

# **BEGINNER FLL PROGRAMMING WORKSHOP**

**BY DROIDS ROBOTICS & EV3LESSONS**

# GOALS FOR THIS WORKSHOP

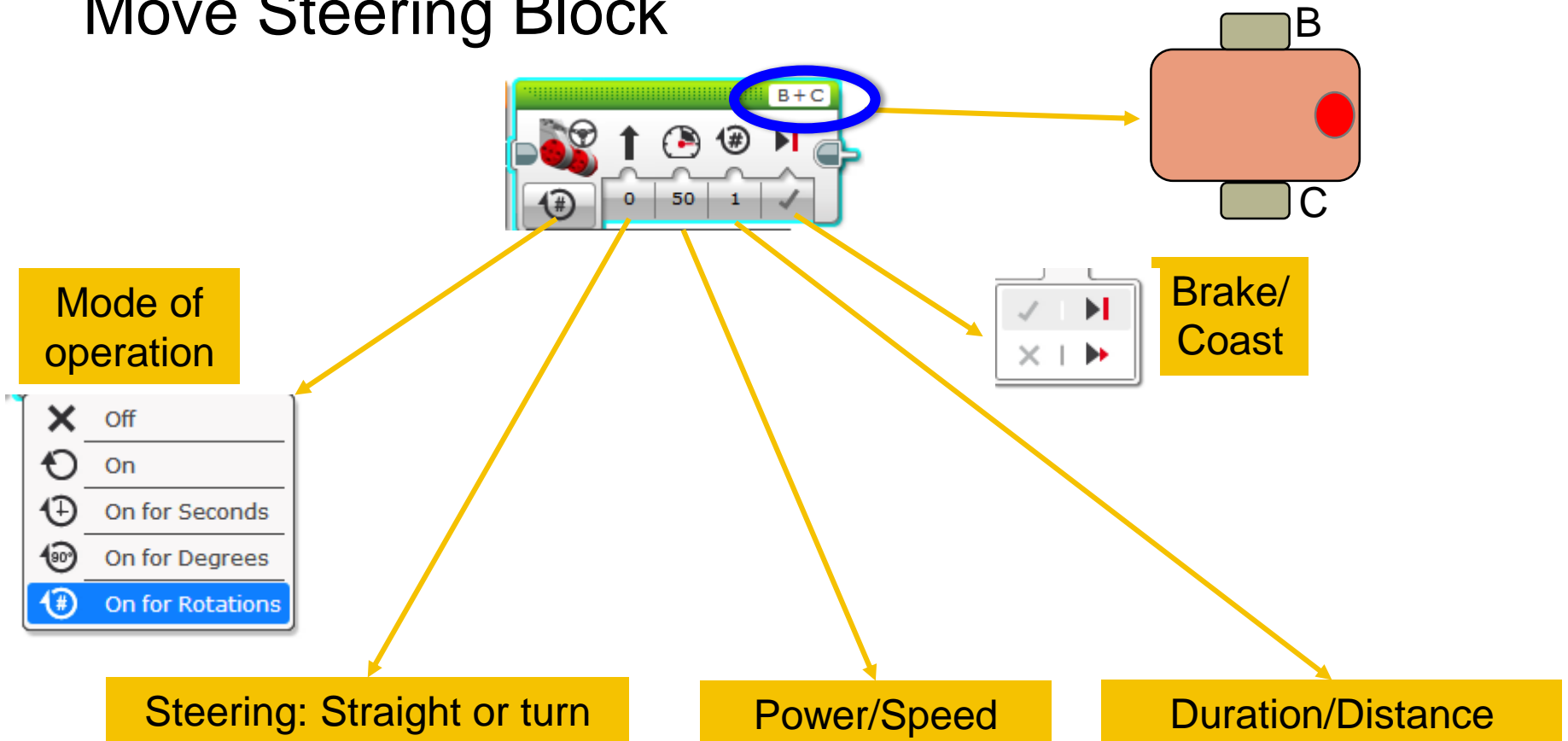
- **Use sensors to solve FLL missions**
  - Wait For Block with a Sensor
  - Line Following with Loops & Switches
- **Learn some tips & tricks in building and programming**
- **Understand where on the Trash Trek Mat you can use these techniques**

# WHERE CAN I LEARN MORE?

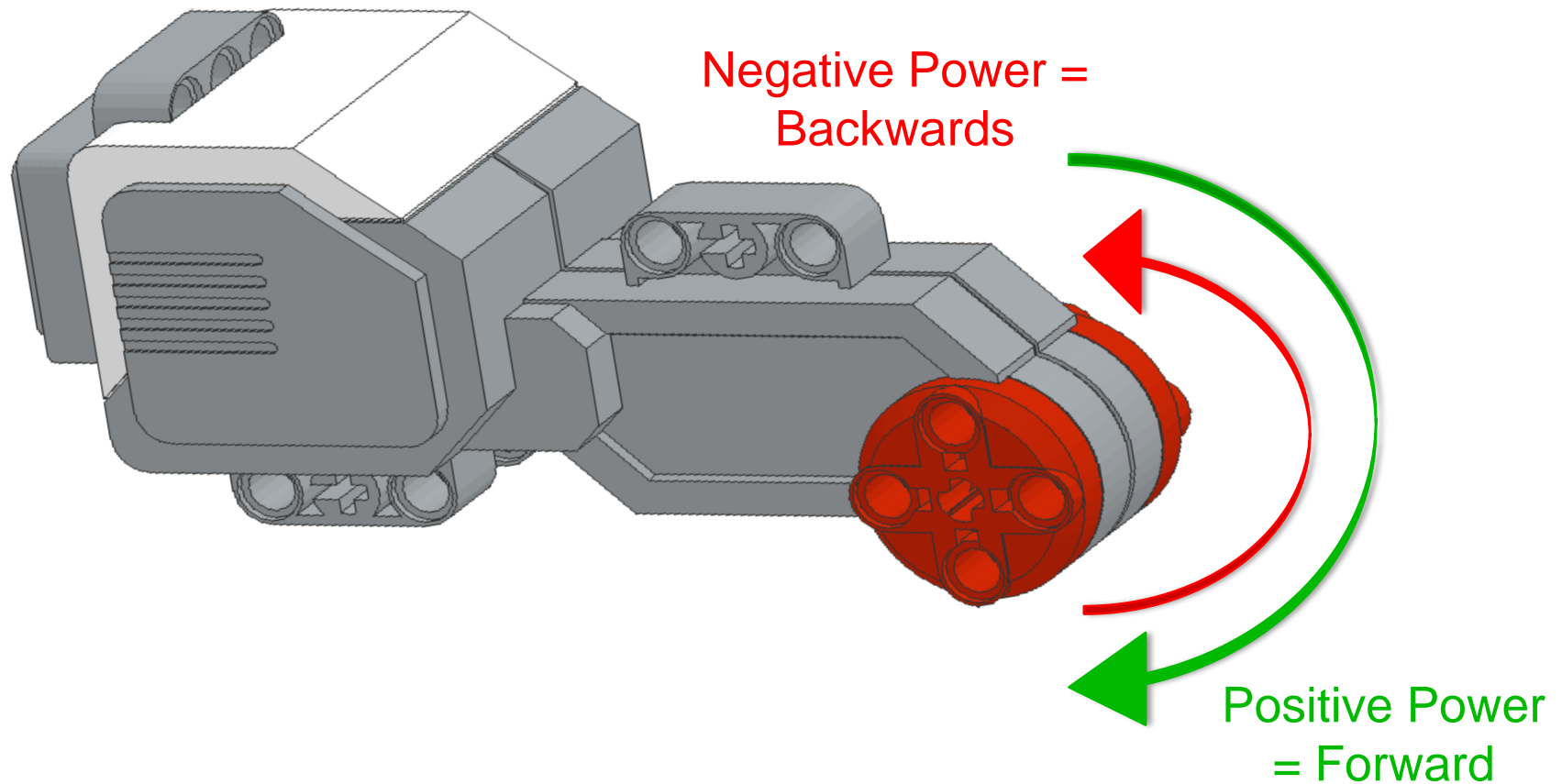
- **During the workshop, we only provided a review of Moving Straight, Turning and Port View. If you want to learn more, please visit [EV3Lessons.com](http://EV3Lessons.com) – Beginner.**
- **We only provided a quick introduction to Loops and Switches during the Line Following lesson. For fun challenges and to learn more, please view the Beginner lessons on Display Blocks, Loops and Switches.**

