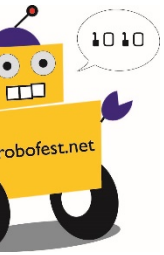



LAWRENCE TECHNOLOGICAL UNIVERSITY

ROBOFEST®

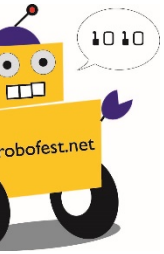
# EV3 Software

# EV3 Robot

# Workshop

Instructor:

Lawrence Technological University
Computer Science



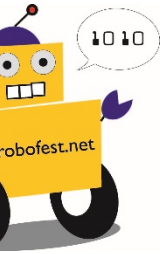

ROBOFEST LAWRENCE TECHNOLOGICAL UNIVERSITY

# Course Overview

- 2019 Robofest competition BinaryBlocks
  - Autonomous robot that arranges white and black blocks to represent a 4-bit binary number
- SPbot introduction
- Using the SPbot to solve the BinaryBlocks challenge

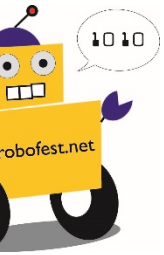
# 2019 Robofest Competition

- Video overview
- Key tasks
  - Understanding binary numbers
  - Finding the edge of the table
  - Following the edge of the table
  - Stop line following
    - When you reach a corner
    - When you reach a given distance
  - Turning the robot
  - Programming EV3 Buttons
  - Aligning the robot to an edge
  - Finding a block
  - Building MyBlocks
  - Stacking blocks




# 2019 Robofest Competition

- Please note that the actual collection and delivery of the binary blocks is beyond the scope of this workshop




# Understanding Binary Numbers

- We typically use decimal numbers in day to day activities. Modern computers typically use binary numbers.
- Decimal numbers are base 10
- Binary numbers are base 2
- Let's examine some examples base 10 and base 2 number to better understand binary numbers

# Understanding Binary Numbers

- Let's examine the base 10 number 357

| | | |
|---|---|---|
| 3 | 5 | 7 |

- Let's examine the base 10 number 13

| | |
|---|---|
| 1 | 3 |

# Understanding Binary Numbers

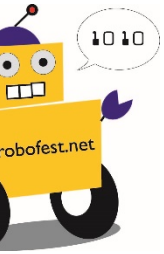
- Represent the base 10 number 13 in base 2:

| 1 | 1 | 0 | 1 | $1101_2$ |
|---|---|---|---|---|

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$

$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13_{10}$

$8 + 4 + 0 + 1 = 13_{10}$



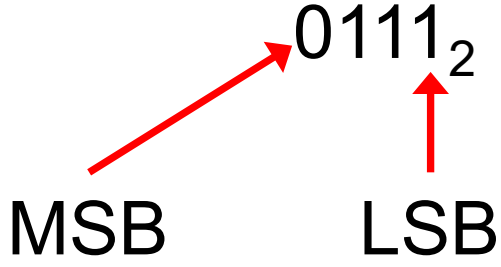

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Understanding Binary Numbers

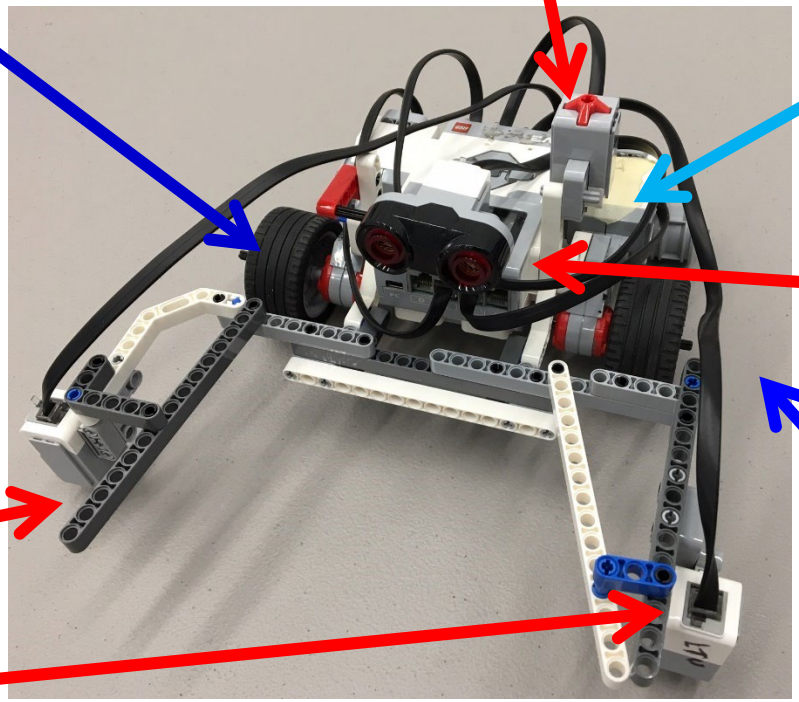
| Decimal/Binary | 8 | 4 | 2 | 1 |
|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 1 |
| **2** | 0 | 0 | 1 | 0 |
| **3** | 0 | 0 | 1 | 1 |
| **4** | 0 | 1 | 0 | 0 |
| **5** | 0 | 1 | 0 | 1 |
| **6** | 0 | 1 | 1 | 0 |
| **7** | 0 | 1 | 1 | 1 |
| **8** | 1 | 0 | 0 | 0 |
| **9** | 1 | 0 | 0 | 1 |
| **10** | 1 | 0 | 1 | 0 |
| **11** | 1 | 0 | 1 | 1 |
| **12** | 1 | 1 | 0 | 0 |
| **13** | 1 | 1 | 0 | 1 |
| **14** | 1 | 1 | 1 | 0 |
| **15** | 1 | 1 | 1 | 1 |

# Understanding Binary Numbers

- Lastly we need to understand the most significant bit and least significant bit
  - The most significant bit (MSB) is the bit in a binary number that is of the greatest numerical value
  - The least significant bit (LSB) is the bit in a binary number that is of the lowest numerical value
- Recall our example with $7_{10} = 0111_2$



# LEGO EV3 robot used – SPbot



# Remember the connections!

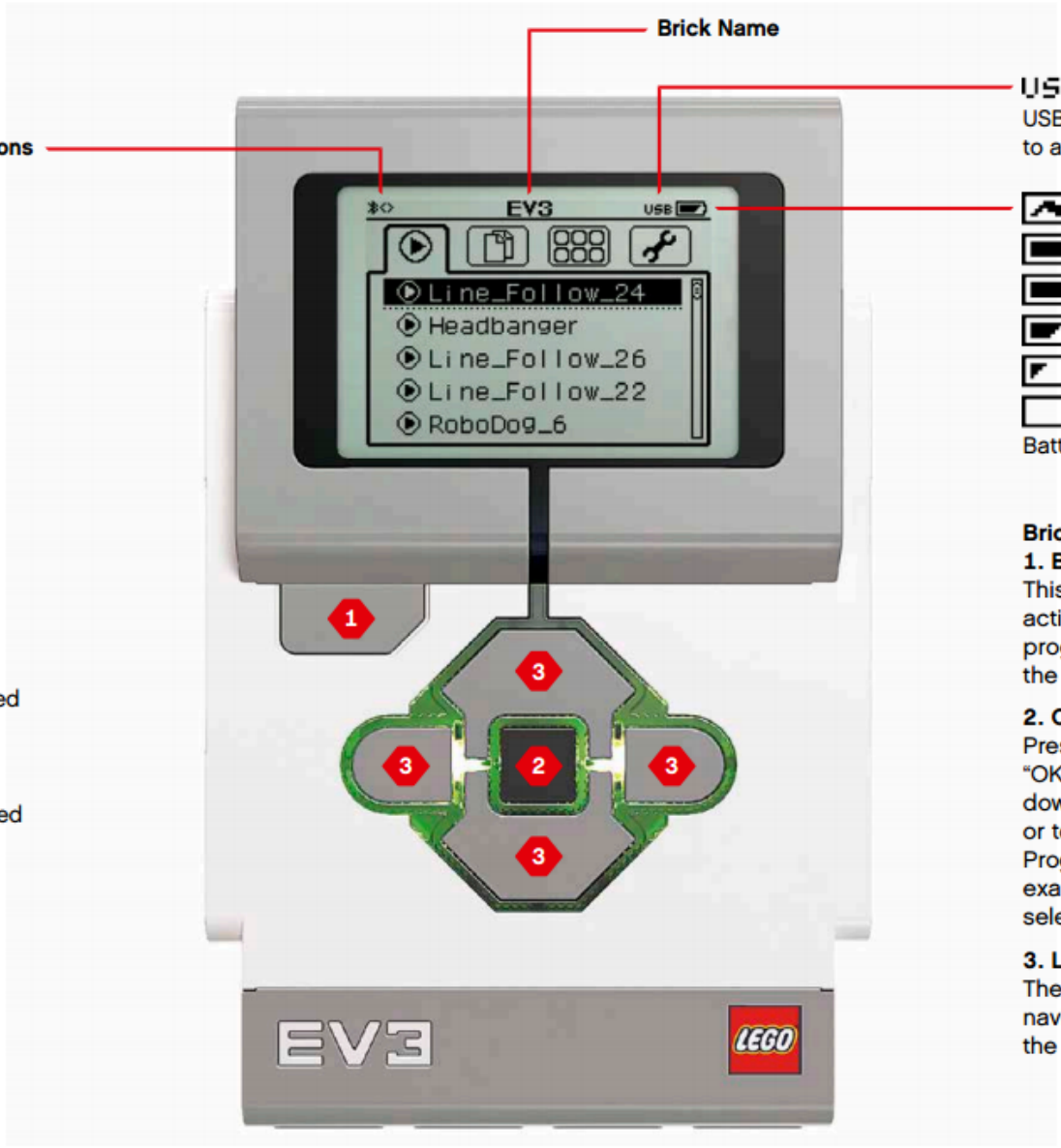
- Left Motor connects to **B**
- Right Motor connects to **C**
  - If your motors are upside down forward will be backwards in your program
- Color sensor 1 connects to port no. **1**
- Color sensor 2 connects to port no. **2**
- Touch sensor connects to port no. **3**
- Ultrasonic sensor connects to port no. **4**

Please note that the retail version of EV3 comes with an infrared sensor, not an ultrasonic sensor.

