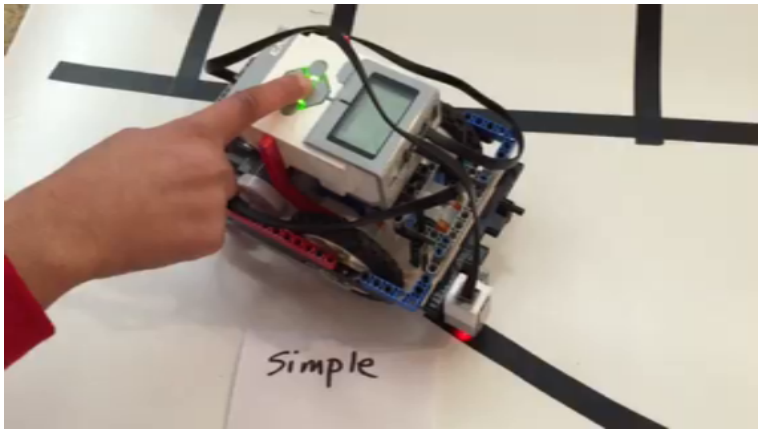
Proportional Follower

- 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

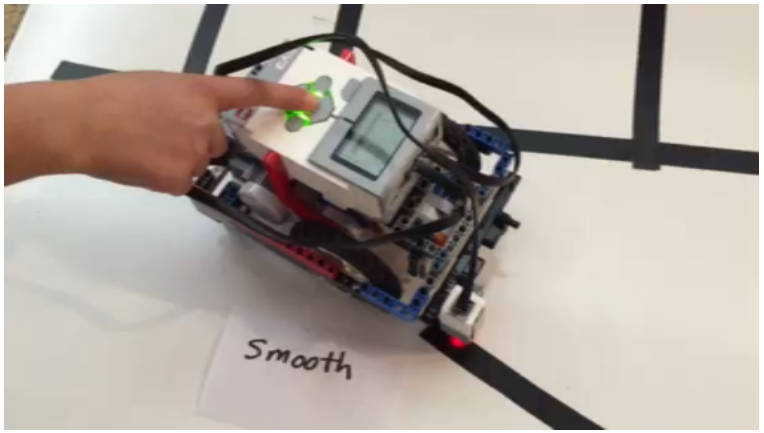
Watch the videos on the next 2 slides to see all four.