# REVIEW: MOVING STRAIGHT

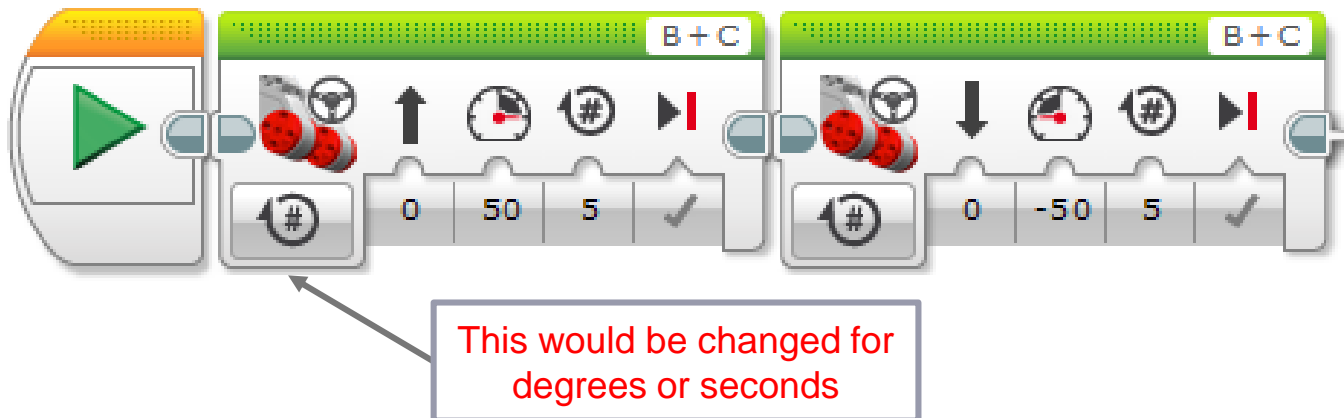
## Move Steering Block



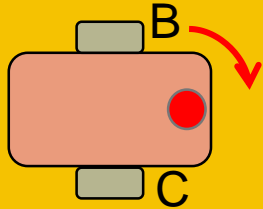
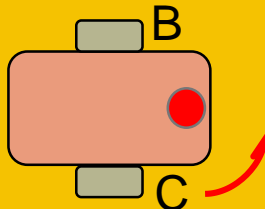
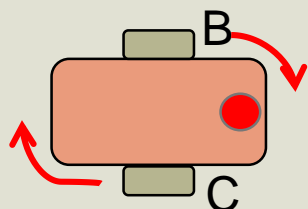
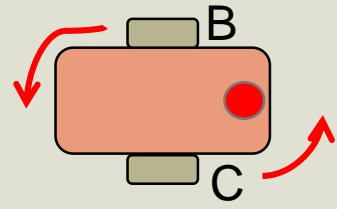
# NEGATIVE & POSITIVE POWER: BACKWARD & FORWARD



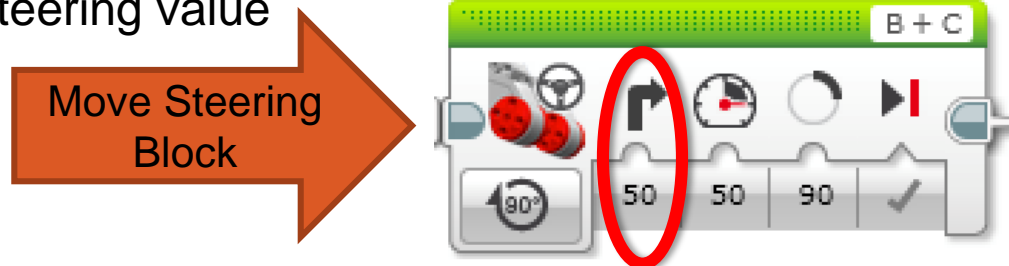
# REVIEW: MOVING FORWARD AND BACKWARDS



# REVIEW: PIVOT & SPIN TURNS

Different Turns and Different Steering Values			
50	-50	100	-100
			
Pivot Turn Right	Pivot Turn Left	Spin Turn Right	Spin Turn Left
More accurate, but takes more space		Good for tight spaces, faster, less accurate	

Change Steering value



# REVIEW: USING ATTACHMENTS

- **Attach a medium motor to Port A or a large motor to Port D as needed.**
- **Move Steering vs. Motor Block**
  - For moving your wheels you should use a Move Steering Block that syncs both wheel motors (see intermediate lesson called Move Blocks to learn about sync)
  - For moving your attachment your arm, you use either a Medium Motor Block or a Large Motor Block because you don't need to sync your motors.

Medium Motor Block

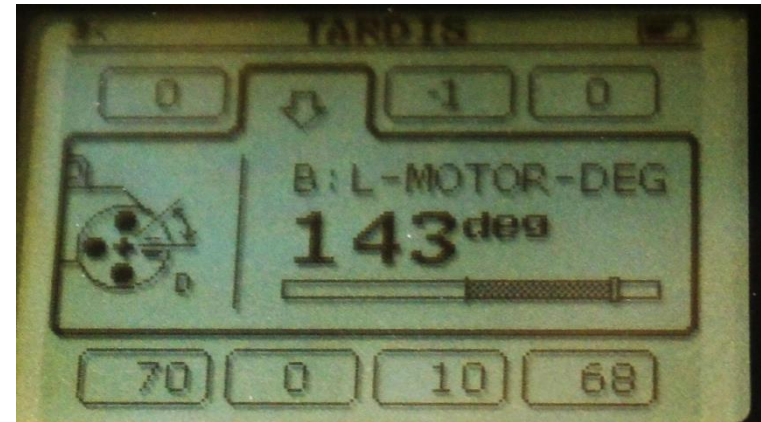
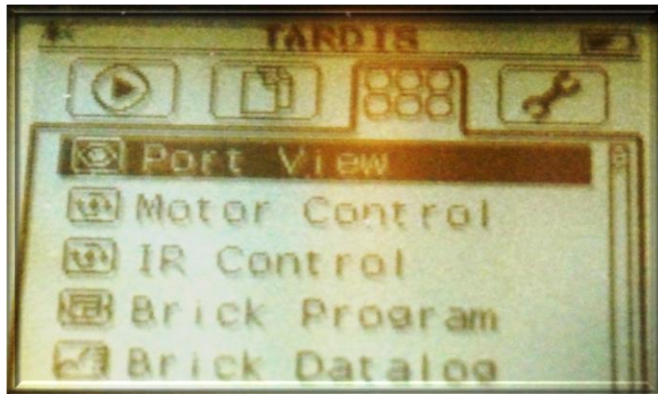


Large Motor Block





# USEFUL TIP: PORT VIEW



**Rookie Tip:** Use Port View on your brick to read sensor values, and measure distances

# WHY USE SENSORS IN FLL?

- 1. Sensors can help you know your position on the FLL table**
- 2. Sensors can help you be more accurate therefore receiving less touch penalties**
- 3. Sensors can help you do accurate turns, straighten up on lines, move until a certain distance from a wall, and know when you are on a wall**
- 4. Moving up to a mission model accurately might need a sensor**

# WHAT IS A SENSOR?

- A sensor lets an EV3 program measure and collect data about its surroundings
- The EV3 sensors for FLL include:
  - Color – measures color and darkness
  - Gyro – measures rotation of robot
  - Ultrasonic – measures distance to nearby surfaces
  - Touch – measures contact with surface

