



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-7zwwqFettag/viewform?usp=sf_link

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



2021 Workshops

Presented by

Lawrence Technological
University
Computer Science

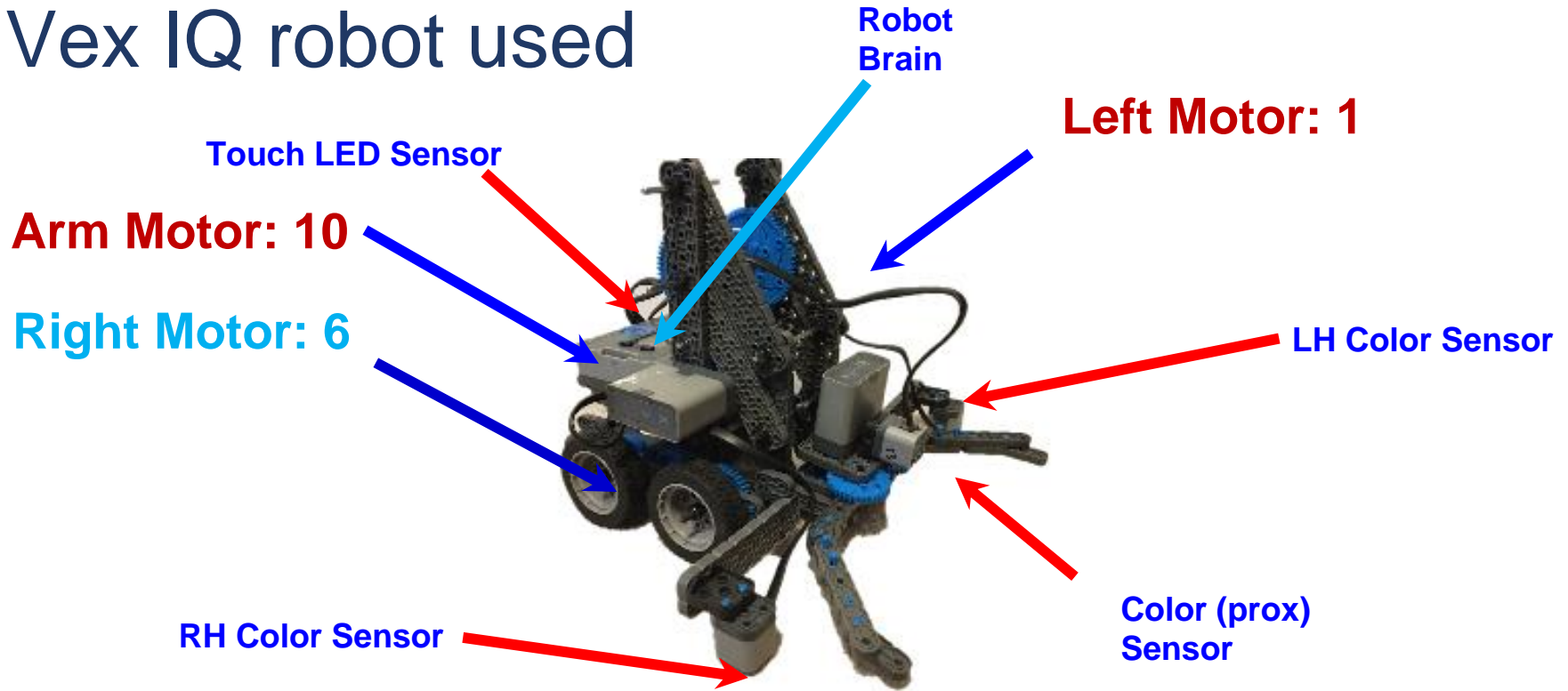
Course Overview

- 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



Getting Started

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

Install RMC plugin...

Log Out

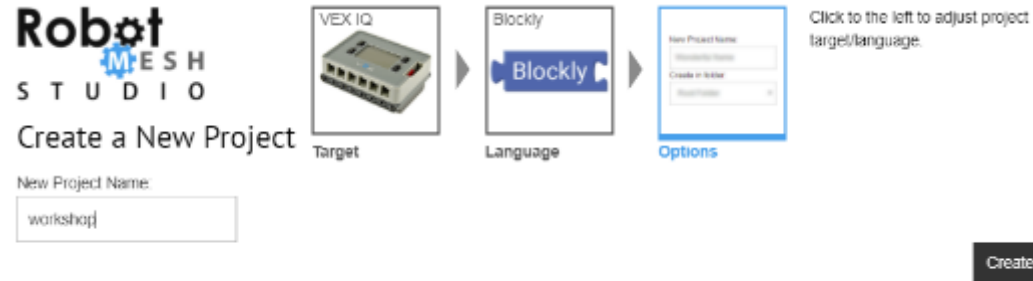


Robot Mesh Connect Installation

Robot Mesh Connect plugin is installed correctly.

Create a New Program

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



Robot MESH STUDIO

Create a New Project

New Project Name:
workshop

Target: VEX IQ

Language: Blockly

Options: New Project Name, Create in folder, [Dropdown]

Click to the left to adjust project target/language.

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”

The screenshot shows the Robot Mesh Studio interface. At the top, there are navigation buttons for 'Run', 'Stop', and 'Full Speed'. Below that, there are tabs for 'Description', 'Blockly', and 'Generated Code'. The main workspace is currently empty, showing a 'step' block. On the right side, there is a 'VEX IQ' panel with a 'Test Drive' button highlighted in red. Below the 'Test Drive' button, there is a list of components including 'motor_1', 'motor_2', 'motor_3', 'motor_4', 'motor_5', 'touch_sensor', and 'drivetrain'. The 'drivetrain' component is expanded, showing 'left' and 'right' motor settings.

