LEGO MINDSTORMS PROGRAMMING CAMP

Robotics Programming 101
Camp Curriculum

Instructor Notes

- Every day of camp, we started with a short video showing FLL robots, real robots or something relevant to the day's lesson
- We switched between using powerpoint slides and showing the actual EV3 software environment
- We created some add-on challenges for students who finished early
- We created a new project each day
- We did not show the challenge solution slide unless someone needed help.

What is a robot?

Robots are machines in factories, play soccer, are in research labs, behave like animals….

We will use the LEGO MINDSTORMS

Lego Mindstorms Robots

You can place a video of your choice here to introduce FLL/Lego robotics

Dos and Do Nots this week

- Share the laptop and the robot
- Work together in your group
- Help each other finish each challenge
- Lift the robot by the handle
- Take Turns
- Do not drop, throw, push on the robot – they are very expensive
- Treat your computer well – they are expensive

What will we do this week?

- Learn basic EV3-G Programming
- Make a Mindstorms robot follow your instructions
- Learn how to make your robot move straight, turn, repeat actions, choose between actions and follow lines
- Have FUN with robots and get inspired to learn more!!

Instructor Notes

- We use the CyberBot NXT design – please note the location of the color sensor
- We use the NXT brick but program with the EV3 software. The NXT brick has less memory.
  - You will have to delete the files often (at the end of every session)
  - Students who create and use a lot of sound files will fill in the brick faster.
- We asked students to make a new project file each day for their programs.

BEGINNER EV3 PROGRAMMING
Lesson: Introduction

Topics Covered:
NXT Basics
Introduction to the NXT Brick and EV3 Software

By: Droids Robotics

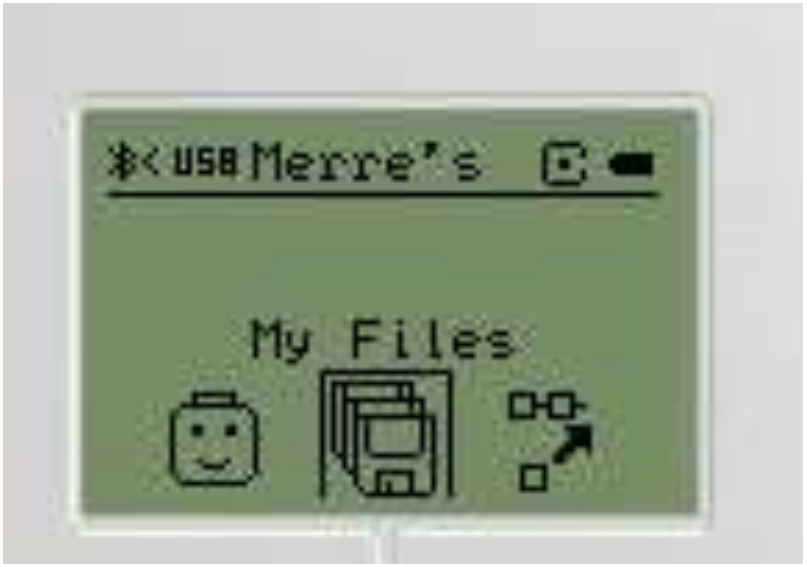
Lesson Objectives

1. Learn how the NXT brick operates
2. Learn about the main components of the EV3 software

The “Brick” Buttons

- 1 = Left, Right → Navigate menus
- 2 = Center Button →
  Select options
  Run Program
  Turn robot on
- 3 = Back →
  Undo
  Stop Program
  Turn robot off

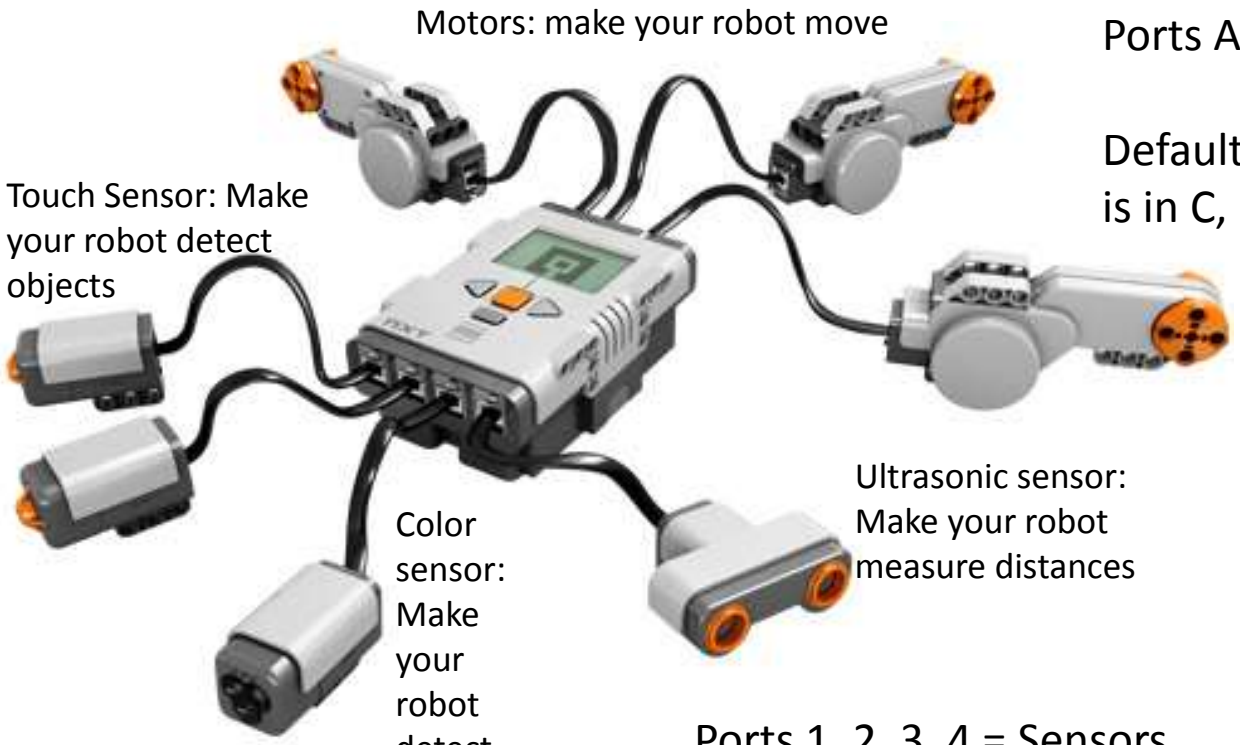
The “Brick” Screen



Menus on Screen

1. **My Files**
   Find programs you downloaded
2. **Use left/right**
   buttons to find other menus such as “View”

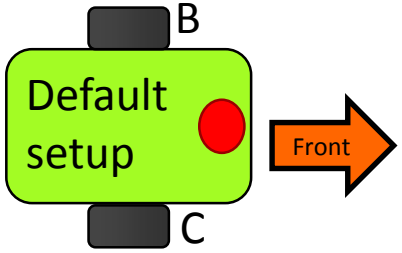
Ports, Sensors, motors



Ports A, B, C = Motors

Default setup assumes Right Motor is in C, Left Motor is in B

Ports 1, 2, 3, 4 = Sensors



EV3 Software

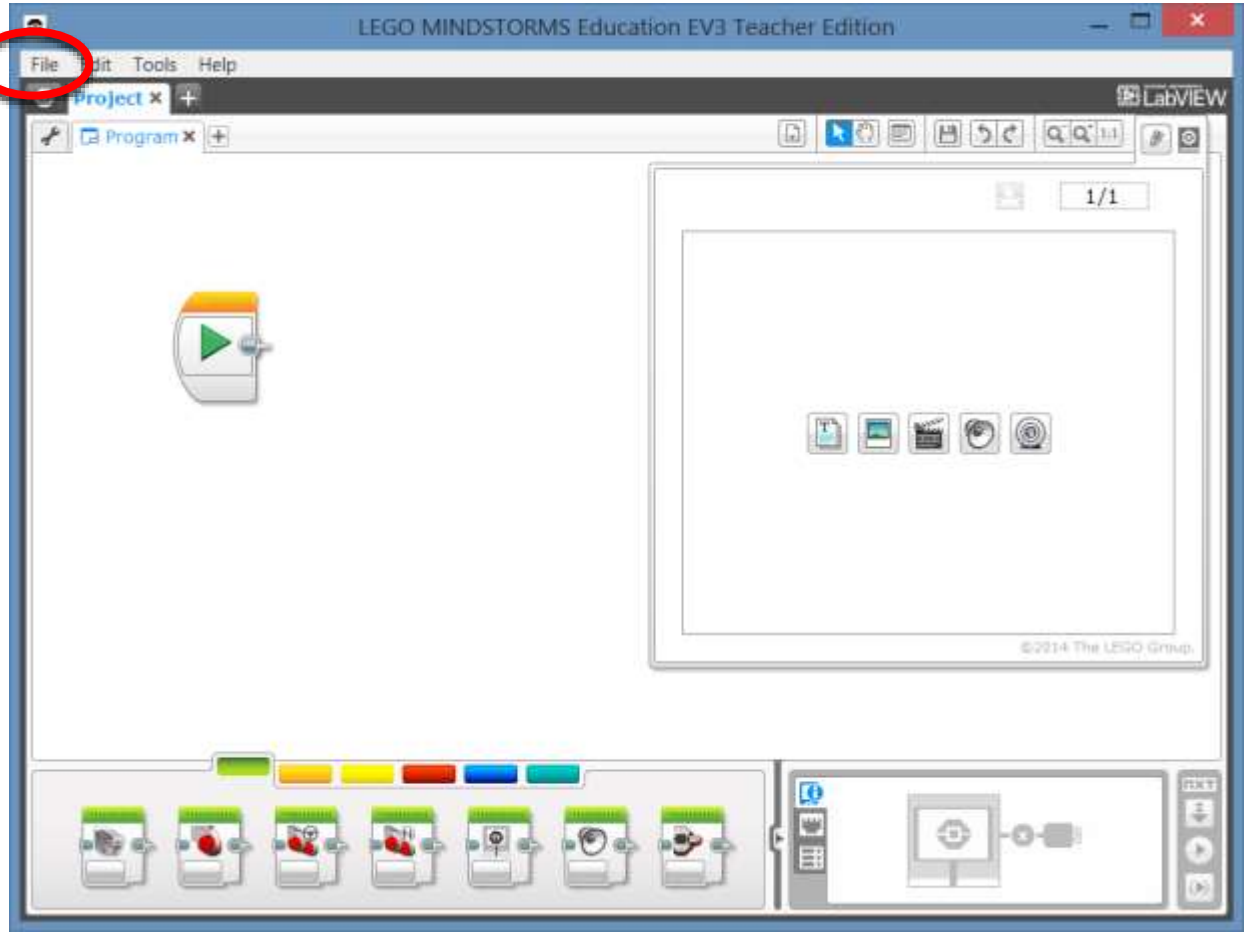
EV3 Software: Saving your project

File..Save Project As

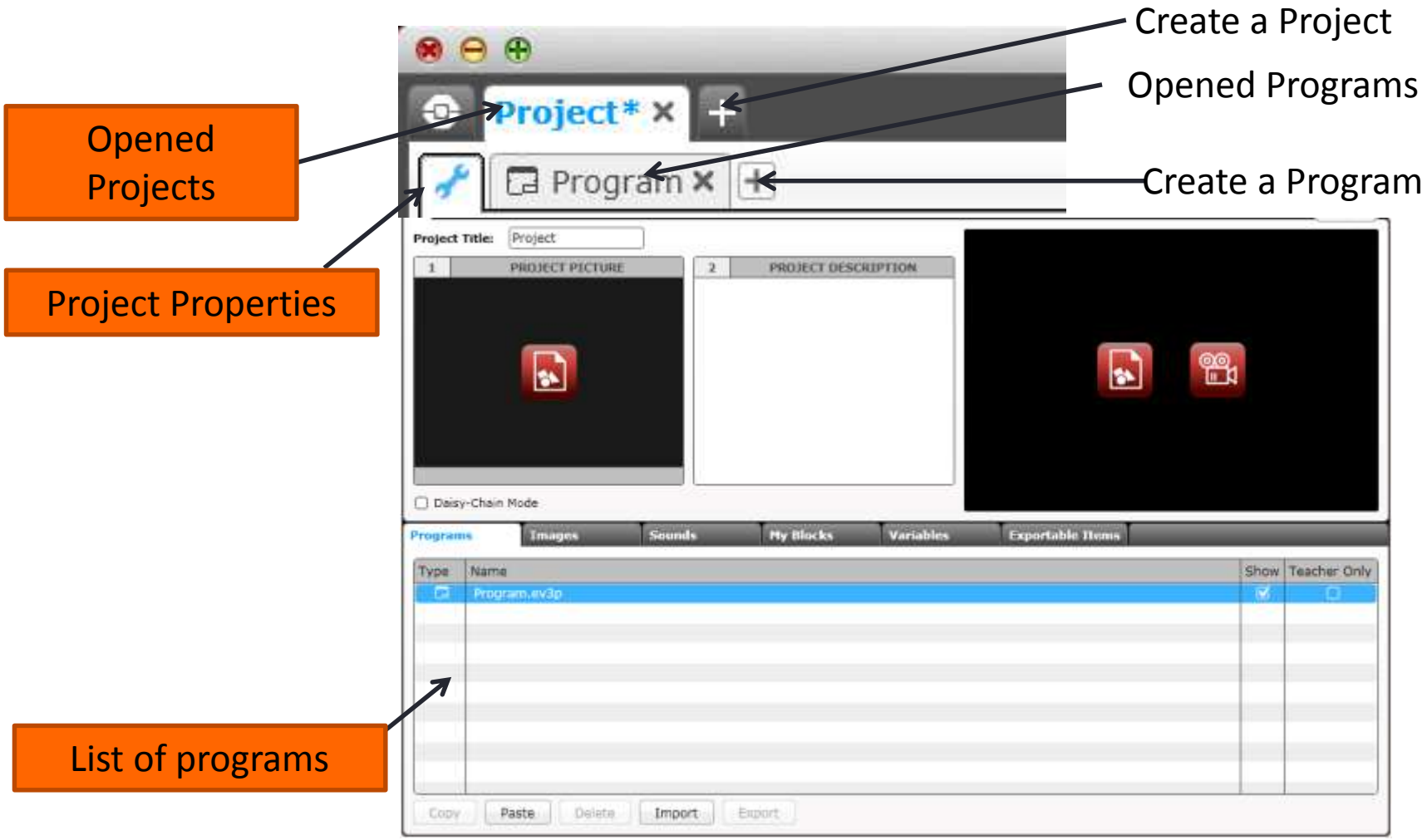
Save project on desktop as “CampPM1”

Each project can contain multiple programs.

You will create a new project for each day this week.



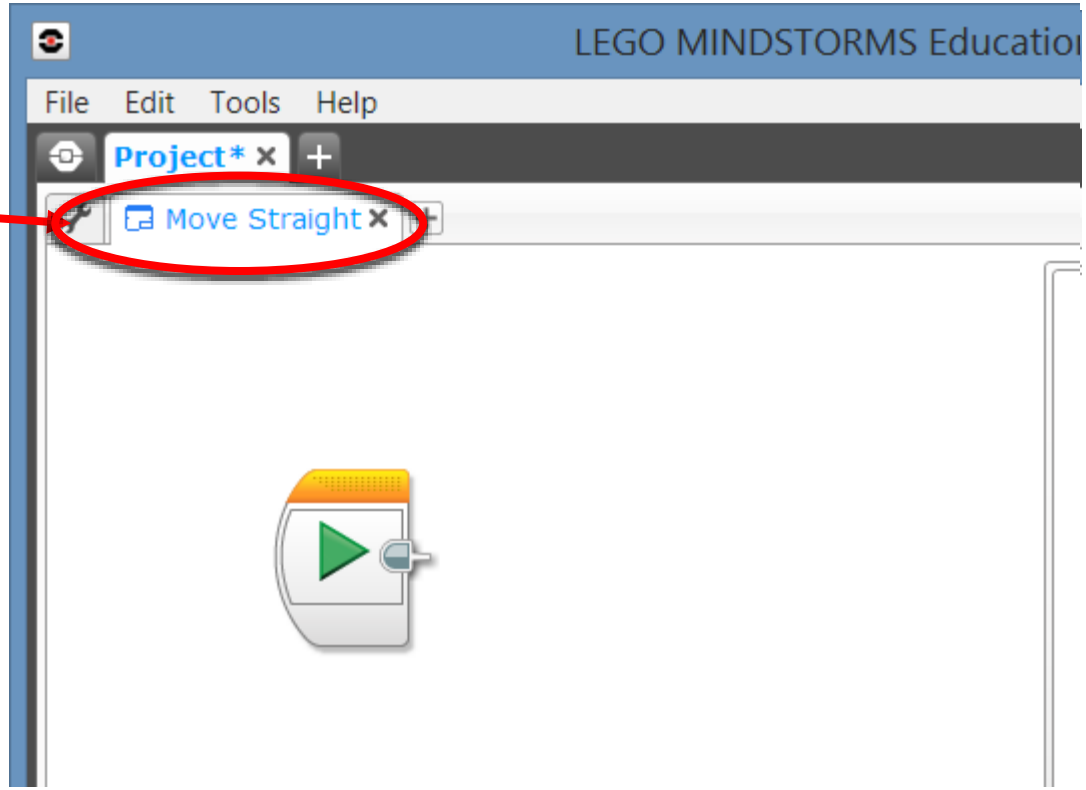
EV3 Software: Starting A new program



EV3 Software: Naming your program

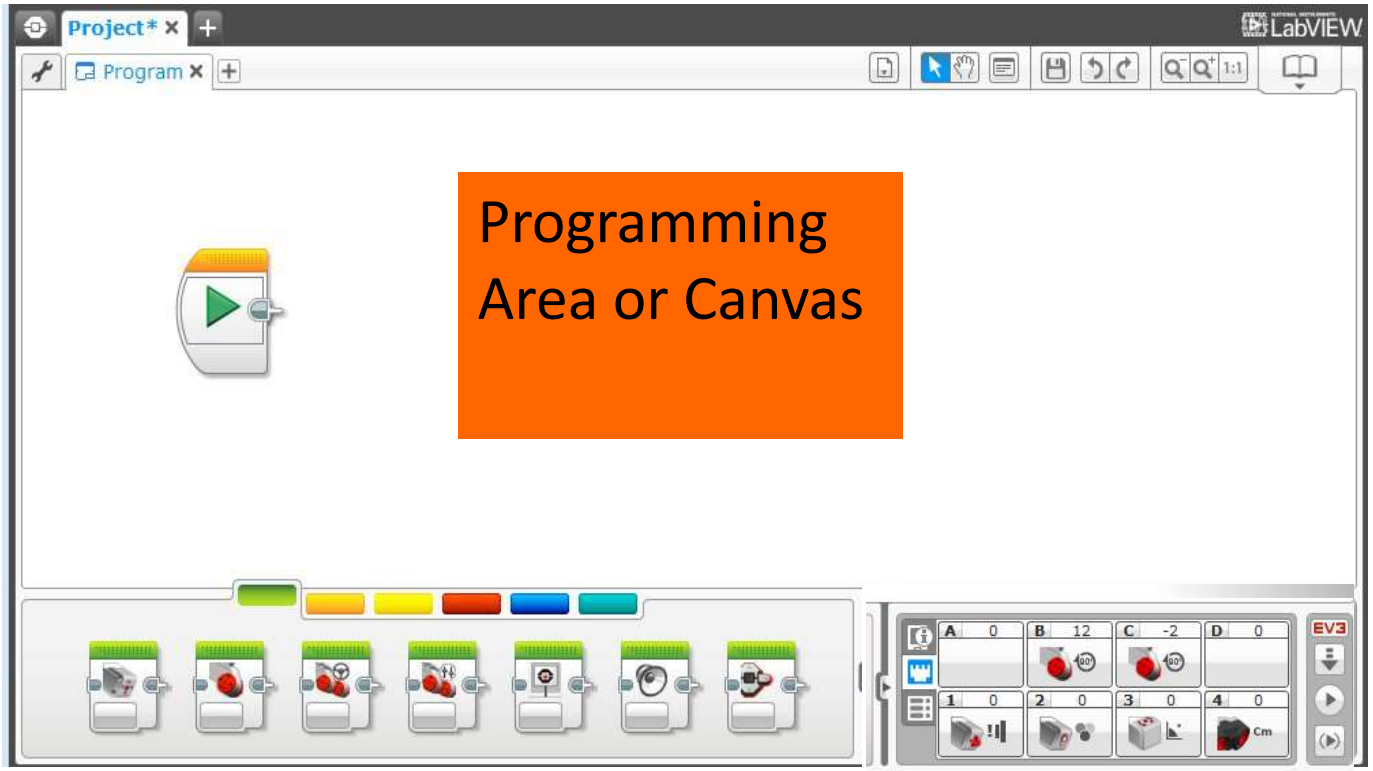
Double click on Program

Change name of program to “Move Straight”



EV3 Software: Programming screen

Click on a program tab to bring up the programming canvas.