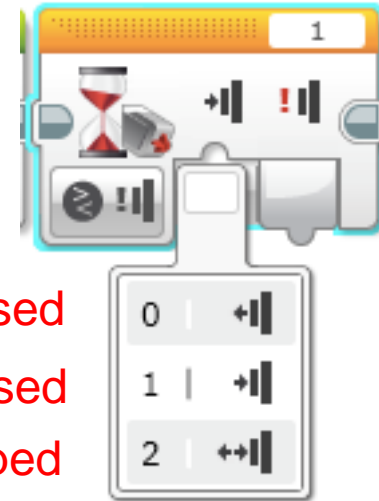


Image from: [http://www.ucalgary.ca/IOSTEM/files/IOSTEM/media\\_crop/44/public/sensors.jpg](http://www.ucalgary.ca/IOSTEM/files/IOSTEM/media_crop/44/public/sensors.jpg)

# WHAT IS A TOUCH SENSOR?

- Touch Sensor can detect when the sensor's red button has been pressed or released
- With this information, you can program an action when the sensor is:

Currently Pressed  
Currently Released  
Bumped



# WHEN MIGHT YOU USE THIS SENSOR IN FLL?

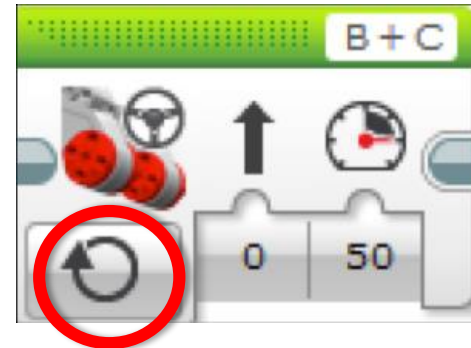
- **Useful for programming “moving until touch sensor is pressed/released”**
- **For example, if you put a touch sensor on the front the robot, you can have it stop moving if it runs into something.**
- **You can also have your program start or stop when a touch sensor is pressed.**

# MOVE ON AND OFF

What would happen if you placed a Move Steering Block and left the motor “On”?

Would the robot...

- 1) Move?
- 2) Move for a little while?
- 3) Not move at all?



**ANS.**

**What does Motor Off do?**

**Rookie Tip:** Motor On needs to be followed by another block (e.g. Wait Block)

# HOW DO YOU PROGRAM WITH THE TOUCH SENSOR?

## Wait For Block



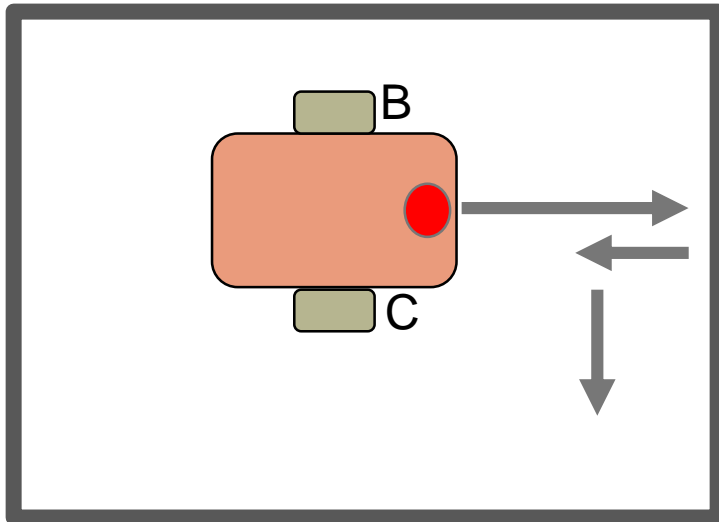
Orange Flow Tab:  
Wait for Block

- Used to wait for a sensor reading (or time)



# CHALLENGE

Program your robot to move until it hits the edge of a wall. Then back up and turn right 90 degrees.



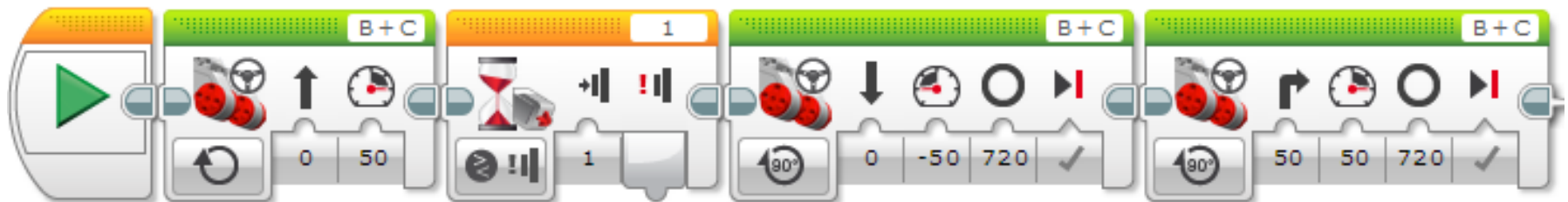
0 = released  
1 = pressed  
2 = bumped

**Hint:** You will combine Move Steering + Turning + Wait Block



# CHALLENGE SOLUTION

The goal of this program is to make your robot move until it hits the edge of a wall. Then back up and turn right 90 degrees



Set move steering block to "on"

Set wait block to Touch-->Compare-->State

Set move steering block to "degrees" and steering to 50. The 720 degrees value will have to be modified for your robot (You measured this in port view earlier beginner lessons).

# HOW DO YOU PROGRAM WITH THE ULTRASONIC?

**Very similar to the touch sensor**

**Just change the wait for block to wait for a reading from the ultrasonic sensor**

**Use it to stay a particular distance away from the wall or mission model.**

# WHAT IS THE COLOR SENSOR?

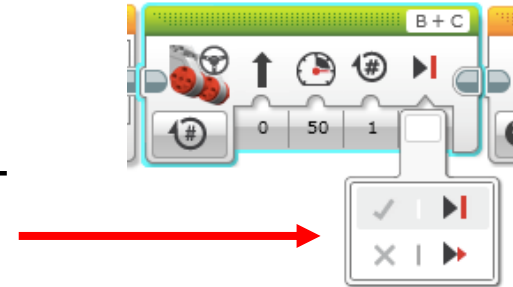
- What are they? Sensors that detect the intensity of light that enters it
- Three modes: Color, Reflected Light Intensity and Ambient Light Intensity
  - **Color Mode:** Recognizes 7 colors (black, brown, blue, green, yellow, red, white) and No Color
  - **Reflected Light:** Measures the intensity of the light reflected back from a lamp that emits a red light. (0=very dark and 100=very light)
  - **Ambient Light:** Measures the strength of the light that enters the sensor from the environment. (0=very dark and 100=very light)





# ANOTHER MOVE STEERING TIP: COAST OR BRAKE?

- Something more about the Move Steering Block
- You will notice you have an option to COAST or BRAKE
- Coast will make the motors keep moving. Brake makes the motors stop immediately.
- Which do you use to stop EXACTLY on a colored line?