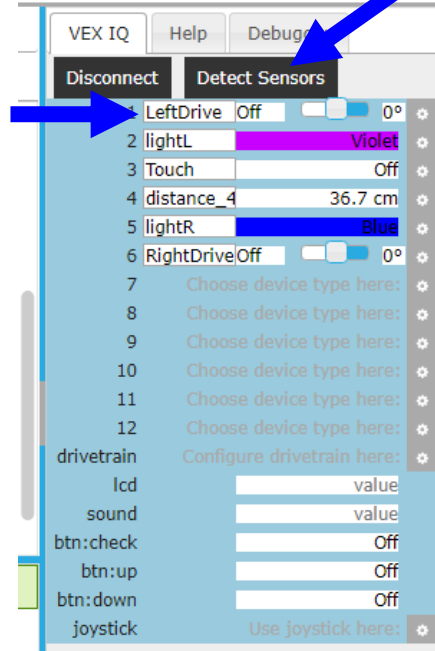
2. Look for “Connected” message at the bottom

Copyright © 2021 Robofest

Motor and Sensor Setup

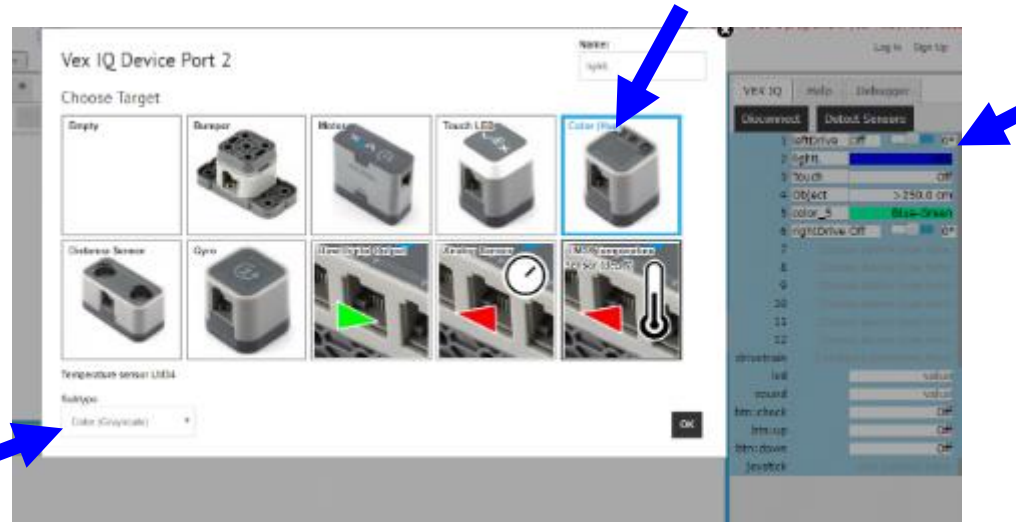
1. Click Here

2. Change to names that have meaning



Motor and Sensor Setup

Change
Color
Sensors to
grayscale
mode



Blockly Programming

Writing Your Blockly Code

The Blockly programming area in the Robot Mesh Studio IDE looks like this:

Robot Mesh
Vex IQ Sensors
Vex IQ Outputs
Vex IQ Motors
Vex IQ Sound
Vex IQ Drivetrain
Logic
Loops
Math
Text
Lists
Variables
Functions
Timer

repeat forever

repeat 10 times

repeat while

count with 1 from 1 to 10 by 1

for each item 1 in list

break out of loop

start

repeat while true

do

motor_left begin running forward Power: 100

[motor] begin running forward Power: 100

Blockly Tool Box
Click a label to show the corresponding block library.

Blockly program. This is where you drag your program blocks to build a program.

Blockly programming blocks. Select by clicking and dragging to the right.

Copyright © 2021 Robofest

Display Sensor Values on LCD

Use the LCD Display to Display the Value of a

```
start
# Sensor Values
repeat forever
  lcd write in row 1 create text with " Left_ " light_L gray level
  lcd write in row 2 create text with " Right_ " light_R gray level
  lcd write in row 3 create text with " Distance_ " distance_2 distance
  sleep for 0.2 seconds
```

The image shows a Scratch script for an LCD display. It starts with a 'start' block, followed by a comment '# Sensor Values'. A 'repeat forever' loop contains three 'lcd write in row' blocks. The first block writes 'Left_' to row 1, using 'light_L' as the sensor value and 'gray level' as the unit. The second block writes 'Right_' to row 2, using 'light_R' as the sensor value and 'gray level' as the unit. The third block writes 'Distance_' to row 3, using 'distance_2' as the sensor value and 'distance' as the unit. The loop ends with a 'sleep for 0.2 seconds' block.

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



Copyright © 2021 Robofest

Drivetrain Configuration

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

Drivetrain Configuration

Enabled

Name:
dl

Left motor:
leftDrive Reverse Polarity

Right motor:
rightDrive Reverse Polarity

Wheel travel (circumference) mm:
200

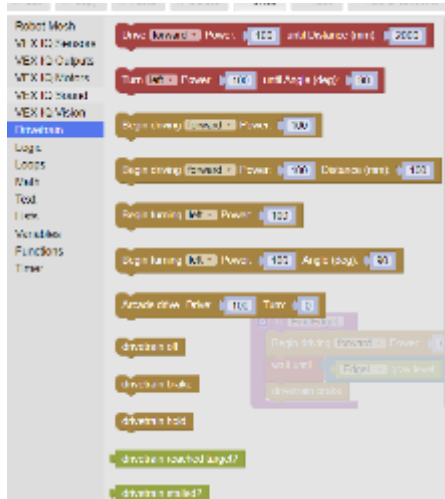
Track width mm:
190

OK Cancel

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



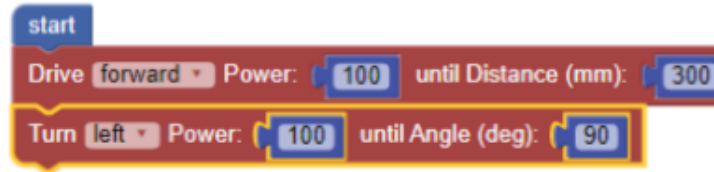
Copyright © 2021 Robofest

Task 0

Finding the edge of the table

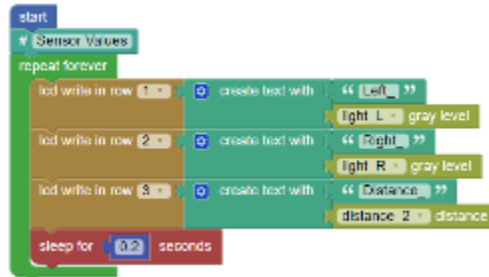
Drive Forward and Turn

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

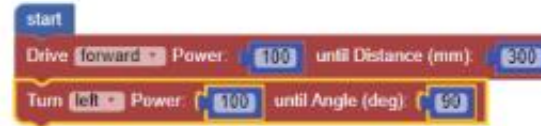


```
start
Drive forward Power: 100 until Distance (mm): 300
Turn left Power: 100 until Angle (deg): 90
```

