



Stack Rolls

VEXcode V5 Software Game Workshop

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



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
- Workshop robot introduction
- Using the workshop robot 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

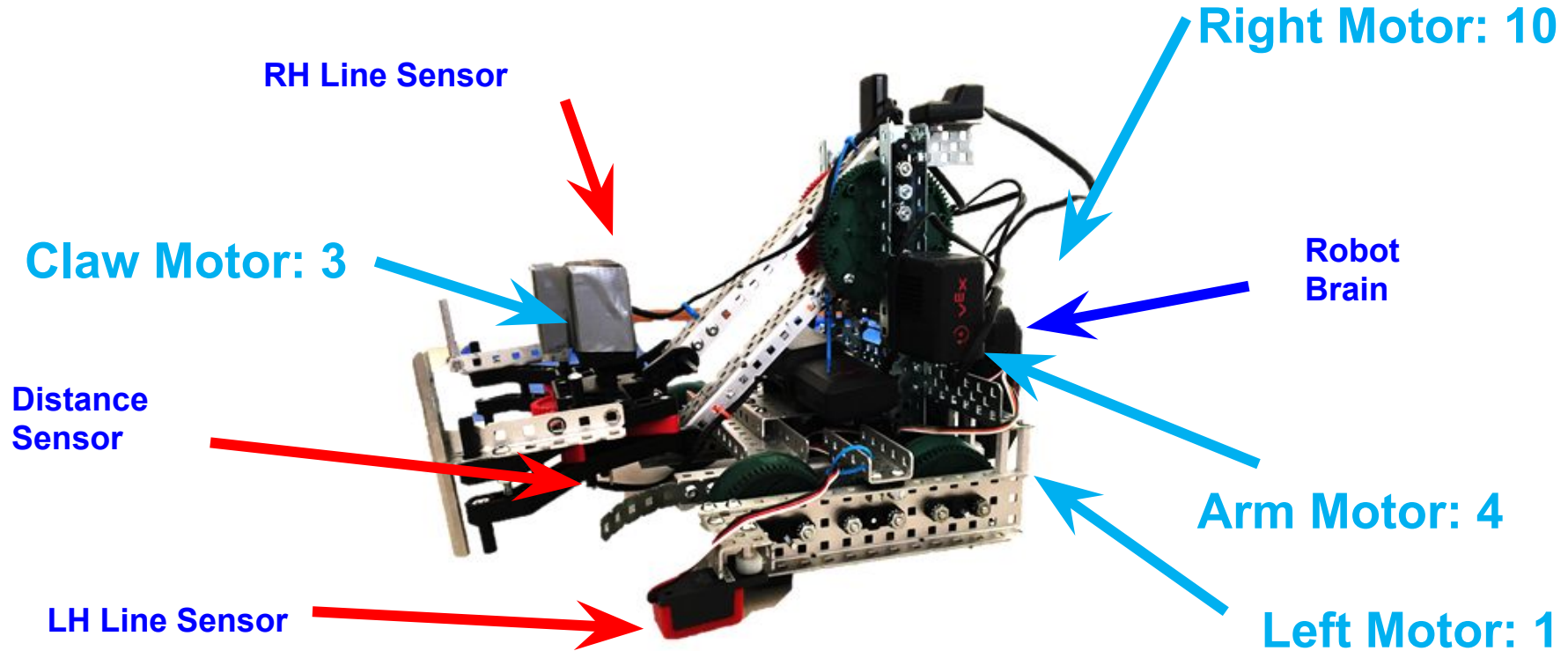
Task 5: Turning the robot

Task 6: Aligning the robot to an edge

Task 7: Manipulating Paper Rolls

Task 8: Building MyBlocks

VEX V5 robot used



Copyright © 2021 Robofest

Remember the connections!