# COLOR SENSOR CHALLENGE

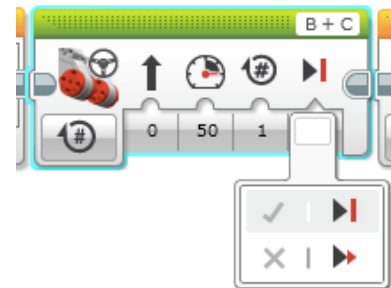
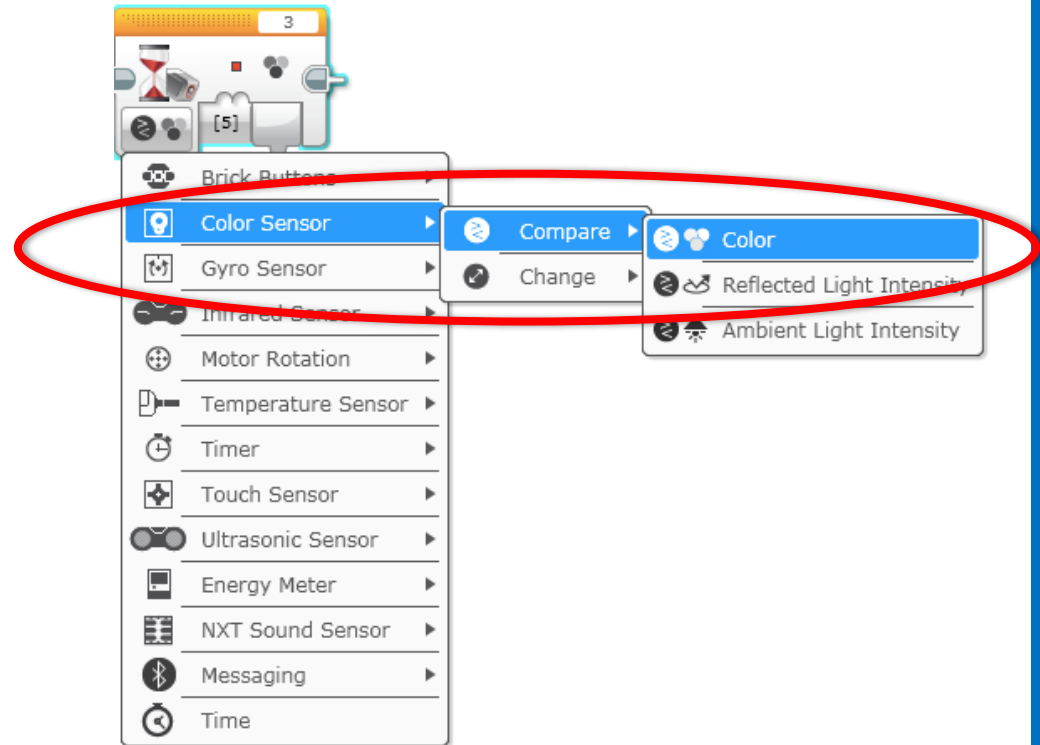
Make the robot move up to a black line using the color sensor?

Step 1: Use Wait For Color

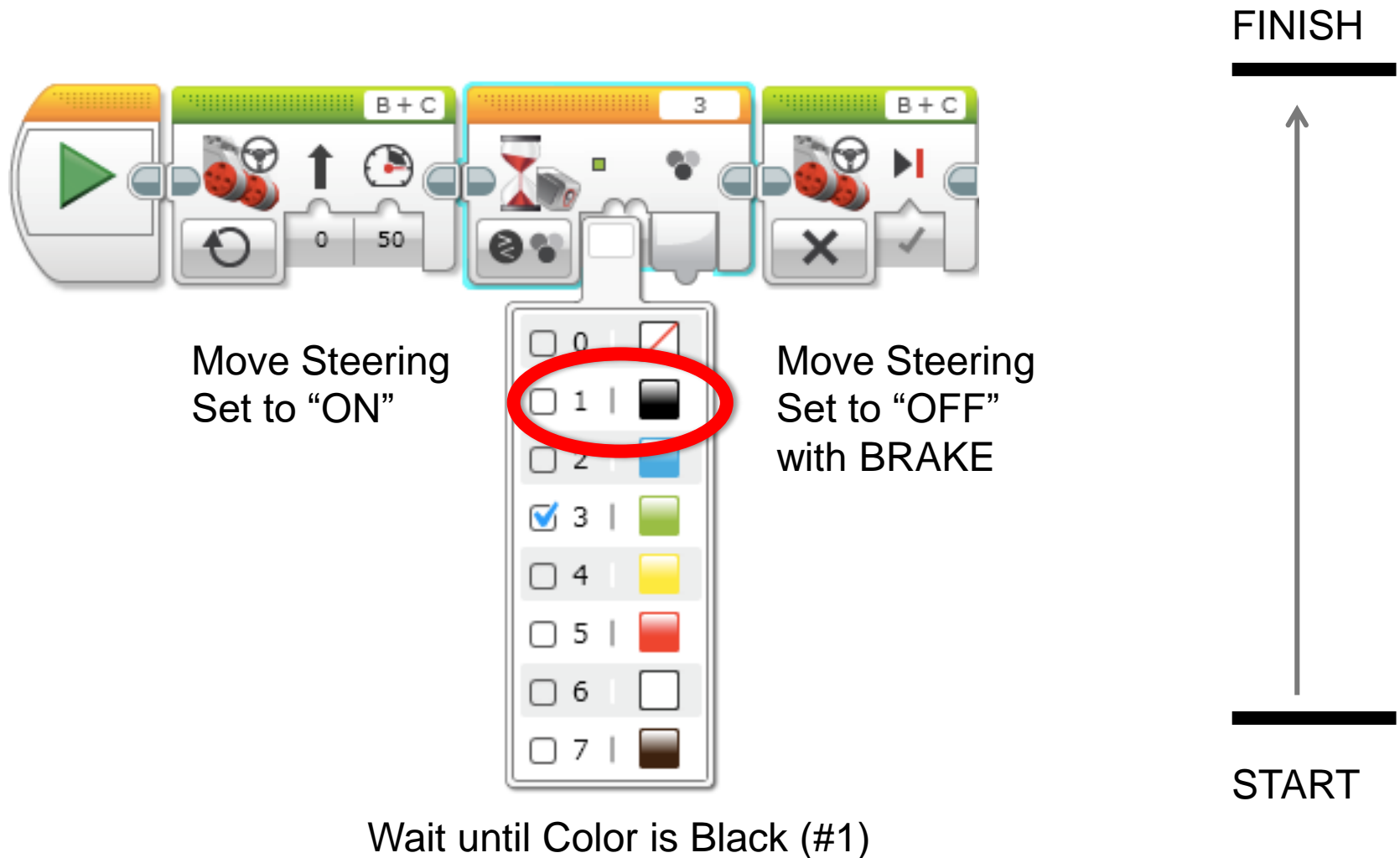
Step 2: Use the color sensor in COLOR MODE

Step 3: Coast or Brake?

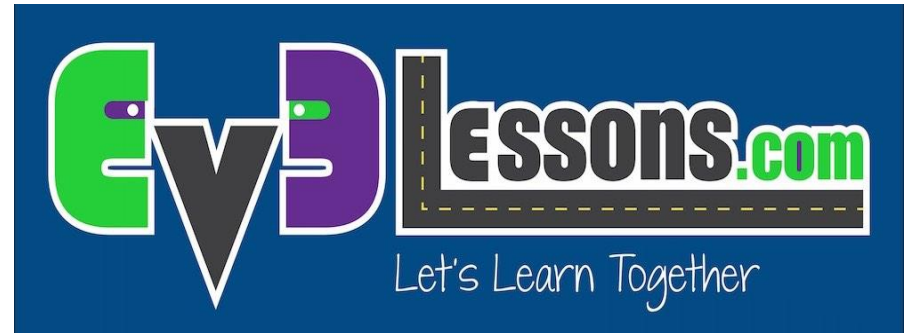
Hint: You will use Move Steering (think about motor on and off) and Wait for "Color"