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



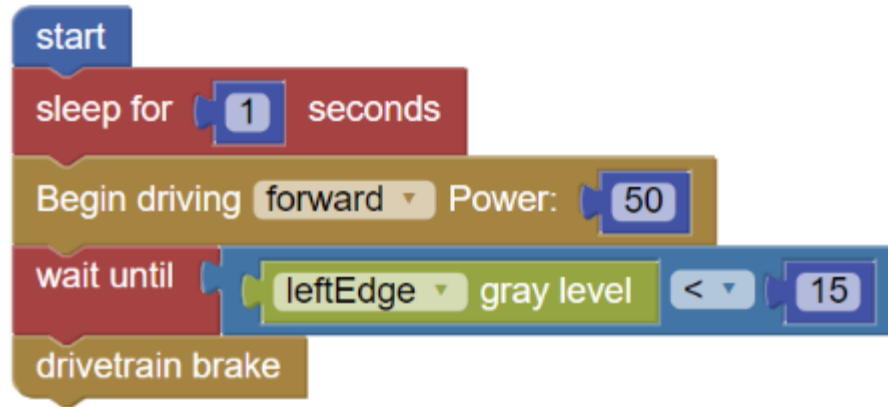
```
start
Sensor Values
repeat forever
  led write in row 1 create text with "Left" light L gray level
  led write in row 2 create text with "Right" light R gray level
  led write in row 3 create text with "Distance" distance 2 distance
  sleep for 0.2 seconds
```



```
start
Drive forward Power: 100 until Distance (mm): 300
Turn left Power: 100 until Angle (deg): 90
```