- Left Motor connects to **1**
- Right Motor connects to **10**
- Claw Motor **3**
- Arm Motor connects to **4**
- LH Line sensor connects to port no. **B**
- RH Line sensor connects to port no. **A**
- Inertial sensor connects to port no. **6**
- Distance sensor connects to port no. **2**

VEXcode V5 Software

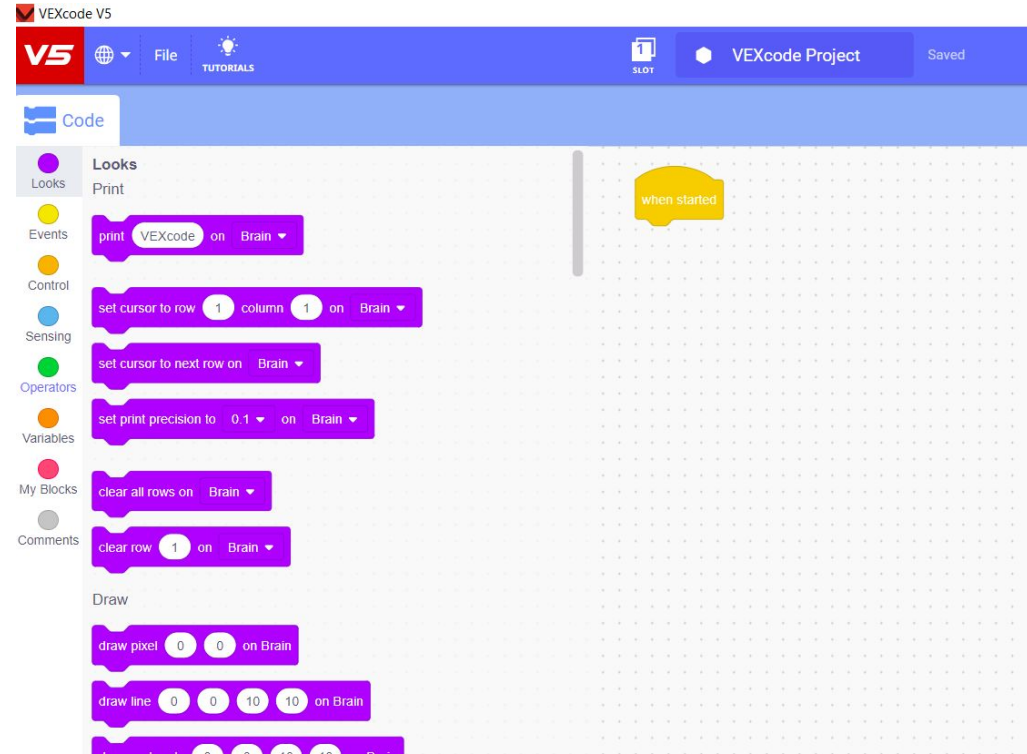
- Programming software for VEX V5
- Uses a drag and drop interface