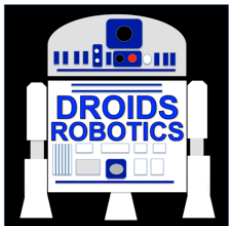
# COLOR SENSOR CHALLENGE SOLUTION



# BEGINNER PROGRAMMING LESSON



## Basic Line Follower



By: Droids Robotics



# WHERE LINE FOLLOWING COULD BE USEFUL IN FLL

**Trash Trek mat is covered with black lines**

**Lines go up to useful regions**

**Lines go up to mission models**

# FOLLOW THE MIDDLE?

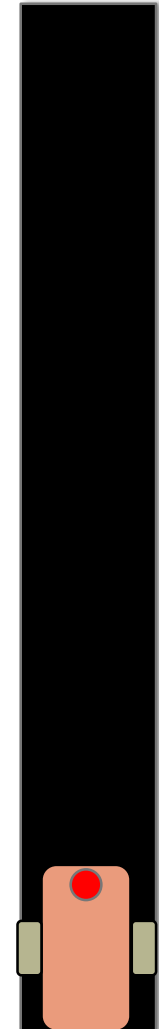


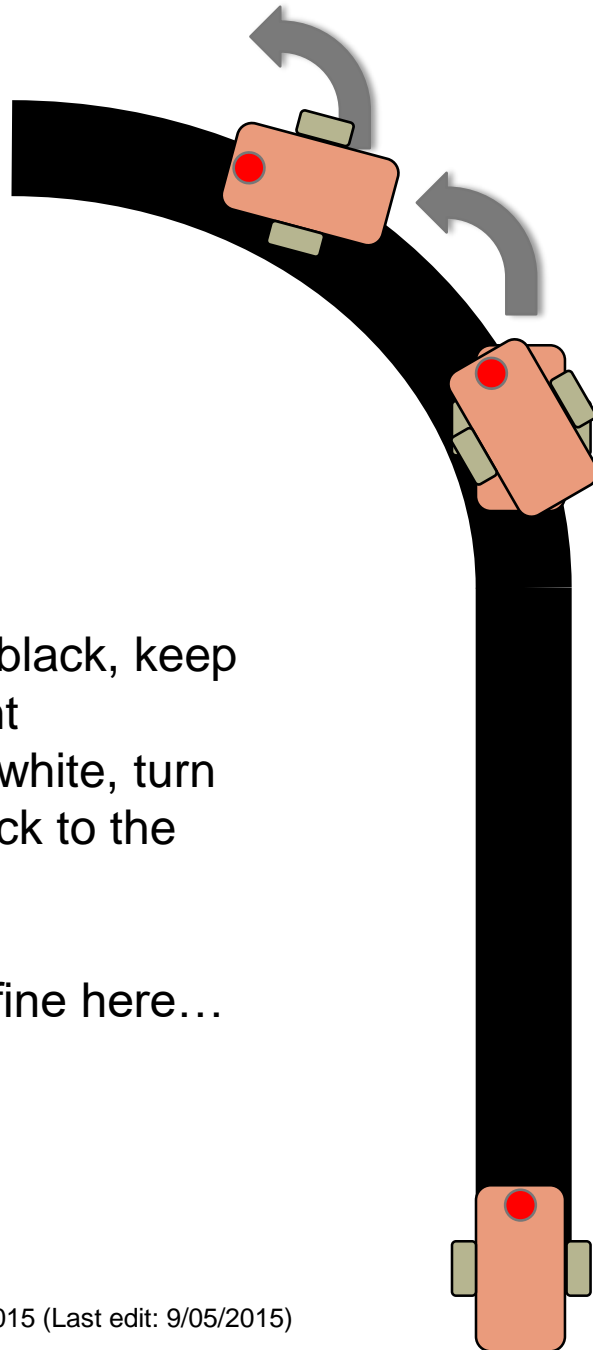
Humans want to follow the line in the middle.

Let's have the robot do the same thing using the **Color Sensor**

What type of questions can we ask using this sensor

- Are you on line or not?





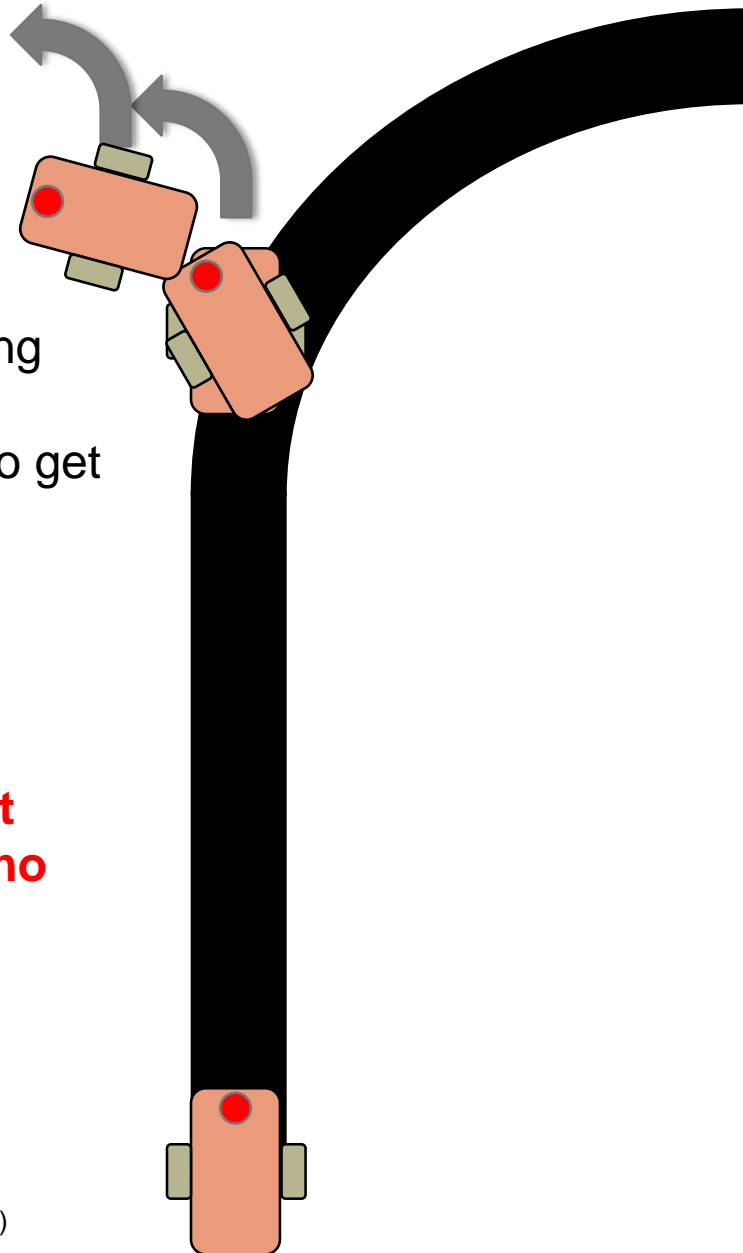
1. If we are on black, keep going straight
2. If we are on white, turn left to get back to the line

Seems to work fine here...

1. If we are on black, keep going straight
2. If we are on white, turn left to get back to the line

**OH NO... my robot is running away....**

**When the robot leaves the left side of the line, the program no longer works!**



# LINE FOLLOWING: ROBOT STYLE

## Why could the Human follow the middle?:

- They can **see ahead**.
- They can **see the whole line and its surroundings**
- They **see both sides** and which side they left

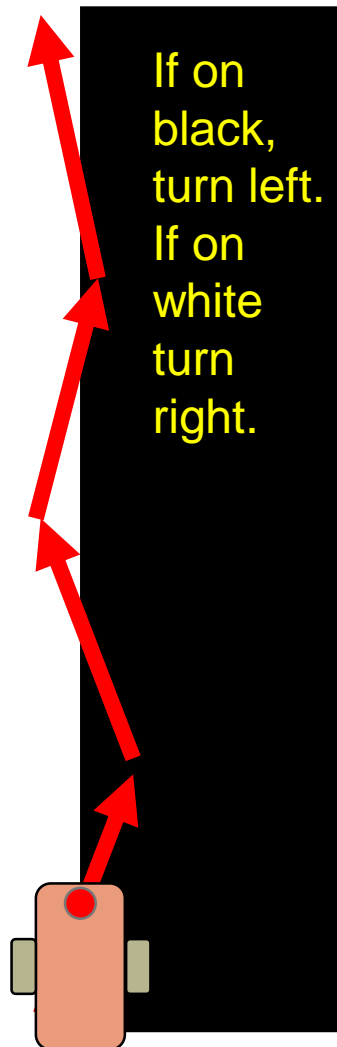
## Why can't the Robot do the same thing?:

- **Can't tell right or left side of the line**
- **How do we make sure the robot always veers off on the SAME SIDE of the line?**
  - Instead of the middle, could the robot follow the "edge"?
- So now the robot will fall off only the same side.
- We will now show you how this works!

