ROBOFEST 2021 GAME

# Stack Rolls

## Vex IQ Workshop
## Using Robot Mesh

This file can be found under the **eAcademy** **Workshops** page on the website

**www.robofest.net** **robofest@ltu.edu** 248-204-3568

**Room J233 Taubman Complex, LTU**
**21000 West 10 Mile Road, Southfield, MI 48075, USA**

# Please take Pre Assessment:

https://docs.google.com/forms/d/e/1FAIpQLScCmm6Fnb3LaKvWiFDGZxELXXc9O5KZ_tYm8M-7zwvqFettag/viewform?usp=sf_link

This file can be found under the **eAcademy** on the website **www.robofest.net** along with online training and certifications

LAWRENCE TECHNOLOGICAL UNIVERSITY
ROBOFEST

# 2021 Workshops

Presented by

Lawrence Technological University

Computer Science

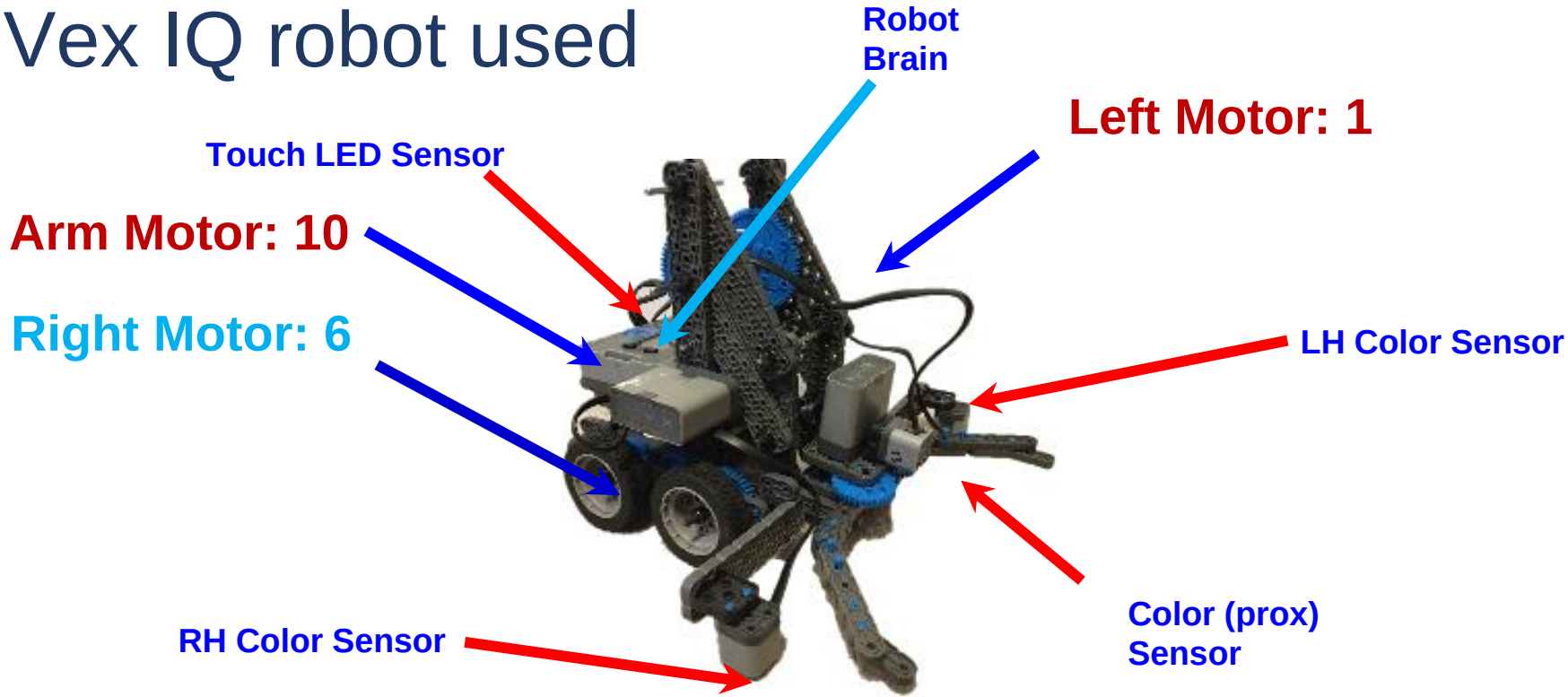
# Course Overview

- 2021 Robofest competition Stack Rolls
  - Autonomous robot that get points by moving and stacking rolls
- SPbot introduction
- Using SPbot to solve the Stack Rolls challenge

# 2021 Robofest Competition

- Video overview
- Key tasks
    - Task 0: Finding the edge of the table
    - Task 1: Following the edge of the table
    - Task 2: Stop line following when you reach a corner
    - Task 3: Stop line following when you reach a given distance
    - Task 4: Finding a Paper Roll
    - Task 5: Turning the robot
    - Task 6: Aligning the robot to an edge
    - Task 7: Manipulating Paper Rolls
    - Task 8: Building MyBlocks

# Vex IQ robot used



# Remember the connections!

- Left Motor connects to **1**
- Right Motor connects to **6**
- LH Color sensor connects to port no. **7**
- RH Color sensor connects to port no. **5**
- Touch LED sensor connects to port no. **4**
- Color(prox) sensor connects to port no. **2**
- Arm Motor connects to **10**

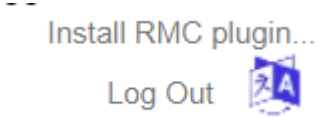
# What is Robot Mesh Studio?

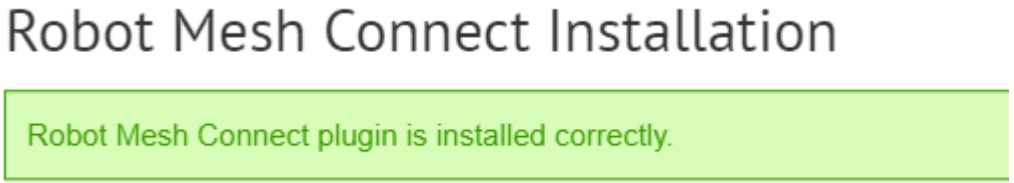
- Programming software for VexIQ
- Uses a drag and drop interface (Blockly, same as Scratch)
- Based on Python programming language

Robot MESH STUDIO Education Edition

# Getting Started

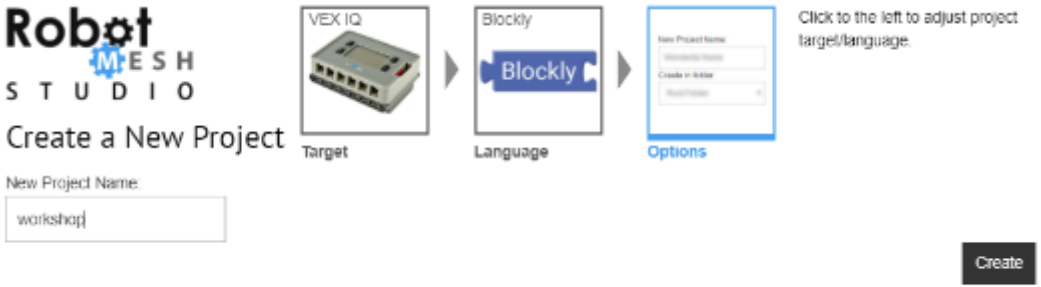
- Go to robotmesh.com/studio
- Go to signup and create an account
- Click on “install RMC plugin” and follow instructions to install



# Create a New Program

- Click on “New Project”
- Make sure these are selected
  1. Target: “Vex IQ”
  2. Language: “Blocky”
- Name the program “Workshop”
- Click on “Create”