VEXcode V5 - v2.0.6

Getting Started

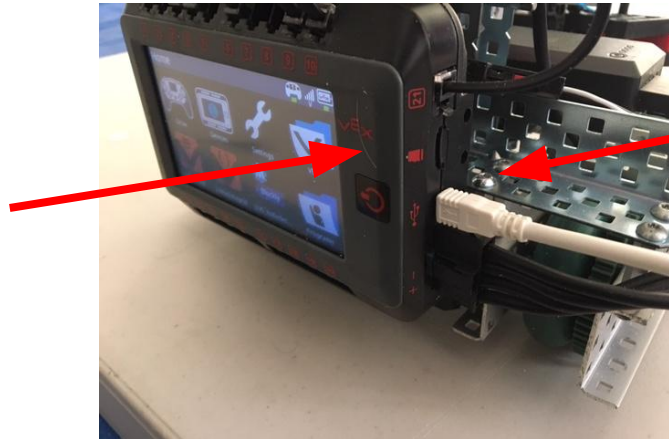
Double click
icon to open
software



Connect To Robot

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

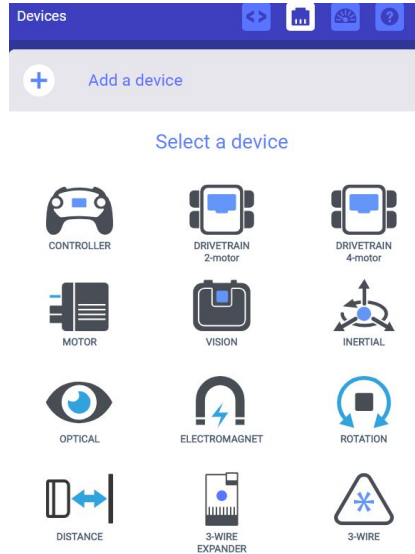
Power button



Programming
cord

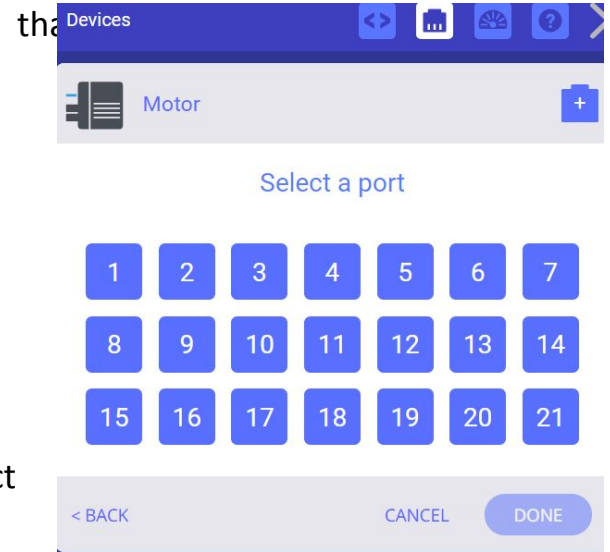
Motor and Sensor Setup

1. Click
“Add a
device”



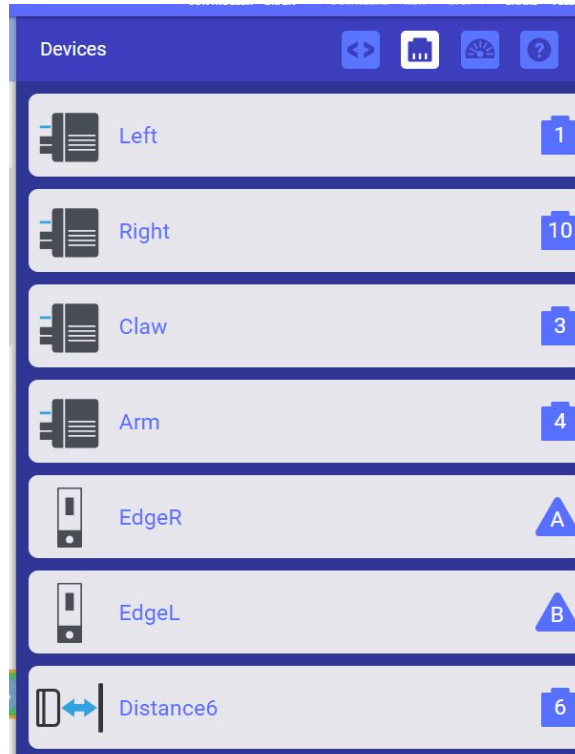
2. Select
device

4. Change to names



3. Select
port

Motor and Sensor Set Up



Copyright © 2021 Robofest

Display Sensor Values on LCD

Use the LCD Display to Display the Value of a Sensor

```
when started
  forever
    clear row 1 on Brain
    set cursor to row 1 column 1 on Brain
    print EdgeL reflectivity in % on Brain
    clear row 2 on Brain
    set cursor to row 2 column 1 on Brain
    print EdgeR reflectivity in % on Brain
    wait .2 seconds
```