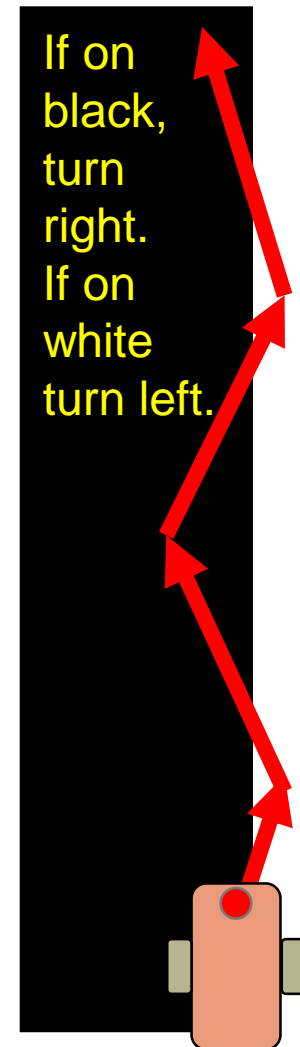


# ROBOT LINE FOLLOWING HAPPENS ON THE EDGES

Left side line following



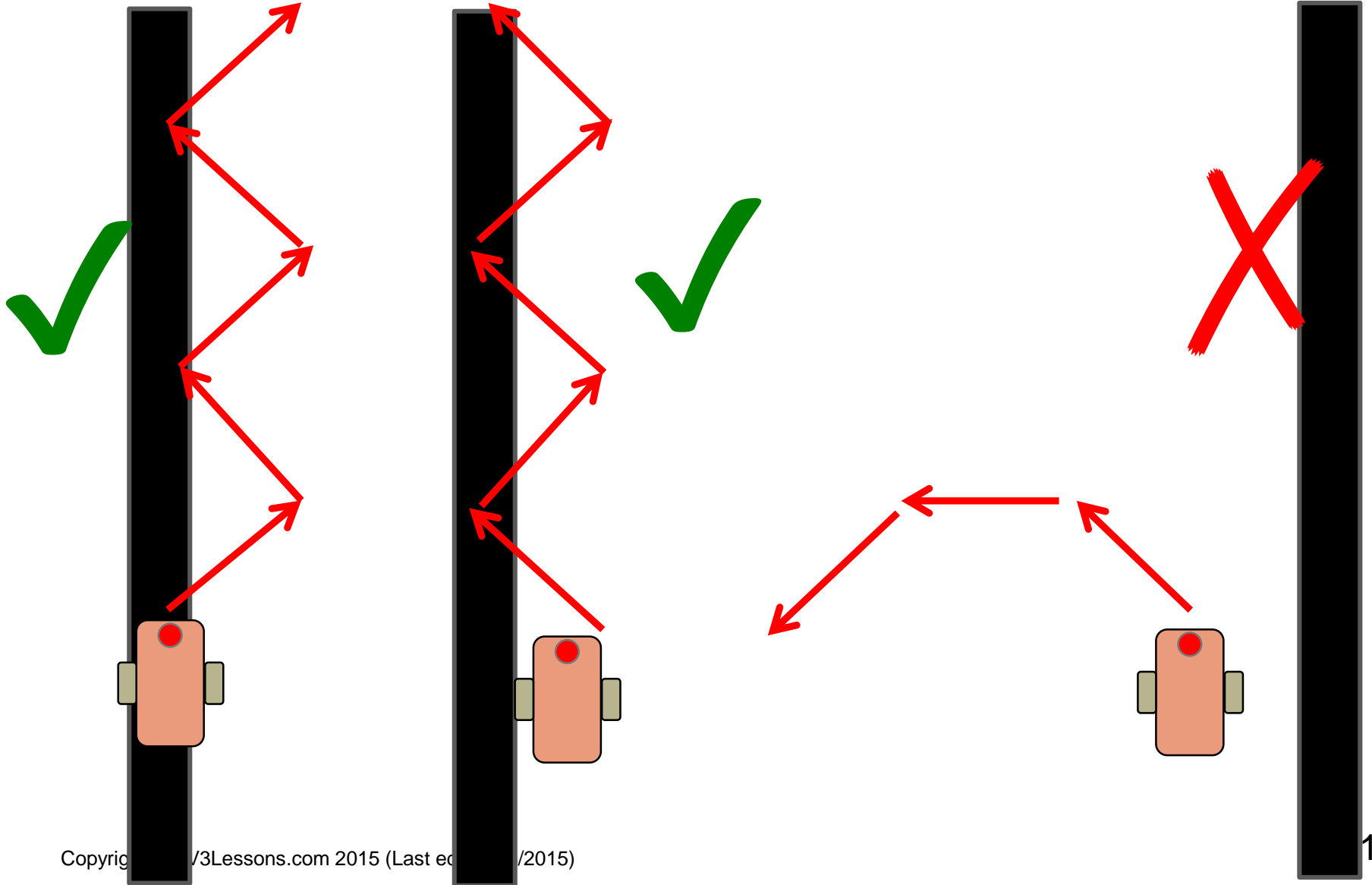
Right side line following



The robot has to choose which way to turn when the color sensor sees a different color.

The answer depends on what side of the line you are following!

# STARTING THE ROBOT ON THE CORRECT SIDE



# HOW DO YOU WRITE A LINE FOLLOWER?

First, the robot will have to decide between 2 actions. What are the two actions?

ANS:

Second, the robot is repeating an action over and over again. What action does the robot repeat again and again?

ANS:



# SWITCH BLOCKS



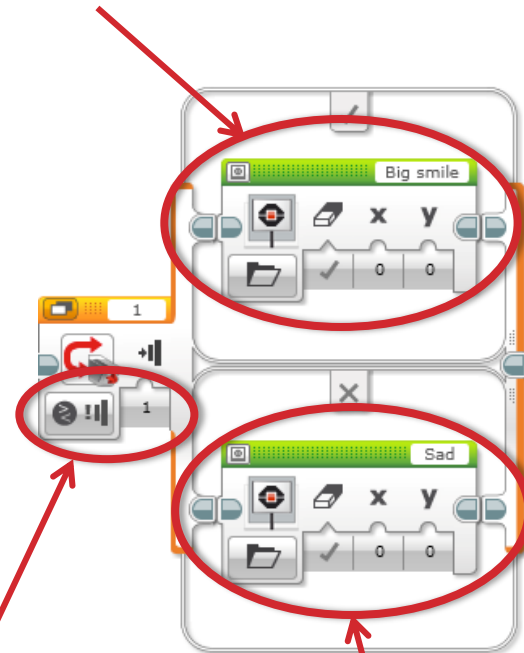
**Asking the robot a question and doing something different based on the answer**

- Example: Does the robot see a line? Or not?