# EV3 Versions Used

- Examples use EV3 Educational Version **1.4.2**
  - Download
    - https://education.lego.com/enus/downloads/mindstorms-ev3
- EV3 Firmware version: **V1.09E**
- PowerPoint and all example programs are available at robofest.net under Tech Resources

# Brick Overview




**Wireless Connection Status icons**
*(from the left)*

Bluetooth enabled but not connected or visible to other Bluetooth devices

Bluetooth enabled and visible to other Bluetooth devices

Bluetooth enabled and your EV3 Brick is connected to another Bluetooth device

Bluetooth enabled and visible and your EV3 Brick is connected to another Bluetooth device

Wi-Fi enabled but not connected to a network

Wi-Fi enabled and connected to a network

**USB:**
USB connection established to another device

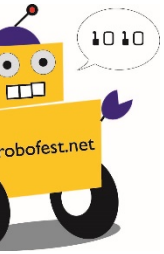
Battery level

**Brick Buttons**

**1. Back**
This button is used to reverse actions, to abort a running program, and to shut down the EV3 Brick.

**2. Center**
Pressing the Center button says "OK" to various questions—to shut down, to select desired settings, or to select blocks in the Brick Program App. You would, for example, press this button to select a checkbox.

**3. Left, Right, Up, Down**
These four buttons are used to navigate through the contents of the EV3 Brick.



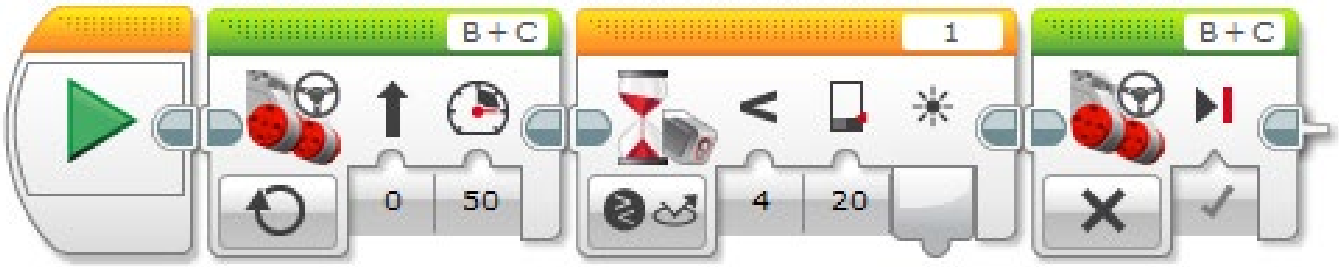

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 0

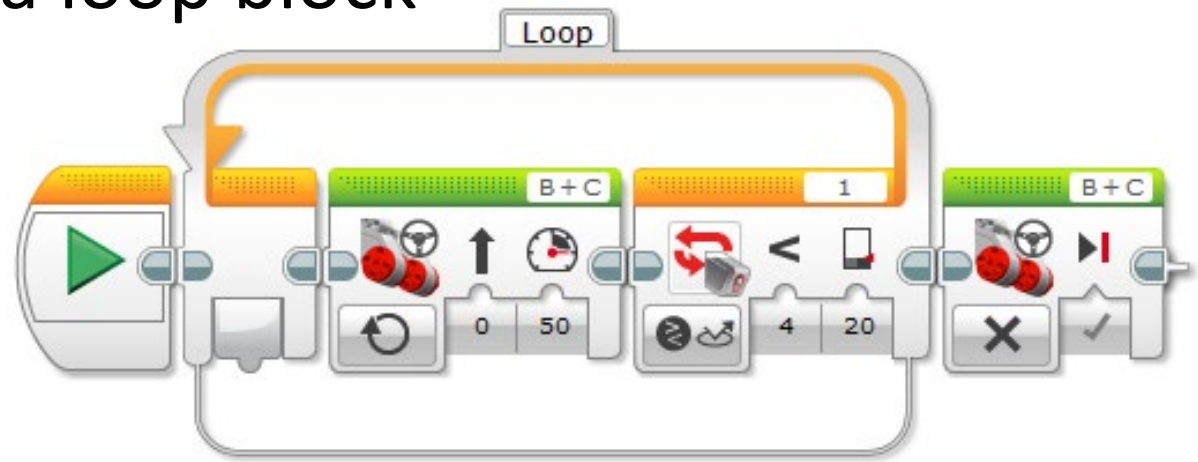
## Finding the edge of the table

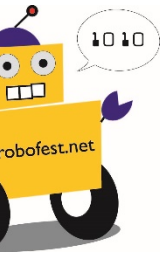
# Task 0: Example Solutions

- Using a wait block




- Using a loop block





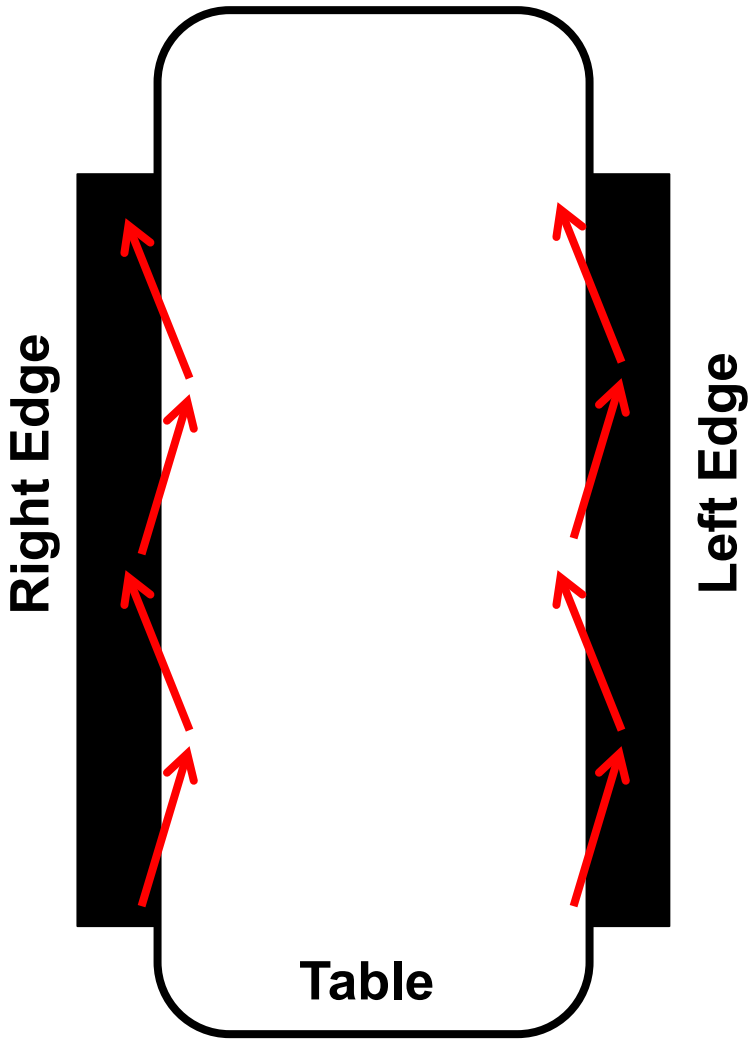

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 1

## Following the edge of the table

# Following The Edge Of The Table

- Use the zig-zag method to follow the edge of the table
- Edge following is also referred to as line following
- We need to determine when the robot is on or off the table



# Following The Edge Of The Table

- Get color sensor values to determine when the robot is **on** or **off** the table. We will use the color sensor in <u>Reflective Light Intensity</u> mode.