Find the edge of the playing field

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

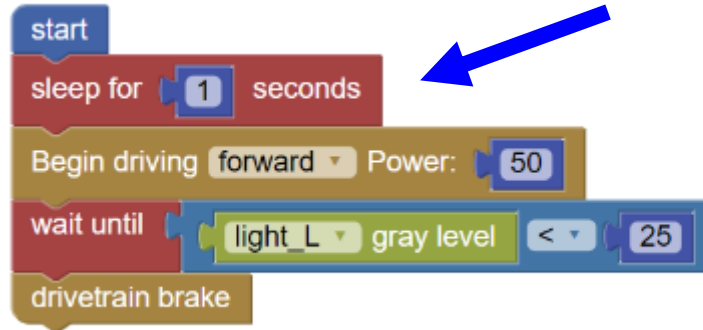


<https://youtu.be/Q-2CG9p8fR0>

Copyright © 2021 Robofest

TIP

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

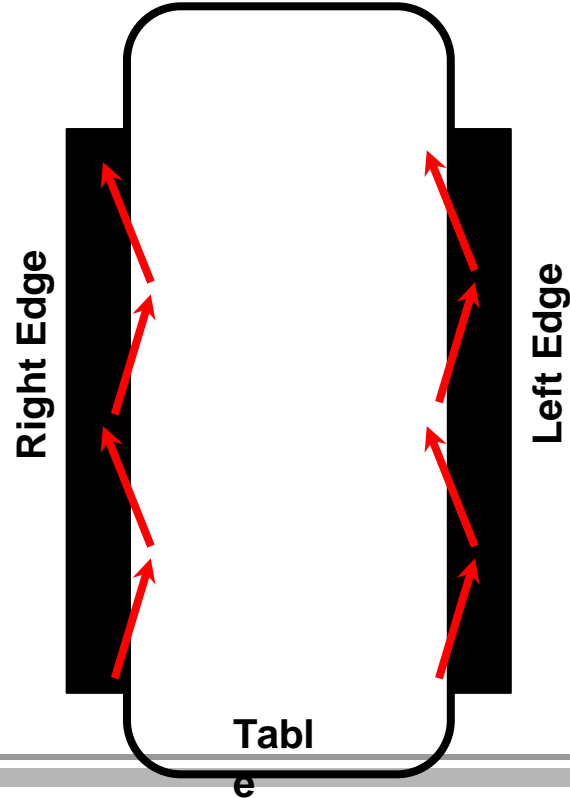


Task 1

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



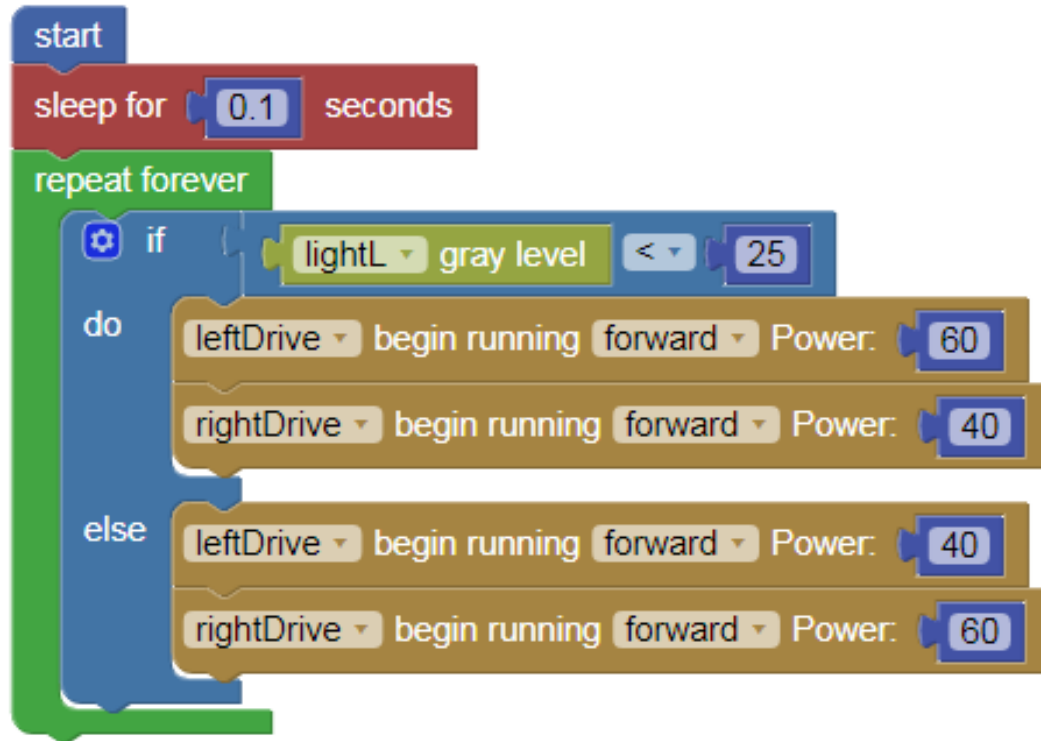
Copyright © 2021 Robofest

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

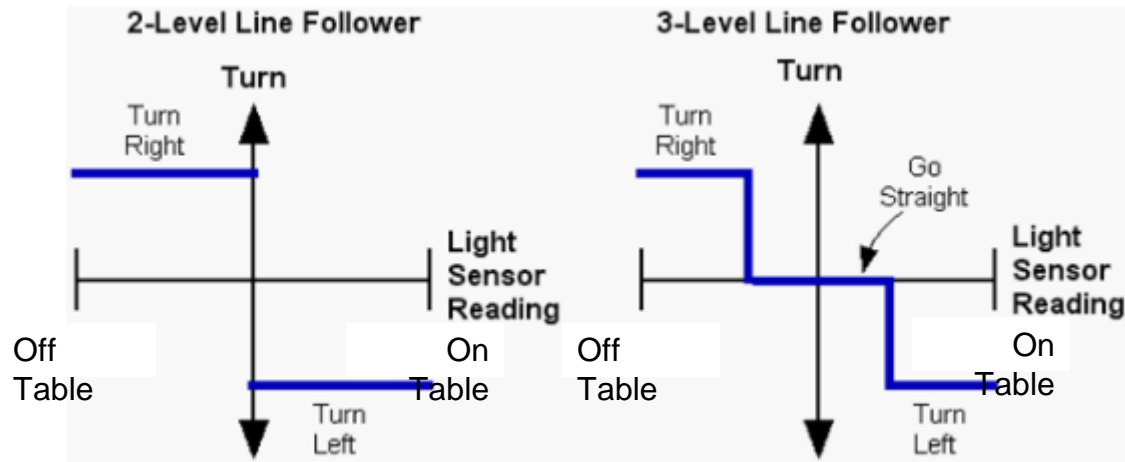


YouTube:

Copyright © 2021 Robofest

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

```
start
sleep for 0.1 seconds
repeat forever
  if lightL gray level < 5
  do
    leftDrive begin running forward Power: 60
    rightDrive begin running forward Power: 40
  else
    if lightL gray level > 40
    do
      leftDrive begin running forward Power: 60
      rightDrive begin running forward Power: 40
    else
      leftDrive begin running forward Power: 60
      rightDrive begin running forward Power: 60
  sleep for 0.05 seconds
```

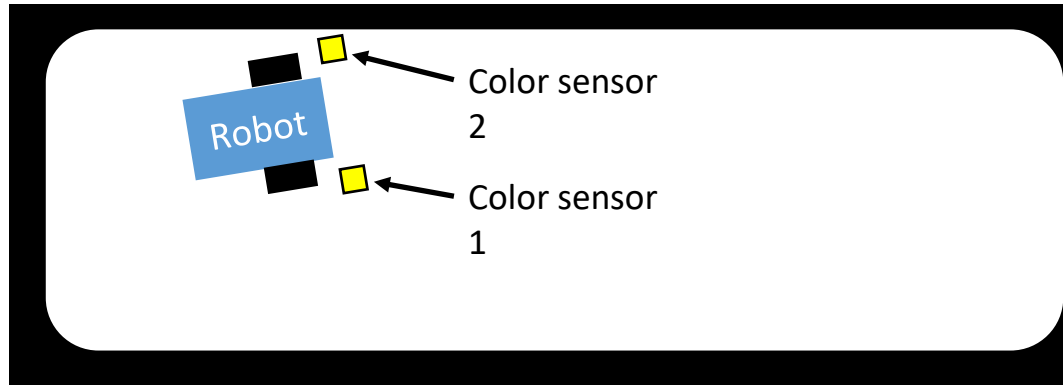
The image shows a sequence of code blocks for a line following algorithm. It starts with a 'start' block, followed by a 'sleep for 0.1 seconds' block. A 'repeat forever' loop contains an 'if' block. The first 'if' block checks 'lightL gray level < 5'. If true, it runs 'leftDrive begin running forward Power: 60' and 'rightDrive begin running forward Power: 40'. If false, it goes to another 'if' block that checks 'lightL gray level > 40'. If true, it runs 'leftDrive begin running forward Power: 60' and 'rightDrive begin running forward Power: 40'. If false, it runs 'leftDrive begin running forward Power: 60' and 'rightDrive begin running forward Power: 60'. The loop ends with a 'sleep for 0.05 seconds' block.

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



```
start
sleep for 0.1 seconds
repeat forever
  if lightL gray level < 25
  do
    leftDrive begin running forward Power: 60
    rightDrive begin running forward Power: 40
  else
    leftDrive begin running forward Power: 40
    rightDrive begin running forward Power: 60
```