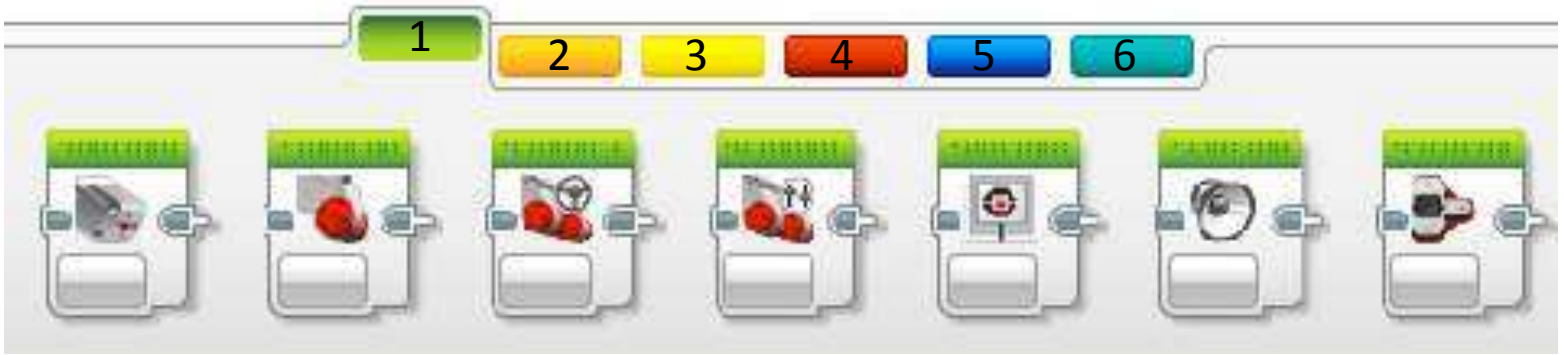


EV3 Blocks: Colored Tabs

ACTION BLOCKS
Move, Large & Medium Motor, Display… 1

FLOW BLOCKS
Start, Wait, Loop, Switch, Loop Interrupt 2

SENSOR BLOCKS
Brick Buttons, Gyro, Color, Ultrasonic 3

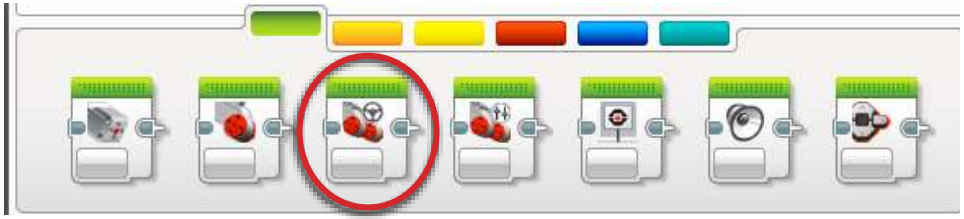


DATA OPERATIONS
Variables, Array, Logic, Math, Compare… 4

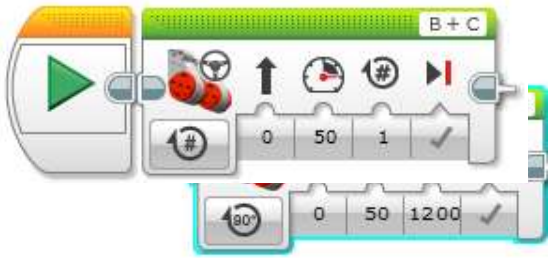
ADVANCED BLOCKS
Data Logging, Unregulated Motor… 5

MY BLOCKS
Custom Blocks you create 6

Adding Blocks To Your Program



STEP 1: Green Block Tab, Click and hold any block and drag to programming area



STEP 2: Drop next to the Start Block (green arrow)
(See animation)

Beginner Lesson: Moving Straight

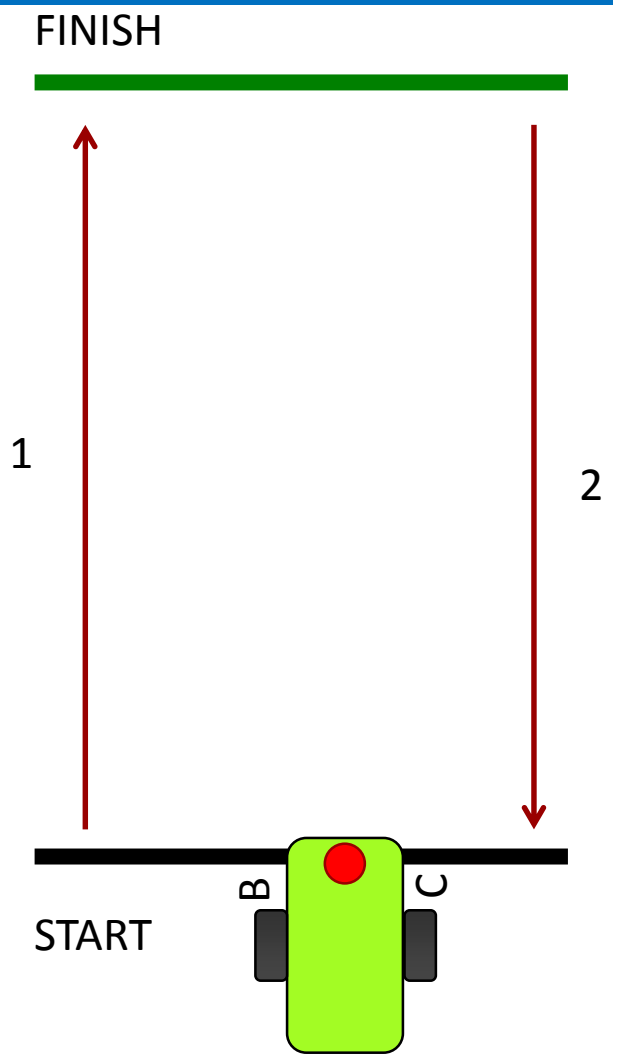
By: Droids Robotics

Lesson Objectives

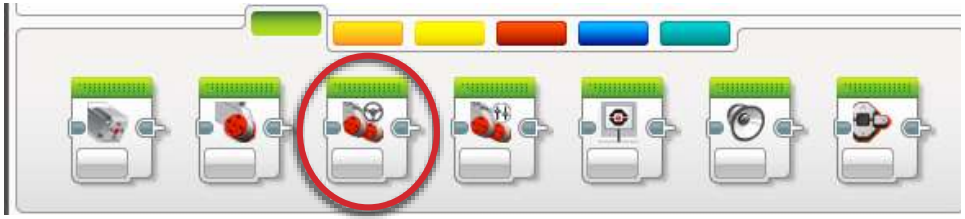
1. Learn how to make your robot go forward and backwards

2. Learn how to use the Move Steering block

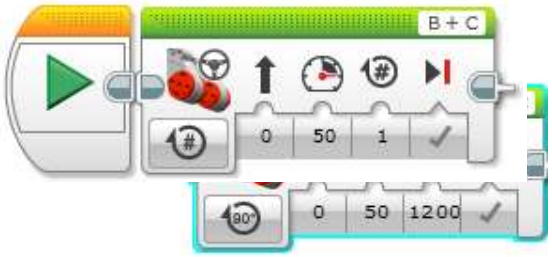
3. Learn how to read sensor values using View



How do you Move Straight?



STEP 1: Green Block Tab, Click and hold Move Steering and drag to programming area

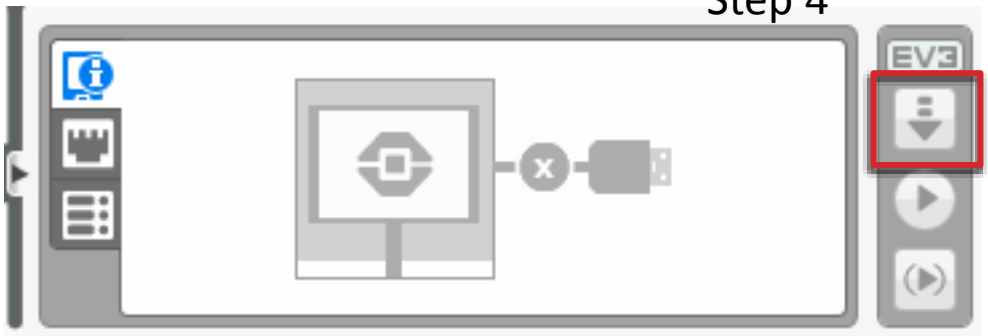


STEP 2: Drop next to the Start Block (green arrow)
(See animation)

CHALLENGE 1: Move Straight

Step 3

Step 4



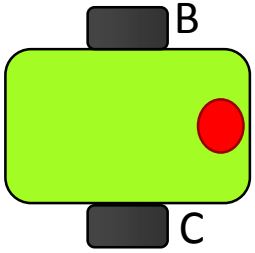
STEP 1: Green Block Tab, Click and hold Move Steering and drag to programming area

STEP 2: Drop next to the Start Block (green arrow)

STEP 3: Connect USB cable to NXT and Laptop.

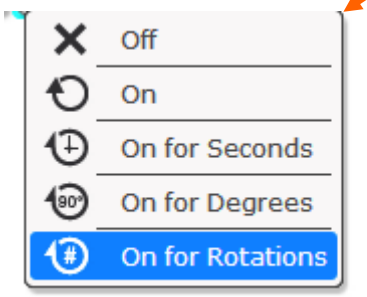
STEP 4: Download to NXT

Move STEERING Block



Brake/ Coast

Mode of operation

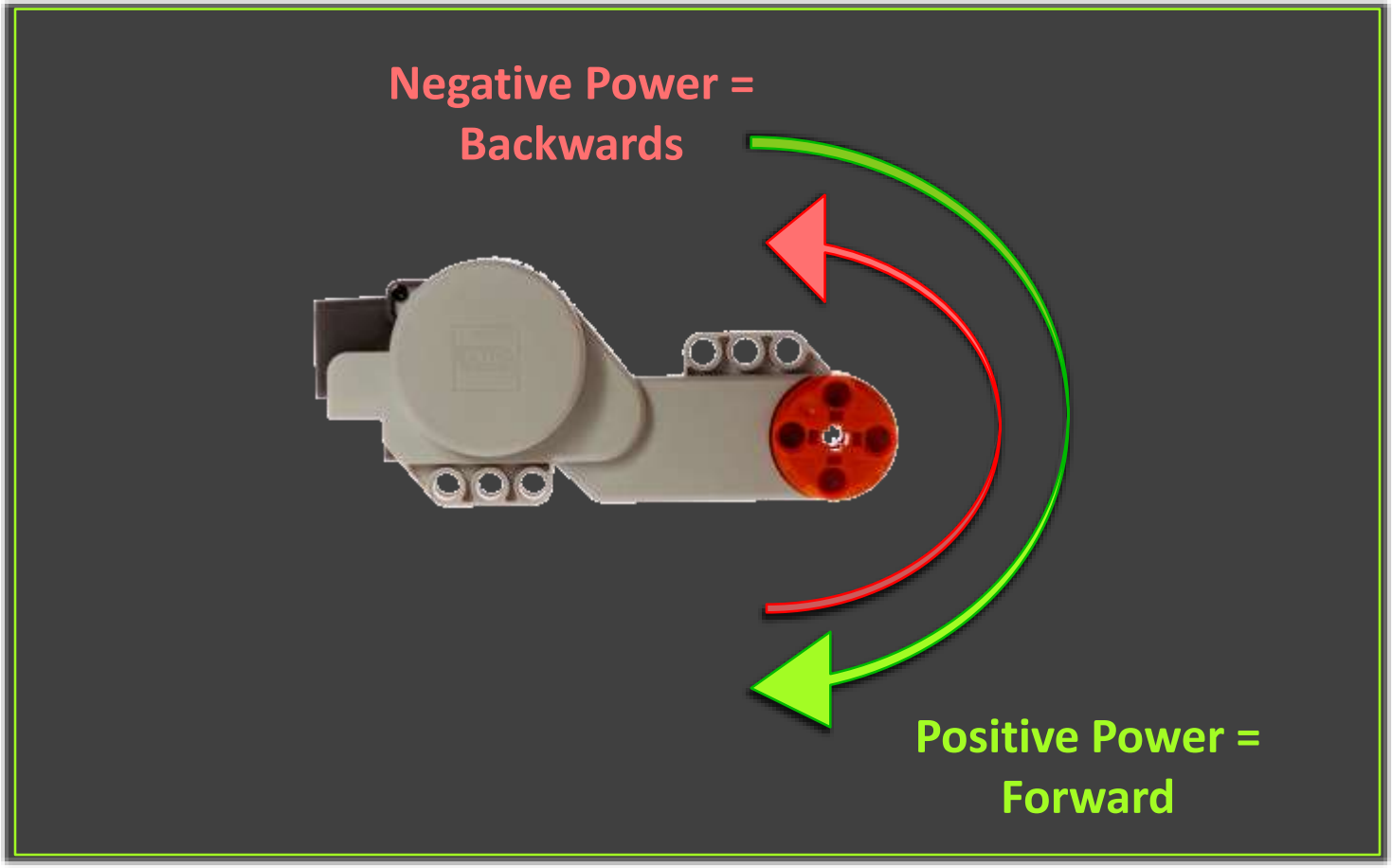


Steering: Straight or turn

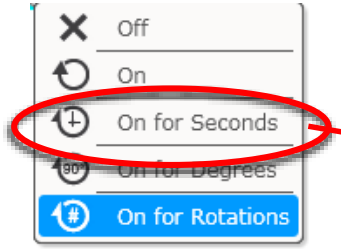
Power/Speed

Duration/Distance

NEGATIVE & POSITIVE POWER: BACKWARD & FORWARD



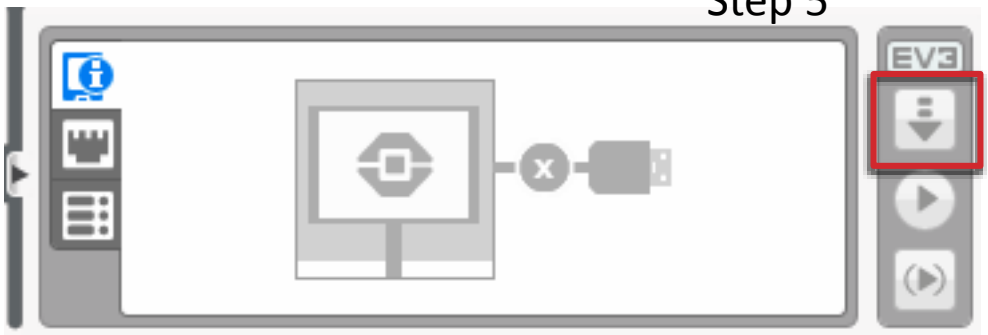
CHALLENGE 2:
Move Straight (3 SECONDS)



Step 3

Step 4

Step 5



STEP 1: Green Block Tab, Click and hold Move Steering and drag to programming area

STEP 2: Drop next to the Start Block (green arrow)

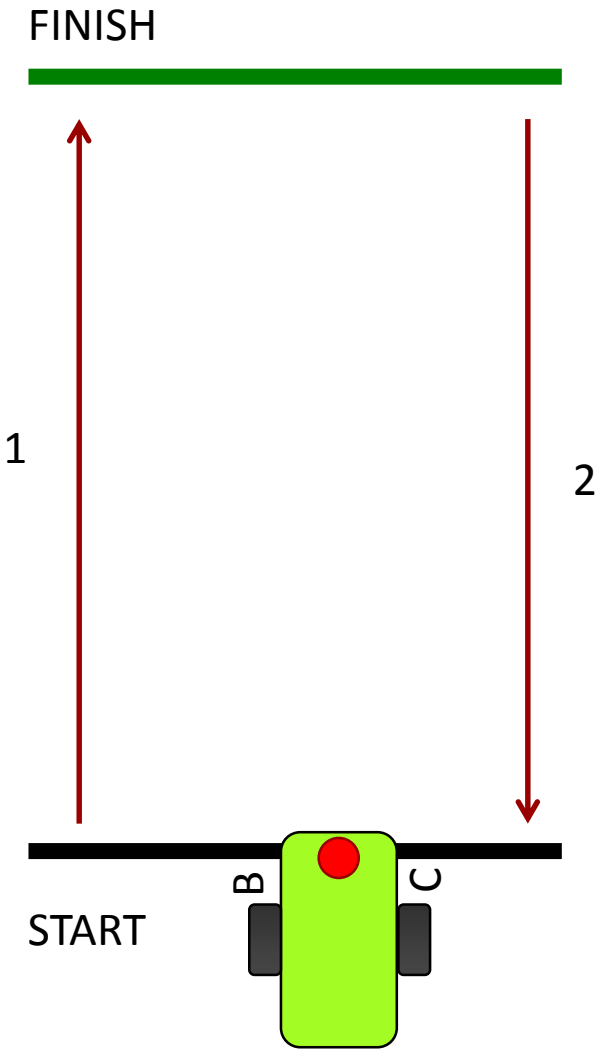
STEP 3: Select Options. Move “3 Seconds”

STEP 4: Connect USB cable to NXTand Laptop.

STEP 5: Download to NXT

Move Straight: Seconds vs. degrees vs. rotations

- CHALLENGE: Move your robot forward from the start line to the finish line (1) and back to the start (2).
- Try mode SECONDS, DEGREES or ROTATIONS and adjust duration/distance
- Try different speeds



MOVE STRAIGHT discussion

Did you guess and check a lot?

Yes. Programming with seconds, rotations and degrees using guess and check takes a lot of time and effort.

Did changing the speed matter?

Yes. When you move in seconds your speed will matter.

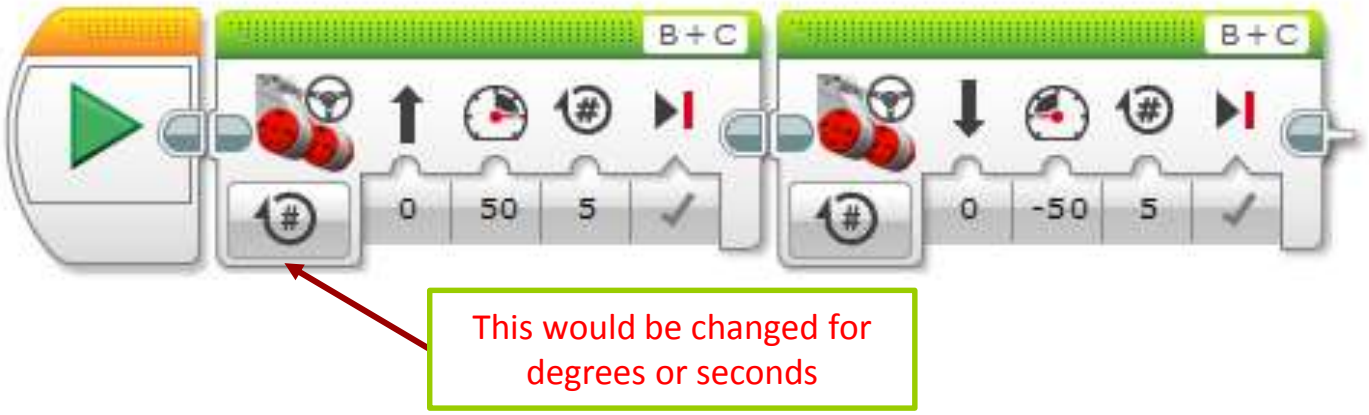
Do you think the wheel size will matter? Why?