**Basically a YES/NO QUESTION**

**Switch blocks are found in the orange/flow tab**

Run this code if the answer is yes



The question being asked: is the touch sensor pressed

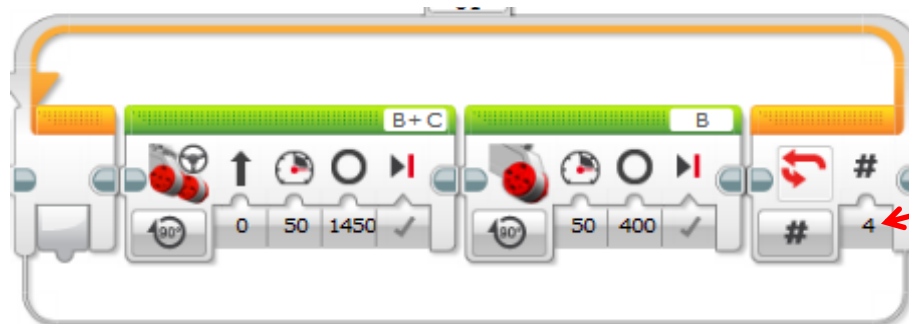
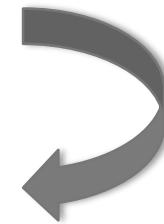
Run this code if the answer is no

# LOOPS



## Loops make repeating a task multiple times easy

- KEEP GOING....Forever, for a Count, Until touch (or something else)

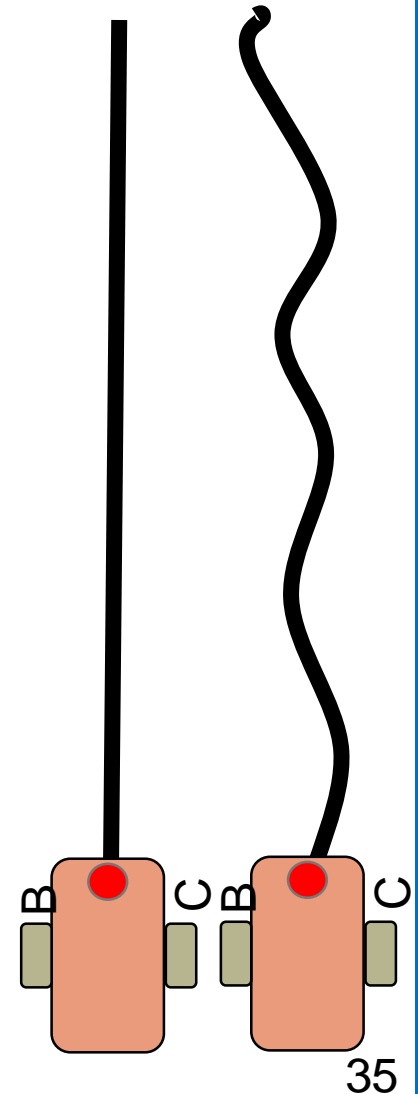
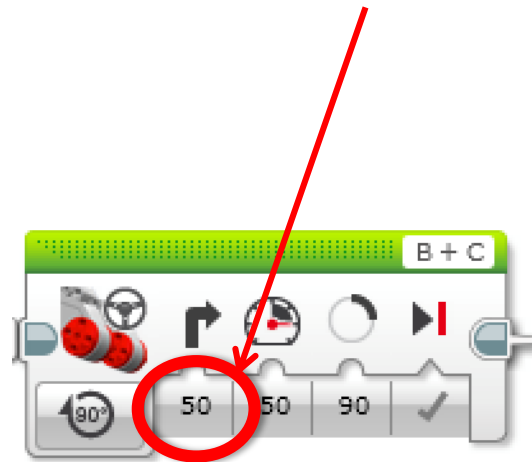


Repeat the loop 4 times

# LINE FOLLOWER CHALLENGE 1

**Step 1:** Write a program that follows the **RIGHT** edge of a line.

**Hints:** If your sensor sees black, turn right. If your sensor sees white, turn left. Use loops and switches!



# OTHER THINGS TO TRY

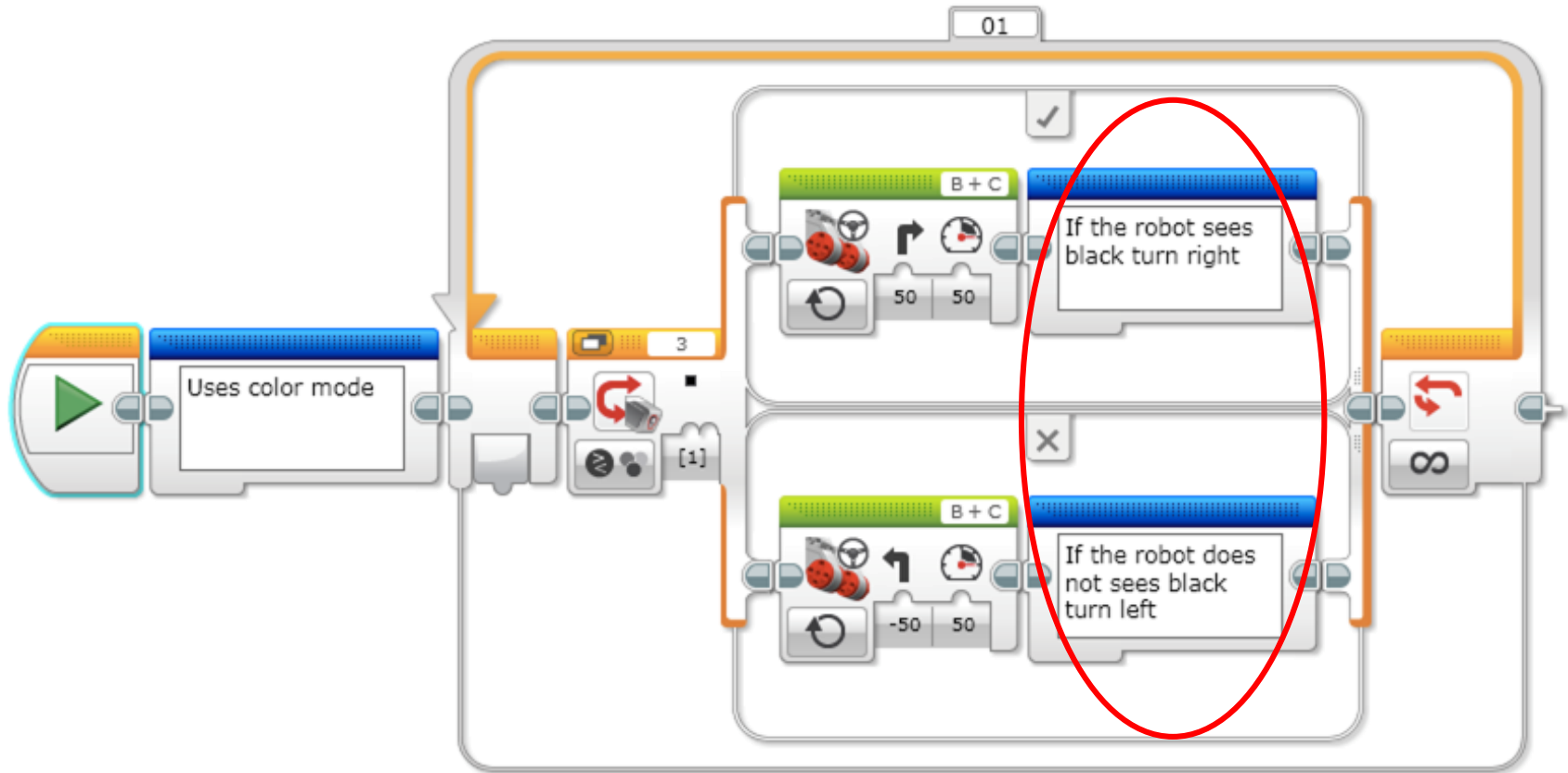
Try the line follower on different lines.

Can you write a smoother line follower (less wiggle)?