Note: these blocks are needed at the start of a program using line following

Task 0

Finding the edge of the table

Find the edge of the playing field

Note: leave this code while adding new code

```
when started
  set Left velocity to 20 %
  set Right velocity to 20 %
  spin Left forward
  spin Right forward
  wait until EdgeL reflectivity in % < 20
  stop Left
  stop Right
```

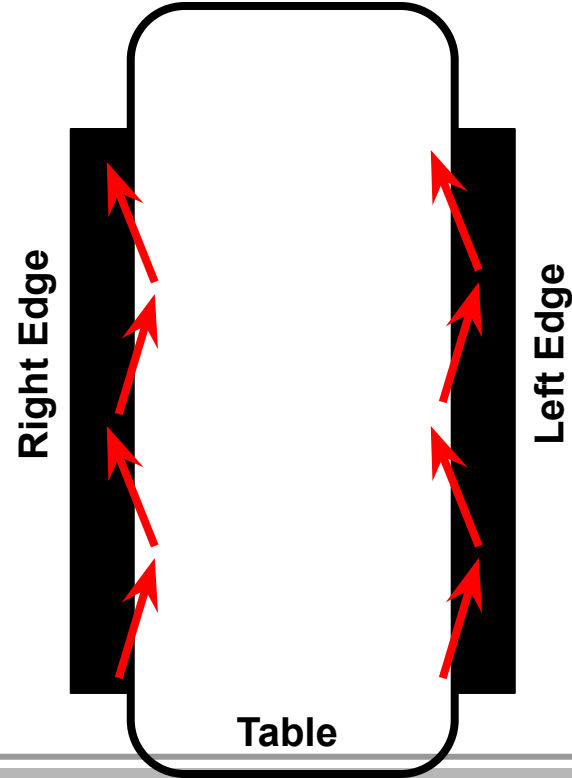
```
when started
  forever
    clear row 1 on Brain
    set cursor to row 1 column 1 on Brain
    print EdgeL reflectivity in % on Brain
    clear row 2 on Brain
    set cursor to row 2 column 1 on Brain
    print EdgeR reflectivity in % on Brain
    wait .2 seconds
```

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 = 35
 - Median threshold = $(5+35)/2 = 20$

- Two cases
 - Light sensor reading > 20 . On table.
 - Light sensor reading < 20 . Off table.

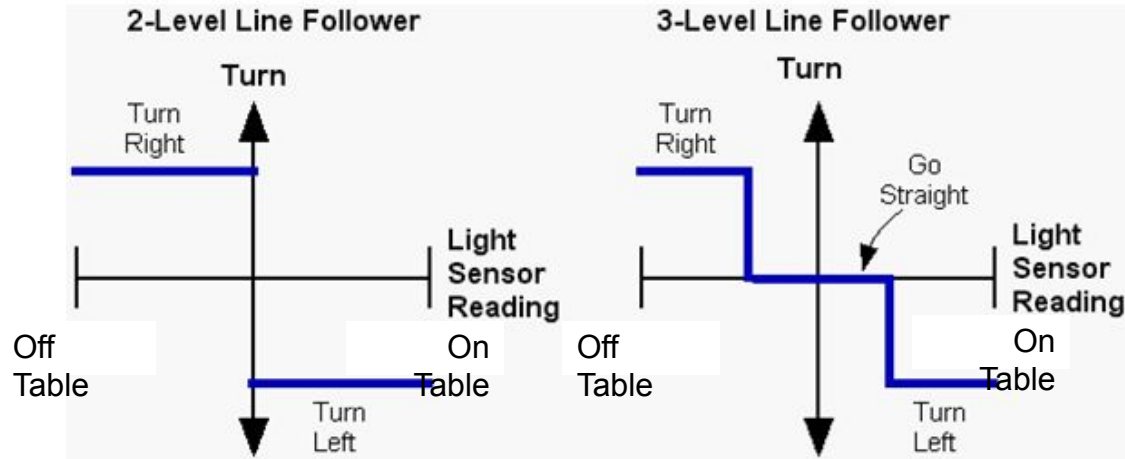
Simple Line Following Algorithm



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)