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

Copyright © 2021 Robofest

Task 3

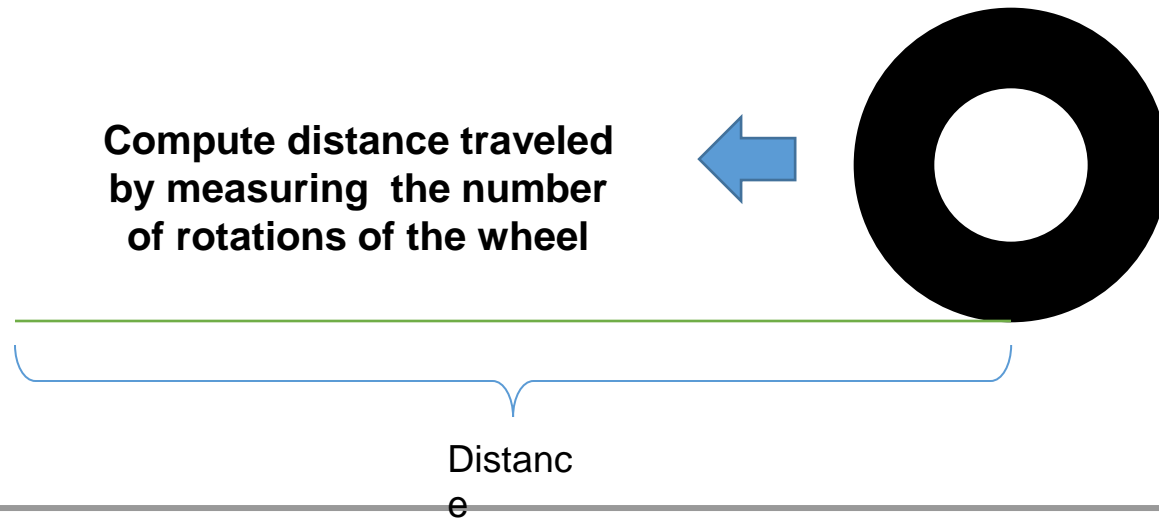
Line following a given distance

Line following a given distance

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

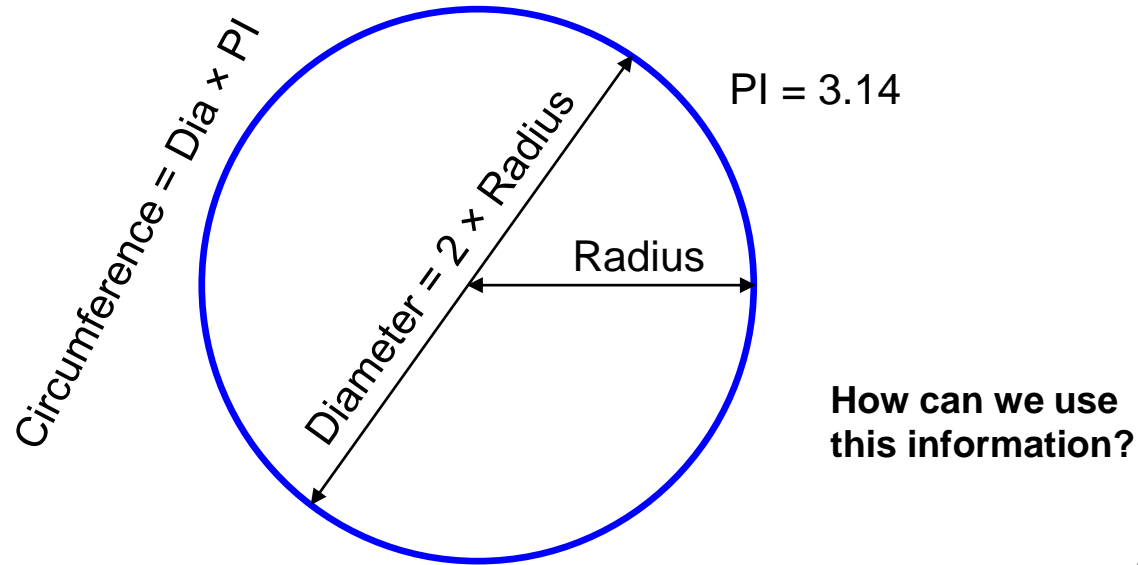
Measuring Distances

- 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



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)

```
start
leftDrive reset position
Begin driving forward Power: 50
sleep for 2 seconds
drivetrain brake
lcd write in row 1 20 x leftDrive position + 360
wait until touch_led is touch?
```

The code consists of the following blocks:

- start** (blue)
- leftDrive reset position** (orange)
- Begin driving forward Power: 50** (orange)
- sleep for 2 seconds** (red)
- drivetrain brake** (orange)
- lcd write in row 1** (orange) with a sub-block: $20 \times \text{leftDrive position} + 360$ (blue)
- wait until touch_led is touch?** (green)

YouTube: <https://youtu.be/9pqK0uBtbzE>

Copyright © 2021 Robofest

Measuring Distances

- Example

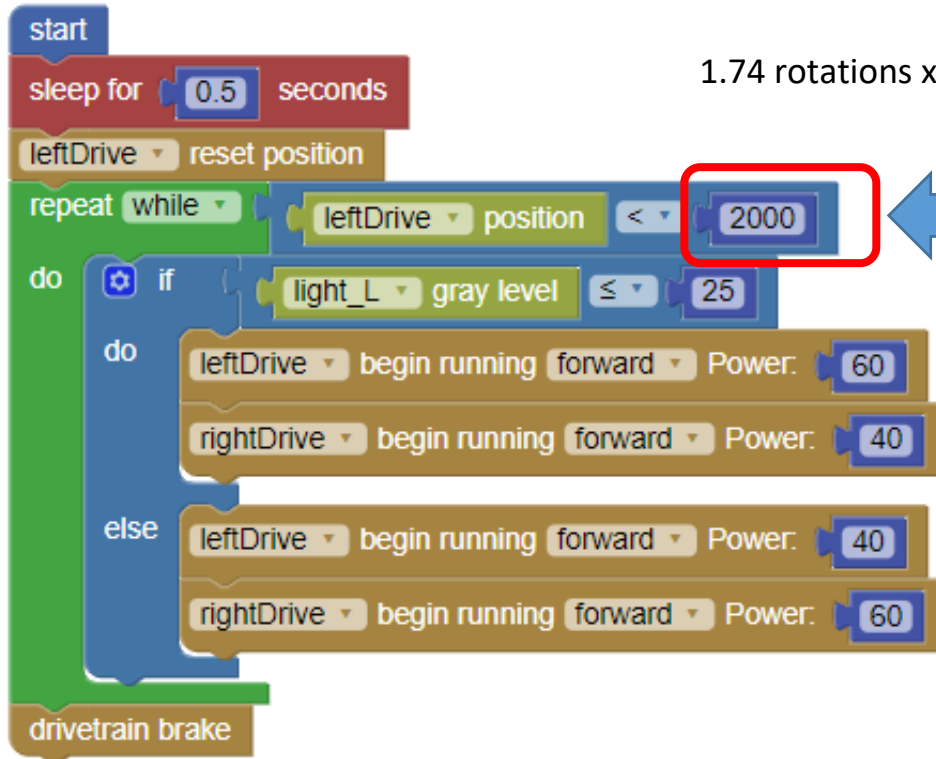
- Let's program the robot to line follow for 30 cm
 - Distance = 30 cm
- Number of rotations
 - Distance = (Wheel Diameter) x (PI) x (# Rotations)
 - Solve for (# Rotations)