Wheel size affects degrees/rotations.

Do you this the battery level will matter? Why?

When you move in seconds, battery levels change the power.

CHALLENGE SOLUTION



- There is a better way (go to next slide) to solve this challenge

SOLUTION: Use View

- Try “view” on brick (instruction below)
  - Move your robot with your hand from your start line to your end line
  - Read how many degrees your robot moved
  - Use this number in the Move Steering Block to move the correct distance.

Browse menu using arrows.

Look for the 'View' option. Select with orrange button.

Browse for ‘Motor degrees’ option

Sepecify the Port on NXT that is connected to the sensor.

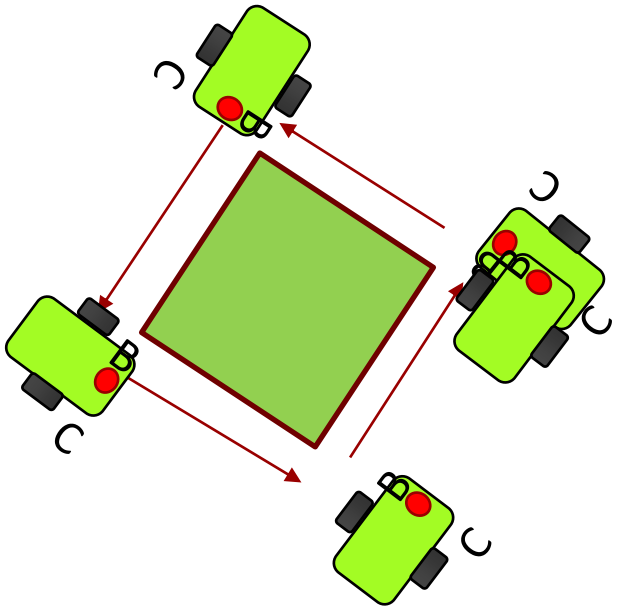
Have fun with your digital ruler!

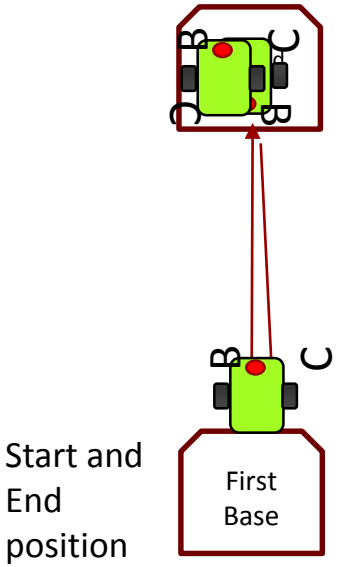
Beginner Lesson: Turning

By: Droids Robotics

Lesson Objectives

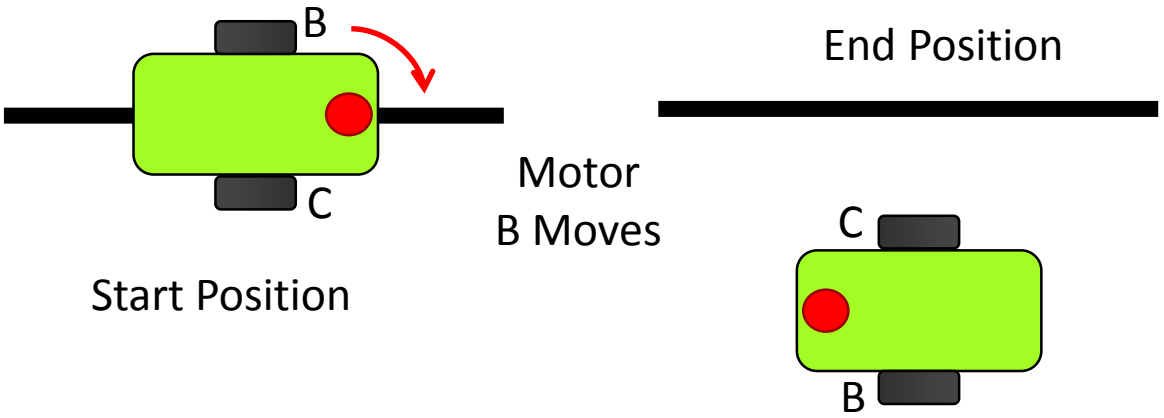
1. Make a robot baseball player
2. Learn how to make the robot turn



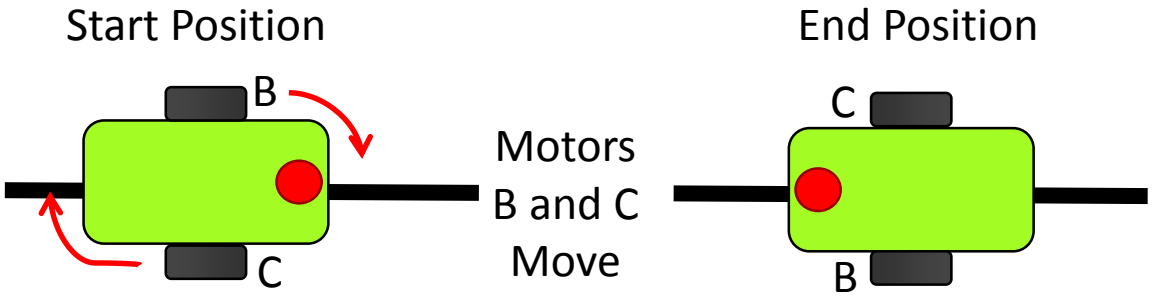


PIVOT Vs. SPIN Turns

180 Degree Pivot Turn



180 Degree Spin Turn

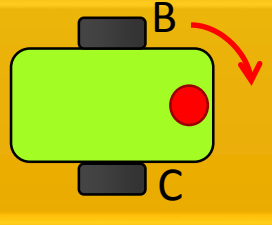
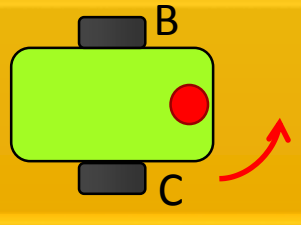
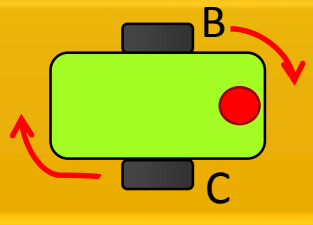
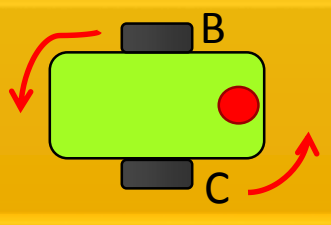


Notice where the robot ends in both pictures after a 180 degree turn.

In the Spin Turn, the robot moves a lot less and that makes Spin Turns are great for tight positions. Spin turns tend to be a bit faster but also a little less accurate.

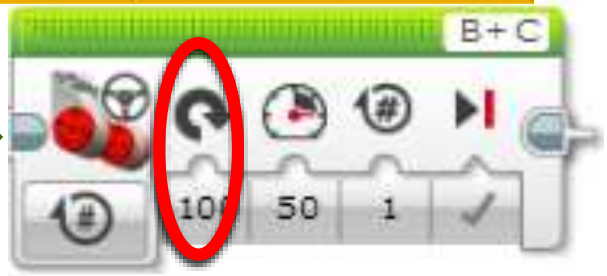
So when you need to make turns, you should decide which turn is best for you!

How to Make Pivot and Spin turns

Move Steering or Large Motor			
Large Motor B	**Large Motor C**	**Move Steering Steering=100**	**Move Steering Steering=-100**
			
Pivot Turn Right	Pivot Turn Left	Spin Turn Right	Spin Turn Left

Large Motor Block

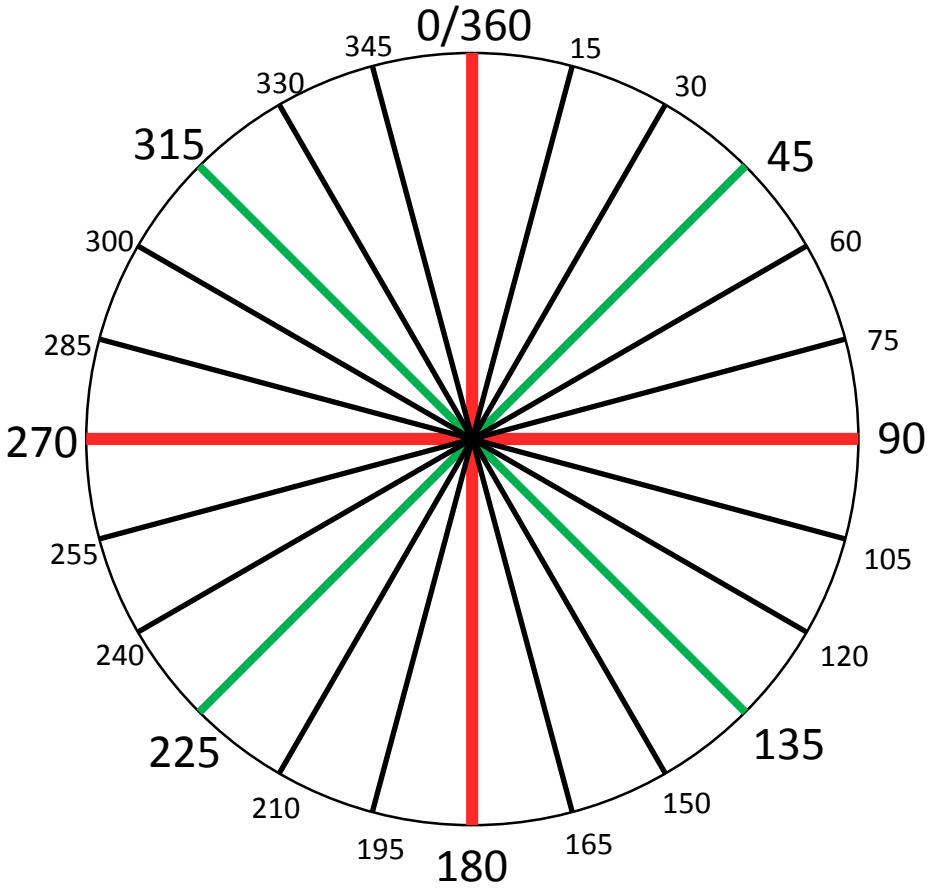
Move Steering Block



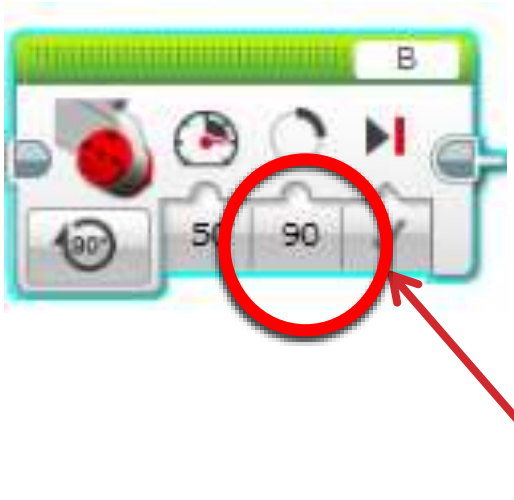
Change Steering value here

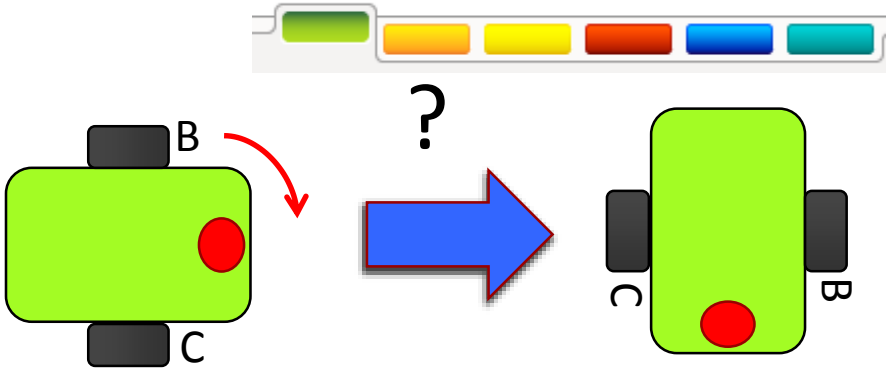
Measuring Turns

- Turns are measured in degrees



Making a Pivot turn for 90 Degrees



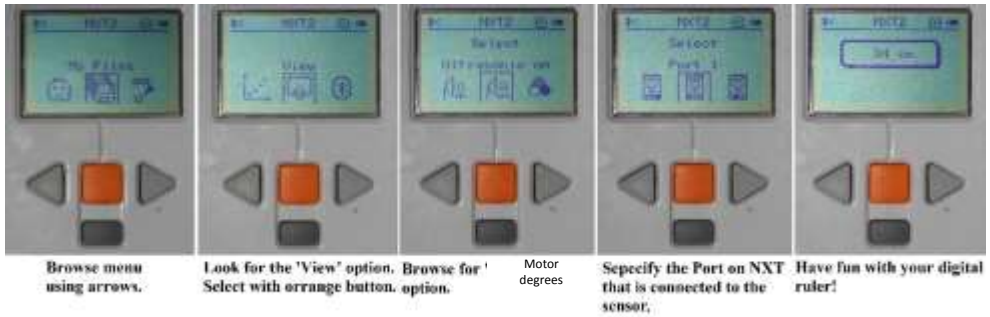


Program your robot to turn 90 degrees....Does the robot actually turn 90 degrees if you just pick 90 degrees for distance?

Ans. NO!
This makes the wheel turn 90 degrees.
The correct solution on next page

How do you make the robot turn 90 degrees?

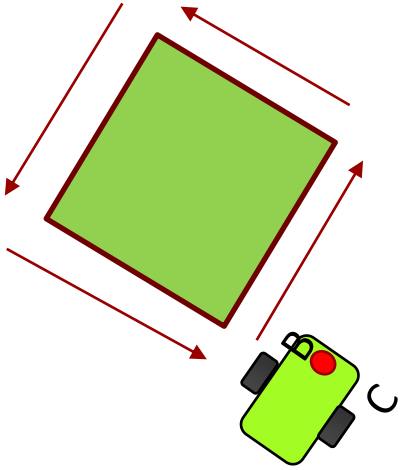
- Ans. Try using the port view to measure the turn and then input the correct number of degrees.



TURNING CHALLENGES

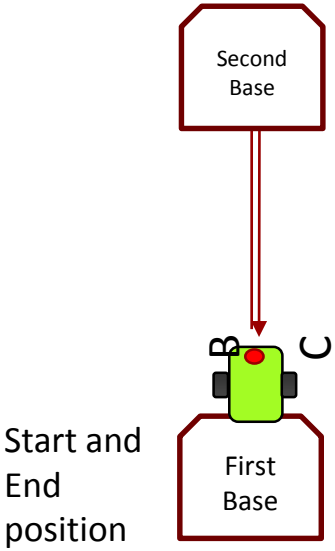
Challenge 1

- Your robot is a baseball player who has to run to all the bases and go back to home plate.
- Can you program your robot to move forward and then turn left?
- Use a square box or tape



Challenge 2

- Your robot baseball player must run to second base, turn around and come back to first.
- Go straight. Turn 180 degrees and return to the same spot.



Class Discussion Guide

- Did you try PIVOT and SPIN turns? What did you discover?

  Pivot turns were fine for Challenge 1, but for Challenge 2, if we used Pivot turns, we were farther away from the base.

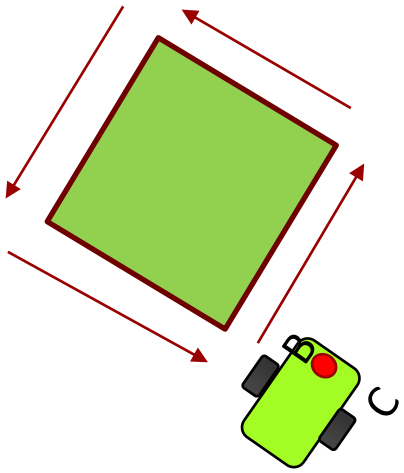
- What situations would one work better than the other?

  Spin turns are better for tight turns (places where there is not enough space) and you stay closer to your original position.

CHALLENGE SOLUTIONS

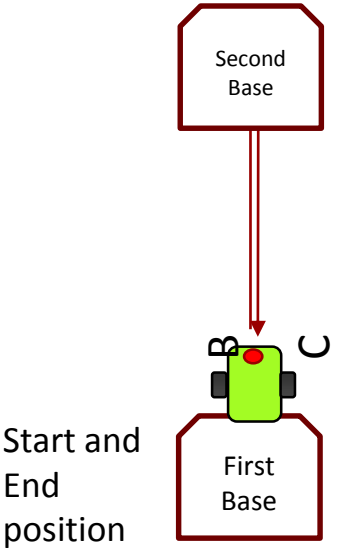
Challenge 1

- You probably used a combination of move steering to go straight and do pivot turns to go around the box.

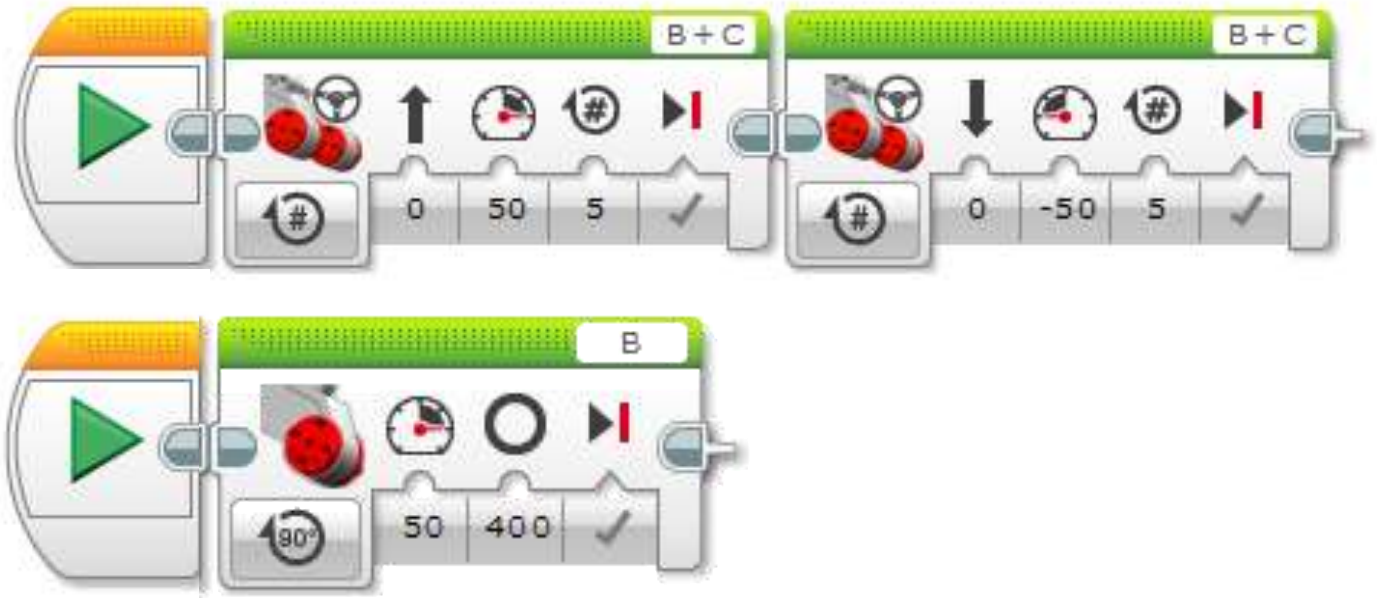


Challenge 2

- You probably used a spin turn because it is better for tighter turns and gets you closer to the starting point!