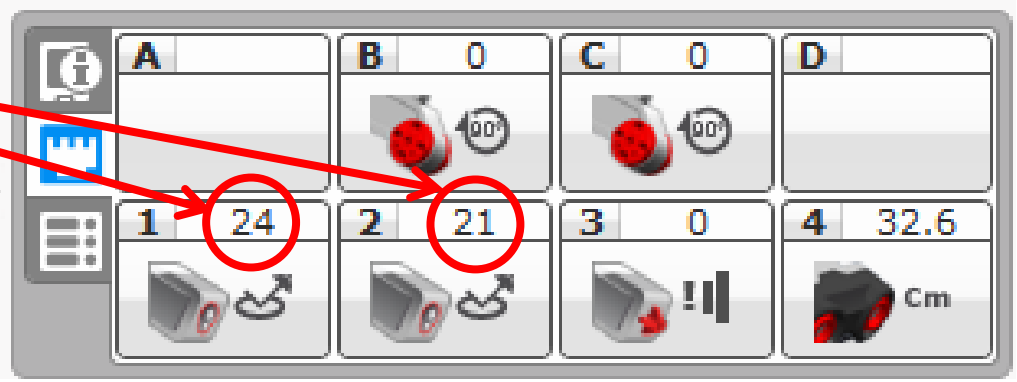
- Color Sensor 1
  - On table = _____ (60)
  - Off table = _____ (20)

Color Sensor 2

On table = _____ (60)

Off table = _____ (10)

Color Sensor Readings

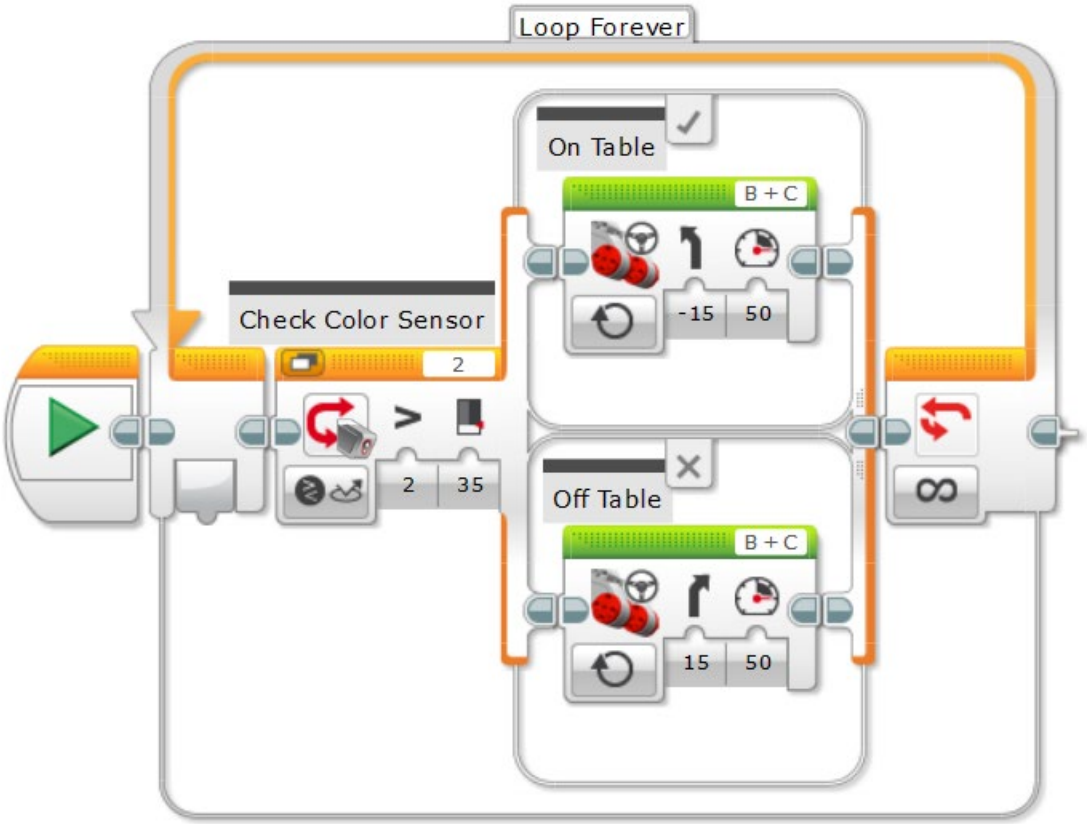


# Following The Edge Of The Table

- For light sensor #2 settings example
  - On table = 60
  - Off table = 10
  - Median threshold = (60+10)/2 = 35

- Two cases
  - Light sensor reading > 35.  On table.
  - Light sensor reading < 35.  Off table.

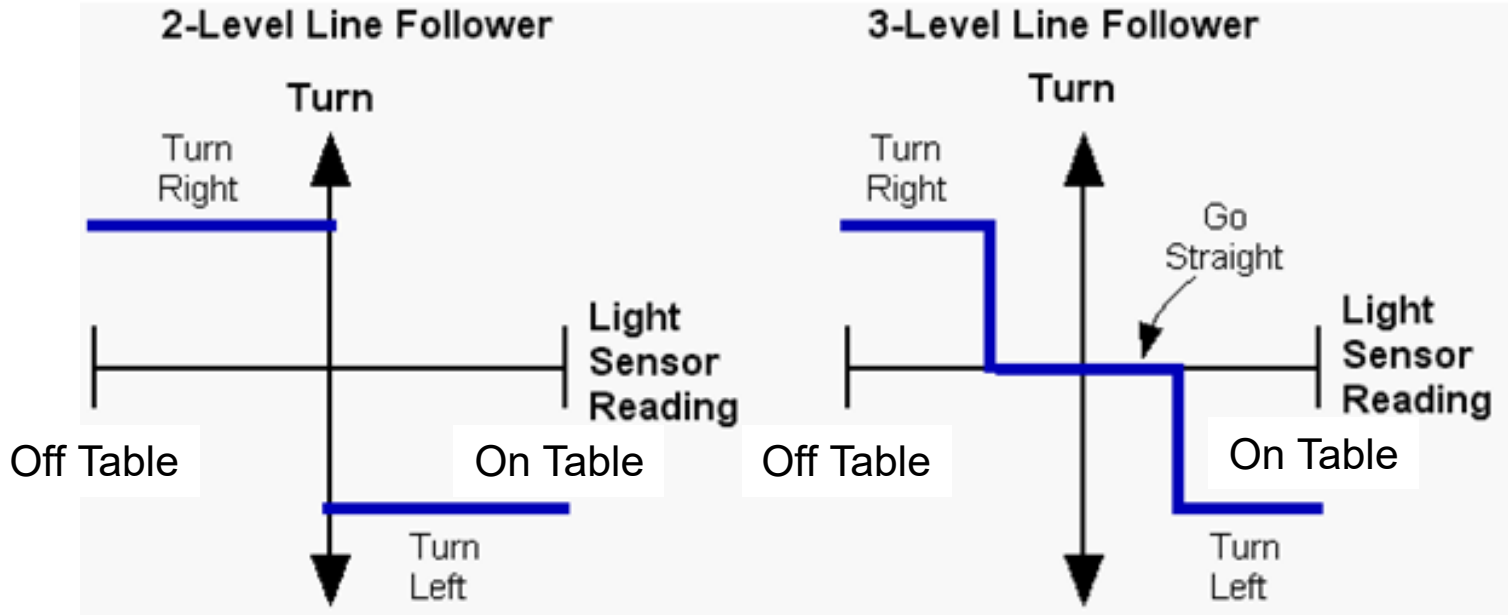
# Simple Line Following Algorithm

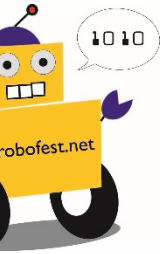



**Program: LineFollowZZ.ev3** **YouTube:** https://youtu.be/pFNnGpIU9GU

# How to improve our line following algorithm

- The zig-zag method can cause a bumpy response
- To improve the response, you can use a 3-level line follower (concept shown below)





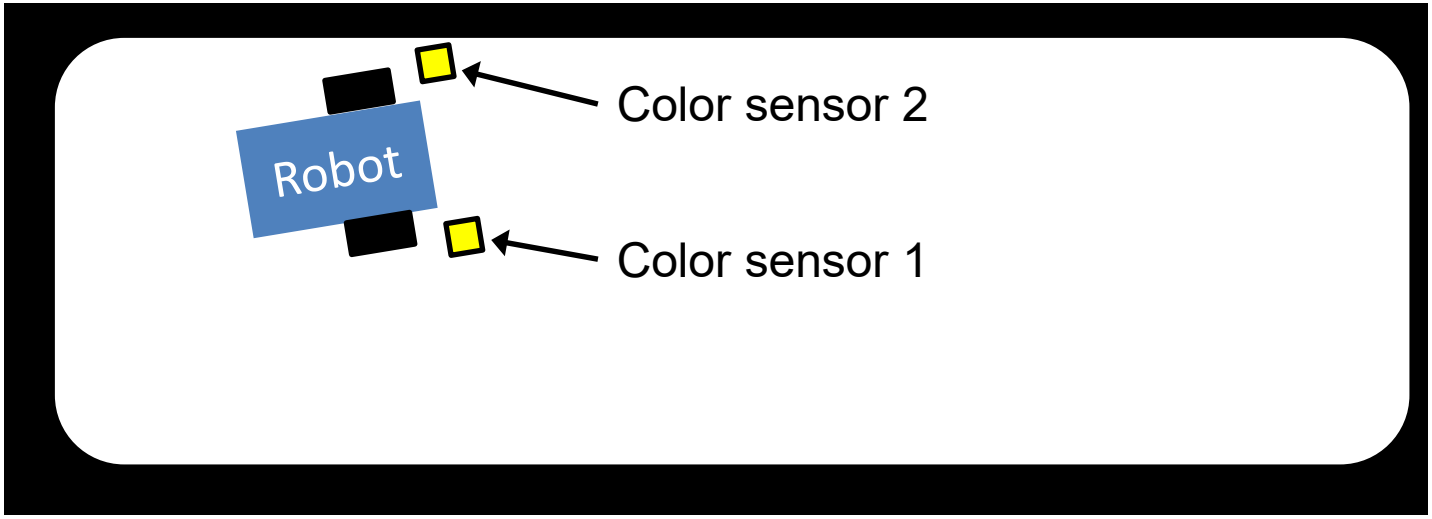

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 2a

Line following to the corner of the table

# Line following to the corner

- One method of line following to the corner is to follow the edge of the table with one color sensor and detect the corners with a other color sensor
  - Sensor 1 used to locate the end of the table
  - Sensor 2 used to follow the edge of the table