# Connect Robot

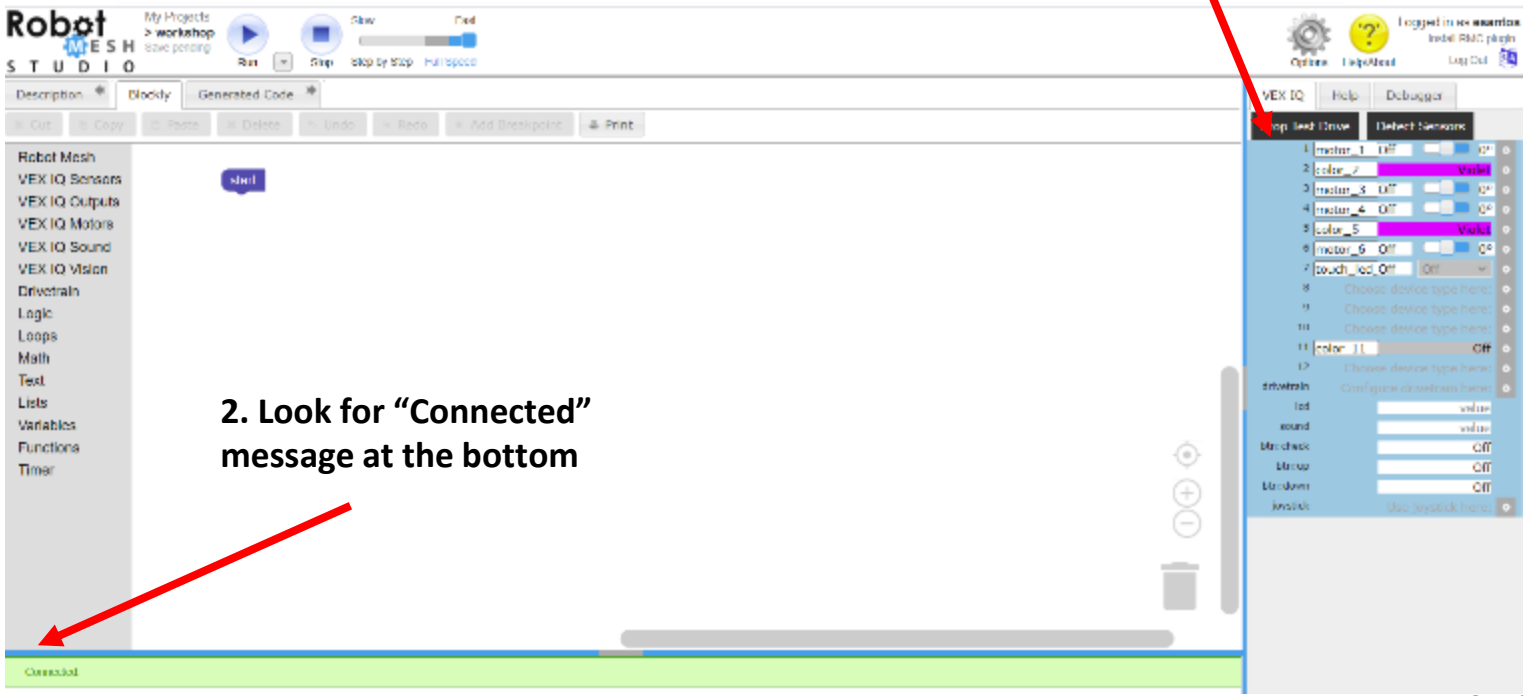
- Connect one end of programming cord into computer
- Connect the other end to the robot brain
- Turn on the Robot by pressing the check button