Review



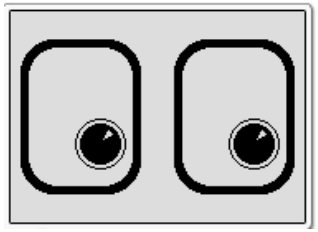
- What do these programs do?
- What are the names of these blocks

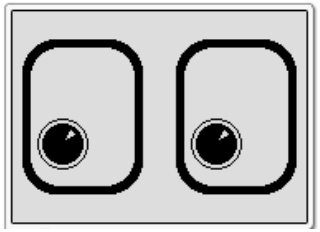
Beginner Lesson: Display Blocks

By: Droids Robotics

LESSON OBJECTIVES

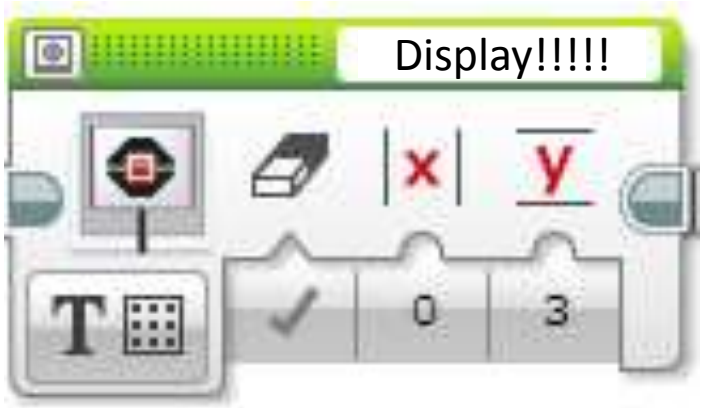
1. Make the robot display some fun faces while moving!
2. Learn to use the Display Block to display text and images
3. Understand why the Display Block can be useful in programming





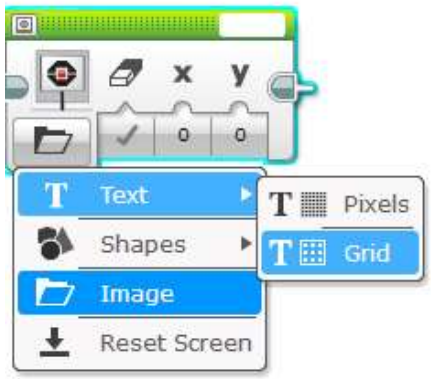
Display Block

- The Display Block to show information and pictures on the screen
- You can control the location and size of text
- You can use this same block to display sensor readings and instructions.
- Located in Green Tab



More on Display Blocks

- Two modes to display
- Pixel mode (Use for displaying images or text)
  - Gives fine control over placement of image/text
- Grid mode (Easier to use, only works for text mode)
  - Accuracy of a single “line”
- Click on top left of block to preview

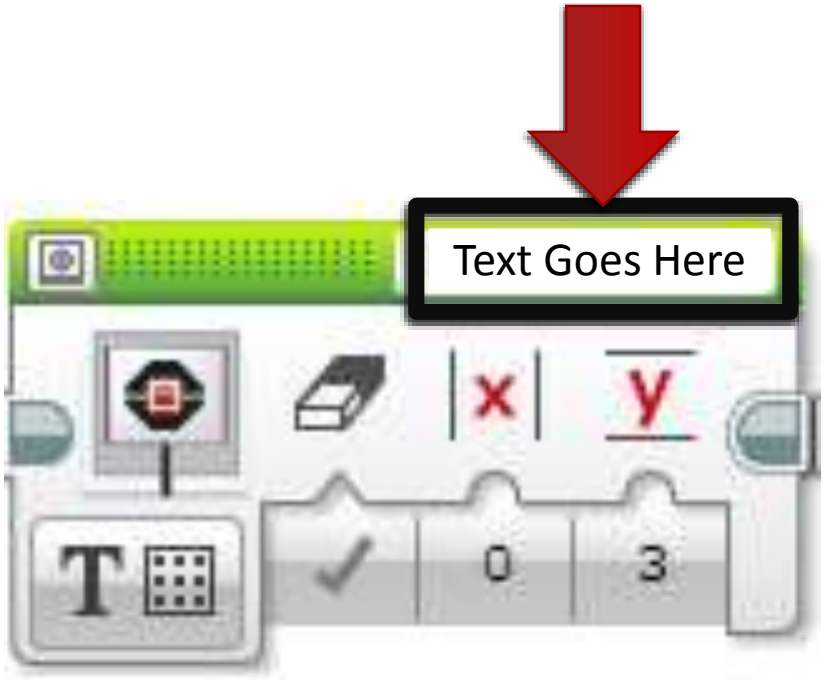


Displaying Text in Grid Mode

Step 1:
Pick Display Block

Step 2:
Click on “Switch Modes” icon and hover over “text”. Then click on “grid”. The icon will change into a square with dots.

Step 3:
Use the box on the top right to enter the text you want to display



DISPLAY BLOCK: CHALLENGE 1

- Can you write a program to display text in the middle of the screen?
  - Display “Hello World”
- How do we get the text to stay on the screen longer?
  - Need to make the program “wait” before exiting

Wait Blocks

- Wait blocks make your program pause for some time (or until something happens) before moving to the next step
- For now, we are just going to wait for time

Choose Time for mode

Enter “3” for seconds to wait

DISPLAY BLOCK: CHALLENGE 2

- Can you write a program to display text in the middle of the screen?
  - Display “Hello World”

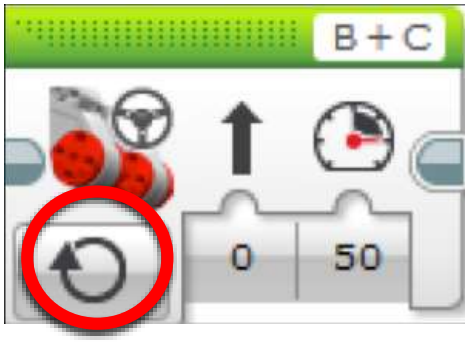
- Make the display block run for 3 seconds

DISPLAY BLOCK: CHALLENGE 3

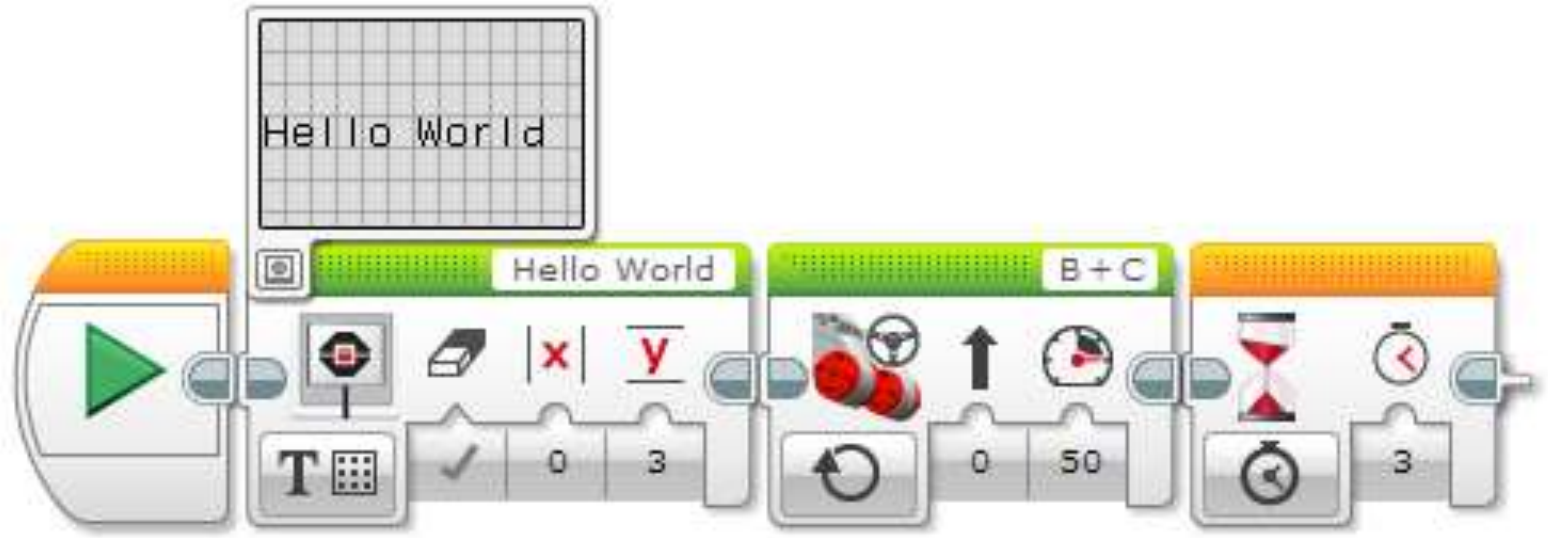
- Can you write a program to display text in the middle of the screen?
  - Display “Hello World”
- Make the display block run for 3 seconds
- Can you also move while doing this?

A tip for Move Steering Blocks With Wait Blocks

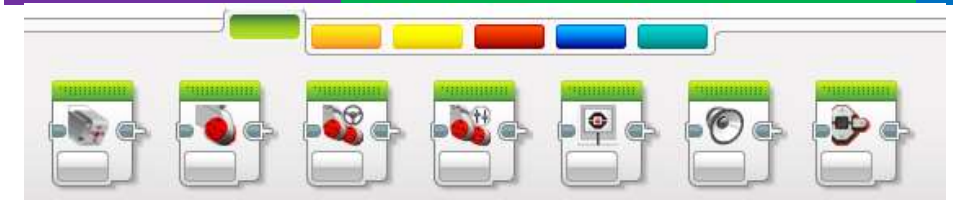
- Leaving the motor “on” and “off”
- Why use the “on” instead of “degrees”?
  - May want the program to do other tasks while moving



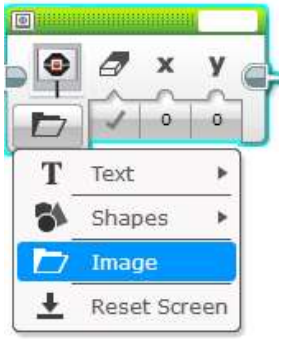
Challenge 3 solution



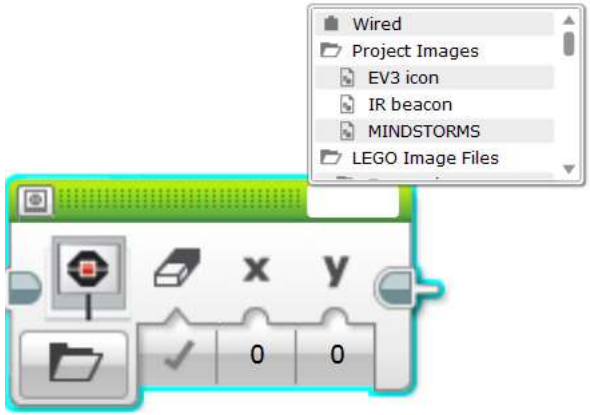
Displaying an image in PIXEL Mode



Step 1:
Pick Display Block



Step 2:
Click on the “Select Mode” which has a folder icon and pick “image”

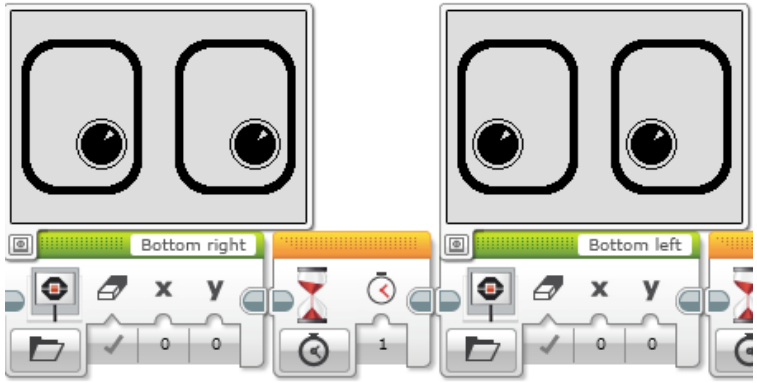


Step 3:
Use the empty box on the top right to pick the image you want to display

Note: the image you choose may not show up correctly on a NXT

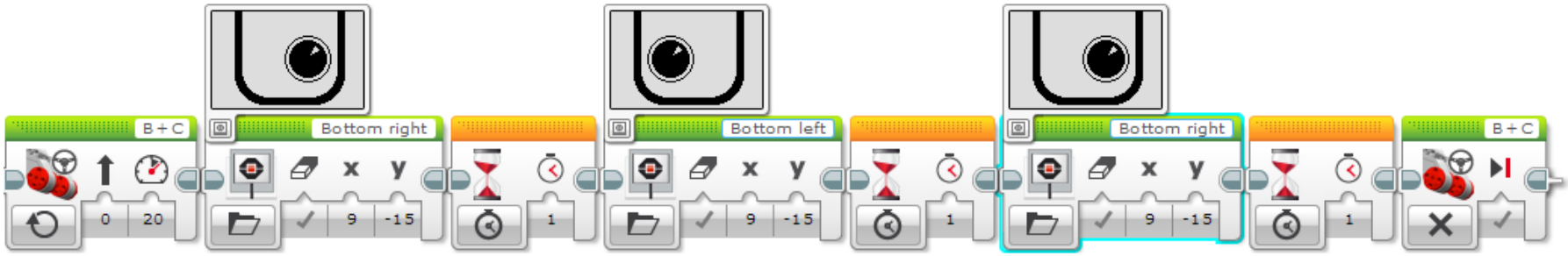
DISPLAY BLOCK CHALLENGE 4

- Can you display eyes on the screen while moving? Alternate eyeballs that look left and right.
  - Use the Display Block, Motor On and Wait Block



- Feel free to have fun with this challenge and make it yours!

CHALLENGE 4 Solution



Motor On

Display Block

Wait Block

Motor Off

BEGINNER EV3 PROGRAMMING: Lesson Touch Sensor

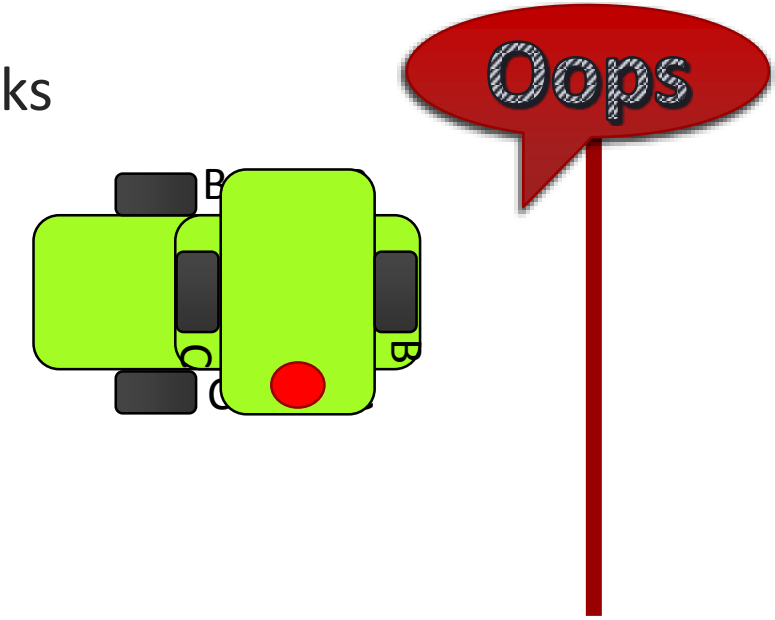
By: Droids Robotics

Lesson Objectives

1. Make a robot that wanders the room – bumps into walls and turns around

2. Learn how to use the Touch Sensor

3. Learn the difference between the Wait For Block and the Sensor Blocks

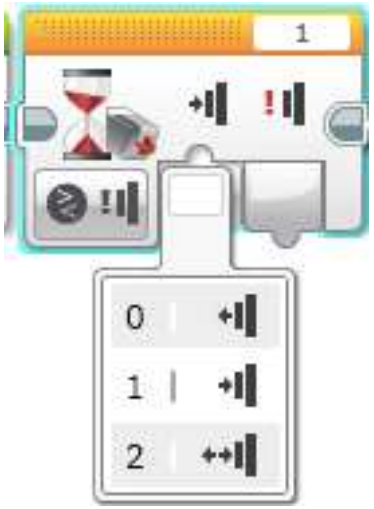


What is a sensor?

- A sensor lets an EV3 program measure and collect data about is surroundings
- The EV3 sensors include:
  - Color – measures color and darkness
  - Ultrasonic – measures distance to nearby surfaces
  - Touch – measures contact with surface

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
  - Pressed and Released Just Before (Bumped)
- When might you use this sensor?
  - Useful for programming "moving until touch sensor is pressed/released/bumped"
  - You can also have your program start or stop when a touch sensor is pressed.

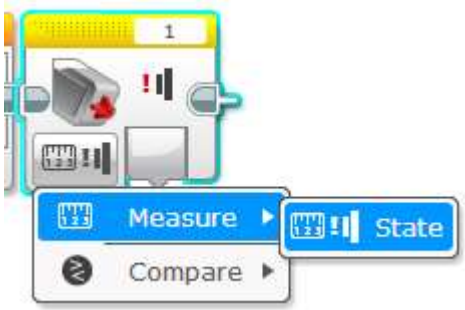


How do you program with the Touch Sensor?

There is a Touch Sensor Block in the Yellow Tab, but there is a Wait for Touch in the Orange Tab. What is the difference!!????!

Yellow Sensor Tab: Sensor Blocks

Used to Read and Compare Sensor Values



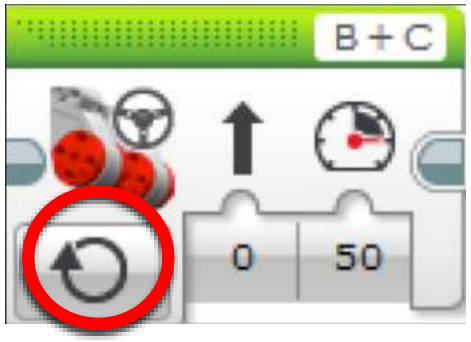
Orange Flow Tab: Wait for Block

Used to wait for a sensor reading (or time)

In this lesson, we will use the Wait For Block

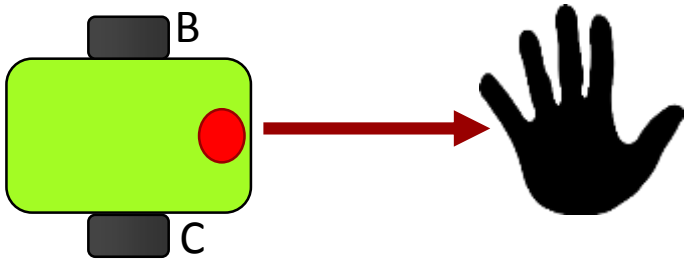
Reminder: A tip for Move Steering Blocks With Sensors

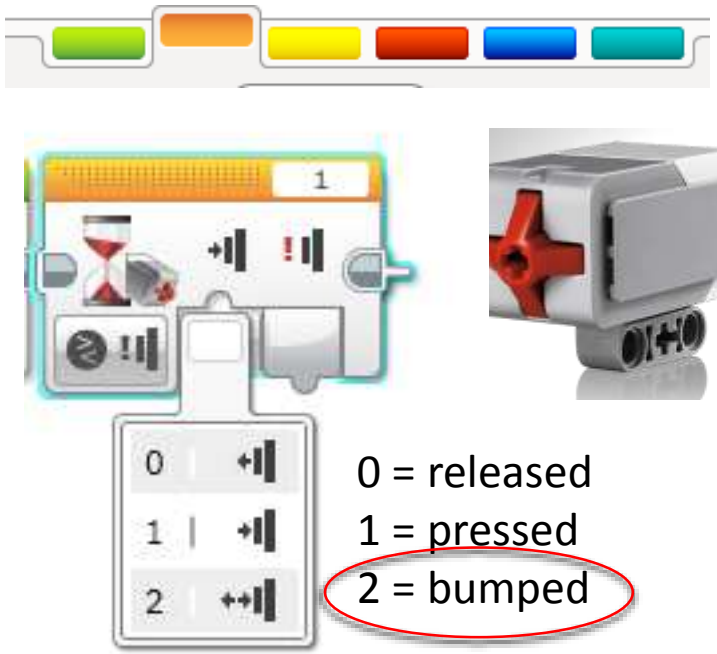
- Leaving the motor “on” and “off”
- Why use the “on” instead of “degrees”?
  - May want the program to do other tasks such as reading a sensor while moving



CHALLENGE 1

- Program your robot to move straight until you tap the sensor with your hand.