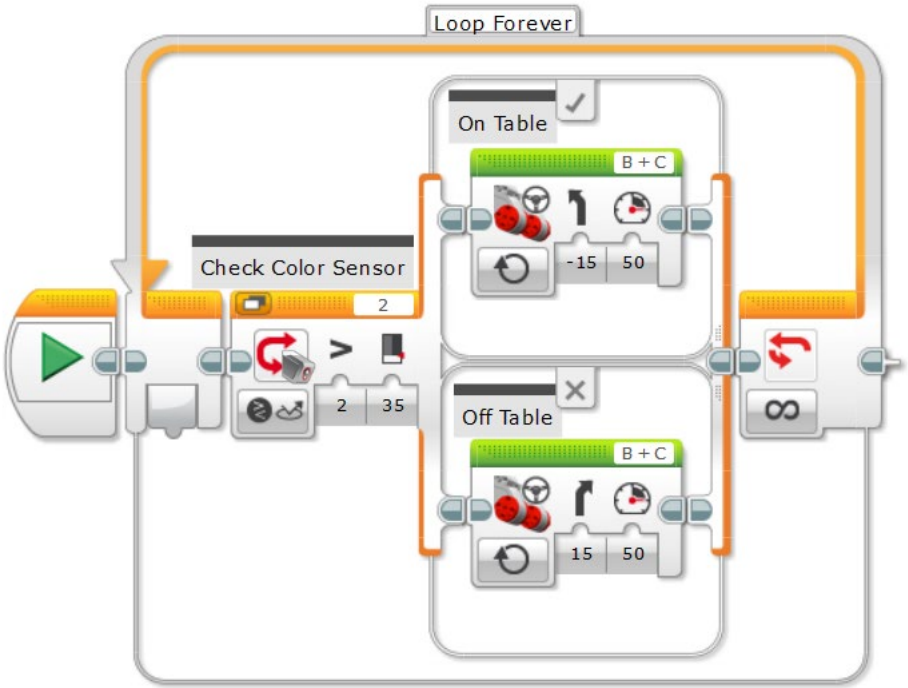


# Line following to the corner

- Couple comments regarding moving around the table
  - It is possible to travel around the edge of the table with only one color sensor, but it is more difficult and potentially less reliable
  - Remember that there are no markers to identify the four corners of the table
    - You need to count the corners has your robot reaches them

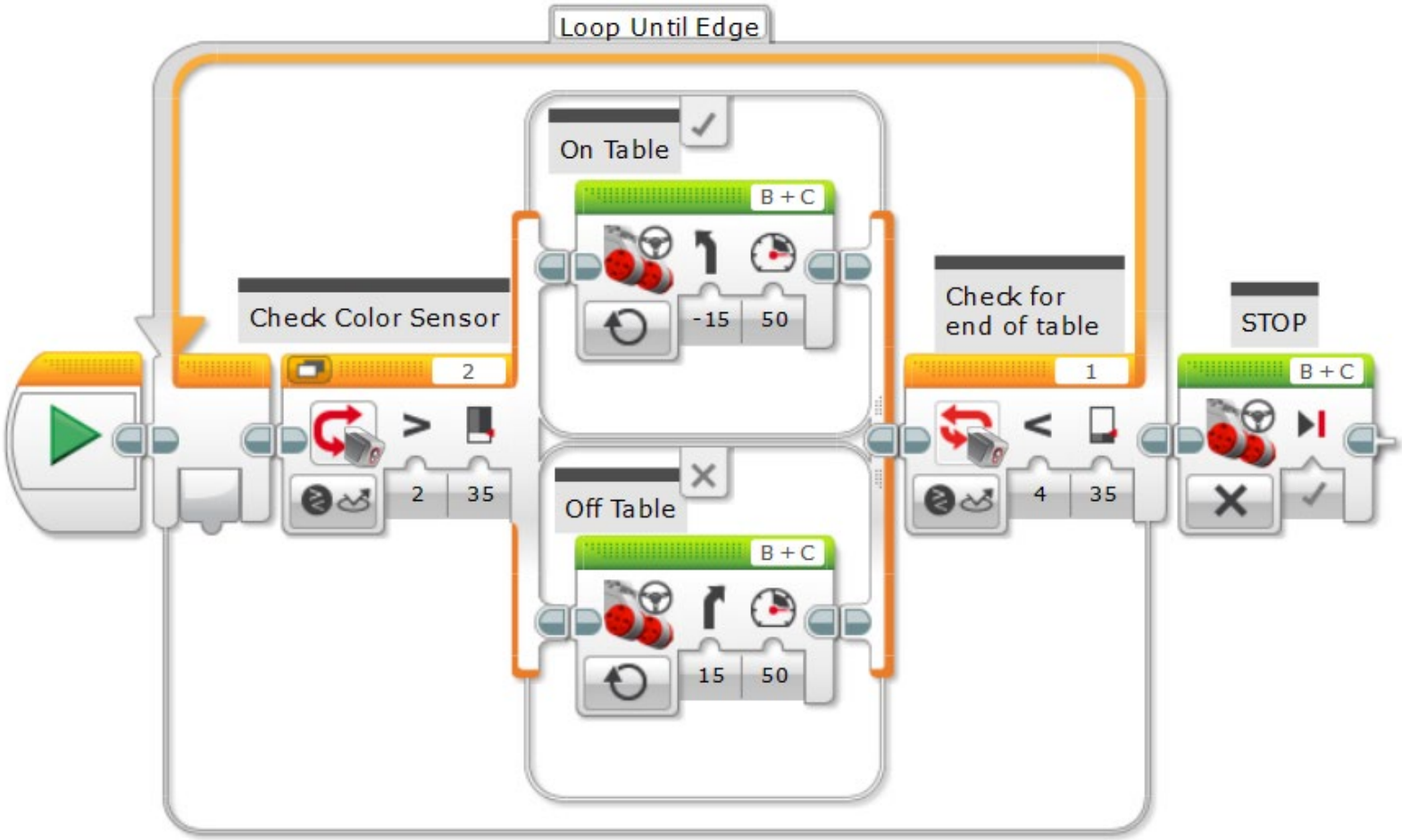
# Line following to the corner

- Recall our line following program LineFollowZZ
  - Let's modify the program to stop when the robot reaches the end of a corner of the table

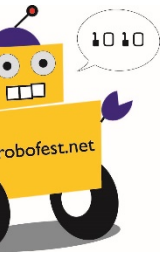



Using this program, the robot will line follow continuously. How can we make the robot stop when it reaches a corner?

# Line following to the corner




**Program: LineFollowZZStop.ev3** **YouTube:** https://youtu.be/Bb-FmLUeS0g




ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 2b

## Line following a given distance

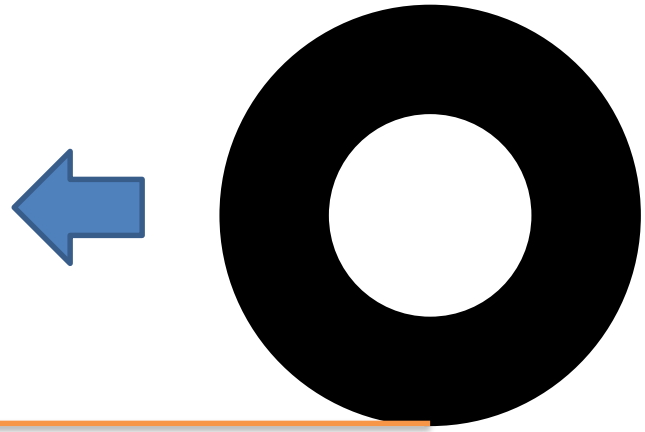
# Line following a given distance

- Approach
  - Modify LineFollowZZStop.ev3 to stop when the robot travels a given distance
- Tools needed
  - Line following
  - Measure distance traveled

# Measuring Distances

- How do we measure distance traveled?
- Let's determine how far the robot travels moving forward for 2 seconds

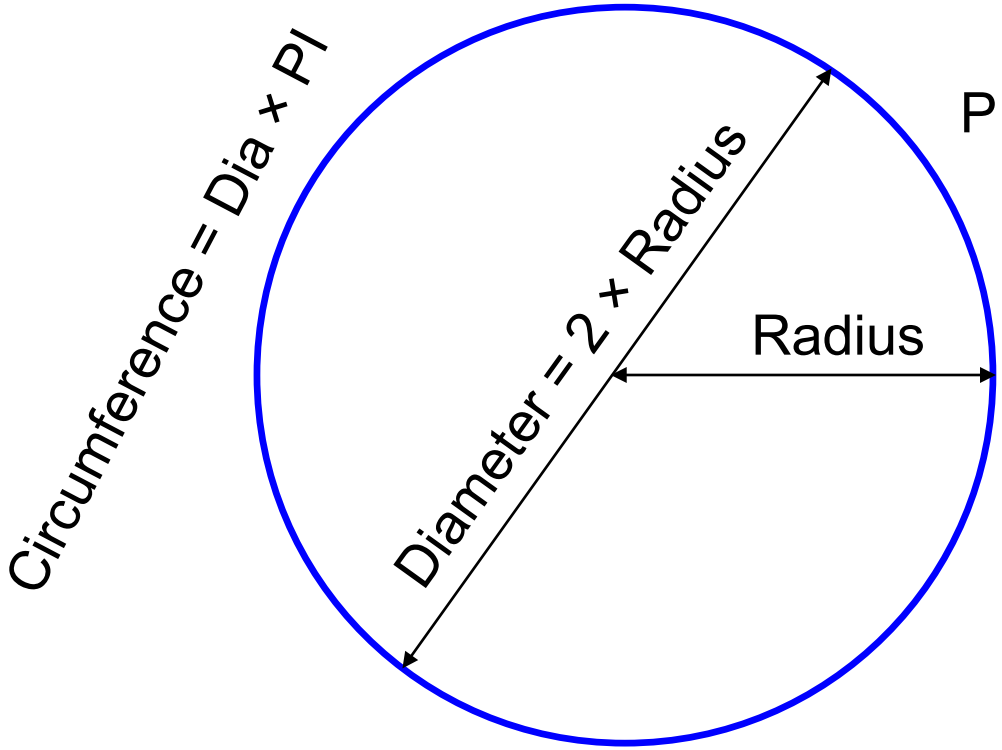
**Compute distance traveled by measuring the number of rotations of the wheel**