Check button

Programming cord

# Check Connection

**1. Click on “Test Drive”**

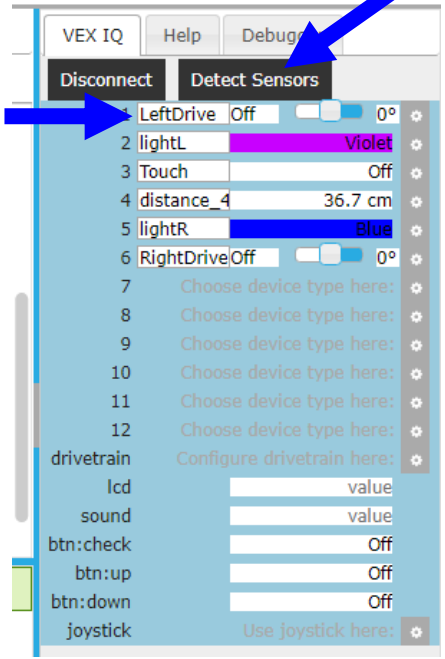



**2. Look for “Connected” message at the bottom**

# Motor and Sensor Setup

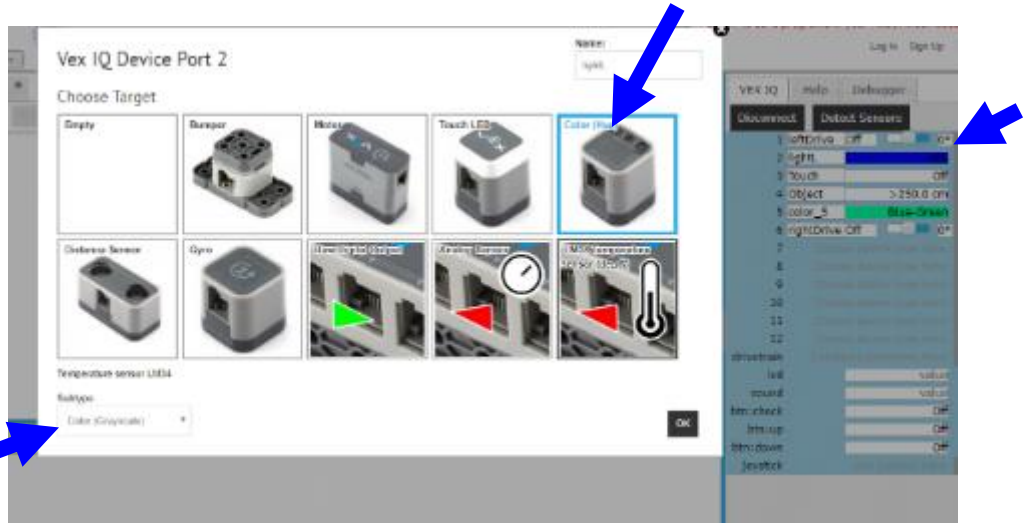
1. Click Here

2. Change to names that have meaning

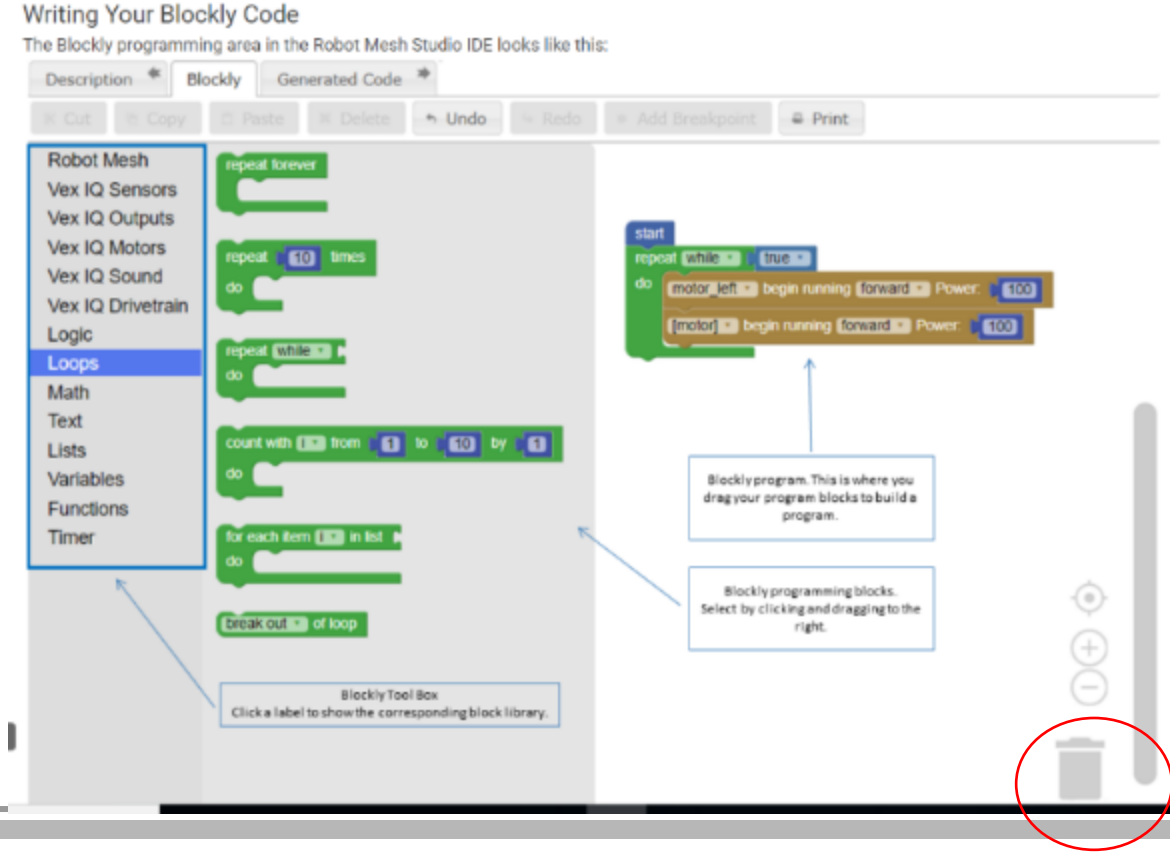


# Motor and Sensor Setup

Change Color Sensors to grayscale mode

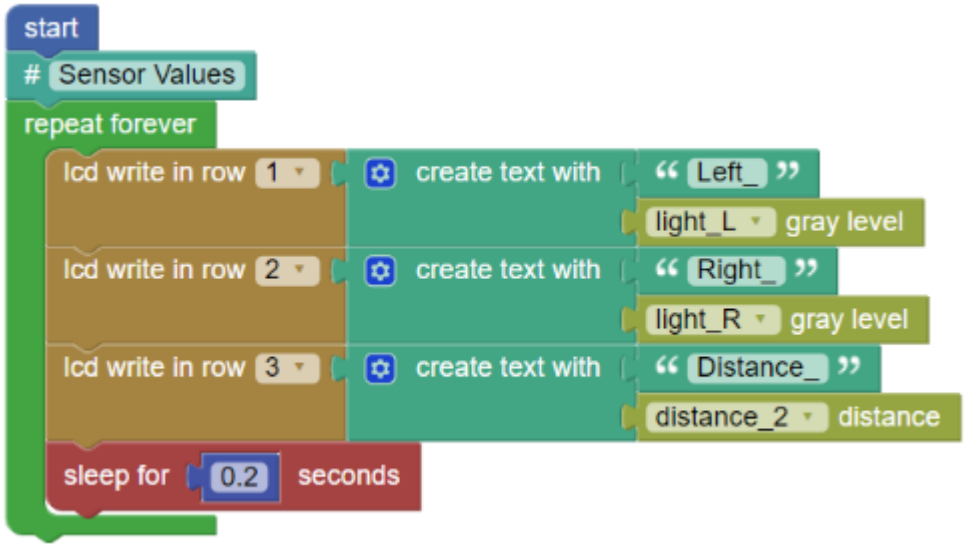


# Blockly Programming



# Display Sensor Values on LCD

Use the LCD Display to Display the Value of a

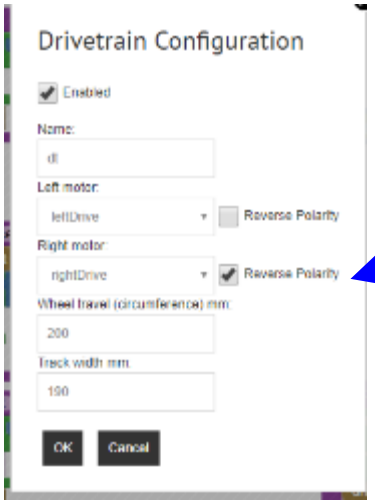


# Drivetrain functions

- Robot Mesh has “Drivetrain” programming blocks
- Blocks enable shorter programs and faster coding
- You need to configure the drivetrain in the right side Interface Panel
- Click on the gear symbol

# Drivetrain Configuration

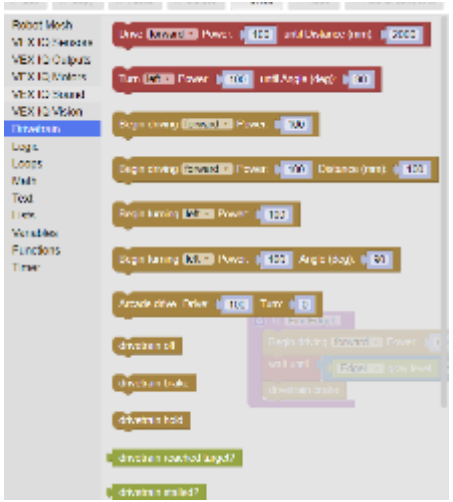
- Configuration dialogue window
- Configuration below should work
- Click “ok” when done




Make sure Left Motor is NOT reversed and Right motor is REVERSED

# Drivetrain blocks

- Found on left menu in “Vex IQ Drivetrain”
- Enables you to program both drive motors with one block



Drive 200 mm using Drivetrain block

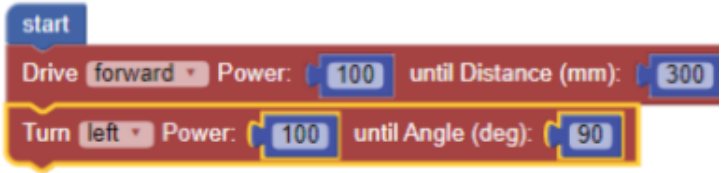

Drive Forward 200 mm without Drivetrain blocks

# Task 0

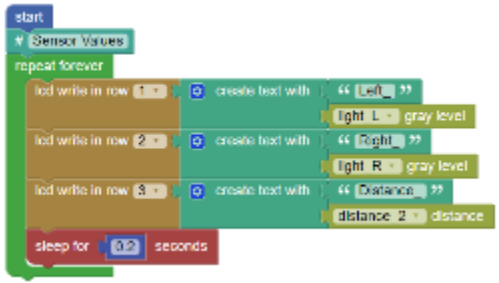
## Finding the edge of the table

# Drive Forward and Turn

- Add this code to our program
- Use a new start block

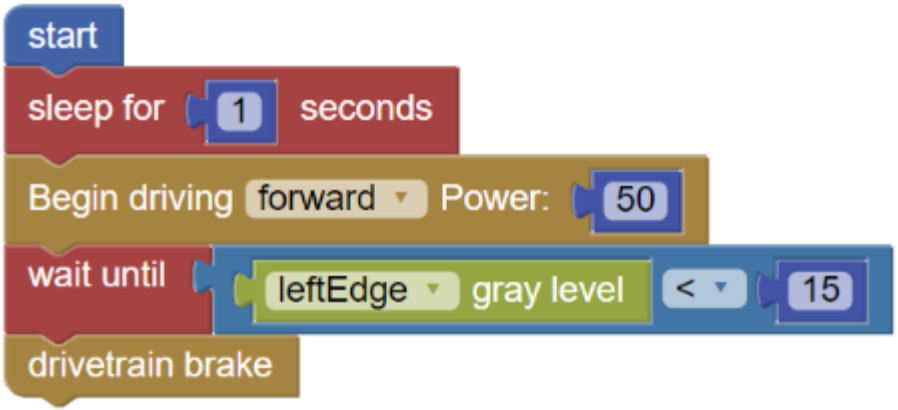



- Both the new code and the display code can be run at the same time




# Find the edge of the playing field

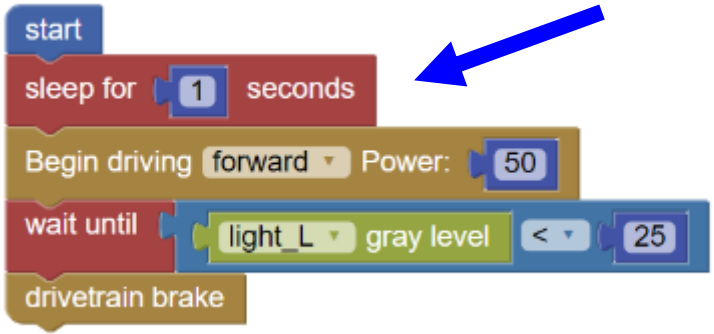
- Delete the “forward” and “turn” blocks and replace with the code below




https://youtu.be/Q-2CG9p8fR0

# TIP

- Put a small delay at the start of the program
- Allows for initialization of sensors
- Without delay, initial sensor readings may be incorrect