Can you have your robot line follow for a particular distance?  
(Hint: Use Port View to measure the distance)

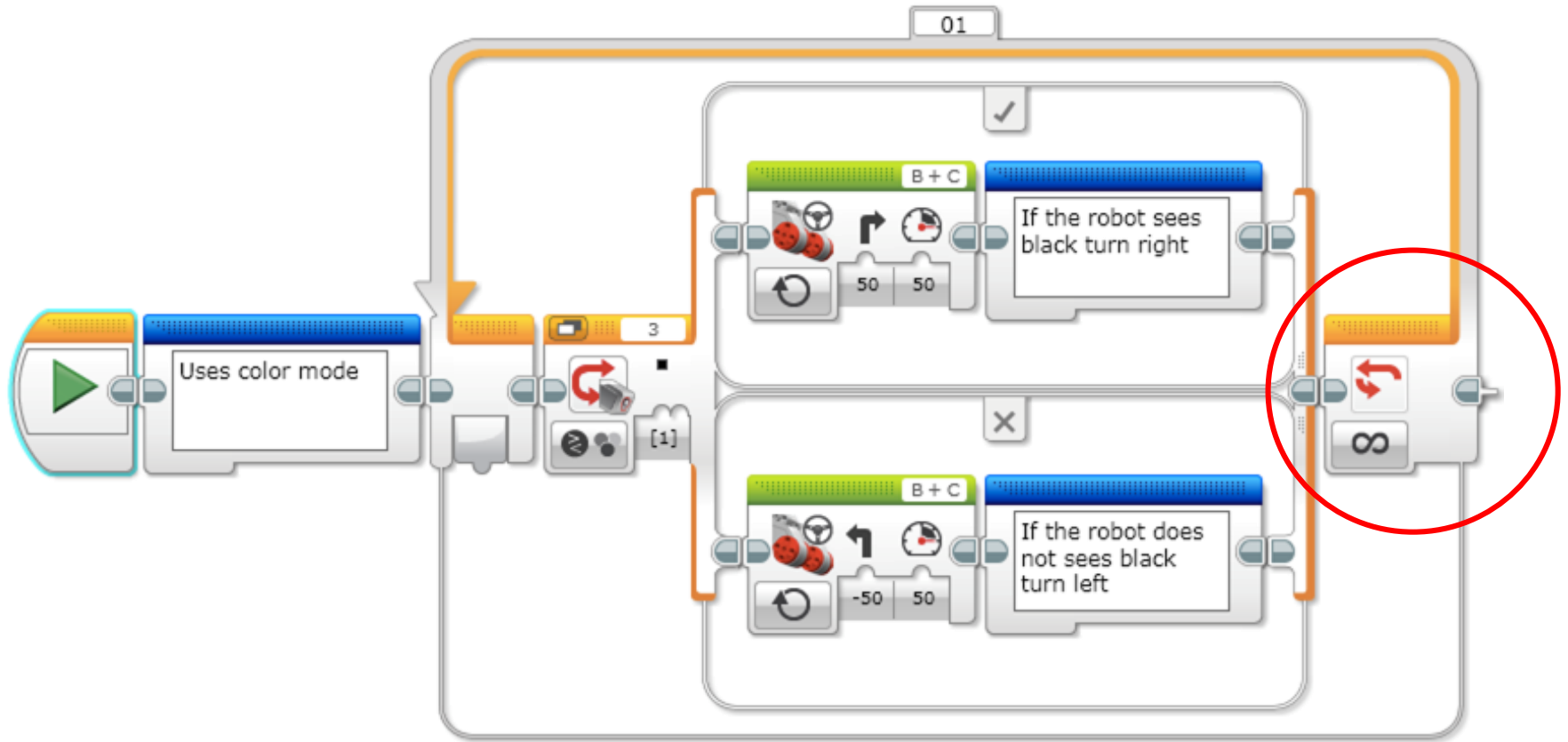
Can you have your robot line follow until it finds an intersection? (Hint: Use the second color sensor)

# LINE FOLLOWING CHALLENGE SOLUTION



Q. Does this program follow the Right or Left side of a line?  
A. The robot is following the Right Side of the line.

# CHALLENGE 1 SOLUTION



Q. This line follower goes forever. How do we make this stop?  
A. Change the end condition on the loop.

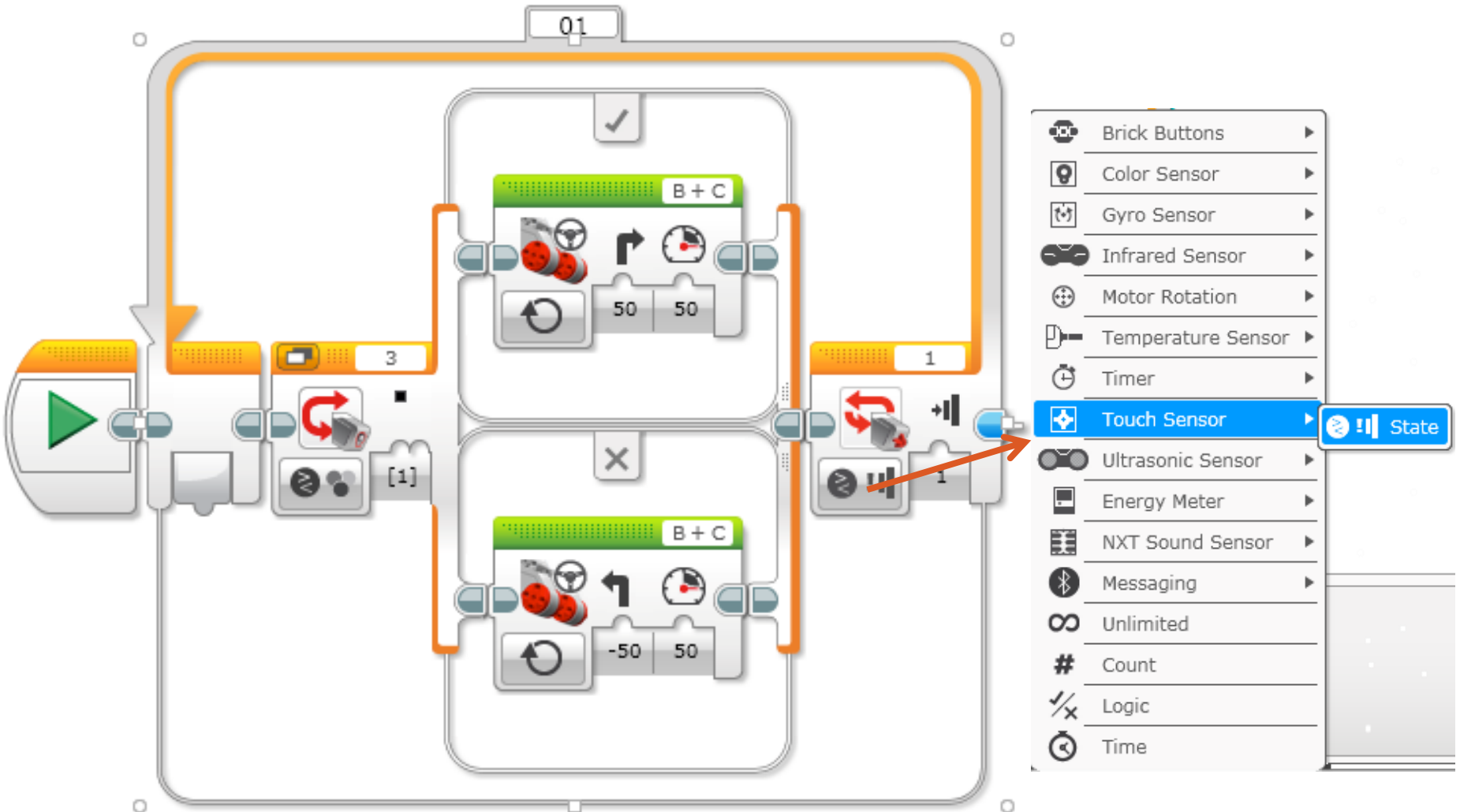
# LINE FOLLOWER CHALLENGE 2

Part 1: Make a line follower that stops when you press the touch sensor

Part 2: Make a line follower that stops after it travels a particular distance

Hint: You will have to use PORT VIEW in your brick to measure the distance first

# CHALLENGE 2 SOLUTION: SENSOR





# CHALLENGE 2 SOLUTION: PARTICULAR DISTANCE