Hint: You will combine: Move Steering + Wait Block

Challenge 1 Solution

The goal of this program is to make your robot move straight until you touch the sensor with your hand.

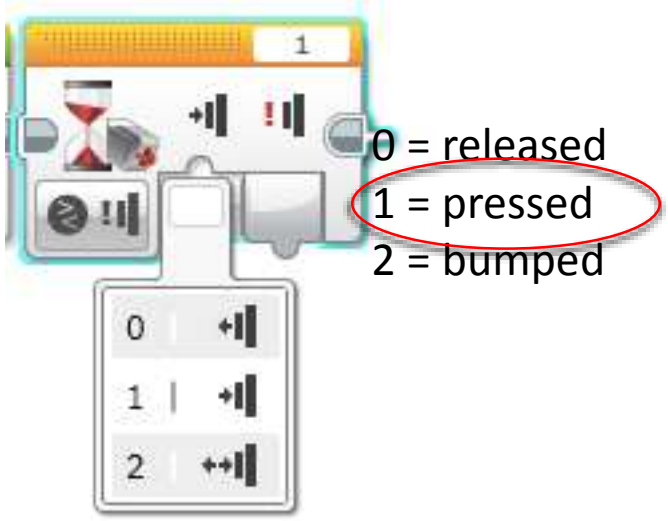
Set move steering block to "on"

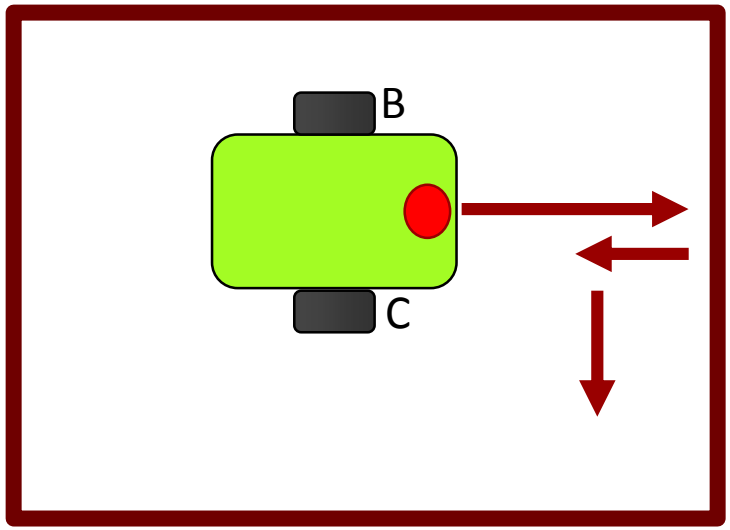
Set wait block to Touch-->Compare-->State

Set steering block to off with brake

CHALLENGE 2

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

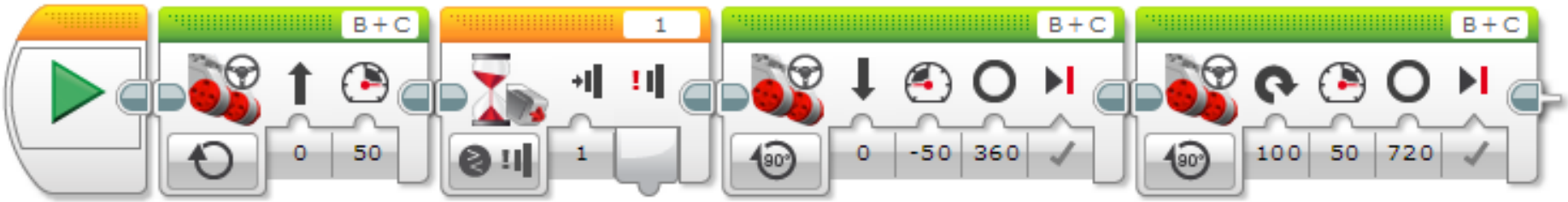




Hint: You will combine Move Steering + Turning + Wait Block

Challenge 2 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 wail block to Touch-->Compare-->State

Set move steering block to "degrees" and steering to 100. The 720 degrees value will have to be modified for your robot (You measured this in port view earlier beginner

DISCUSSION

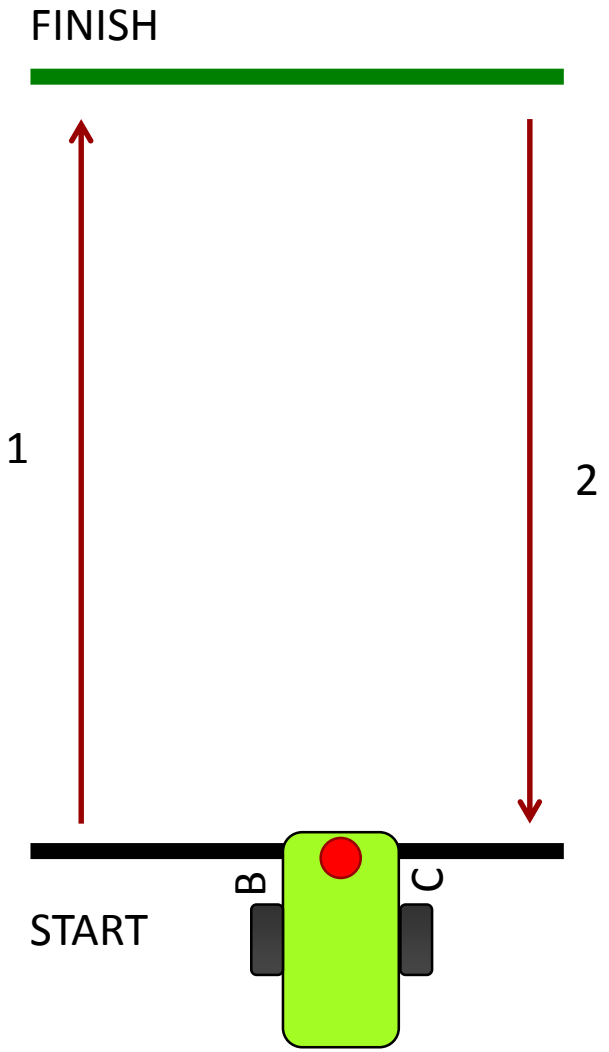
- Why did you use MOTOR ON for these challenges?
  - You want to read the sensor while the motor is on.
- Why do we use the WAIT FOR BLOCK in these challenges?
  - We need to program to wait for the correct reading
- What is the difference between PRESSED, RELEASED and BUMPED?
  - PRESSED = pushed in, RELEASED = not pushed,
    BUMPED = pressed and released recently
- What are some situations you might want to use each of these for?
  - PRESSED = running into a wall, BUMPED = tapped by hand
    RELEASED = no longer touching a wall

BEGINNER EV3 PROGRAMMING
Lesson: Color Sensor

By: Droids Robotics

Lesson Objectives

1. Make the robot move up to the green line without doing any measurement!
2. Learn how to use the Color Sensor
3. Learn about Coast and Brake



What is the color sensor?

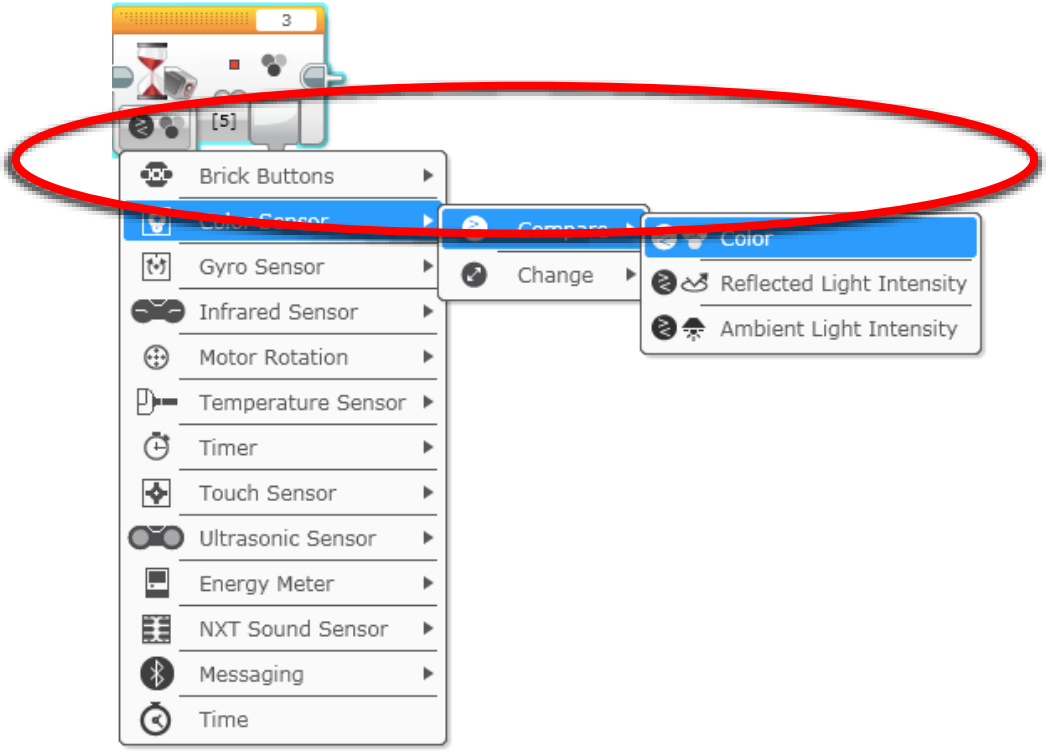
- What are they? Sensors that detect the intensity of light that enters it
- Three modes: Color, Reflected Light Intensity and Ambient Light Intensity
  - **Color Mode:** Recognizes 7 colors (black, brown, blue, green, yellow, red, white) and No Color
  - **Reflected Light:** Measures the intensity of 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)

- USES:
  - Move until a line
  - Follow a line

We will use COLOR MODE in this Lesson

COLOR SENSOR CHALLENGE

- How do you make the robot move up to a green line using the color sensor?
- Step 1: Use Wait For Color
- Step 2: Use the color sensor in COLOR MODE
- Hint: You will use Move Steering (think about motor on and off) and Wait for “Color”

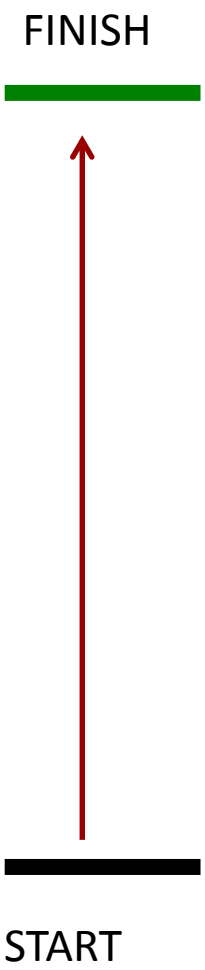


Color Sensor Challenge Solution

Move Steering
Set to “ON”

Move Steering
Set to “OFF”
with BRAKE

Wait until Color is Green (#3)

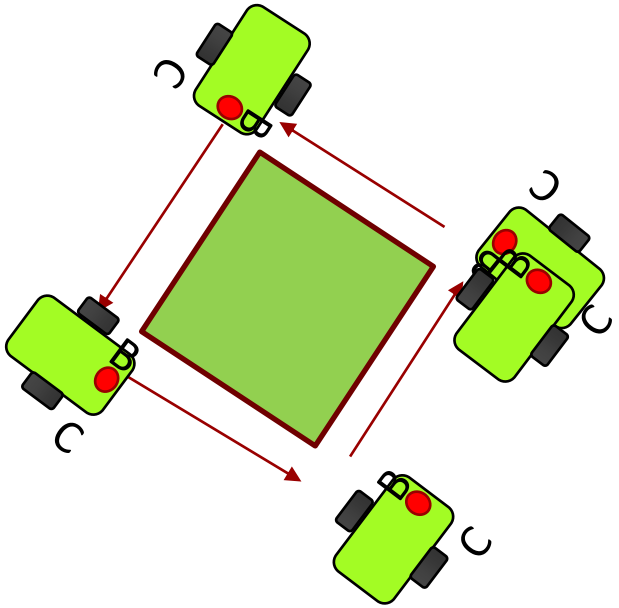


BEGINNER EV3 PROGRAMMING
Lesson: Loops

By: Droids Robotics

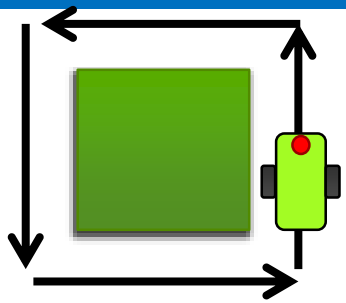
Lesson Objectives

1. Can we improve our baseball robot by just programming it to go to the “next” base and repeating this action?
2. Learn how to repeat an action
3. Learn how to use Loop Blocks



Repeating an Action

- How can we move around bases using the commands we already know?
  - **(move + turn) + (move + turn) + (move + turn) …**



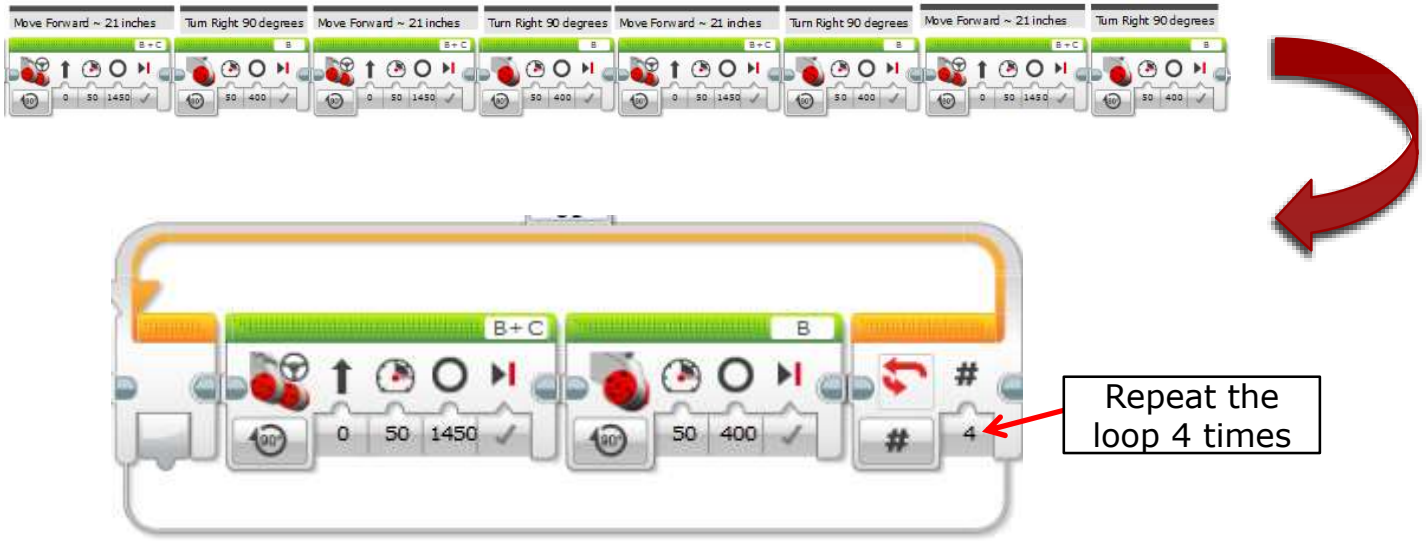
- Loops make repeating a task multiple times easy

Is there an easier way?

Hint:

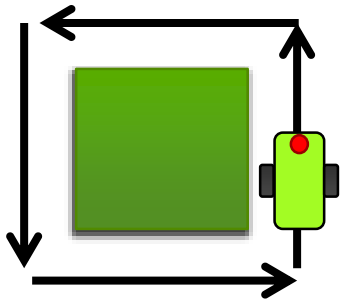
Loops

- Loops make repeating a task multiple times easy
  - KEEP GOING….Forever, for a Count, Until touch (or something else)



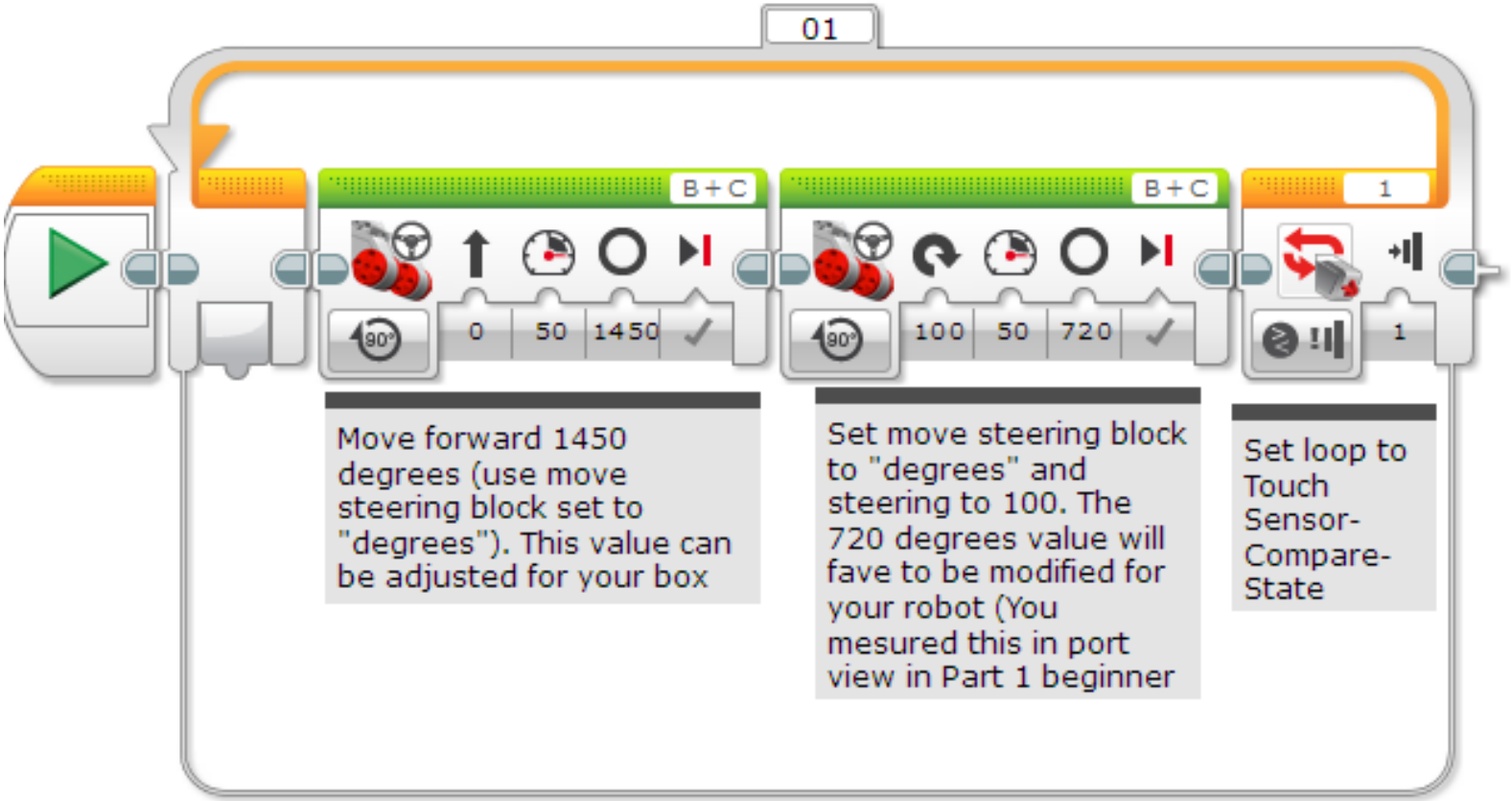
LOOP CHALLENGE

- Challenge: Write a program to go around bases until touched. You must use a LOOP