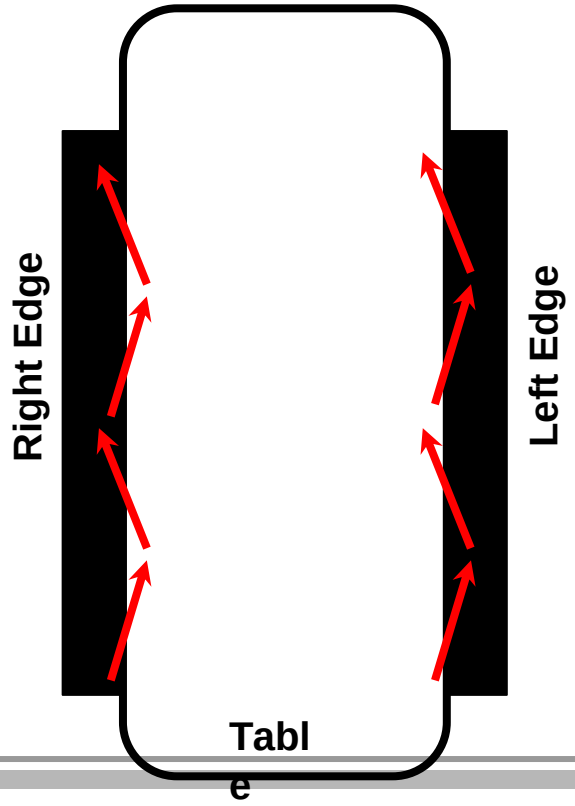


# Task 1

## 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

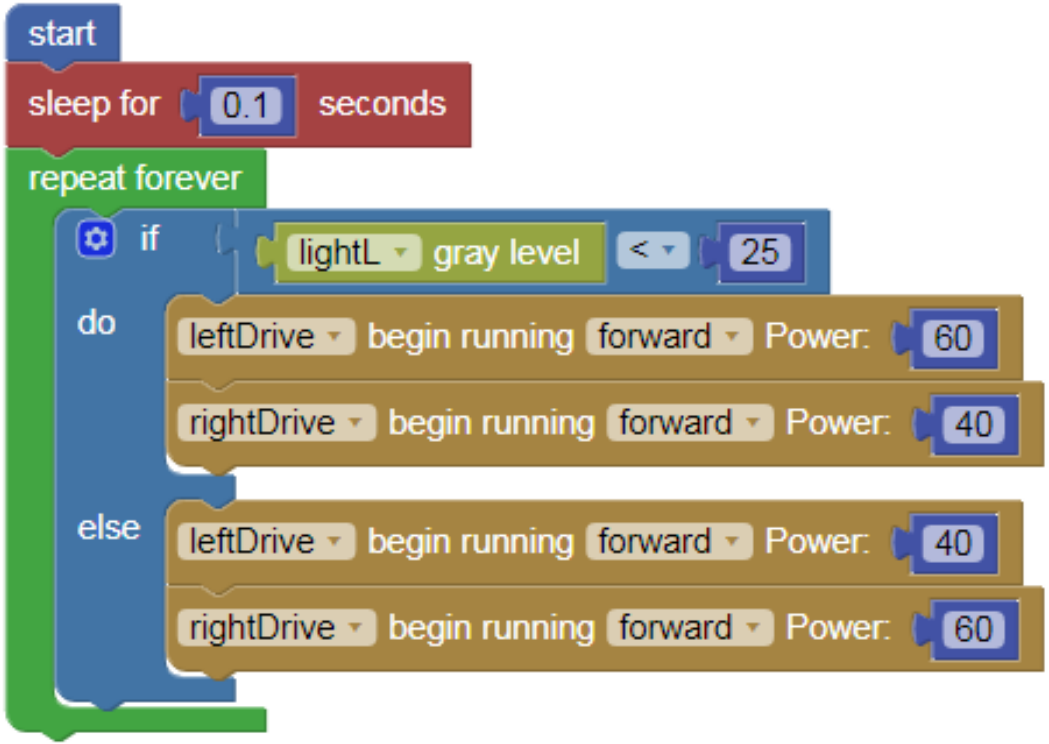


# Follow The Edge Of The Table

- Light sensor settings example
  - Off table = 5
  - On table = 45
  - Median threshold = (5+45)/2 = 25

- Two cases
  - Light sensor reading > 25.  On table.
  - Light sensor reading < 25.  Off table.