Copyright © 2021 Robofest

How to improve our line following algorithm

Determine upper and lower limits

Go straight if in between limits

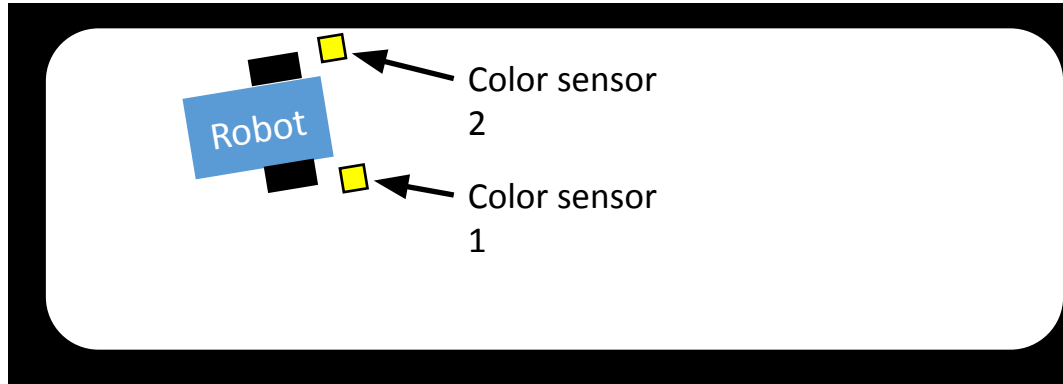
```
when started
if EdgeL reflectivity in % < 10 then
  set Left velocity to 35 %
  set Right velocity to 25 %
  spin Left forward
  spin Right forward
else
  if EdgeL reflectivity in % > 20 then
    set Left velocity to 25 %
    set Right velocity to 35 %
    spin Left forward
    spin Right forward
  else
    set Left velocity to 30 %
    set Right velocity to 30 %
    spin Left forward
    spin Right forward
```

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



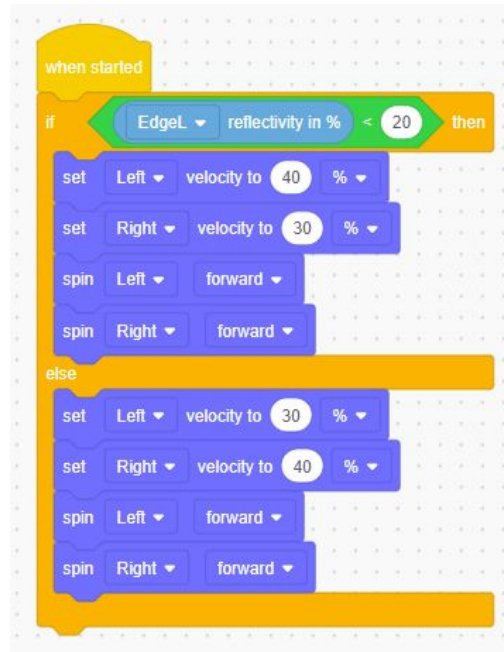
Line following to the corner

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 as your robot reaches them

Travel around the table

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



```
when started
if EdgeL reflectivity in % < 20 then
  set Left velocity to 40 %
  set Right velocity to 30 %
  spin Left forward
  spin Right forward
else
  set Left velocity to 30 %
  set Right velocity to 40 %
  spin Left forward
  spin Right forward
```

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

```
when started
while EdgeR reflectivity in % < 20
if EdgeL reflectivity in % < 20 then
set Left velocity to 40 %
set Right velocity to 30 %
spin Left forward
spin Right forward
else
set Left velocity to 30 %
set Right velocity to 40 %
spin Left forward
spin Right forward
stop Left
stop Right
```