Distance

# Measuring Distances

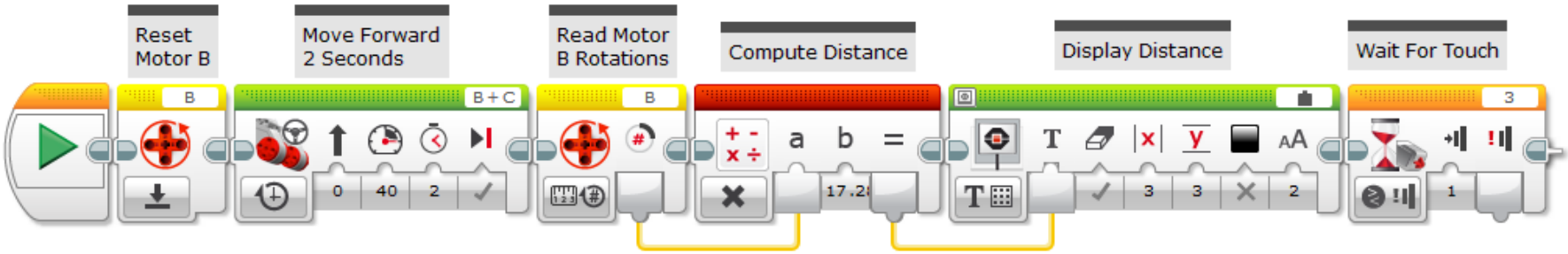
- Use the wheel geometry




**How can we use this information?**

# Measuring Distances

- For each rotation of the wheel, the robot will travel (Wheel Diameter) x (PI)
  - Distance = (Wheel Diameter) x (PI) x (# Rotations)
  - Distance = (5.5 cm) x (PI) x (# Rotations)
  - Distance = (17.28 cm) x (# Rotations)




**Program: MeasureDistance.ev3** **YouTube:** https://youtu.be/3fTWEjgACE0

# Measuring Distances

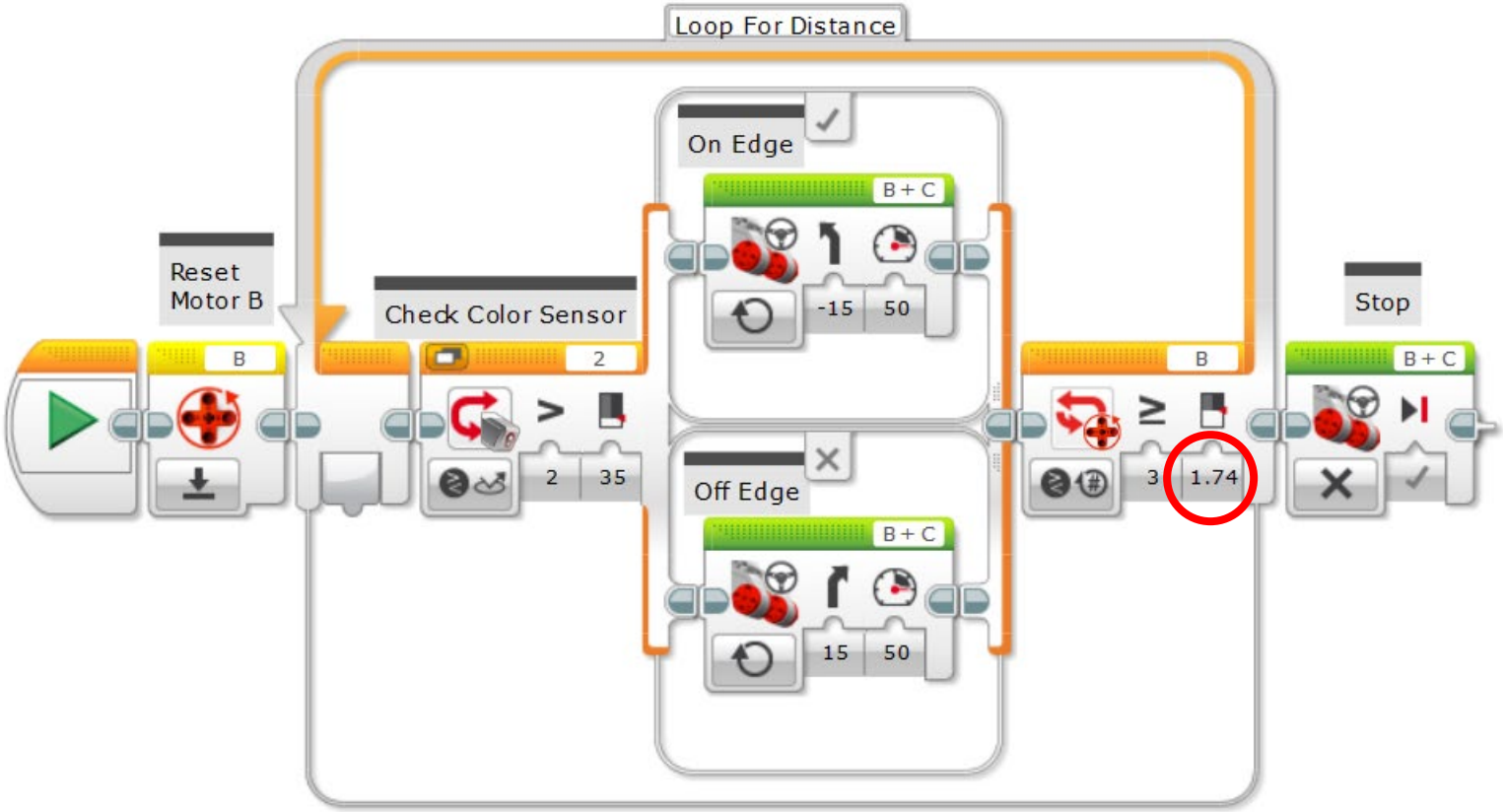
- Example
  - Let's program the robot to line follow for 30 cm
    - Distance = 30 cm
  - Number of rotations
    - Distance = (Wheel Diameter) x (PI) x (# Rotations)
    - Solve for (# Rotations)

$$(\#\ \text{Rotations}) = \frac{\text{Distance}}{(\text{Wheel Diameter}) \times (\text{PI})}$$

$$(\#\ \text{Rotations}) = \frac{30\ \text{cm}}{(5.5\ \text{cm}) \times (\text{PI})} = 1.74\ \text{rotations}$$

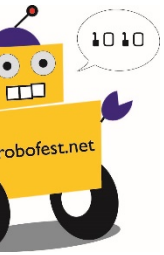
# Line following a given distance

- Line follow a desired distance




**Program: LineFollowDistance.ev3**

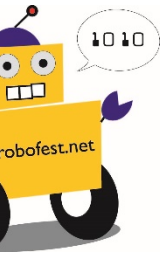
**YouTube:** https://youtu.be/JCvT_0vtYUo




ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 3

# Turning the robot




# Turning The Robot

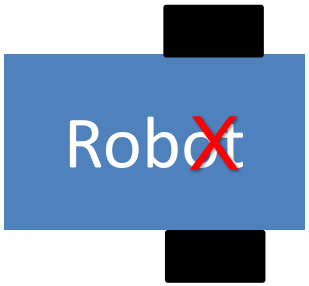
- For our example here, we wish to turn the robot 90 degrees

- There are many methods for turning a tripod robot. We will focus on two methods
  - "Spin" turn
  - "Swing" turn

# 90 Degree Spin

- Let's have the robot spin 90 degrees CCW
- The robot will rotate about center of the drive wheels (denoted by red X)

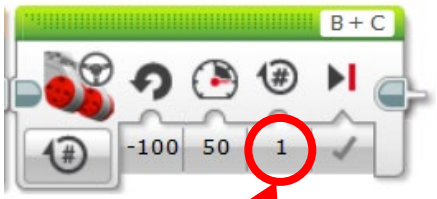
**Starting Position**




**Final Position**

# 90 Degree Spin

- To spin 90 degrees CCW, we use the Move Steering block as shown here




- Set the steering to -100. This causes:
  - Right wheel to rotate forward
  - Left wheel to rotate reward
  - Equal and opposite rotations