Curved Line: Watch Videos

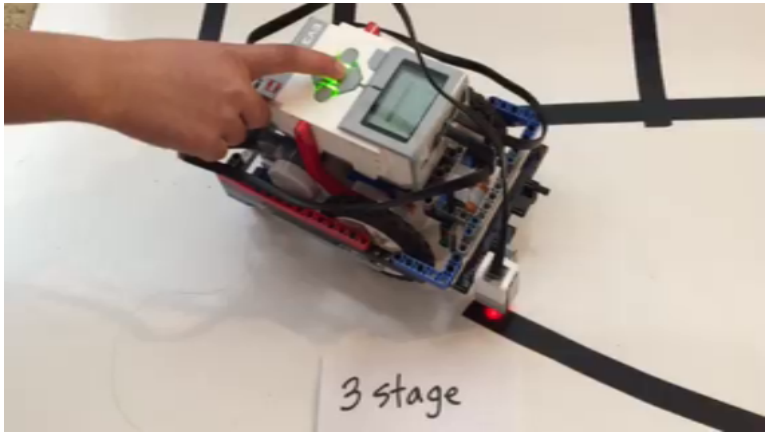
Simple Line Follower



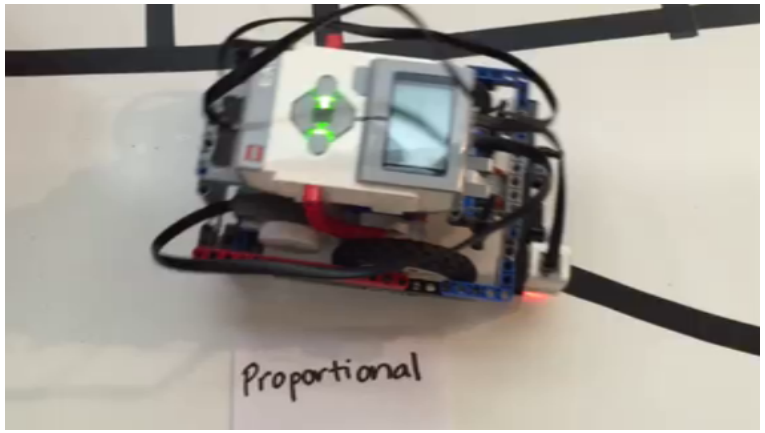
Smooth Line Follower



3-Stage Follower

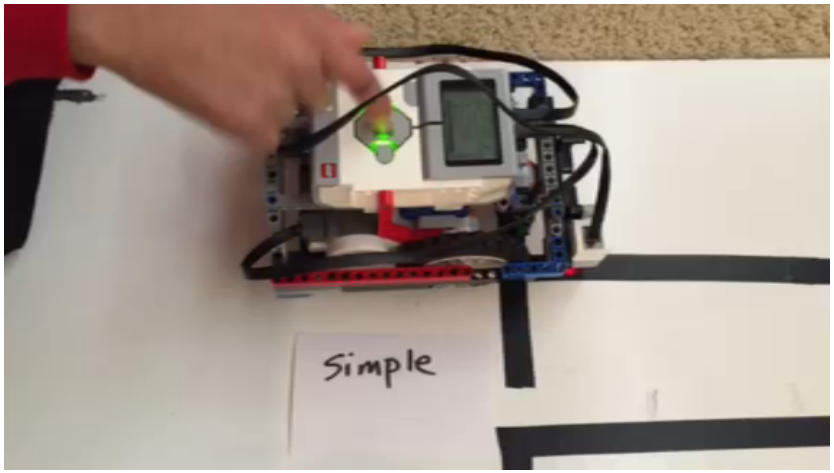


Proportional Follower

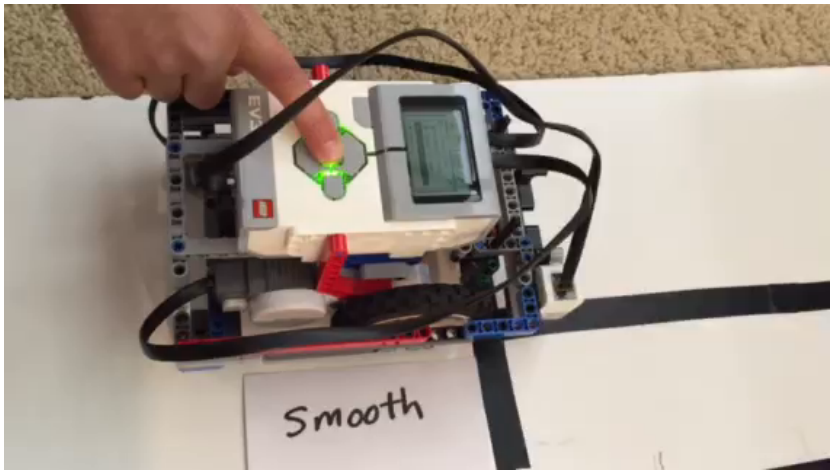


Straight Line: Watch Videos

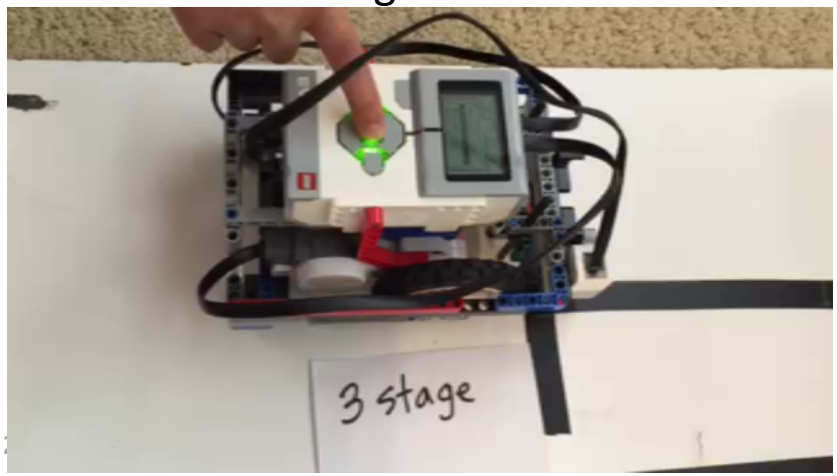
Simple Line Follower



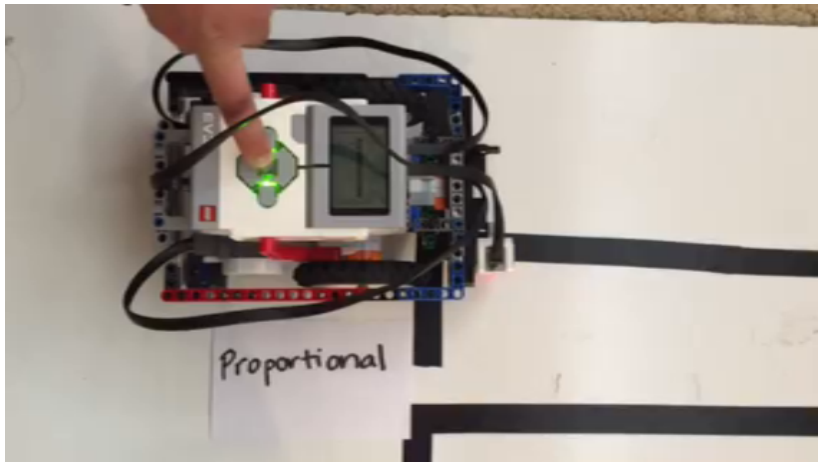