# Simple Line Following Algorithm

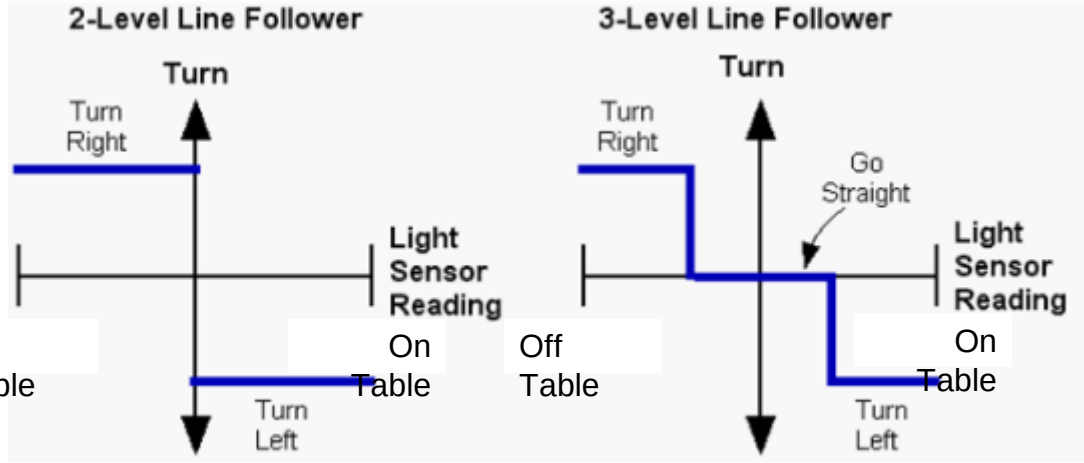



**YouTube:**

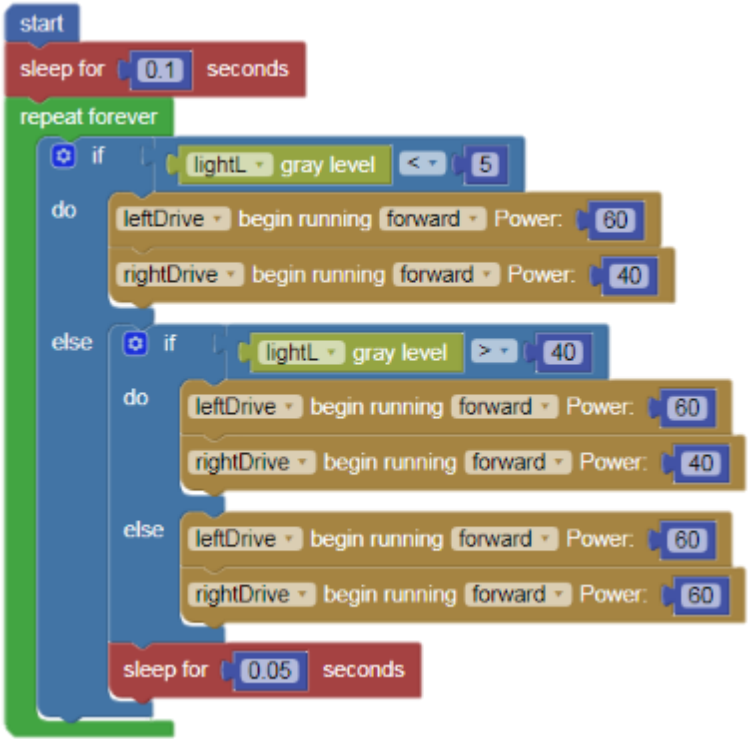
https://youtu.be/AFRbJX_IpcE

# 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)



# How to improve our line following algorithm

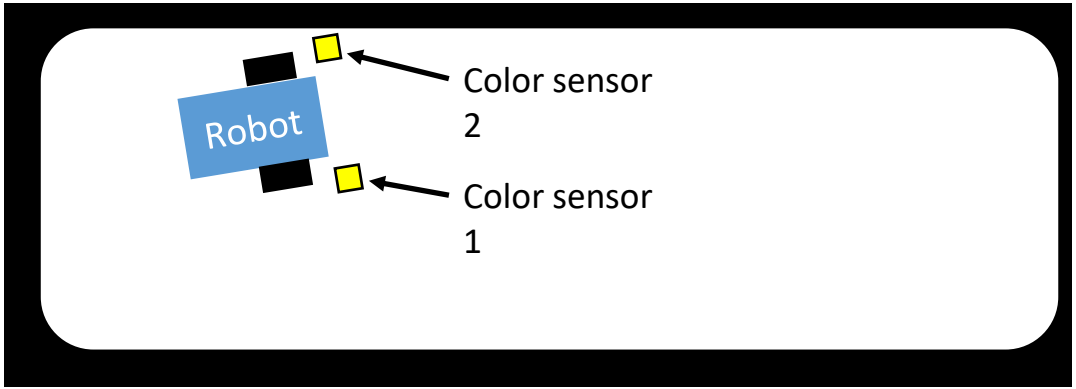


# Task 2

## 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 second color sensor
  - Sensor 1 used to follow the edge of the table
  - Sensor 2 used to locate the end of the table



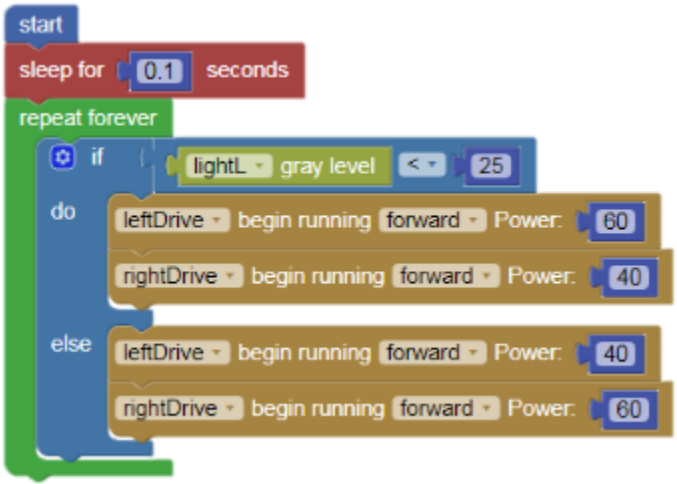
# Line following to the corner

# 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

# Travel around the  table

- Recall our line following program LineFollow
  - Let's modify the program to stop when the robot reaches the end of an edge of the table




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

# Follow the playing field edge until corner


**YouTube: https://youtu.be/1z_GVJNjD94**

# Task 3

# Line following a given distance

# Line following a given distance

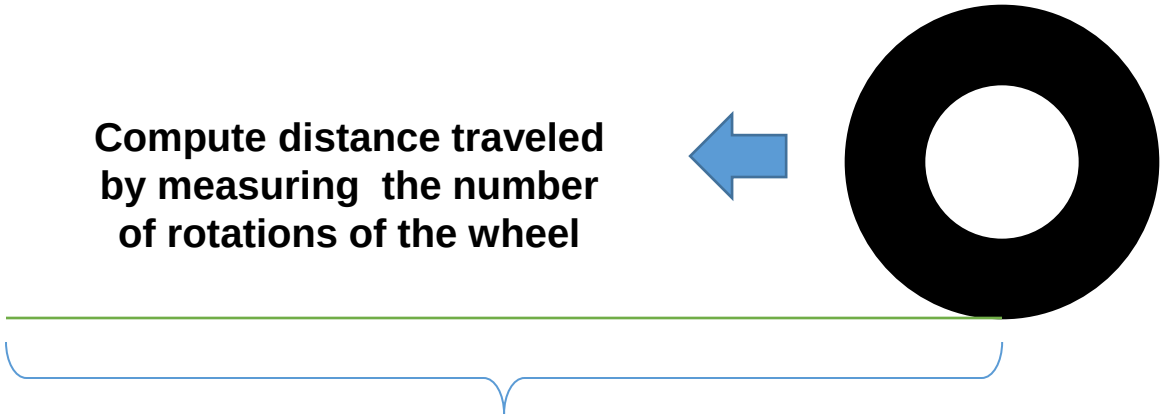
- Approach
  - Modify LineFollow to stop when the robot travels a given distance

- Tools needed
  - Line following
  - Measure distance traveled

# Measuring Distances

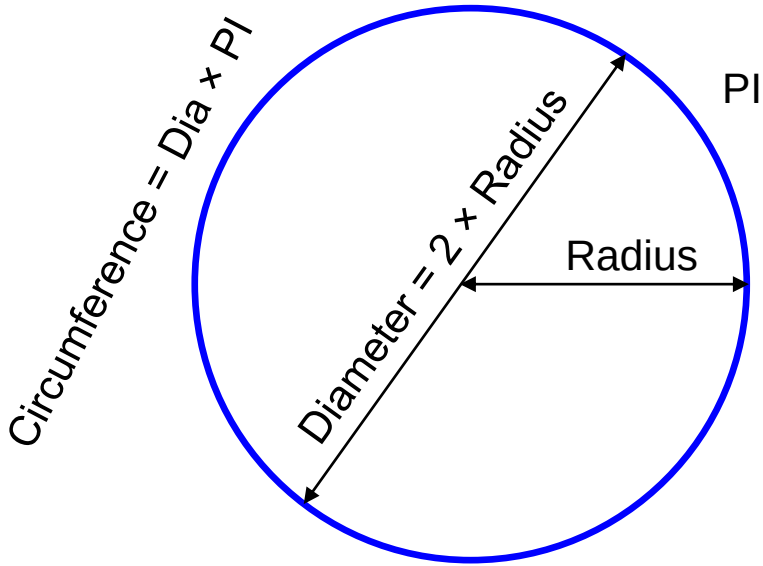
- How do we measure distance traveled?