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

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

Rotations must be converted to degrees

$$1.74 \text{ rotations} \times 360 \text{ degrees} = 626$$



Change this number to go the desired distance

<https://youtu.be/K1suXAzZuCA>

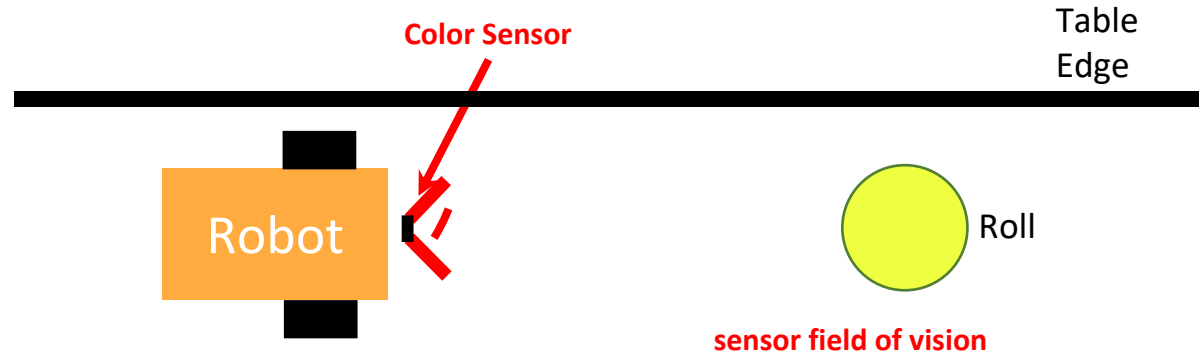
Copyright © 2021 Robofest

Task 4

Finding a Roll

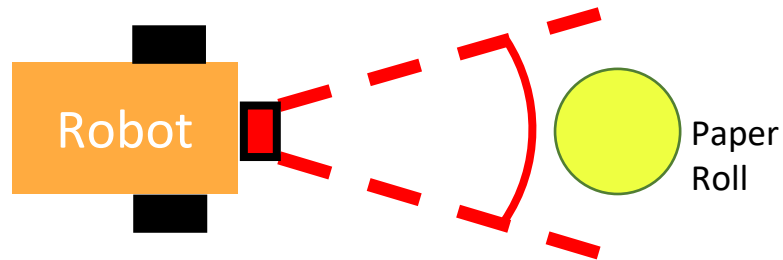
Finding a Rolls

- 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



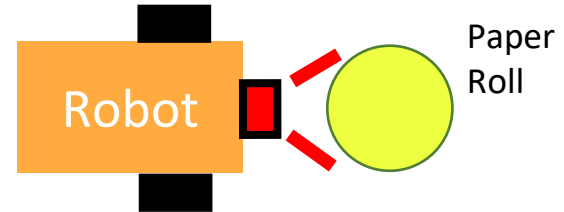
Finding a Paper Roll

- 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

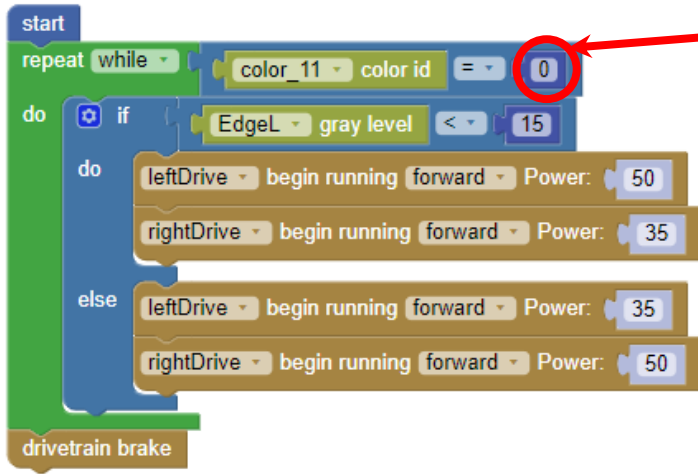


Color sensor will read values when the Paper Roll is close to the robot

Final Position

Finding a Paper Roll- Color Mode

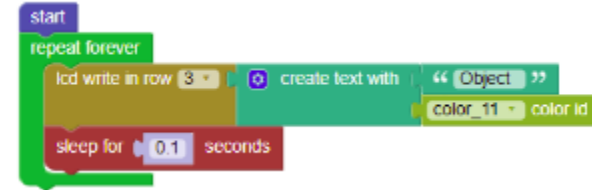
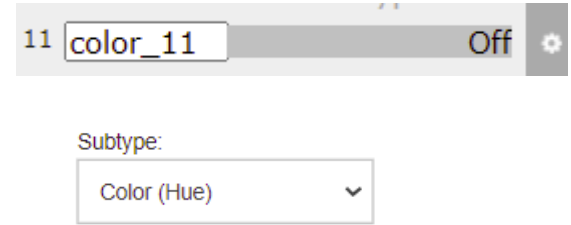
- Now, we travel along the edge of the table and stop if we find a Paper Roll



```
start
repeat while
  color_11 color id = 0
do
  if
    EdgeL gray level < 15
  do
    leftDrive begin running forward Power: 50
    rightDrive begin running forward Power: 35
  else
    leftDrive begin running forward Power: 35
    rightDrive begin running forward Power: 50
drivetrain brake
```

The code block shows a 'repeat while' loop with the condition 'color_11 color id = 0'. The value '0' is circled in red, with a red arrow pointing to it from the text box on the right. Inside the loop, there is an 'if' statement: 'if EdgeL gray level < 15'. If true, the left drive runs forward at power 50 and the right drive at power 35. If false, the left drive runs forward at power 35 and the right drive at power 50. The code ends with a 'drivetrain brake' block.