Smooth Line Follower



3-Stage Follower



Proportional Follower



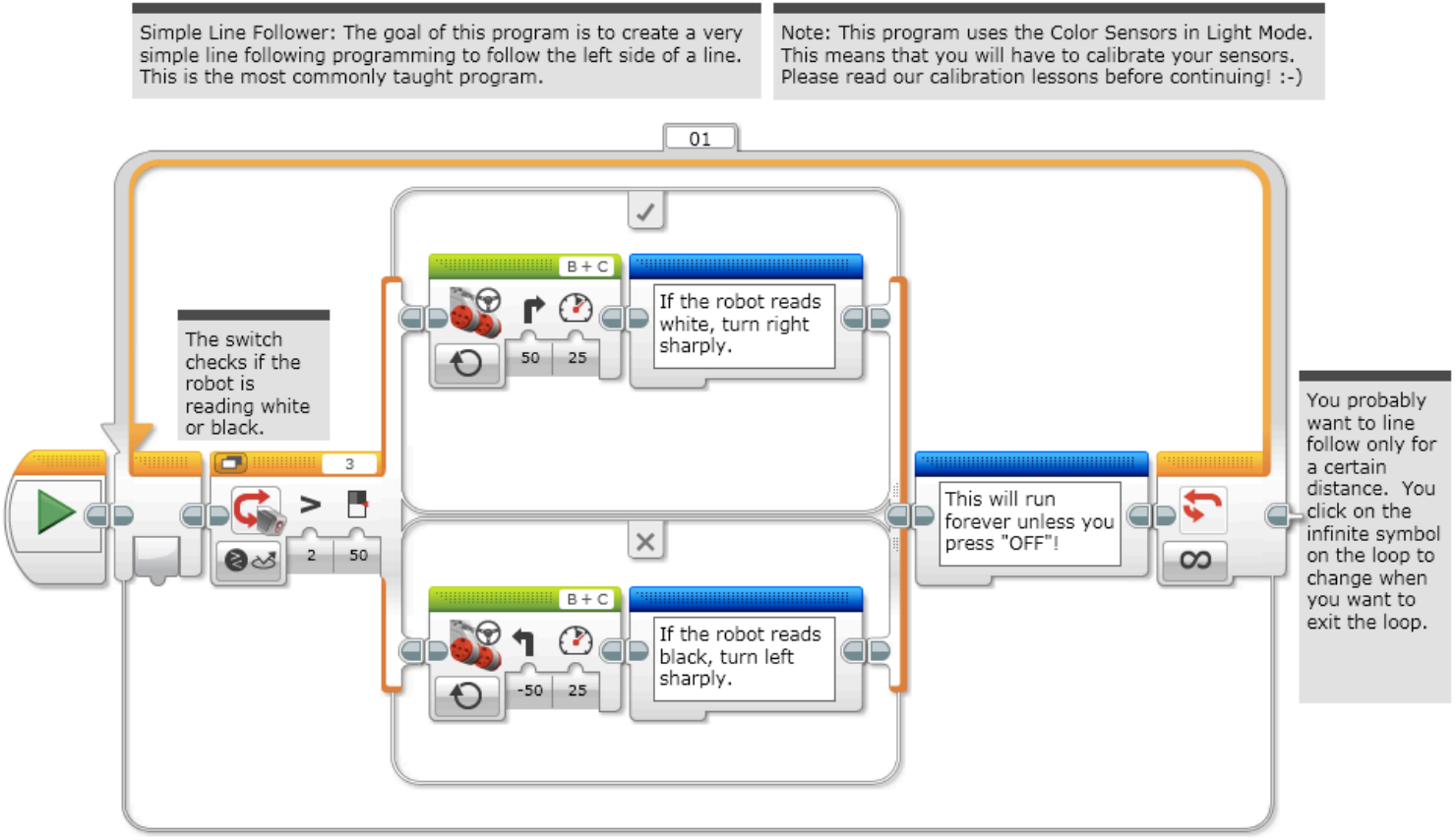
3 Line Follower Challenges

- Challenge 1: Can you write a simple line follower? Hint: Review Beginner: Basic Line Follower lesson
- Challenge 2: Can you write a smoother line follower? Hint: Change how sharp the turns are in a simple line follower.
- Challenge 3: Can you write a three-stage line follower where the