- Let’s determine how far the robot travels moving forward for 2 seconds



# Measuring Distances

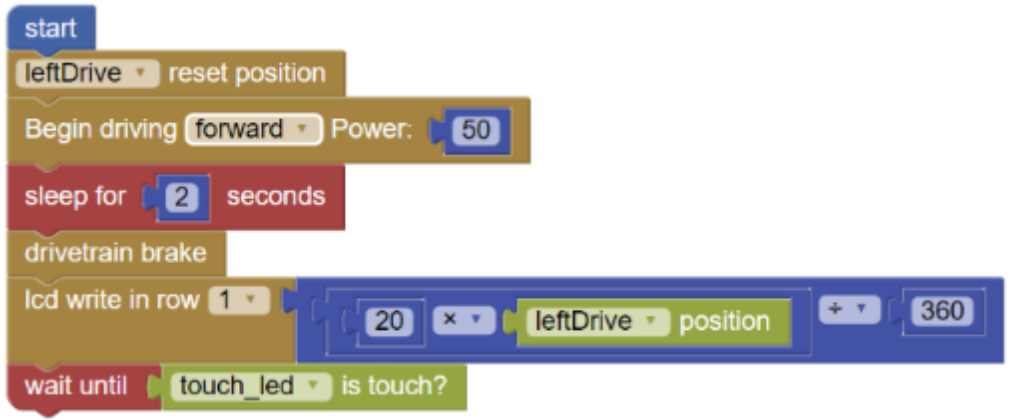
- Use the wheel geometry




**How can we use this information?**

# Measure Distances

- For each rotation of the wheel, the robot will travel (Wheel Diameter) x (PI)
  - Distance = (Wheel Diameter) x (PI) x (# Rotations)
  - Distance = (63 mm) x (PI) x (# Rotations)
  - Distance = (20cm) x (# Rotations)




**YouTube:** https://youtu.be/9pqK0uBtbzE

# Measuring Distances

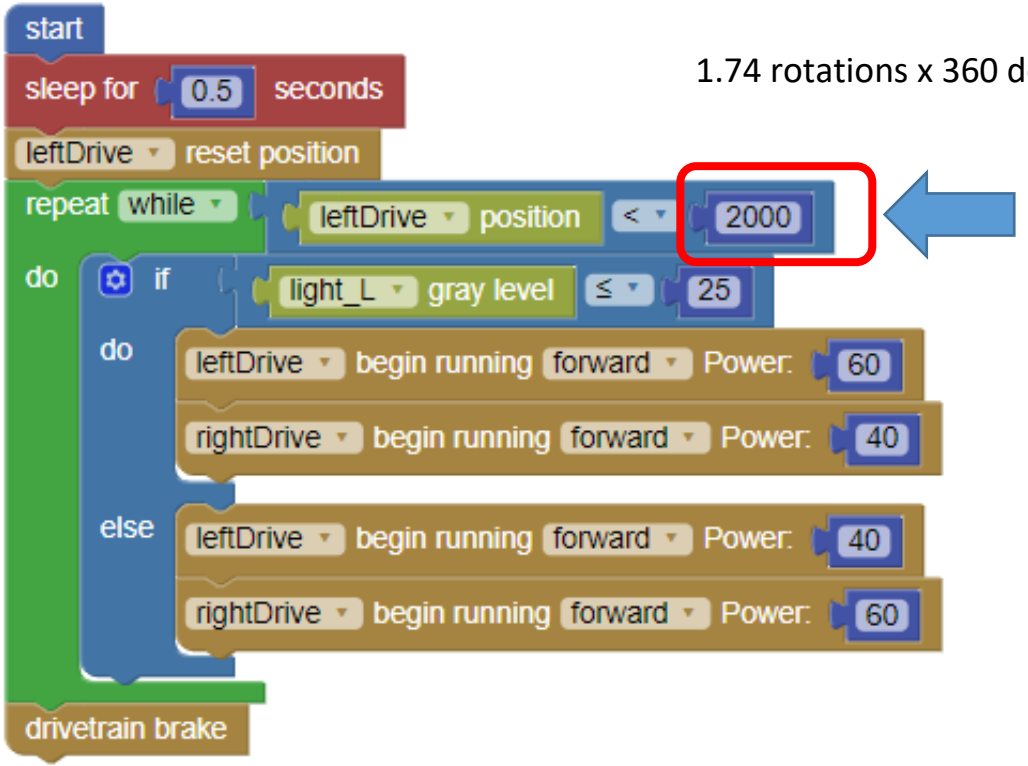
- Example
  - Let's program the robot to line follow for 30 cm
    - Distance = 30 cm
  - Number of rotations
    - Distance = (Wheel Diameter) x (PI) x (# Rotations)
    - Solve for (# Rotations)

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

$$(\#\ \text{Rotations}) = \frac{36\ \text{cm}}{(5.5\ \text{cm}) \times (\text{PI})} = 1.74\ \text{rotations}$$

Rotations must be converted to degrees

1.74 rotations x 360 degrees = 626




Change this number to go the desired distance

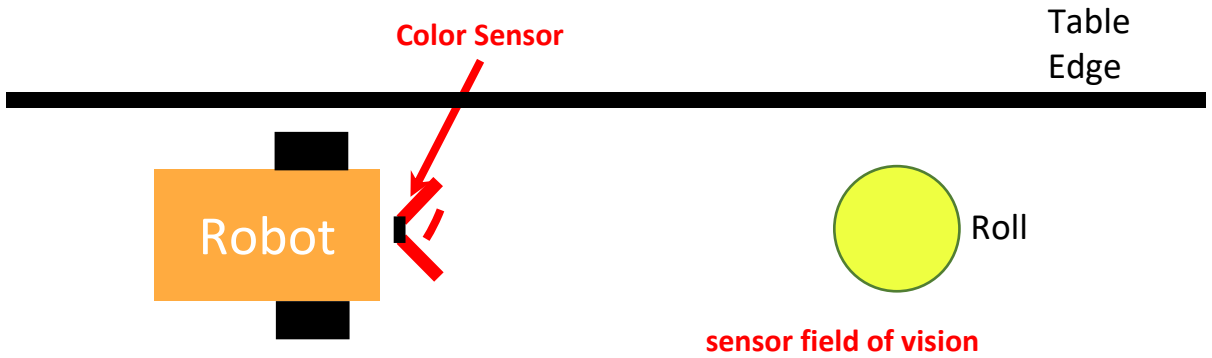
https://youtu.be/K1suXAzZuCA

# Task 4

# Finding a Roll

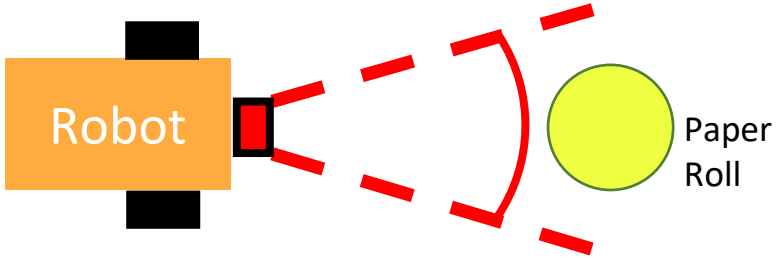
# Finding a Rolls

- We can use a color sensor to determine if an object is near the robot
- Here we will assume that we are following the edge of the table and wish to stop the robot once a Paper Roll close to the robot



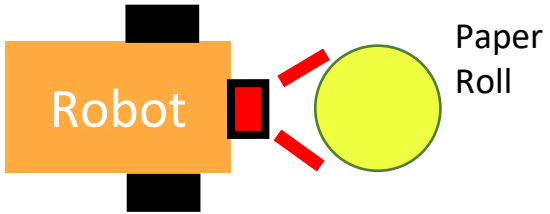
# Finding a Paper Roll

- Here we will use our line following program to follow the edge of the table and stop the robot when the Paper Roll is close to the color sensor




Color sensor will read no values when the Paper Roll is far away

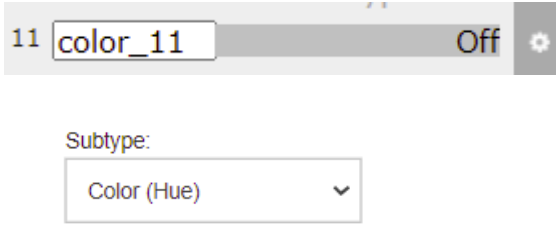
**Starting Position**



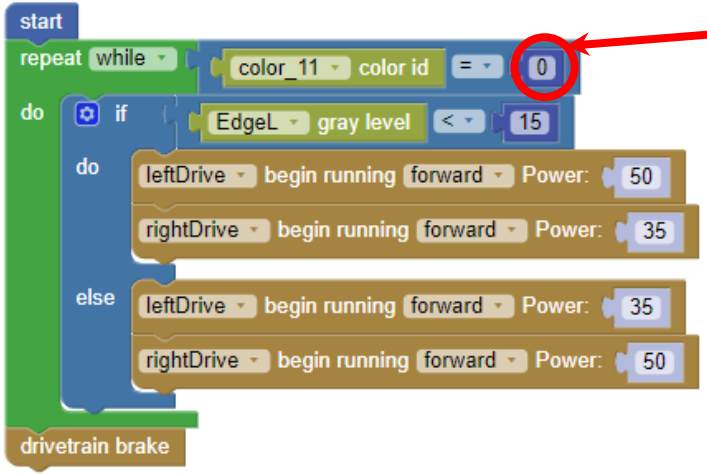

Color sensor will read values when the Paper Roll is close to the robot

**Final Position**

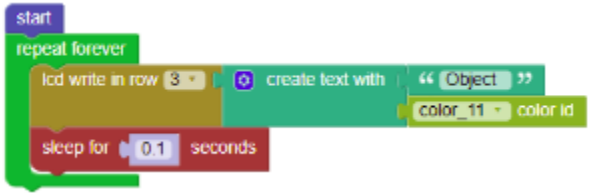
# Finding a Paper Roll- Color Mode




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




We can determine the appropriate value by testing the sensor readings with a Paper Roll near the front of the robot.