Loop CHALLENGE Solution

Challenge: Write a program to go around a box until touched



BEGINNER EV3 PROGRAMMING
Lesson: Switches

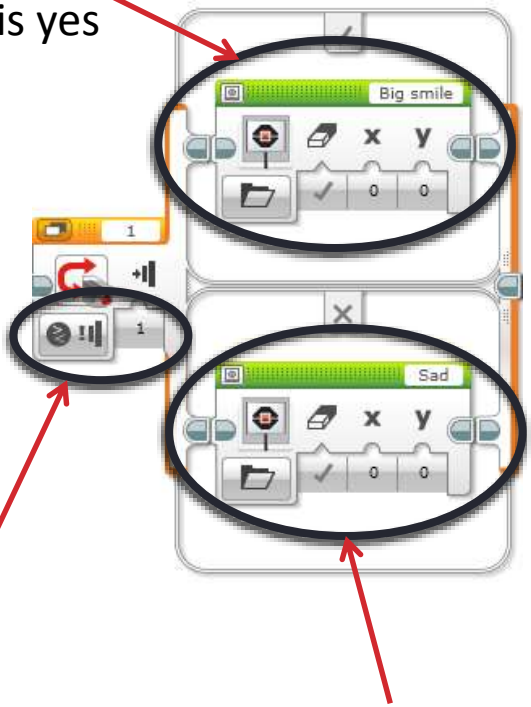
By: Droids Robotics

Lesson Objectives

1. Get the robot to show a happy face when you press its button and a sad face when you don't
2. Learn how to make your robot decide what to do out of different choices
3. Learn how to use a Switch Block

Switch Blocks

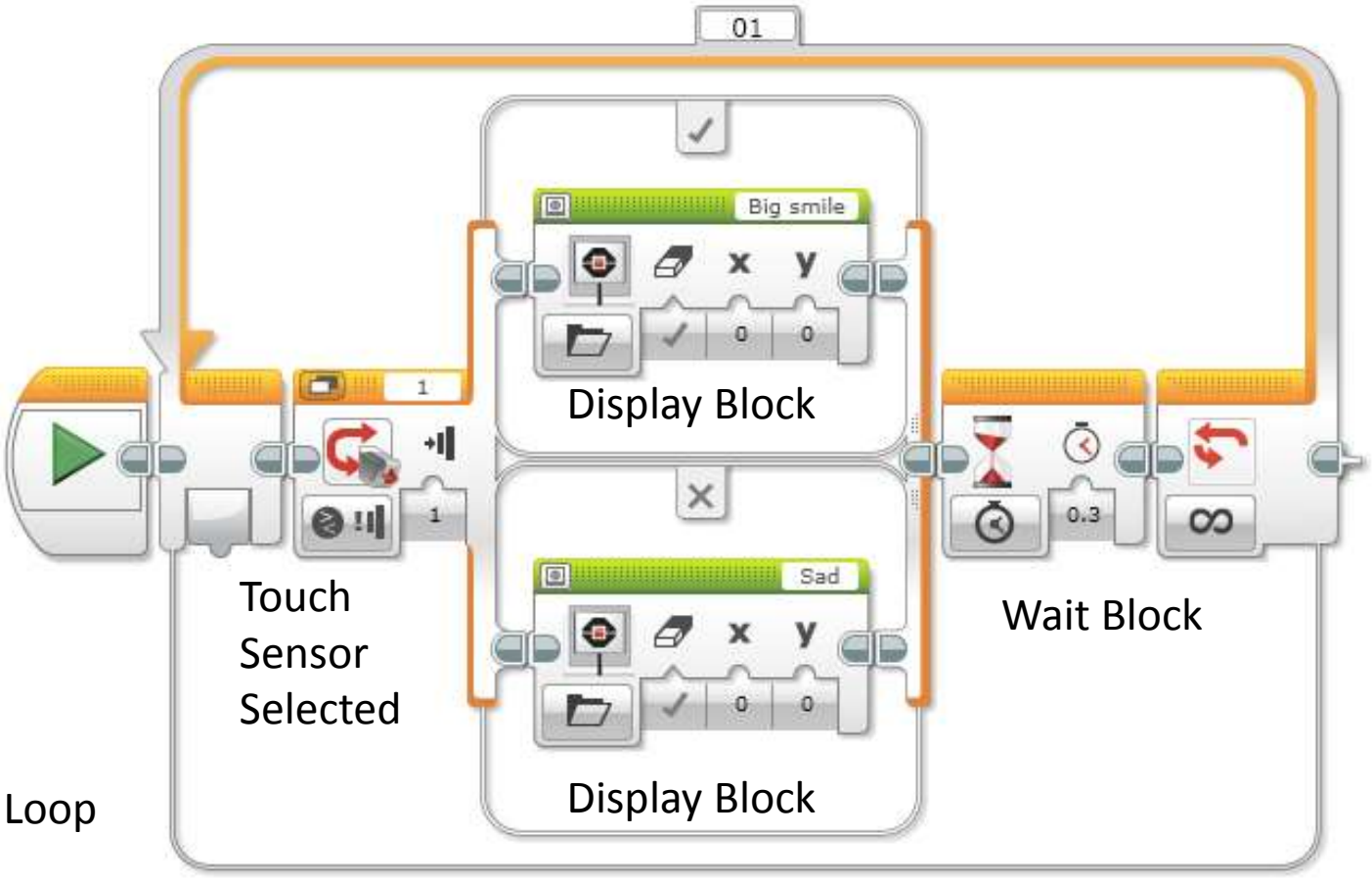
- Asking the robot a question and doing something different based on the answer
  - Example: Does the robot see a line? Or not?
- Basically a YES/NO QUESTION
- Switch blocks are found in the orange/flow tab



Switch Block CHALLENGE 1

- Challenge: Write a program that changes the display based on if the touch sensor is pressed or not pressed.
- If pressed, your EV3 is happy! Display a smiley face. If not pressed, the EV3 is sad! Display a sad face.
- Hint: You will need to use the display block, loops and switch blocks!

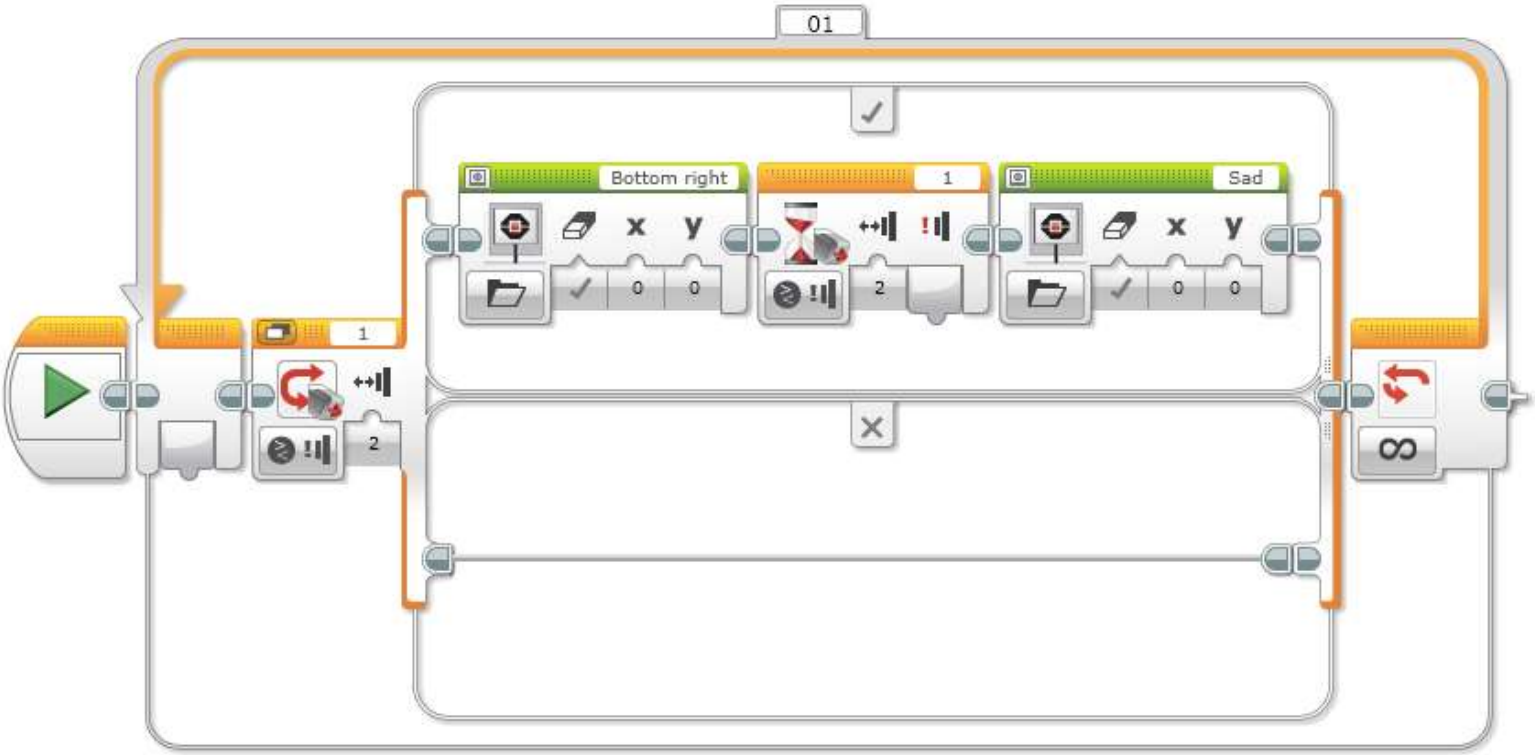
Challenge 1 SOLUTION



Switch Block Challenge 2

- Can you write a program that display big eyeballs if you touch it once and a sad face if you touch it a second time and toggles back and forth.

Challenge 2 solution



BEGINNER EV3 PROGRAMMING
Lesson: Ultrasonic Sensor

By: Droids Robotics

LESSON OBJECTIVES

1. Make a robot that follows you around like a dog
2. Learn about the Ultrasonic Sensor
3. Learn how to use Wait Until Ultrasonic Block
4. Learn the difference between the Wait Until Ultrasonic Block and the Ultrasonic Block

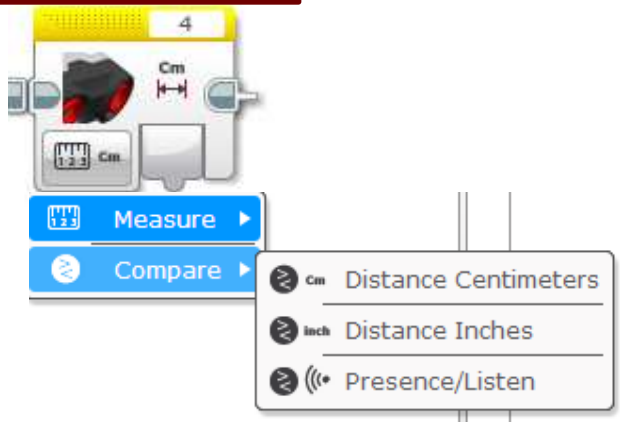
ULTRASONIC

- An ultrasonic sensor measures distance.
- You use it when you need to make sure you are a certain distance away from a target.
- The distance can be measured in inches or centimeters.

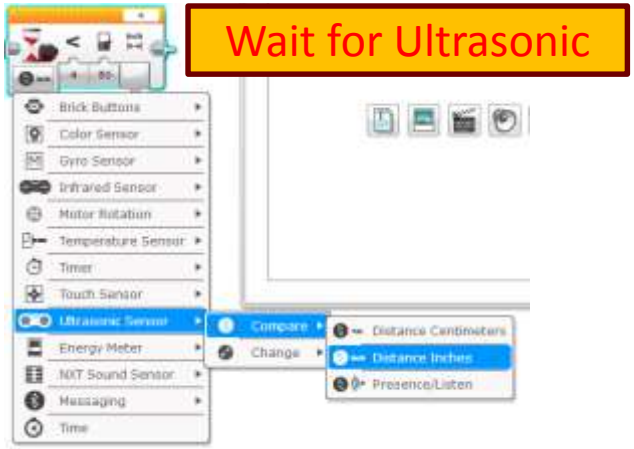
VS.

- To read the ultrasonic sensor, you use the Ultrasonic Block. To use the ultrasonic to do an action until a distance, you use “Wait Until”

Read Ultrasonic

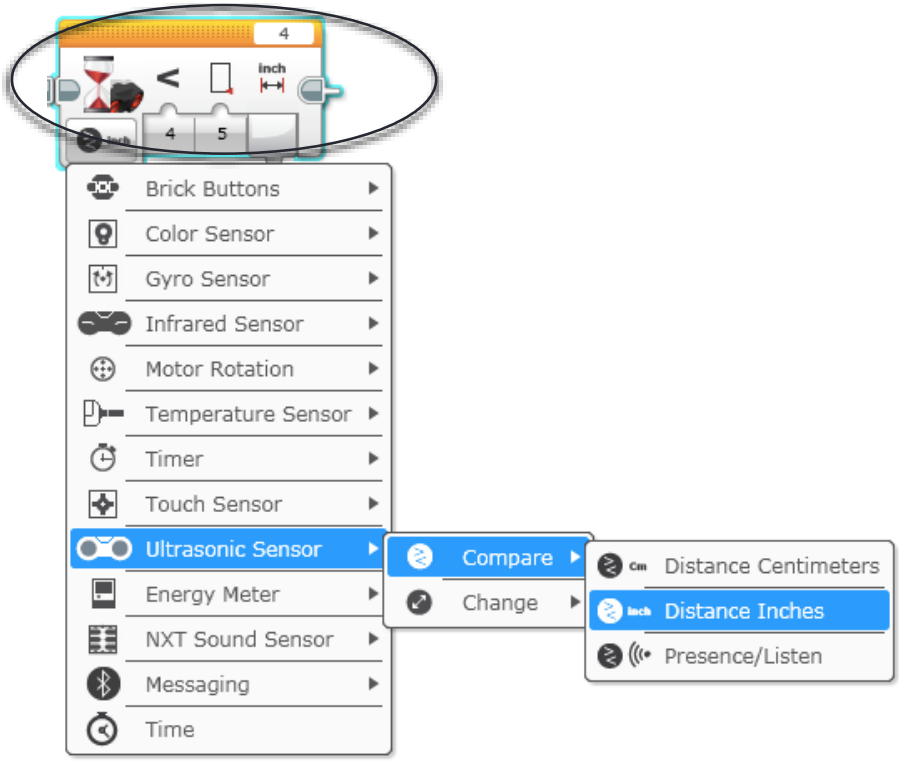


Wait for Ultrasonic

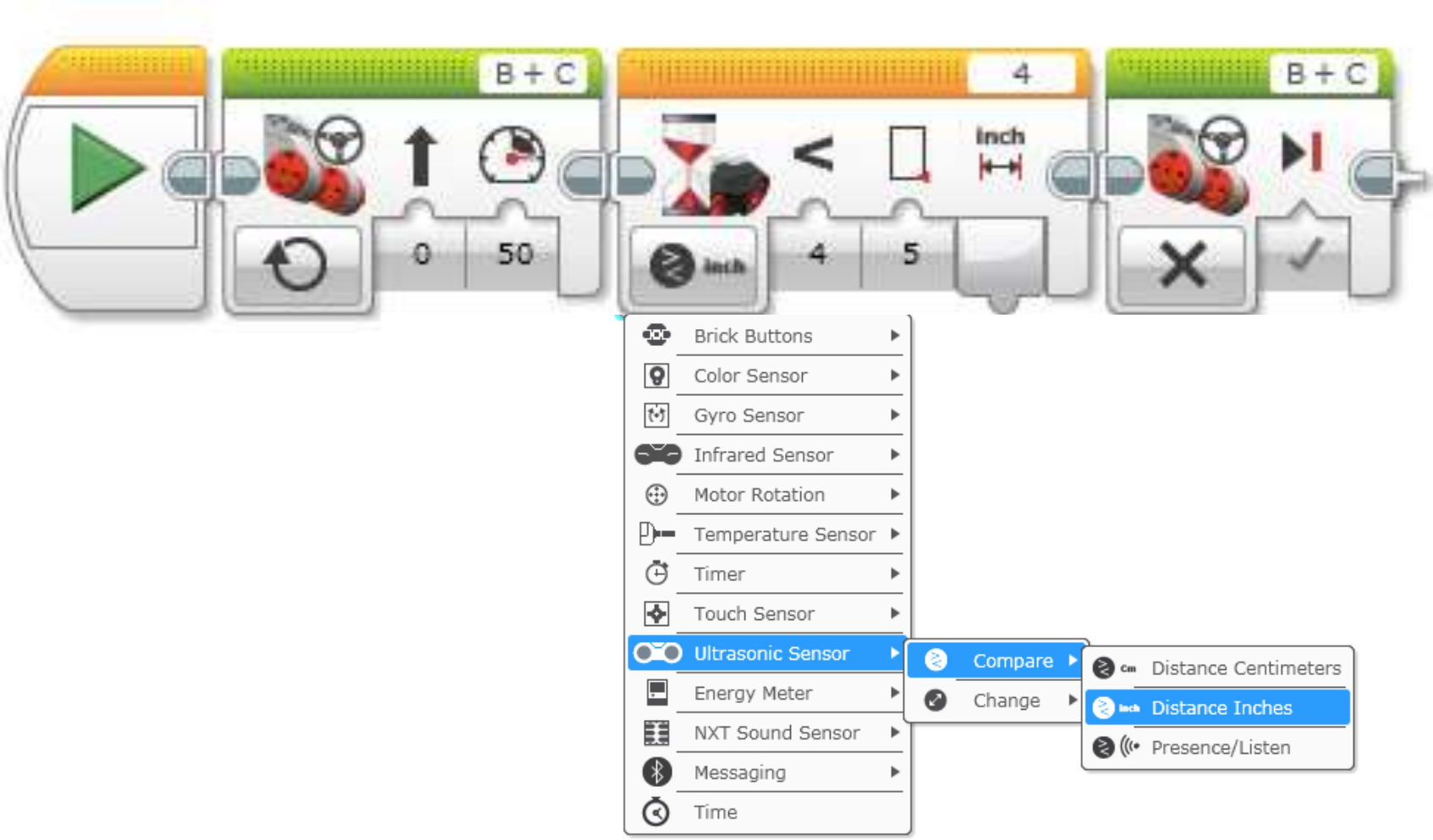


Ultrasonic Challenge 1

- Challenge: Make the robot move until it is 5 inches away from the wall.
- Step 1: Make a new program
- Step 2: Set move to “on”
- Step 3: Set wait block to use the Ultrasonic
- Step 4: Set move block to “off”