robot moves different 3 different ways (left, right or straight) based on the reading from the color sensor?

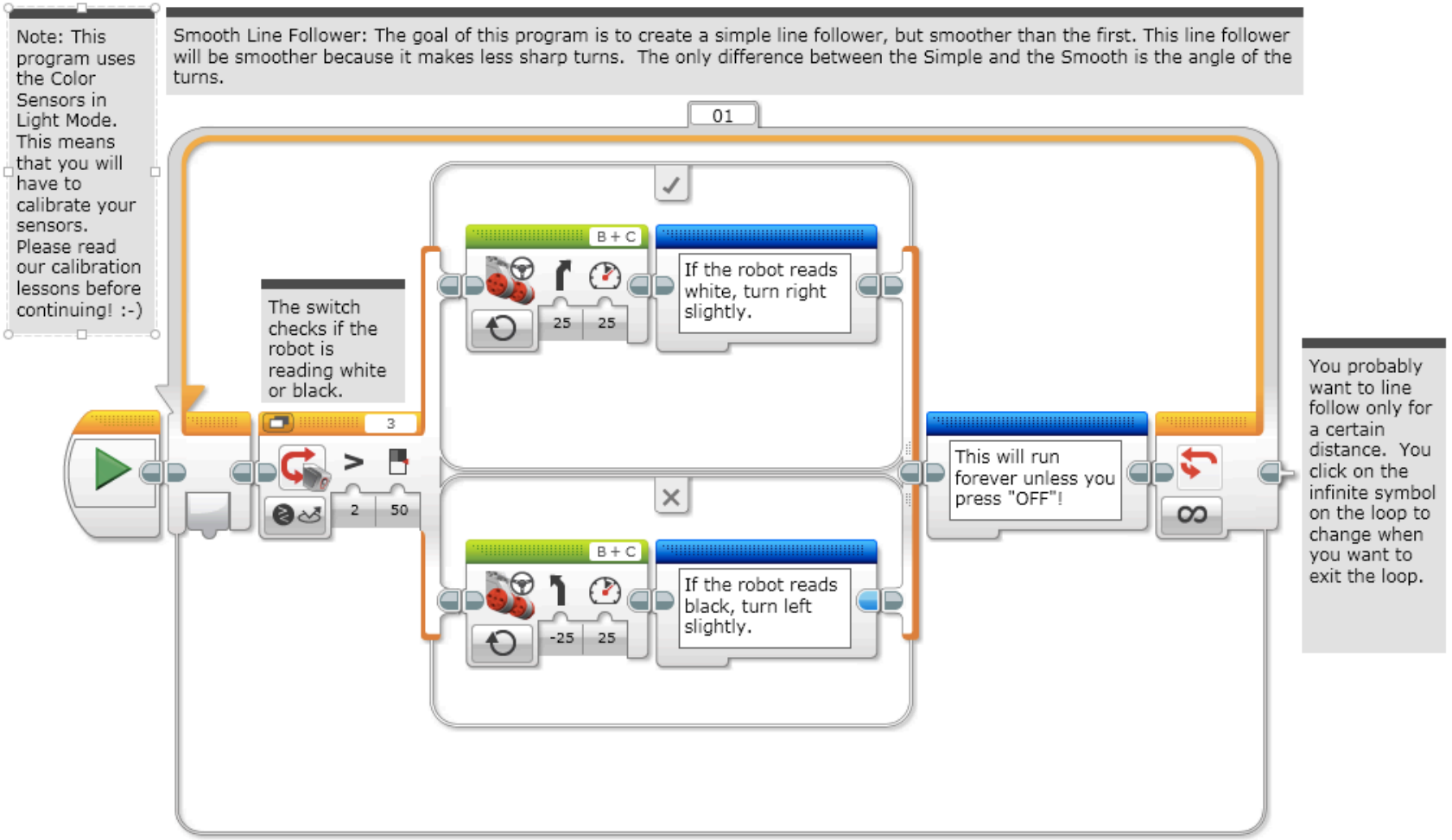
A Note About Our Solutions

- CALIBRATE:
  - The programs use the EV3 Color Sensor in Light Sensor mode
  - You will have to calibrate your sensors.
  - Please refer to Intermediate: Color Sensor Calibration Lesson
- 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