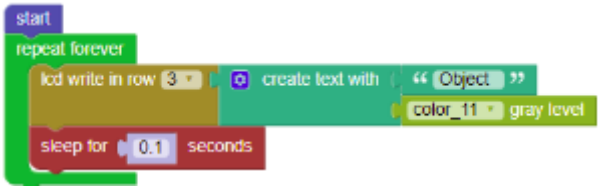


# Finding a Paper Roll- Grayscale

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


We can determine the appropriate value by testing the sensor readings with a Paper Roll near the front of the robot.






https://youtu.be/-zrYApNv5jI

# Task 4

# Turning the robot

# 90 Degree Spin

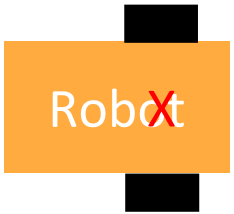
- You can determine the proper number of rotations mathematically; however, the result typically needs some adjustment due to lash in the motors

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

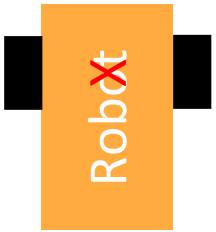
# 90 Degree Spin

- 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**

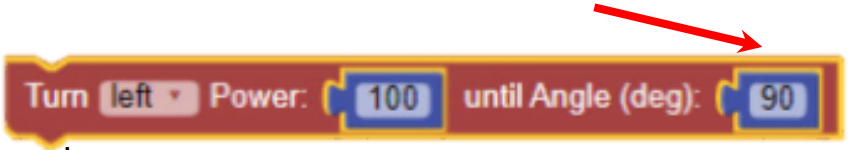



**Final Position**



# Spin 90 Degrees

- To spin 90 degrees CCW, we use the turn block from the “Drivetrain” menu
- Set “Turn” to “left” and




  - Right wheel to rotate forward
  - Left wheel to rotate reward
  - Equal and opposite rotations
- Now, we need to determine the correct number of degrees

# Spin 90 Degrees

- 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 trail and error to find the number of degrees that cause the robot to turn 90 degrees

# Spin 90 Degrees

- We can use one block to spin the robot

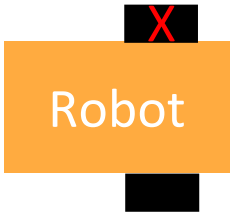

- For our sample robot, it takes 130 degrees to spin the robot 90 degrees

**YouTube:**
https://youtu.be/AUqXzsnf0NI

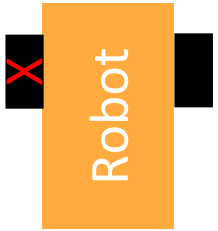
# 90 Degree Swing

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

**Starting Position**

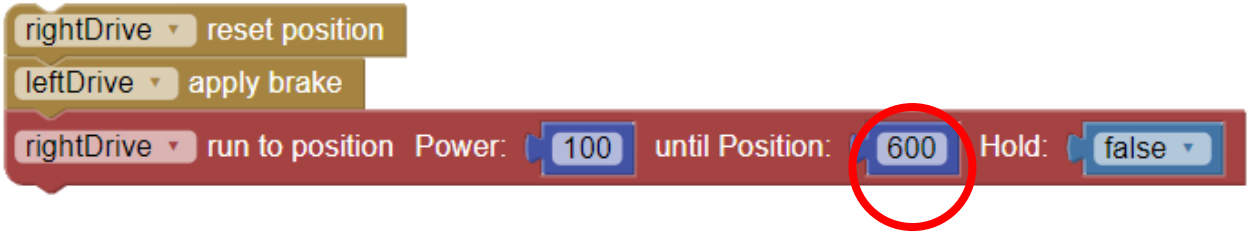



**Final Position**



# 90 Degree Swing

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



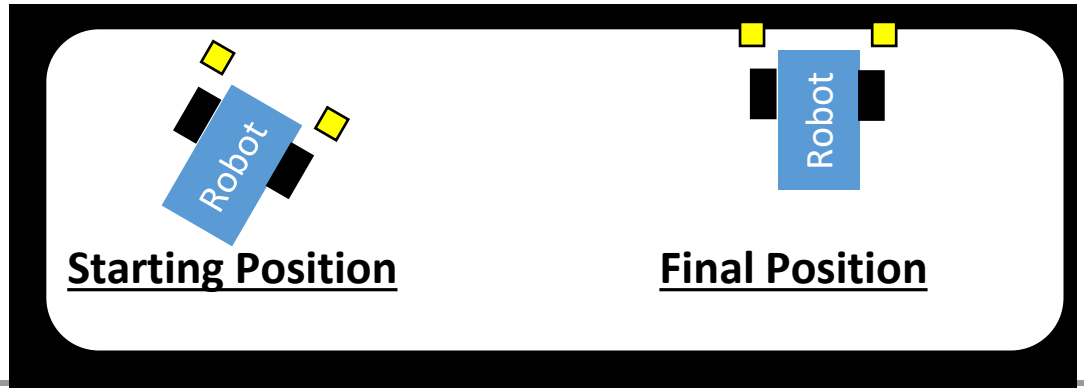

- For our sample robot, it takes 600 degrees to swing the robot 90 degrees

# Task 6

# Aligning the robot to an edge

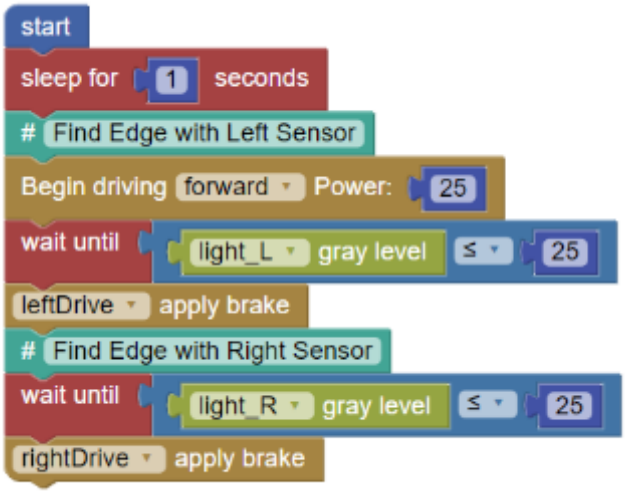
# Aligning the robot to an edge

- 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

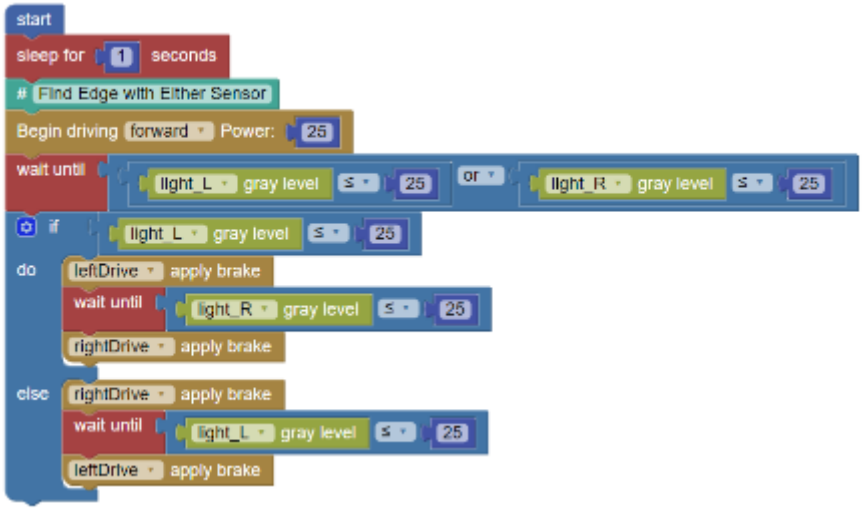
- Travel until LH color sensor reaches the edge, swing robot until it is aligned with the edge