The code is a Scratch script starting with a 'when started' block. It enters a 'while' loop with the condition 'EdgeR reflectivity in % < 20'. Inside this loop, there is an 'if' block with the condition 'EdgeL reflectivity in % < 20'. If this condition is true, it sets the left motor velocity to 40% and the right motor velocity to 30%, then spins both motors forward. If the condition is false, it sets the left motor velocity to 30% and the right motor velocity to 40%, then spins both motors forward. After the 'if' block, the 'while' loop continues. At the end of the script, there are two 'stop' blocks for the left and right motors.

youtube: https://youtu.be/1z_GVJNjD94

Copyright © 2021 Robofest

Follow the Edge to the Target Zone

What is the light sensor reading for the target zone?

```
when started
  set Left position to 0 degrees
  while Left position in degrees < 600
    if EdgeL reflectivity in % < 20 then
      set Left velocity to 40 %
      set Right velocity to 30 %
      spin Left forward
      spin Right forward
    else
      set Left velocity to 30 %
      set Right velocity to 40 %
      spin Left forward
      spin Right forward
  stop Left
  stop Right
```

The code is written in a Scratch-like block-based language. It starts with a 'when started' block. The first block is 'set Left position to 0 degrees'. This is followed by a 'while' loop with the condition 'Left position in degrees < 600'. Inside the while loop, there is an 'if' block with the condition 'EdgeL reflectivity in % < 20'. If this condition is true, the code sets 'Left velocity to 40 %' and 'Right velocity to 30 %', and then spins both 'Left' and 'Right' wheels 'forward'. If the condition is false, it sets 'Left velocity to 30 %' and 'Right velocity to 40 %', and spins both wheels 'forward'. After the while loop, there are two 'stop' blocks for 'Left' and 'Right'.