- Now, we need to determine the correct number of rotations

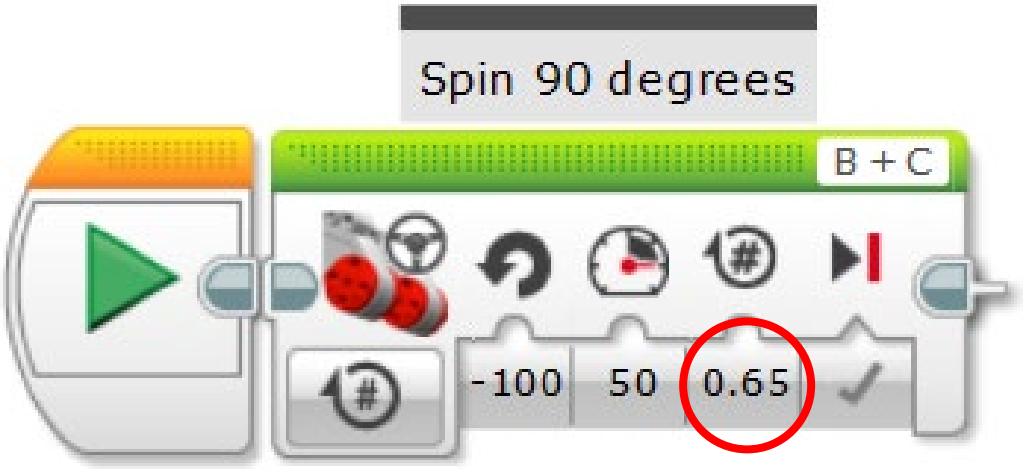
# 90 Degree Spin

- You can determine the proper number of rotations mathematically; however, the result typically needs some adjustment due to lash in the motors

- For today's class, we will use trial and error to find the number of rotations that cause the robot to turn 90 degrees

# 90 Degree Spin

- We can use one block to spin the robot




- For our sample robot, it takes 0.65 rotations to spin the robot 90 degrees

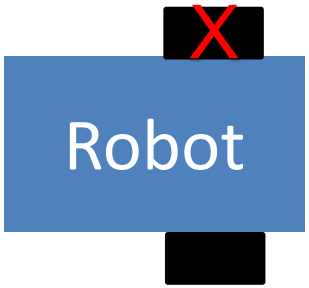
**Program: Spin90.ev3**

**YouTube:** https://youtu.be/I9ffG4Wz7Ek

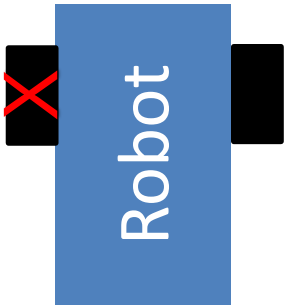
# 90 Degree Swing

- Let's have the robot swing 90 degrees CCW
- The robot will rotate about a locked wheel (denoted by red X)

**Starting Position**

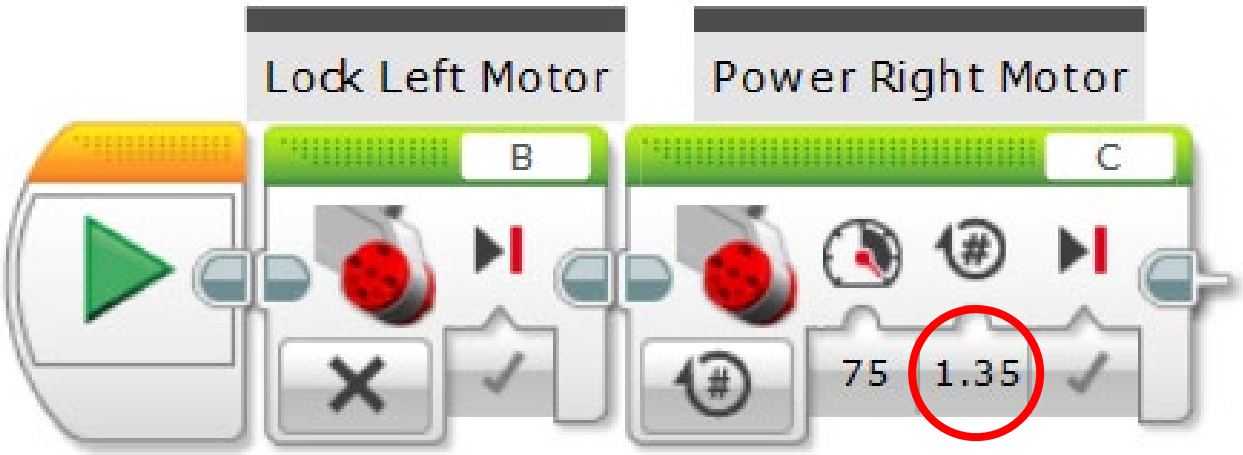



**Final Position**



# 90 Degree Swing

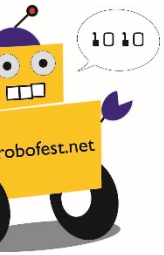
- To swing, we lock the left motor and power the right motor to turn the robot




- For our sample robot, it takes 1.35 rotations to swing the robot 90 degrees

**Program: Swing90.ev3**

**YouTube:** https://youtu.be/I9ffG4Wz7Ek




ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 4

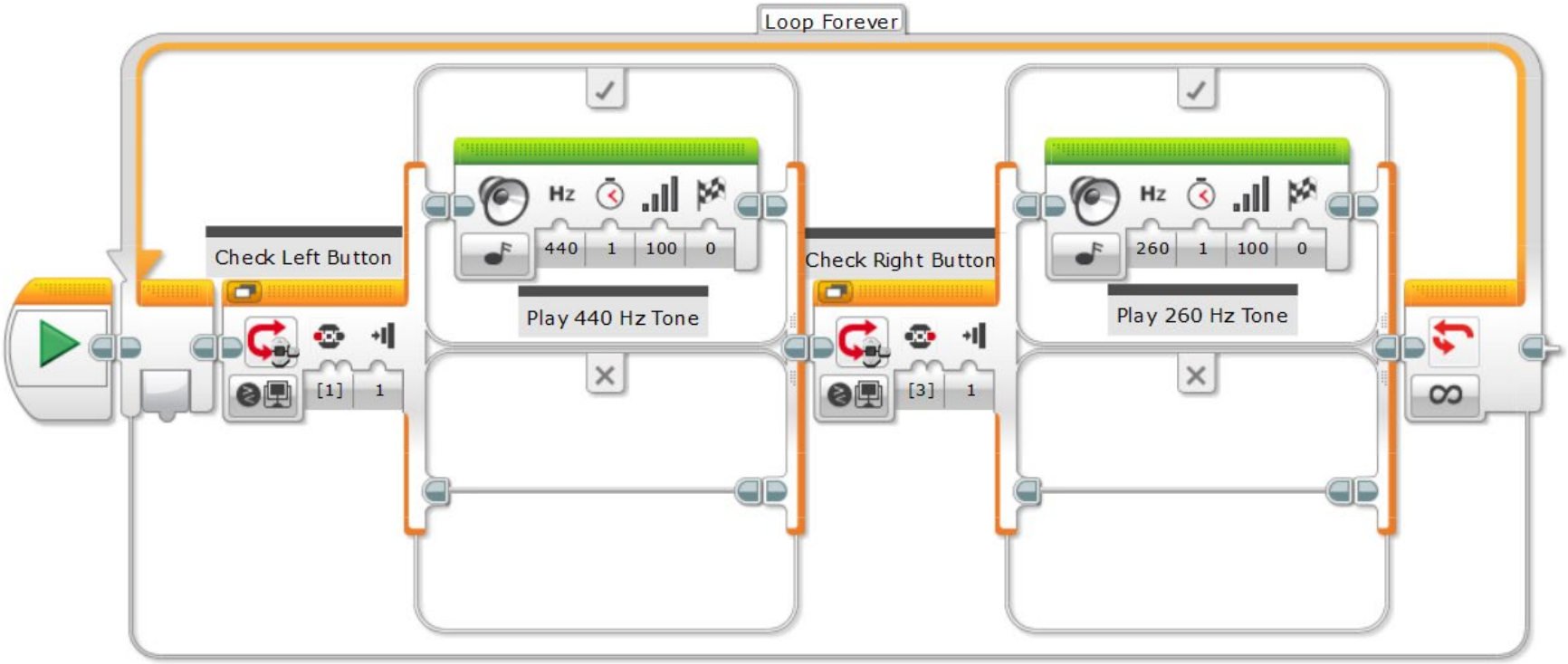
# Programming EV3 buttons

# Programming EV3 Buttons

- Now we will learn how the program the EV3 buttons
- Let's make the robot beep for one second when a button pressed
  - Left button pressed = 440 Hz tone
  - Right button pressed = 260 Hz tone
- Button identification
  - Left button = #1
  - Right button = #3

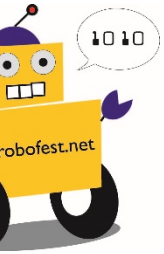
# Programming EV3 Buttons

- Inside an infinite loop, we use multiple switch blocks to play a tone when a button is pressed