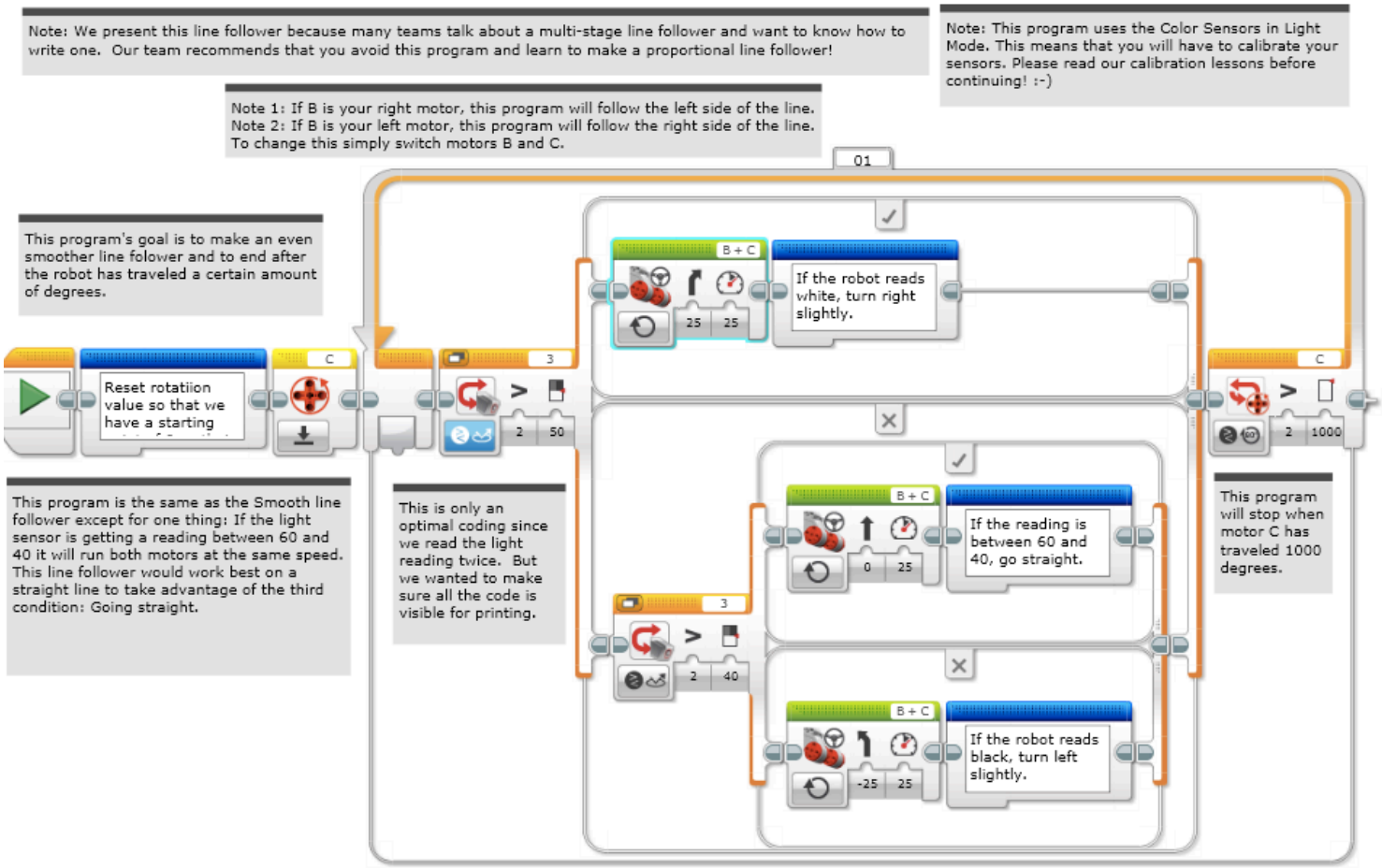
Solution 1: Simple Line Follower



Solution 2: Smooth Line Follower



Solution 3: Three-Stage Line Follower



Challenge 4: Proportional Line Follower

Challenge 4: Can you write a proportional line follower that changes the angle of the turn depending on how far away from the line the robot is?