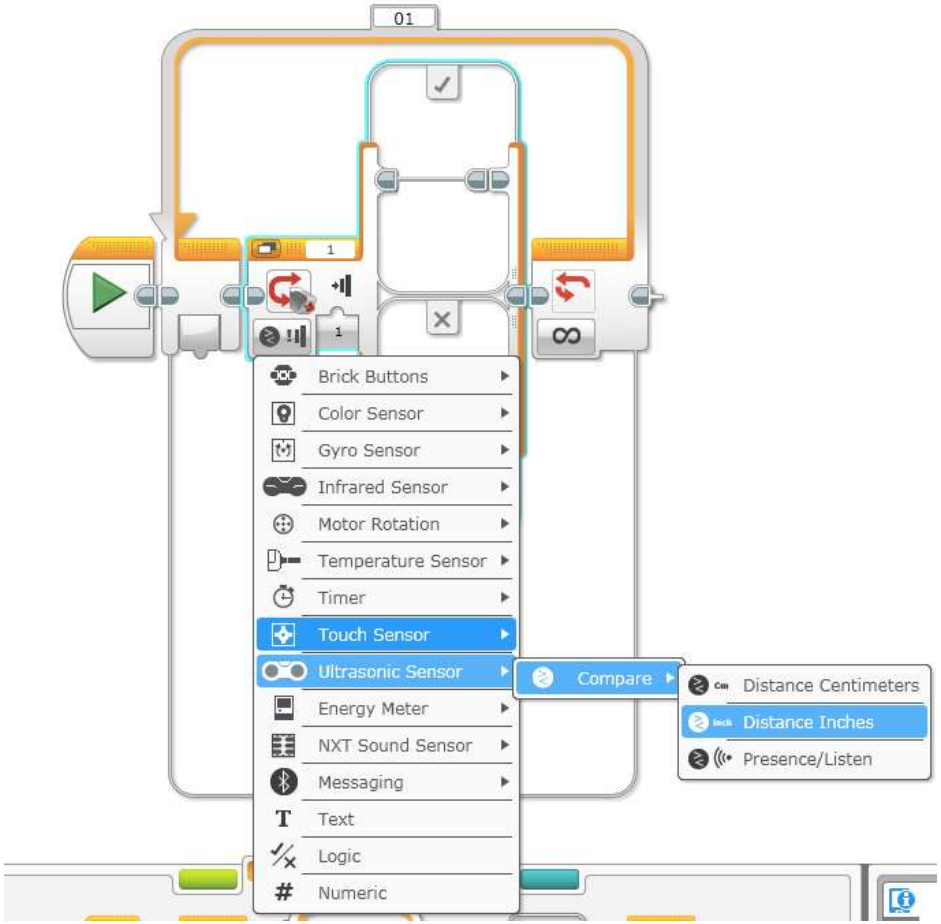


Challenge 1 solution

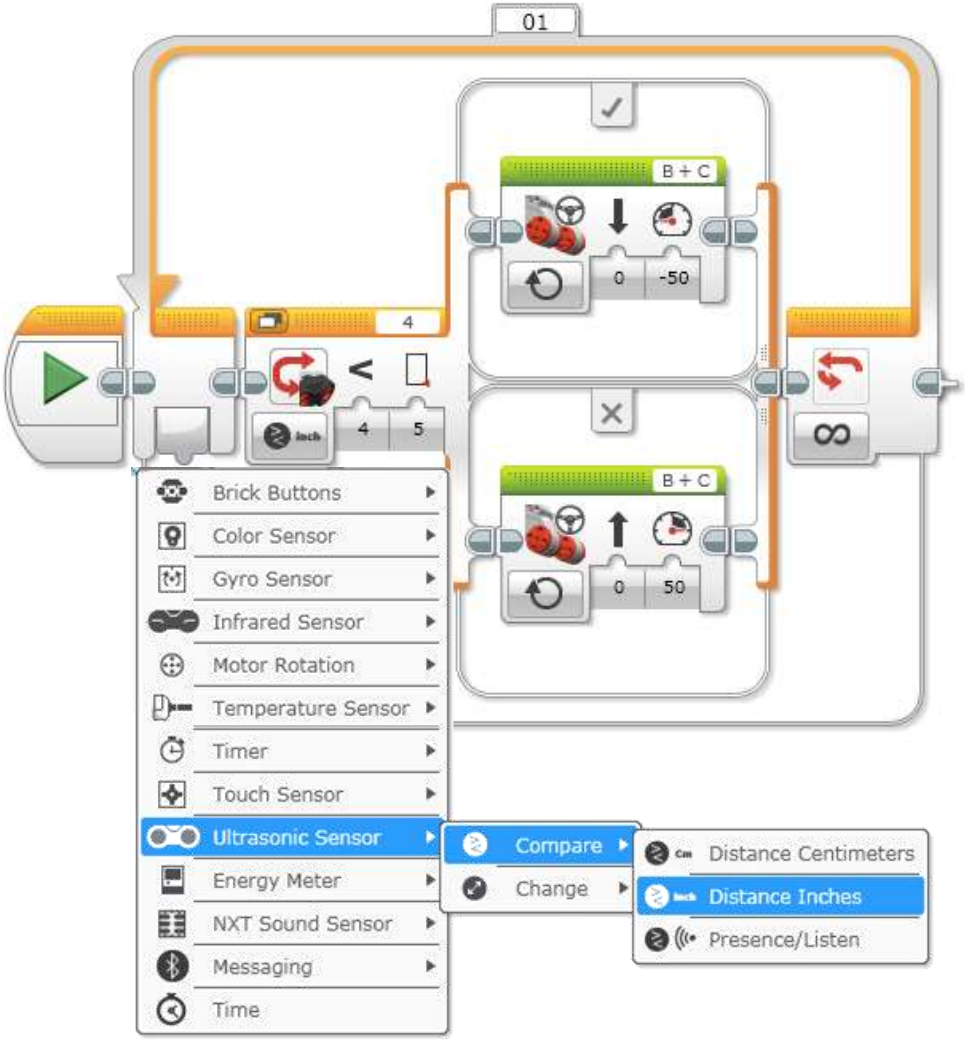


Challenge 2: Dog Follower

- If the robot is closer than 5 inches away from your hand move backward, otherwise move forward.
- Step 1: Drag a loop from the orange tab
- Step 2: Drag switch inside loop
- Step 3: Set switch to Ultrasonic
- Step 4: Set move steering block to ON and place in TRUE
- Step 4: Set move steering block to OFF and place in FALSE



Challenge 2 Solution



BEGINNER PROGRAMMING
Lesson: Basic Line Follower

By: Droids Robotics

LESSON OBJECTIVES

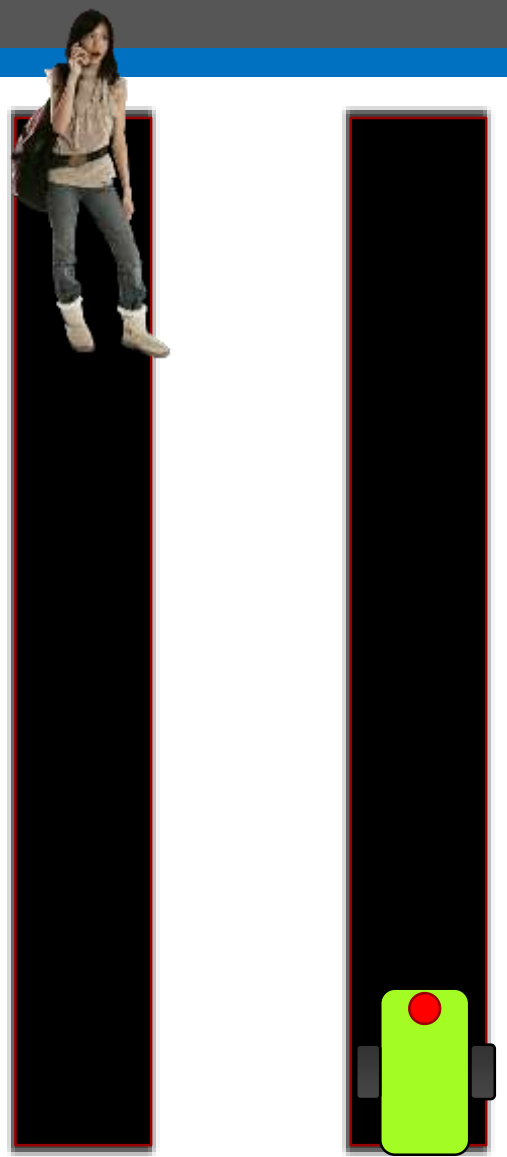
1. Learn how to make a robot follow a line
2. Learn how to get a robot to follow a line using Color Mode on the EV3 Color Sensor
3. Learn how to follow a line until a sensor is activated
4. Learn how to follow a line for a particular distance
5. Learn how to combine sensors, loops and switches

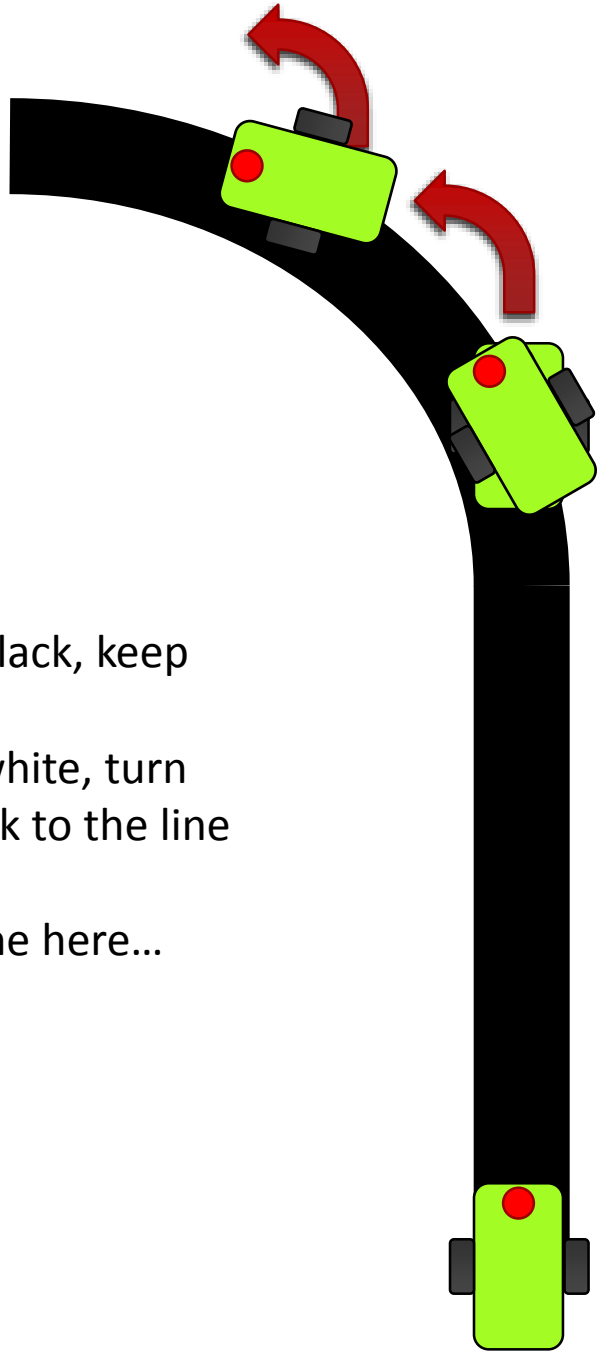
Instructor Notes

- We used CyberBot (see EV3Lessons.com Robot Design page)
  - CyberBot has color sensors behind the wheel
  - Therefore, students will have to line follow backwards (negative power)
- Programming a line follower on an NXT brick with an EV3 requires some adjustments
  - We found that Move Steering with an angle does not work with the NXT for pivot turns
  - Therefore, the code uses Large Motor blocks instead for turning

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?





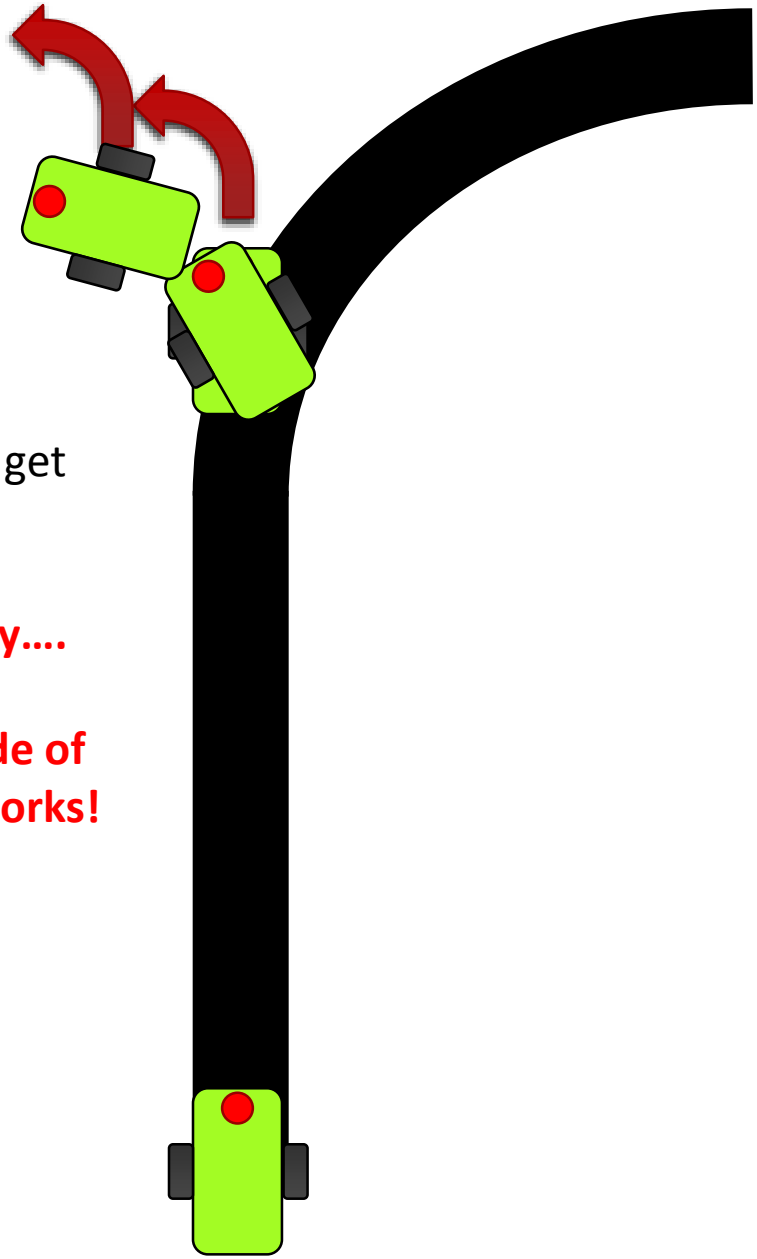
1. If we are on black, keep going straight
2. If we are on white, turn left to get back to the line

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….

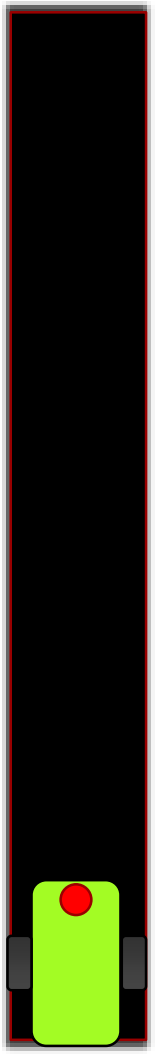
When the robot leaves the left side of the line, the program no longer works!



Line Following: ROBOT STYLE

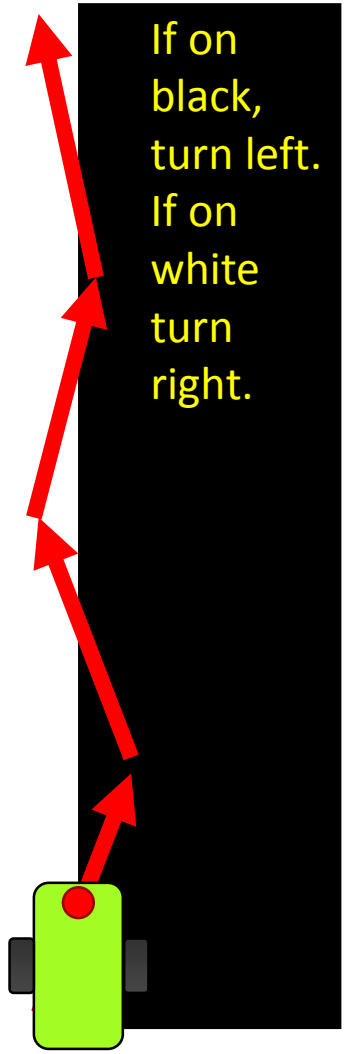
- Why could the Human follow the middle?:
  - They can see ahead.
  - They can see the whole line and its surroundings
  - They see both sides and which side they left

- 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!



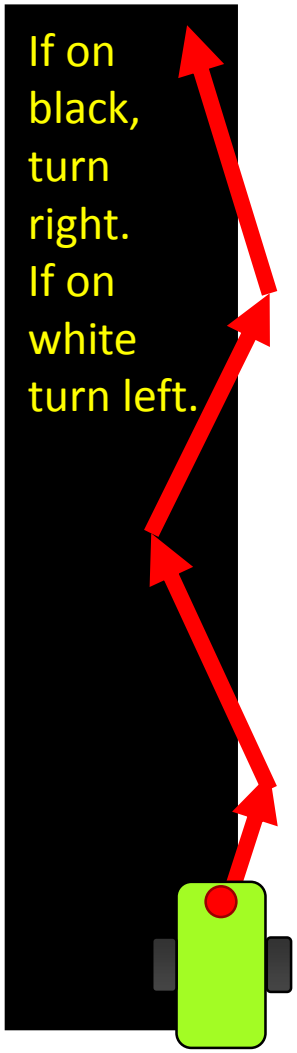
ROBOT LINE FOLLOWING Happens on the edges

Left side line following



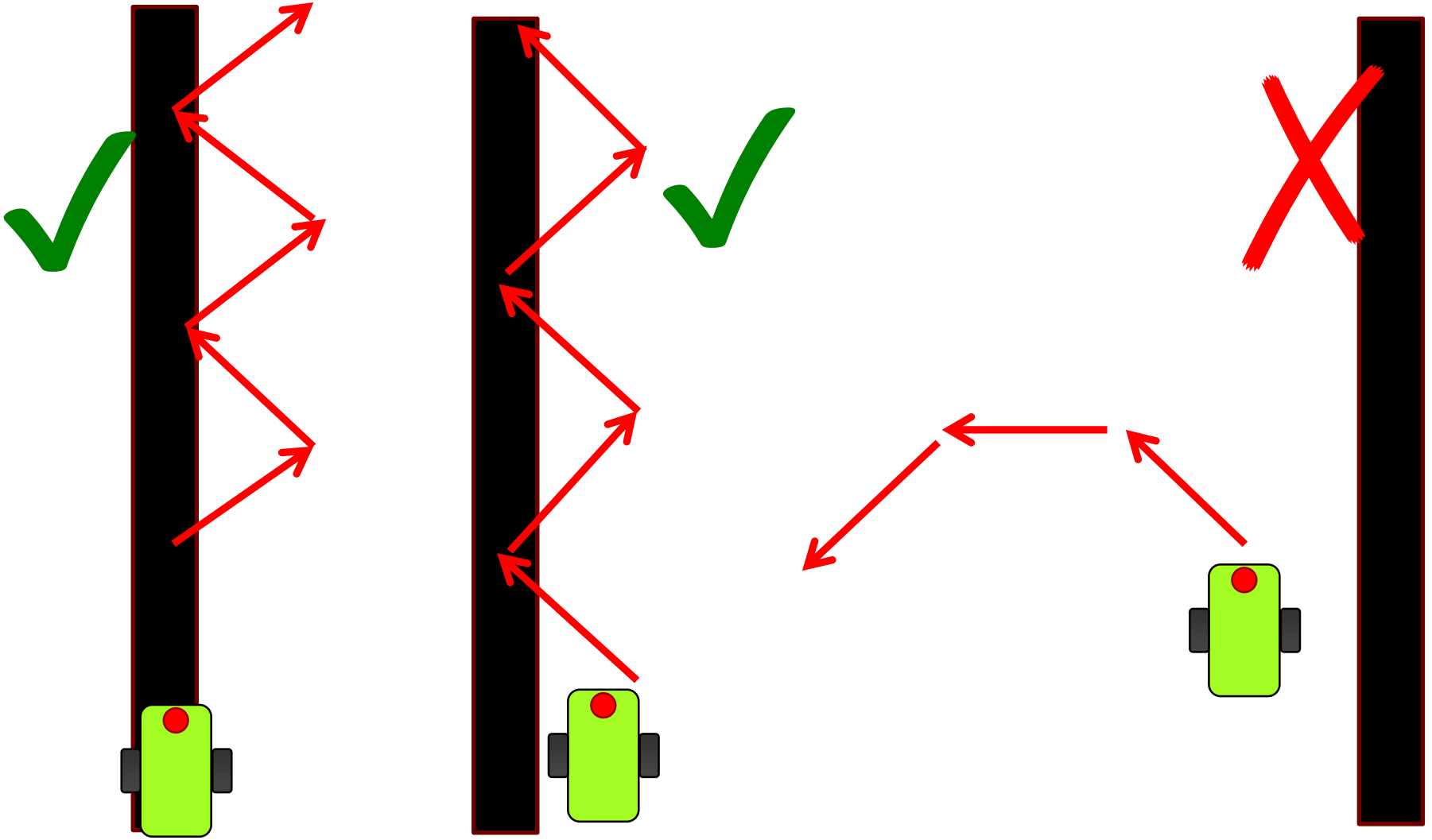
The robot has to choose which way to turn when the color sensor sees a different color.