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



```
start
repeat forever
  lcd write in row 3
  create text with " Object "
  color_11 color id
  sleep for 0.1 seconds
```

The code block shows a 'repeat forever' loop. Inside the loop, it performs 'lcd write in row 3', 'create text with " Object "', 'color_11 color id', and 'sleep for 0.1 seconds'.

Finding a Paper Roll- Grayscale

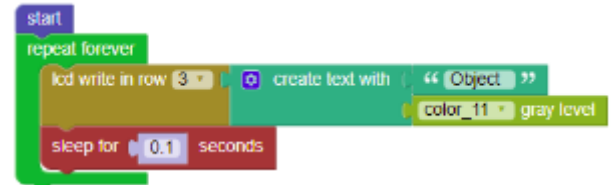
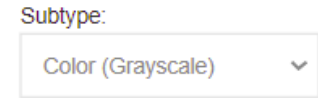
- Now, we travel along the edge of the table and stop if we find a Paper Roll



```
start
repeat while color_11 gray level < 50
do
  if EdgeL gray level < 15
  do
    leftDrive begin running forward Power: 50
    rightDrive begin running forward Power: 35
  else
    leftDrive begin running forward Power: 35
    rightDrive begin running forward Power: 50
drivetrain brake
```

The code block shows a 'repeat while' loop with the condition 'color_11 gray level < 50'. A red circle highlights the number '50' in the condition. Inside the loop, there is an 'if' block with the condition 'EdgeL gray level < 15'. If true, it runs 'leftDrive begin running forward Power: 50' and 'rightDrive begin running forward Power: 35'. If false, it runs 'leftDrive begin running forward Power: 35' and 'rightDrive begin running forward Power: 50'. The code ends with 'drivetrain brake'.

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



```
start
repeat forever
  lcd write in row 3
  create text with " Object "
  color_11 gray level
  sleep for 0.1 seconds
```

The code block shows a 'repeat forever' loop. Inside the loop, it runs 'lcd write in row 3', 'create text with " Object "', 'color_11 gray level', and 'sleep for 0.1 seconds'.

<https://youtu.be/-zrYApNv5jI>

Task 4

Turning the robot

4
8

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

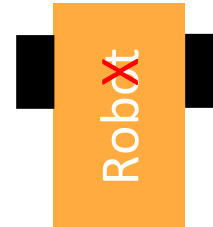
90 Degree Spin

- 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

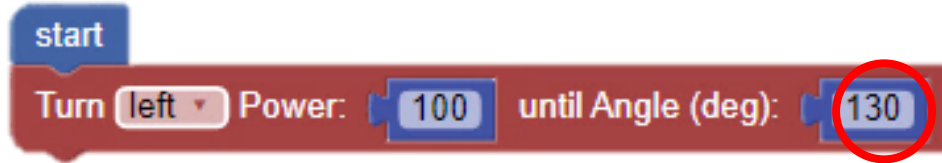
- Now, we need to determine the correct number of degrees

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 trial 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

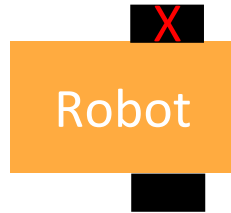
YouTube:

<https://youtu.be/AUqXzsnf0NI>

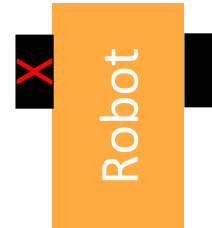
90 Degree Swing

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

Starting Position

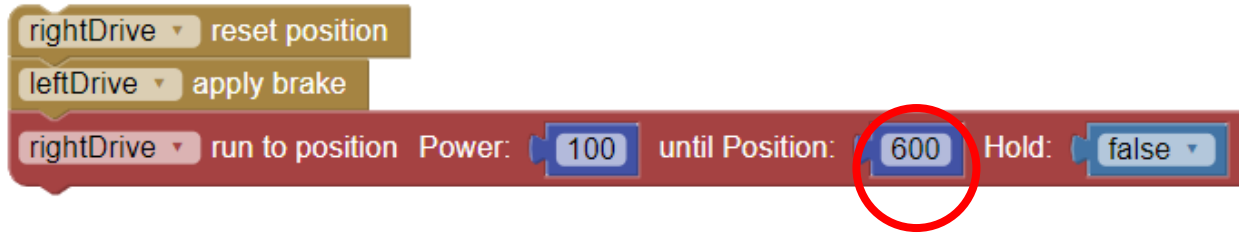


Final Position



90 Degree Swing

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



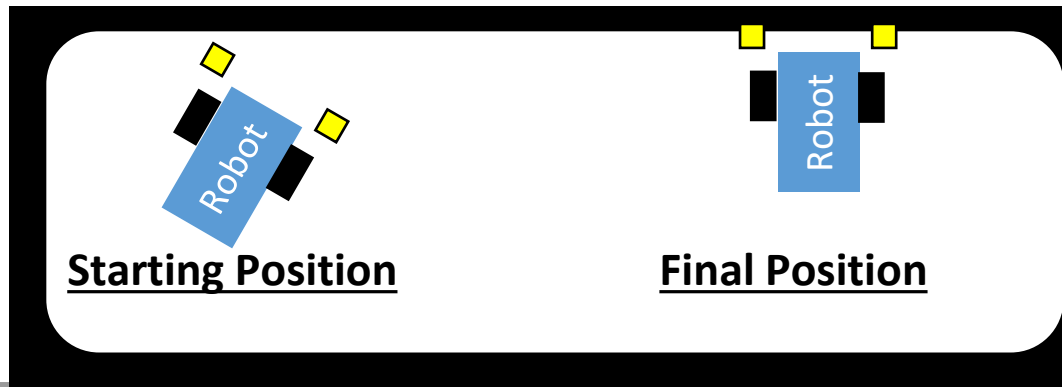
- For our sample robot, it takes 600 degrees to swing the robot 90 degrees

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

```
start
sleep for 1 seconds
# Find Edge with Left Sensor
Begin driving forward Power: 25
wait until light_L gray level ≤ 25
leftDrive apply brake
# Find Edge with Right Sensor
wait until light_R gray level ≤ 25
rightDrive apply brake
```

The code block sequence is as follows: a blue 'start' block, a red 'sleep for 1 seconds' block, a teal comment block '# Find Edge with Left Sensor', a brown 'Begin driving forward' block with a power of 25, a red 'wait until' block where 'light_L gray level' is less than or equal to 25, a brown 'leftDrive apply brake' block, a teal comment block '# Find Edge with Right Sensor', a red 'wait until' block where 'light_R gray level' is less than or equal to 25, and a brown 'rightDrive apply brake' block.