**YouTube**: https://youtu.be/Jq0L2ekQARk

# Aligning the robot to an edge-either sensor

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





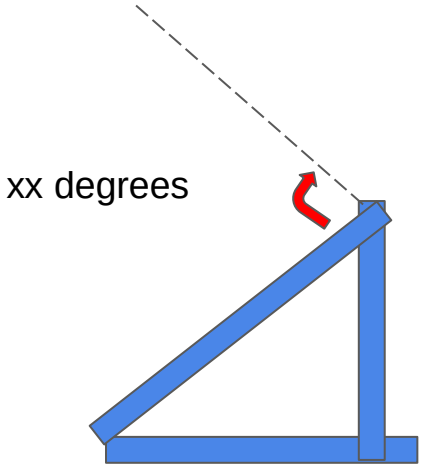

**YouTube**: https://youtu.be/Jq0L2ekQARk

# Task 7

## Manipulating Paper Rolls
## Moving the Arm and Claw

# How to Control the Arm and Claw

- Time
- Rotation(encoder degrees)




Arm all the way down
0 degrees

# Moving the Arm- Using time

- Advantages
  - Simple
  - Easy to program
  - Will not get stuck
- Disadvantages
  - Can be imprecise
  - Repeatability
- How to do it
  - Set a motor block to “seconds”
  - Select the motor port
  - Select the direction
  - Select the duration

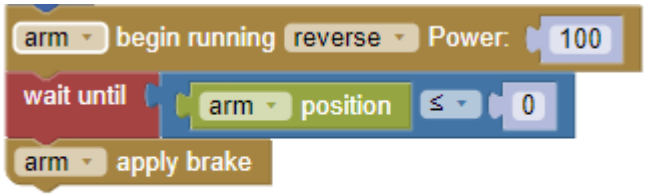

Raise Arm


Lower Arm

# Moving the Arm- Using encoder

- Advantages
  - More precise
  - More repeatable
- Disadvantages
  - More difficult to program
  - Can get stuck
- How to do it
  - Establish a “zero” point
  - Determine direction of motor
  - Set limits


Raise Arm

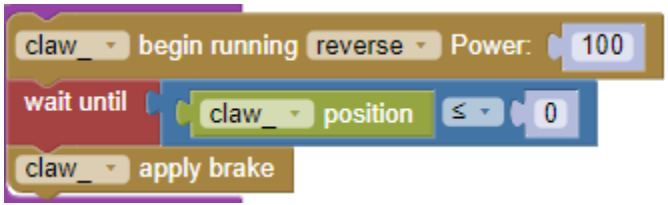



Lower Arm

# Moving the Claw

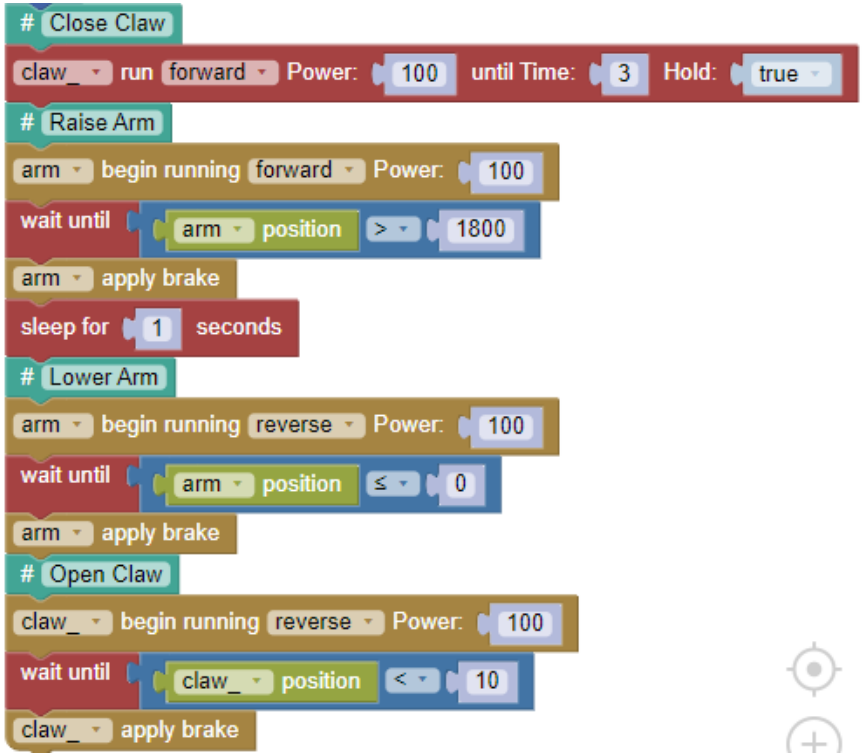
- May need to overdrive motor to get enough grip
- Use encoder to open back to zero position


Use time to close




Use rotation to open

# Combine Arm and Claw Movement




https://youtu.be/bgfusgzjxiM

# Task 8

## Building Functions

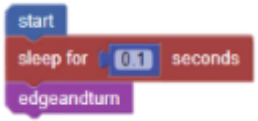
# Functions

- Solving the Robofest Game challenge will typically require a fairly large program (around 100 blocks is not unreasonable)
- Very large programs can be difficult to understand, navigate and use
- To alleviate this issue, the Robot Mesh Studio software has “Functions” to create custom blocks that can replace sections of your program
- Variables can be used to make the function more flexible

# Functions

- For example, let's assume you have a section code that completes the following:
  - Move forward until the edge of the table is found with color Left Color Sensor, 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



# Create a Edge Following Function

- Go to Functions Menu
- Get a “do something” block
- Name it “edgeFollow”
- Drag the edge following blocks into the function block


# Use the Function in the Program


# Putting It All Together

- In this course we learned about
  - 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
  - Finding a Paper Roll
  - Turning the robot
  - Aligning the robot to an edge
  - Manipulating Paper Rolls
  - Building MyBlocks

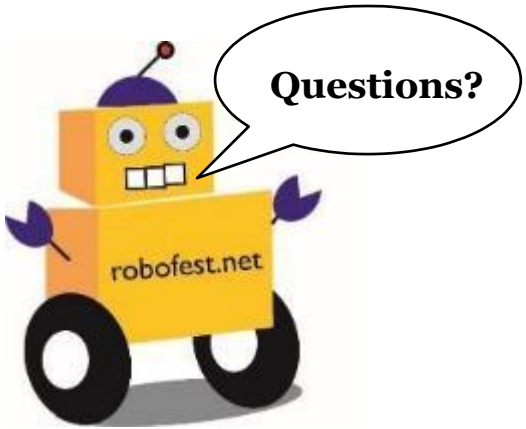
Workshop code: https://www.robotmesh.com/studio/5fbd44bece773b05bc8f9740

# Test for Knowledge

Post Assessment Link: https://docs.google.com/forms/d/e/1FAIpQLScPtA3T0-W60EPbIth1zEfTHWrLWnQRci4jl88zHvhmK_LflQ/viewform?usp=sf_link

LAWRENCE TECHNOLOGICAL UNIVERSITY ROBOFEST

# Little Robots, Big Missions




robofest@LTU.edu
LTU Computer Science