**Program: ButtonBeep.ev3**

**YouTube:** https://youtu.be/CINfVUoCJ04



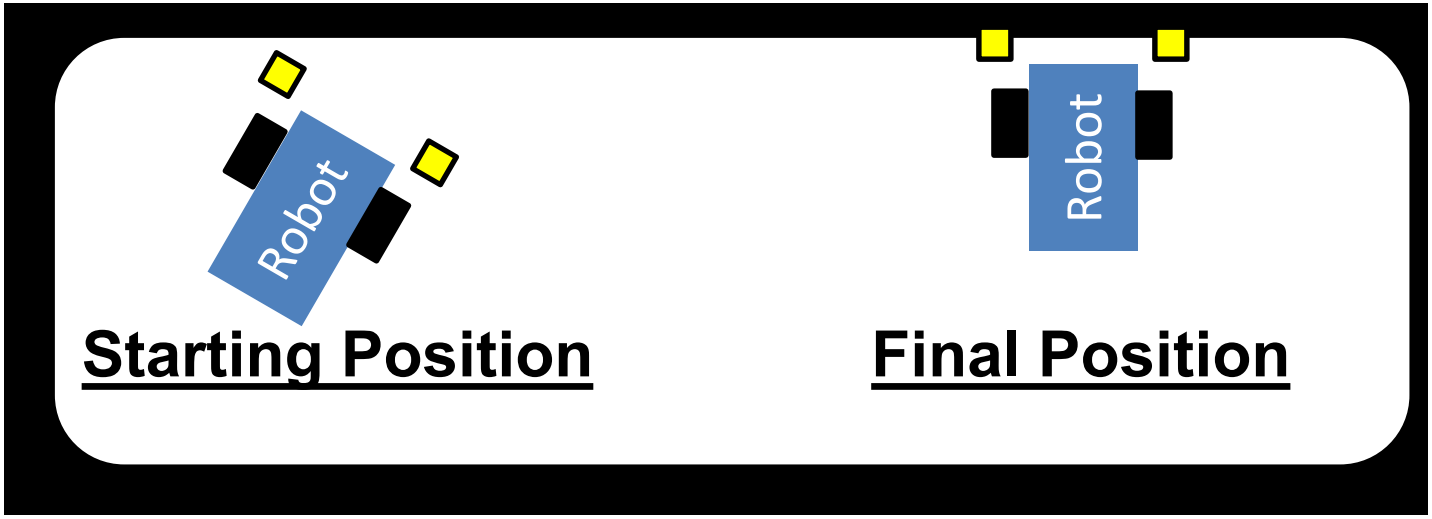

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 5

## Aligning the robot to an edge

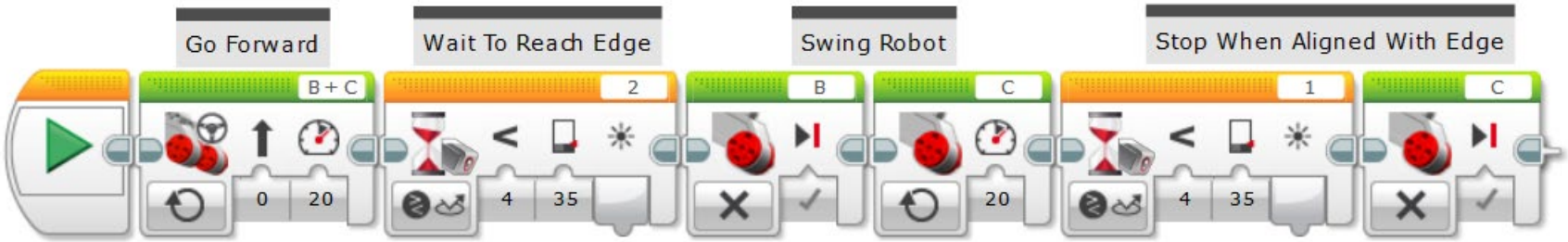
# Aligning the robot to an edge

- In some situations we desire align with robot to an edge of the table as shown below
- Assuming the starting position below, how can we program the robot to reach the final position that is aligned with the edge of the table?



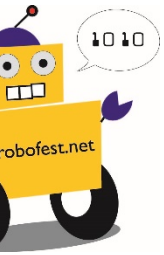
# Aligning the robot to an edge

- Travel until color sensor #2 reaches the edge, swing robot until it is aligned with the edge




**Program: AlignToEdge.ev3**

**YouTube:** https://youtu.be/FkGYni6PU0I



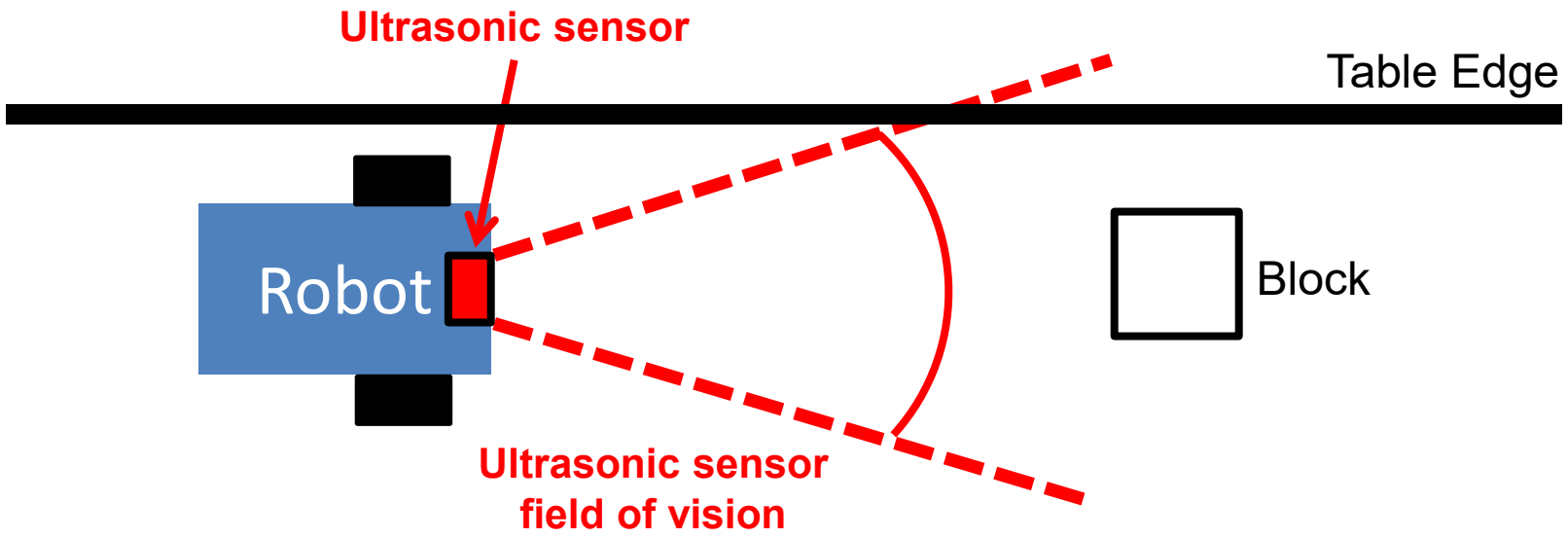

ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 6

# Finding a block

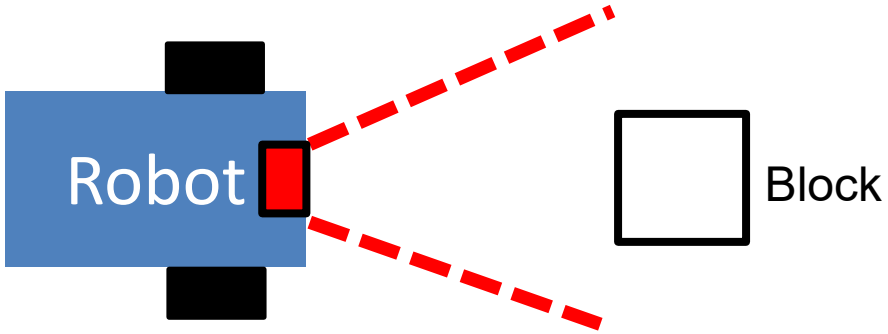
# Finding blocks

- We can use the ultrasonic sensor to determine if an object near the robot
- Here we will assume that we are following the edge of the table and wish to stop the robot once a block is “captured” by the robot
- Please note that you cannot use an ultrasonic sensor alone to determine if the robot captured a white or black block



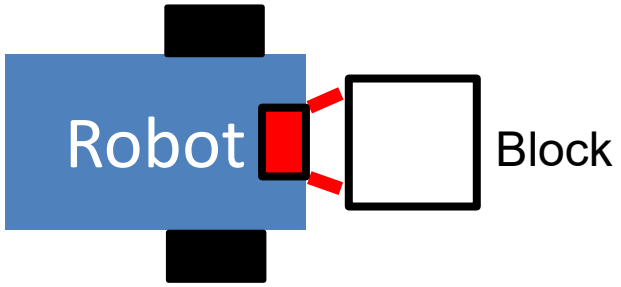
# Finding blocks

- Here we will using our line following program to follow the edge of the table and stop the robot when the block is very close to the ultrasonic sensor