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

```
start
sleep for 1 seconds
# Find Edge with Either Sensor
Begin driving forward Power: 25
wait until light_L gray level <= 25 or light_R gray level <= 25
if light_L gray level <= 25
do
leftDrive apply brake
wait until light_R gray level <= 25
rightDrive apply brake
else
rightDrive apply brake
wait until light_L gray level <= 25
leftDrive apply brake
```

Copyright © 2021 Robofest

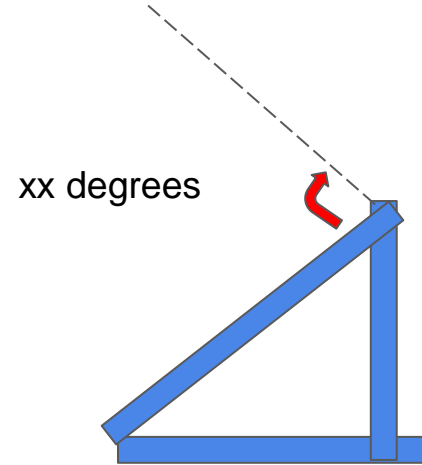
YouTube: <https://youtu.be/JqOL2ekQARk>

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



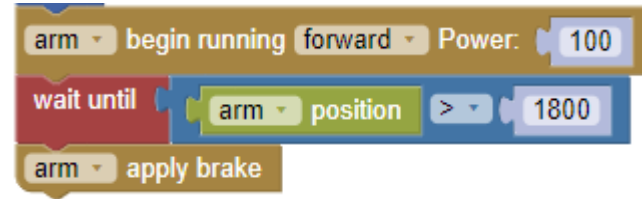
Raise
Arm



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



```
claw_ run forward Power: 100 until Time: 3 Hold: true
```

Use time to close



```
claw_ begin running reverse Power: 100
```



```
wait until claw_ position <= 0
```



```
claw_ apply brake
```

Use rotation to open

Combine Arm and Claw Movement

```
# Close Claw
claw_ run forward Power: 100 until Time: 3 Hold: true

# Raise Arm
arm begin running forward Power: 100
wait until arm position > 1800
arm apply brake
sleep for 1 seconds

# Lower Arm
arm begin running reverse Power: 100
wait until arm position ≤ 0
arm apply brake

# Open Claw
claw_ begin running reverse Power: 100
wait until claw_ position < 10
claw_ apply brake
```

<https://youtu.be/bgfusgzxiM>



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

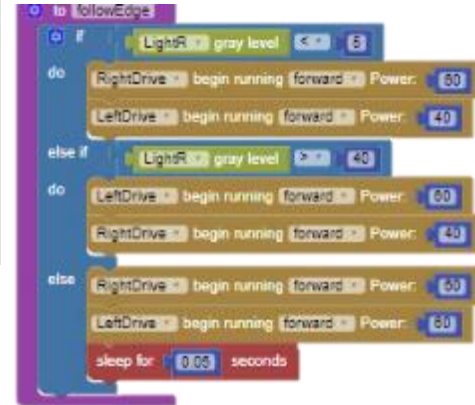
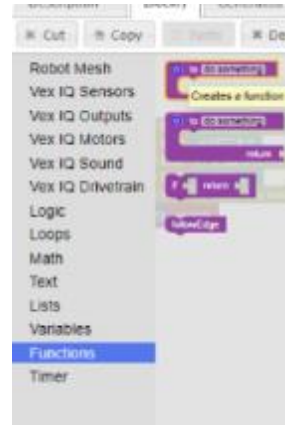


- My blocks will allow us to convert this to a single block



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

```
start
sleep for 0.1 seconds
repeat while LightL gray level > 10
do followEdge
drivetrain brake
```

```
to followEdge
if LightR gray level < 5
do
RightDrive begin running forward Power: 60
LeftDrive begin running forward Power: 40
else if LightR gray level > 40
do
LeftDrive begin running forward Power: 60
RightDrive begin running forward Power: 40
else
RightDrive begin running forward Power: 60
LeftDrive begin running forward Power: 60
sleep for 0.05 seconds
```

Copyright © 2021 Robofest

Putting It All Together

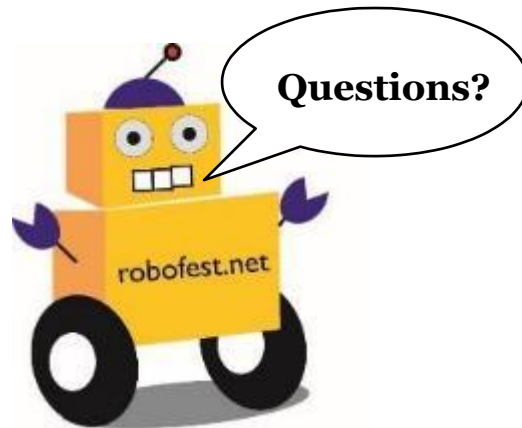
- In this course we learned about
 - Finding the edge of the table
 - Following the edge of the table
 - Stop line following
 - When you reach a corner
 - When you reach a given distance
 - Finding a Paper Roll
 - Turning the robot
 - Aligning the robot to an edge
 - Manipulating Paper Rolls
 - Building MyBlocks

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

Test for Knowledge

Post Assessment Link: https://docs.google.com/forms/d/e/1FAIpQLScPtA3T0-W60EPbIth1zEfTHWrLWnQRci4jl88zHvhmK_LfIQ/viewform?usp=sf_link

Little Robots, Big Missions



robofest@LTU.edu
LTU Computer Science

Copyright © 2021 Robofest