Pseudocode:

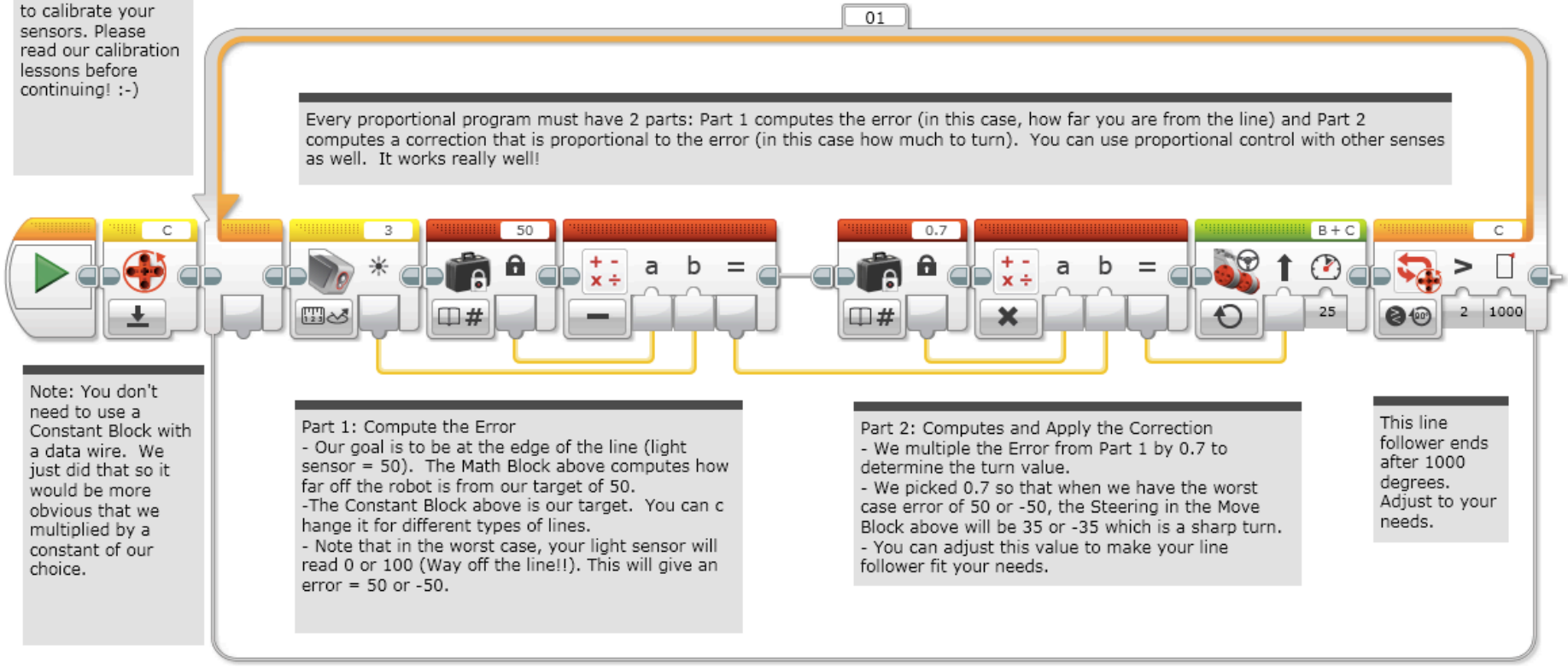
1. Reset the Rotation sensor (Only required for line following for a total distance)
2. Compute the error = Distance from line = (Light sensor reading – Target Reading)
3. Scale the error to determine a correction amount. Adjust your scaling factor to make you robot follow the line more smoothly.
4. Use the Correction value (computer in Step 3) to adjust the robot's turn towards the line.

Solution: 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.

Tips

- You will get better results
- ….if your color sensors are closer to the ground
- ….if you shield your color sensors
- ….remember to calibrate

Discussion Guide

Simple Line Follower

+

+

-

-

Smooth Line Follower

+

+

-

-

Three-Stage Line Follower

+

+

-

-

Proportional Follower

+

+

-

-

Fill in the above with positives and negatives of each technique. Consider if the line follower is best for curved or straight lines. Consider if the robot will wiggle a lot.

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