**Ultrasonic sensor will read high values when the block is in far in front of the robot**

**Starting Position**

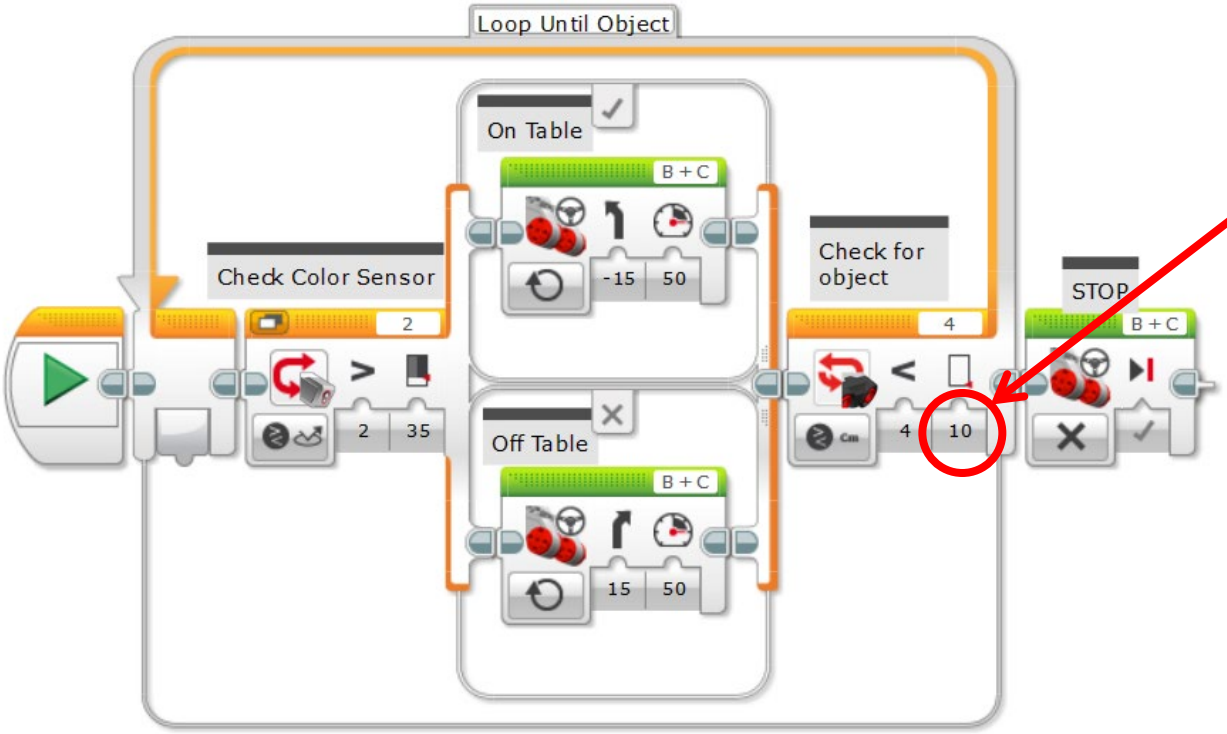



**Ultrasonic sensor will read low values when the block is in immediately in front of the robot**

**Final Position**

# Finding blocks

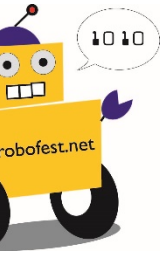
- Now, we travel along the edge of the table and stop if we find a block




We chose a low threshold. We can determine the appropriate value by testing the sensor readings with a block near the front of the robot.

**Program: DetectBlock.ev3**

**YouTube:** https://youtu.be/VHOUspGSNCQ




ROBOFEST
LAWRENCE TECHNOLOGICAL UNIVERSITY

# Task 7

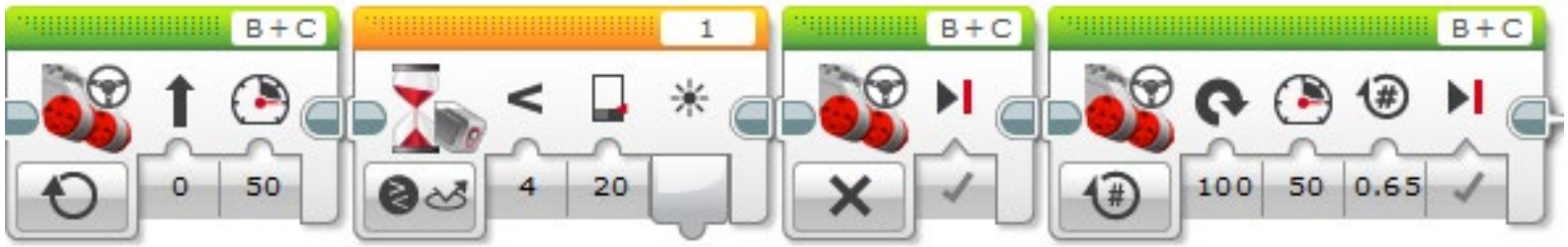
## Building MyBlocks

# My Blocks

- Solving the Robofest Game challenge will typically require a fairly large EV3 program (around 100 blocks is not unreasonable)
- Very large programs can be difficult to understand, navigate and use
- To alleviate this issue, the EV3 software has a My Block Builder to create custom blocks that can replace sections of your program

# My Blocks

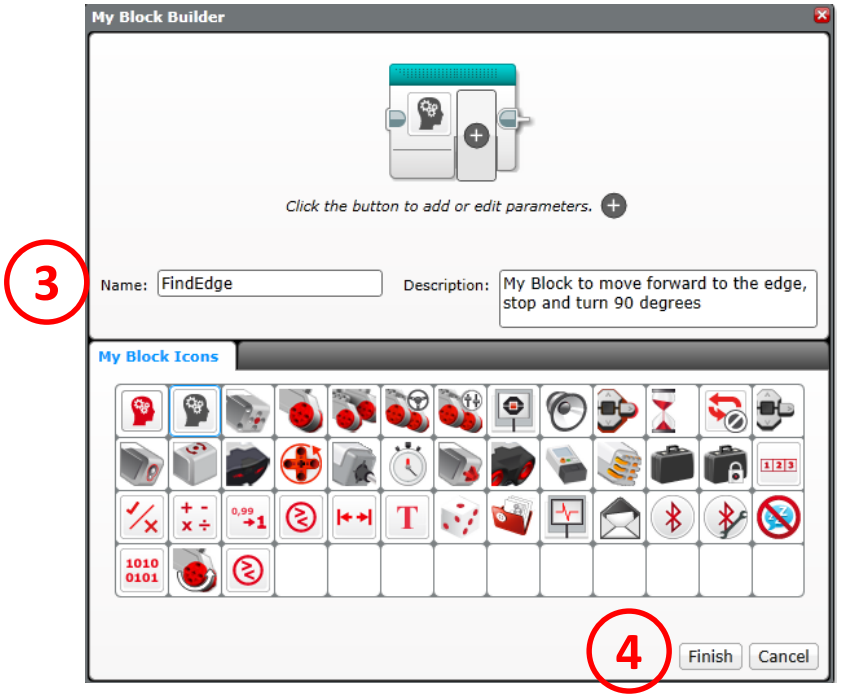
- For example, let's assume you have a section of code that completes the following:
  - Move forward until the edge of the table is found with color sensor 1, then stop
  - After stopping, rotate the robot 90 degrees
- The code may look like this




- My blocks will allow us to convert this to a single block

# My Blocks

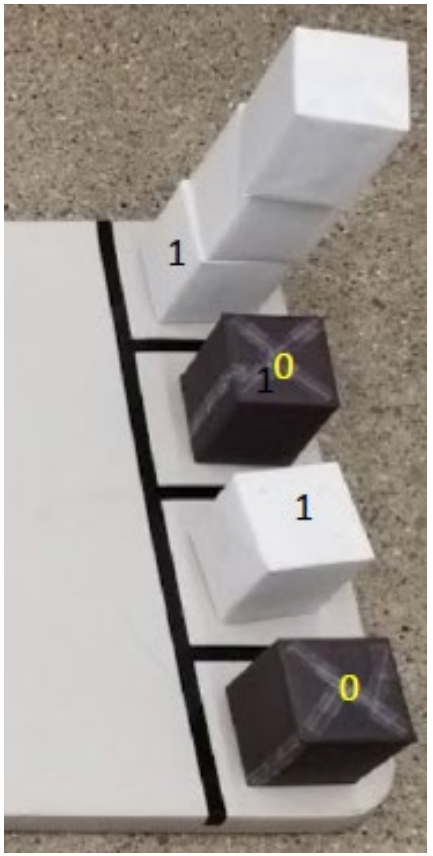
- Creating a My Block
  1. Select the section to convert
  2. Go to Tools -> My Block Builder
  3. Enter the block name, description and select icon
  4. Hit Finish

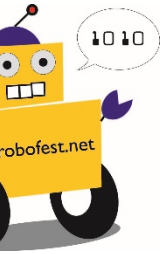




This creates a My Block called FindEdge that will be located in the My Blocks Pallet

# Stacking Blocks

- So far, we have discussed some key concepts in getting the bocks to the to the bit storage location
- The next challenge is to stack blocks as necessary
- Ideas for stacking blocks…
  - Fork lift concepts
  - Multi-level robot
  - Drop/throw the block





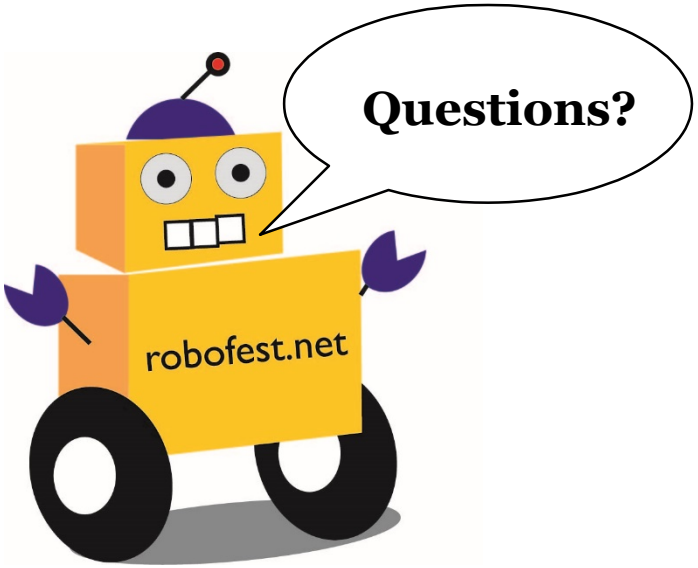

ROBOFEST LAWRENCE TECHNOLOGICAL UNIVERSITY

# Putting It All Together

- In this course we learned about
  - Understanding binary numbers
  - Finding the edge of the table
  - Following the edge of the table
  - Stop line following
    - When you reach a corner
    - When you reach a given distance
  - Turning the robot
  - Programming EV3 Buttons
  - Aligning the robot to an edge
  - Finding a block
  - Building MyBlocks

LAWRENCE TECHNOLOGICAL UNIVERSITY
ROBOFEST

# Little Robots, Big Missions




robofest@LTU.edu
LTU Computer Science