Task 3

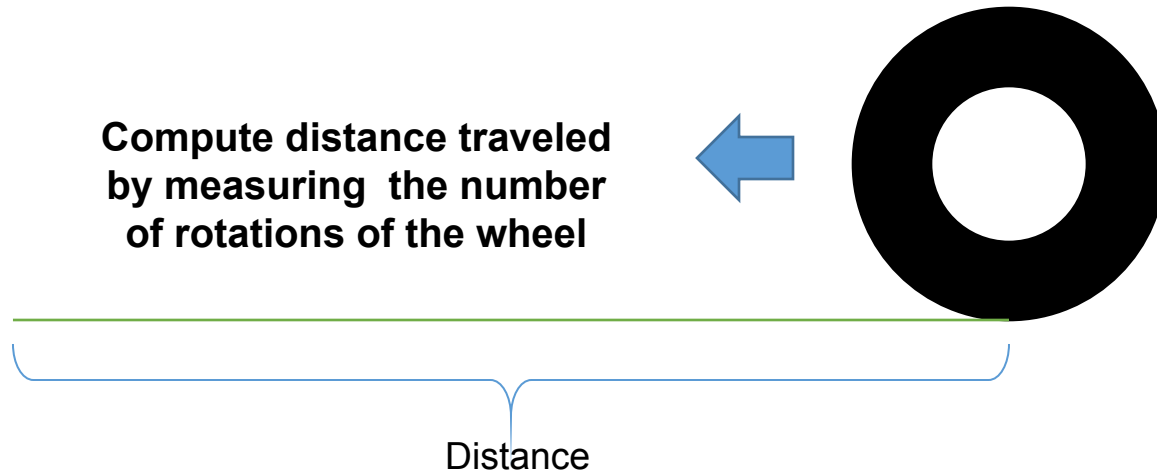
Line following a given distance

Line following a given distance

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

Measuring Distances

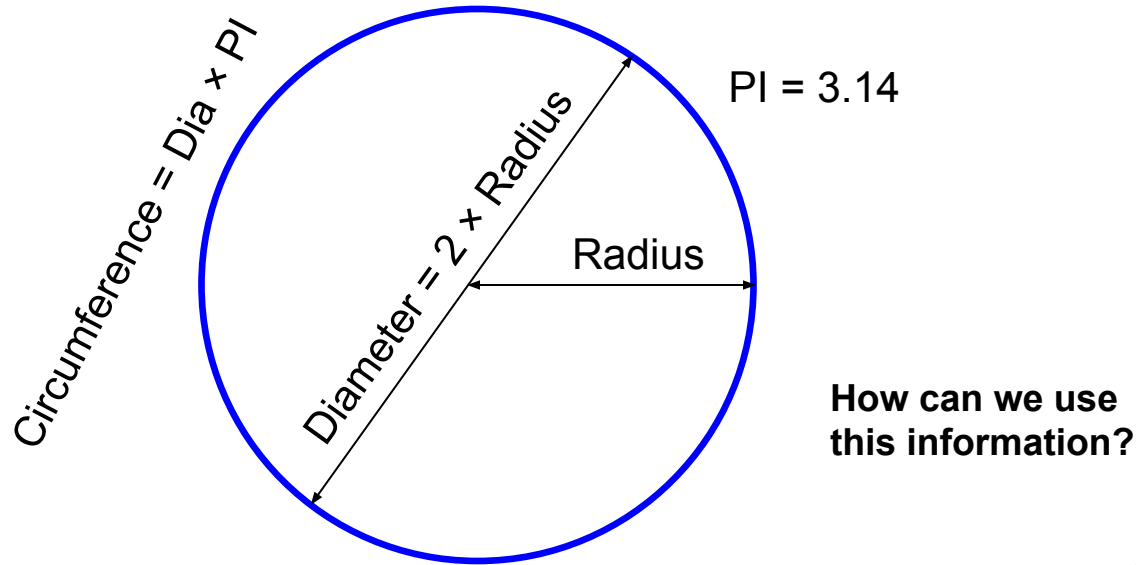
- How do we measure distance traveled?
- Let's determine how far the robot travels moving forward for 2 seconds



Copyright © 2021 Robofest

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 = (69.85 mm) x (PI) x (# Rotations)
□ Distance = (21.9cm) x (# Rotations)

```
when started
set Left velocity to 20 %
set Right velocity to 20 %
spin Left forward
spin Right forward
set Left position to 0 degrees
wait 2 seconds
stop Left
stop Right
set Distance to 21.9 * (Left position in degrees / 360)
print Distance on Brain
wait 100 seconds
```