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



Right side line following

Starting the robot on the correct side

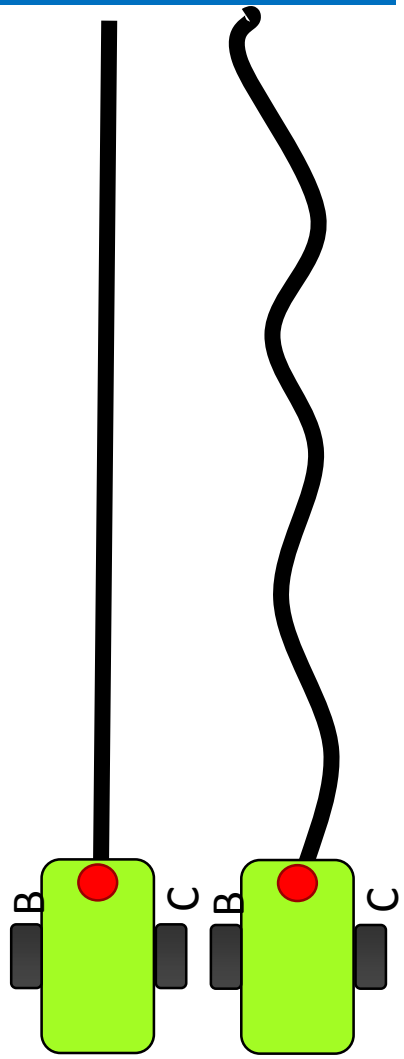


Line Follower challenge 1

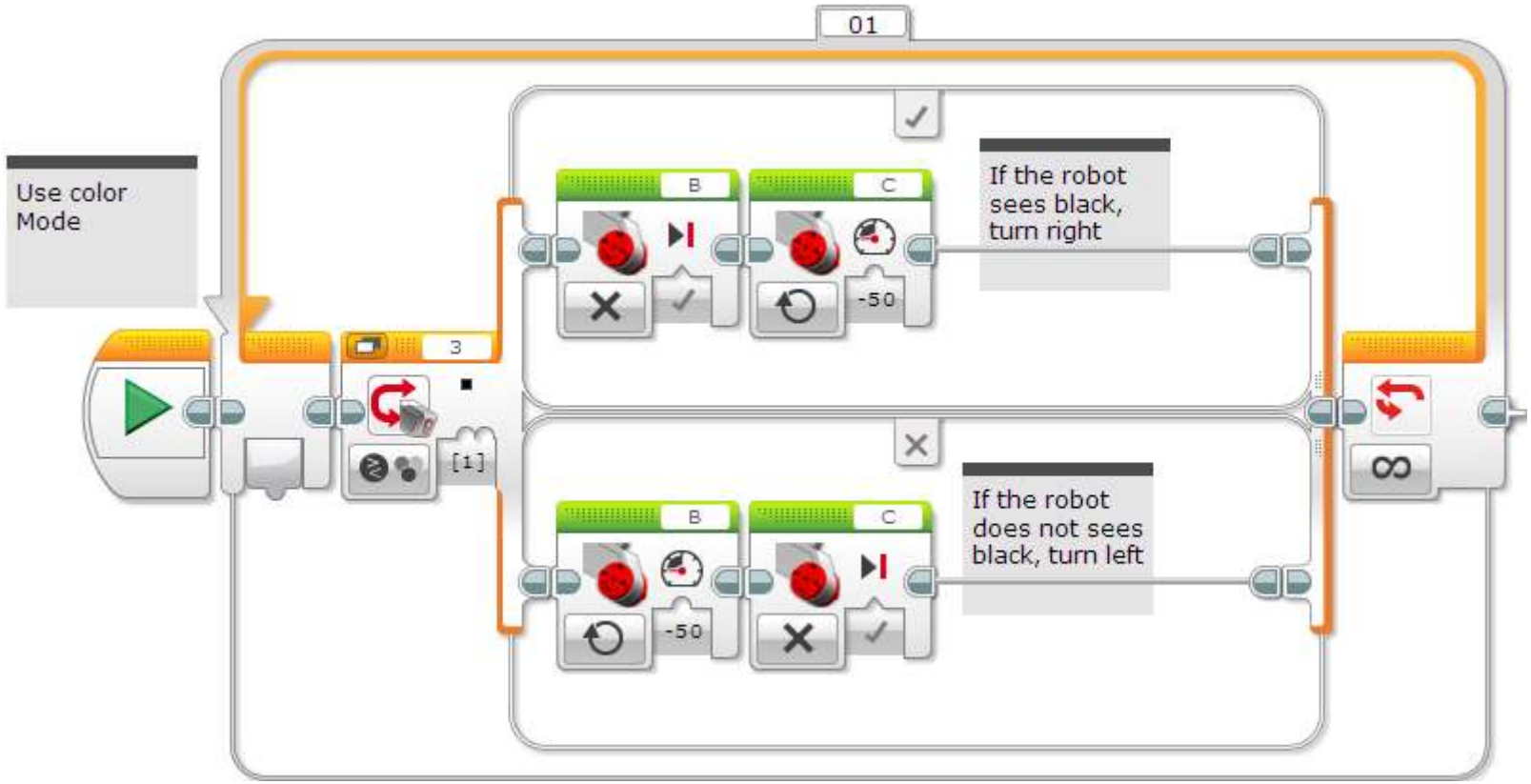
- Step 1: Write a program that follows the RIGHT edge of a line.

1. If your sensor sees black, turn right
2. If your sensor sees white, turn left
3. Use loops and switches!
4. You will need to use Large Motor block in “ON” Mode.
5. You will need to control each motor (B and C) separately.

- Step 2: Try it out on different lines.



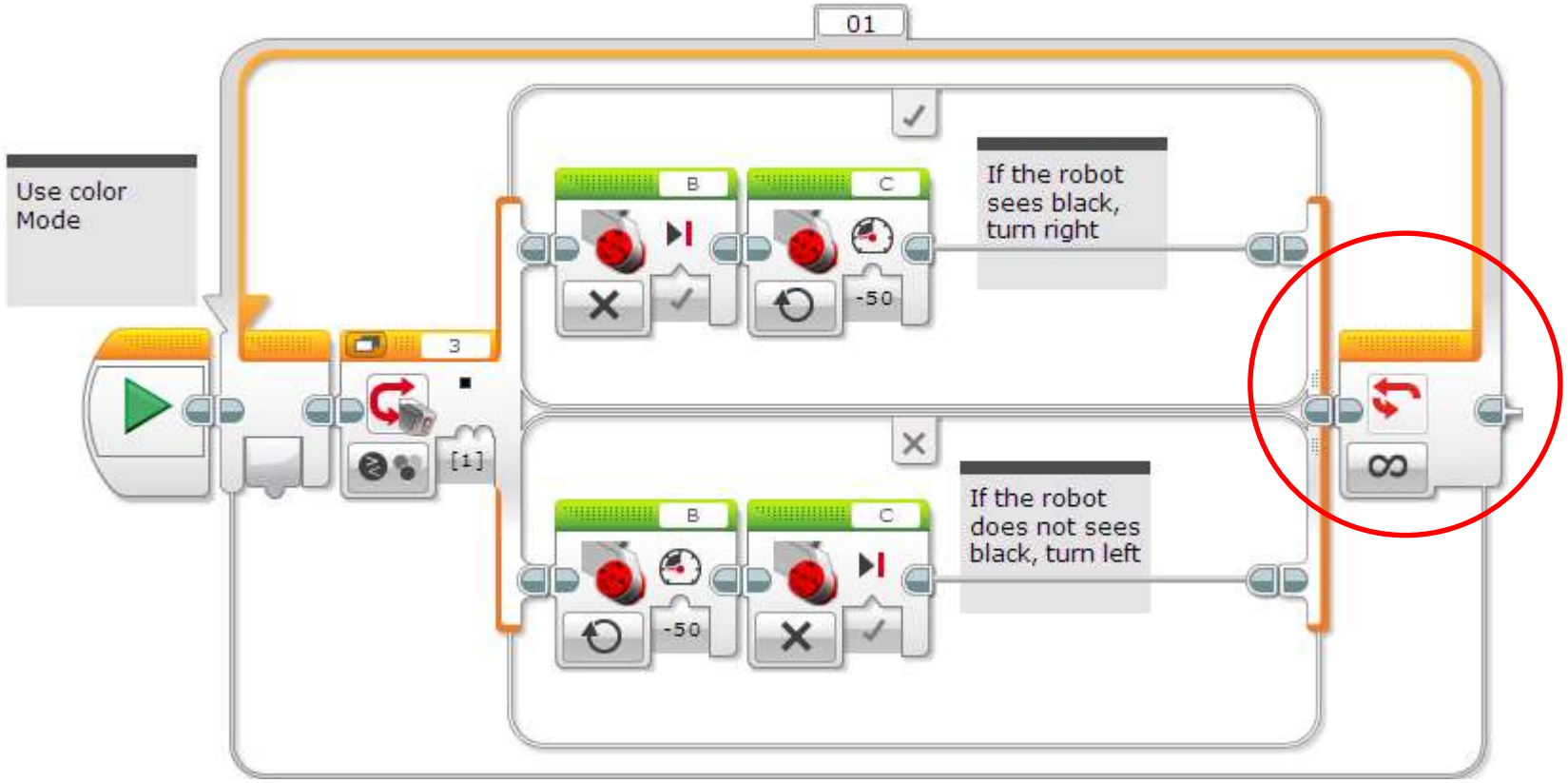
Line Following challenge Solution



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?

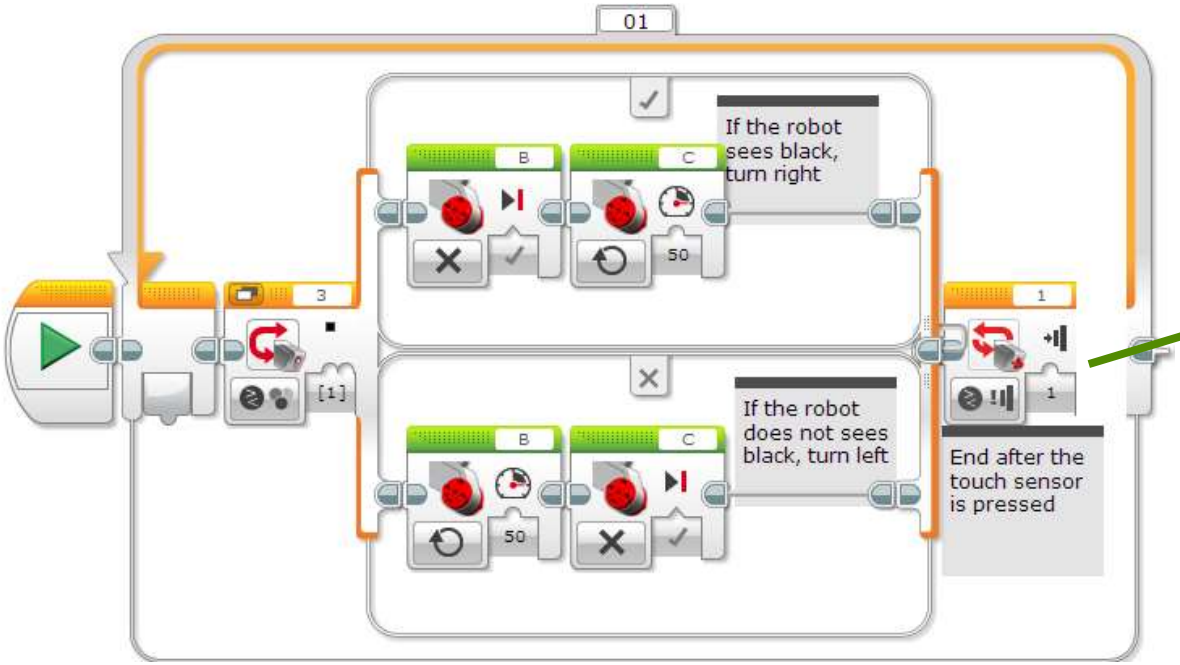
A. Change the end condition on the loop.

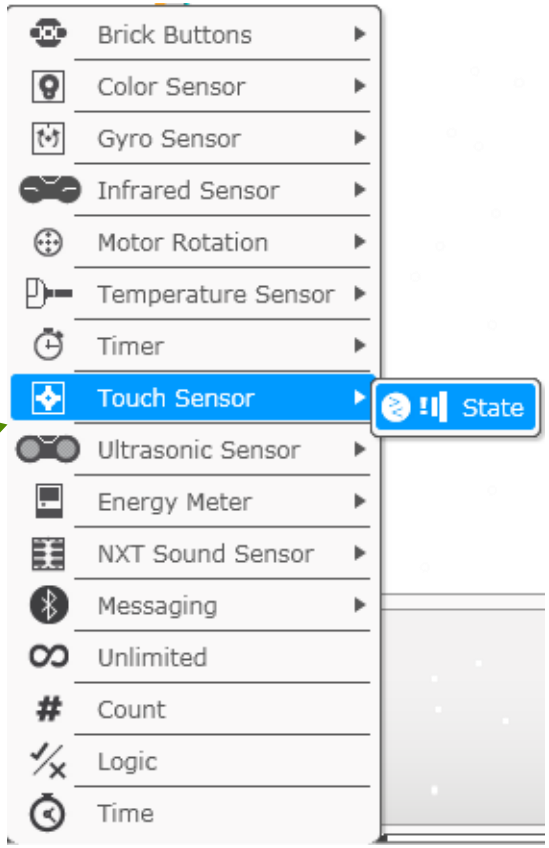
Line follower challenge 2

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

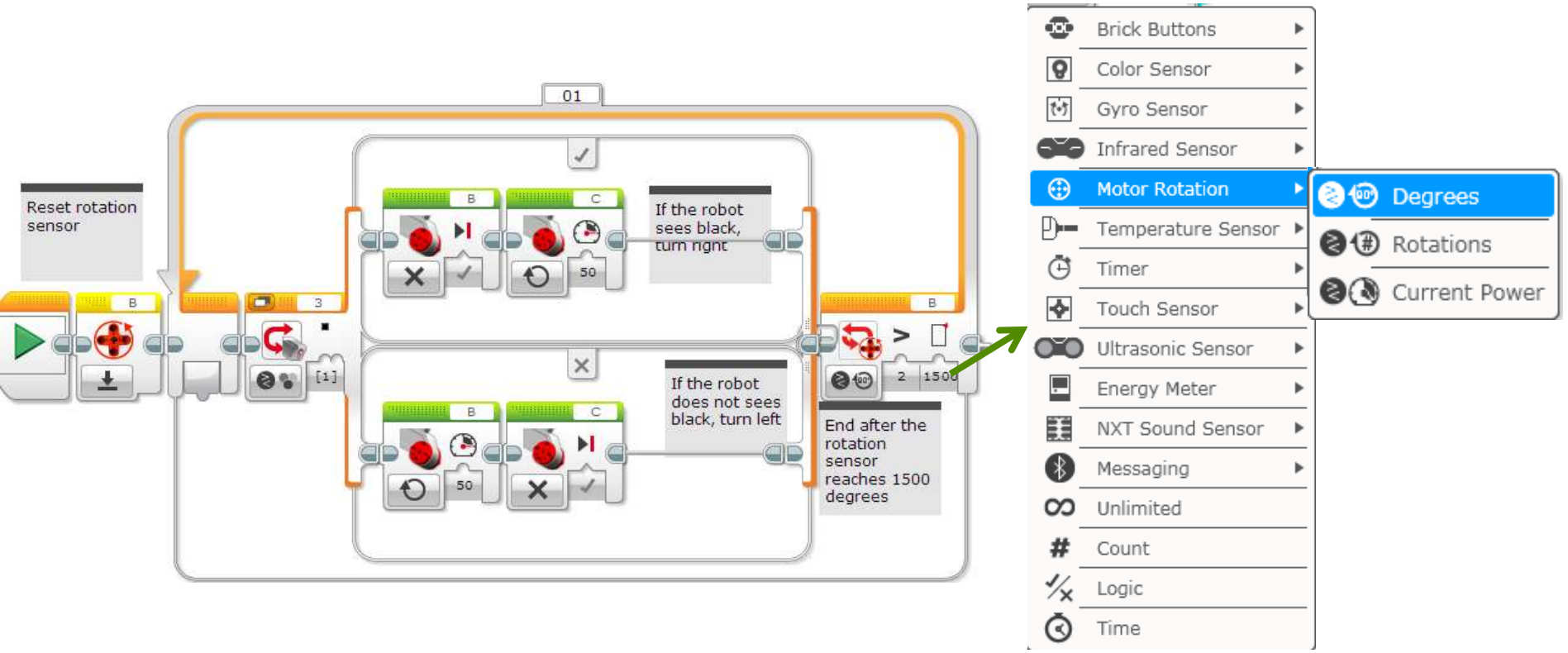
Part 2: Make a line follower that stops after it travels a particular distance

Challenge 2 SOLUTION: Sensor





Challenge 2 Solution: Particular distance



DISCUSSION GUIDE

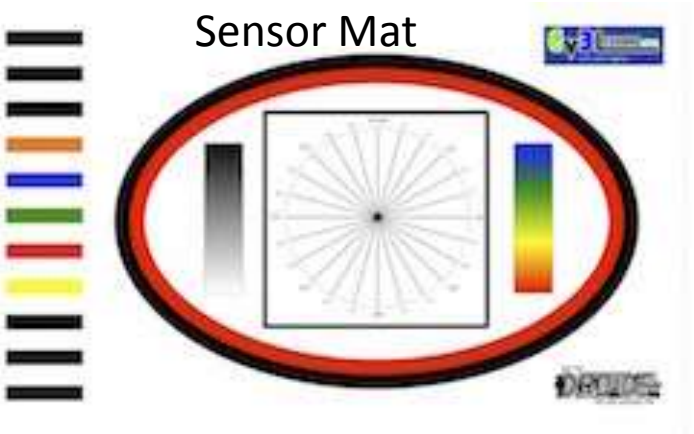
- Why is it important for the robot to follow the same side of the line?
- The robot only knows to check if it is on or off the line.
- This is a basic line follower. What are some things that were not good about this line follower? Do you think the line follower can be improved?
- It wiggles a lot. Smoother line followers are described in the Advanced lessons.
- What sensor measures how far you have travelled?
- The rotation sensor used in Challenge 2 solution measures how much the wheels have turned.
- How would you write a line follower that will stop when it sees a line? Or another color?
- Change the loop exit condition to use the color sensor.

Final Challenge

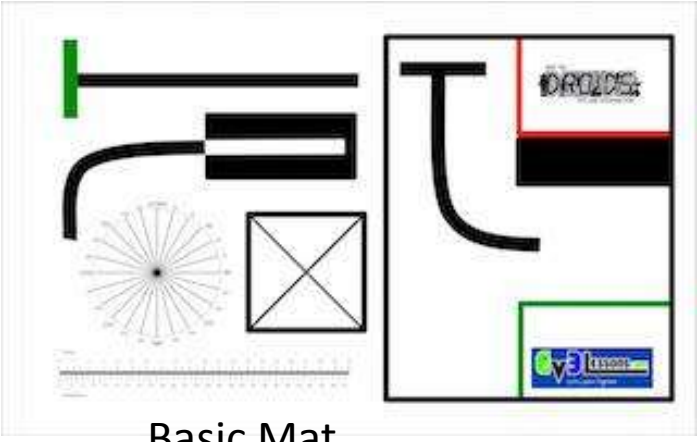
By: Droids Robotics

Using EV3Lessons Training Mats

- Black line follower
- Red line follower
- Move until a certain color
- Drive over all the colors and have the robot name the color
- Have robots follow each other around the track, but stay 5inches away



- Line follower races
- Try your robot on different lines



Basic Mat

- Line follower until green
- Line follow black then white
- Go around the box
- Start in green box and end in red without knocking over box (place light box in black rectangle)

CREDITS

- This tutorial was created by Sanjay Seshan and Arvind Seshan from Droids Robotics.
- More lessons are available at www.ev3lessons.com
- Author's Email: team@droidsrobotics.org