Copyright © 2021 Robofest

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

Measuring Distances

- Example

- Let's program the robot to line follow for 30 cm

- Distance = 30 cm

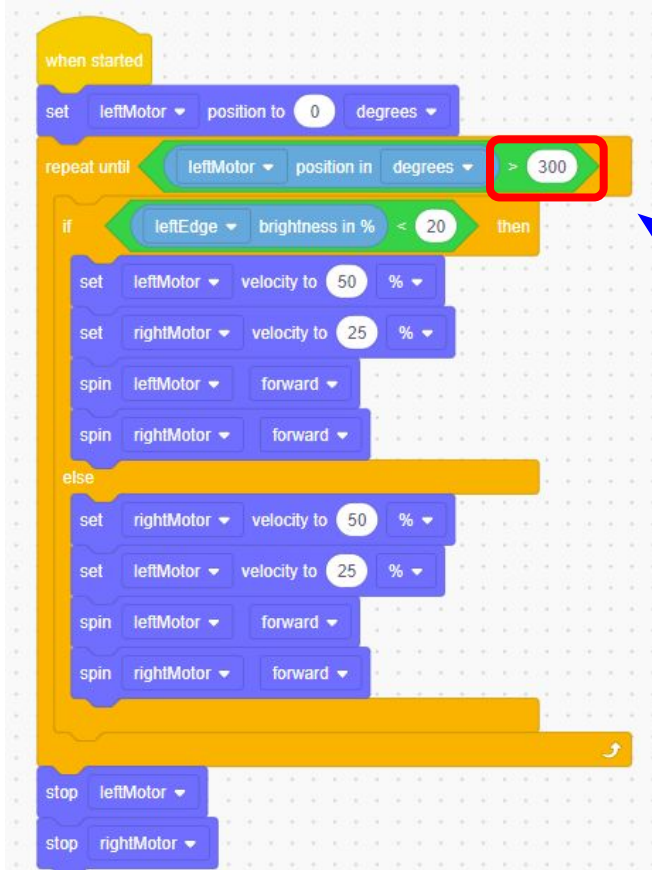
- Number of rotations

- 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

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

Change
this
number to
go the
desired
distance

YouTube:

<https://youtu.be/K1suXAzZuCA>

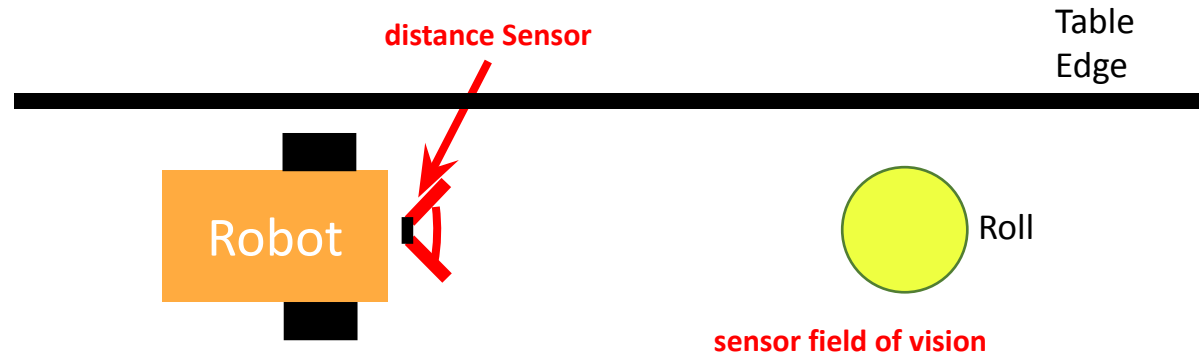
Copyright © 2021 Robofest

Task 4

Finding a Roll

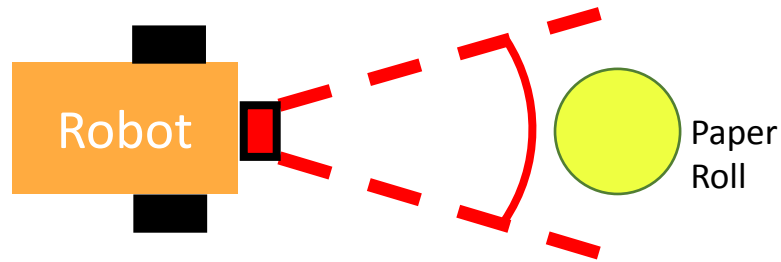
Finding a Rolls

- We can use a distance 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



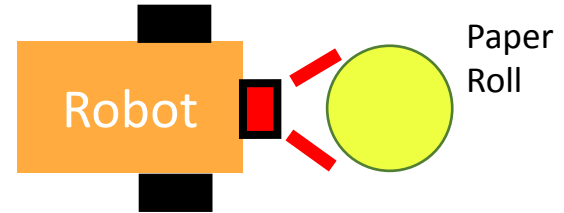
Finding a Paper Roll

- 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 distance sensor



**Distance sensor will read no value
when the Paper Roll is far away**

Starting Position



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

Final Position

Finding a Paper Roll

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

```
when started
while Distance6 < distance in mm < 30
  if EdgeL reflectivity in % < 20 then
    set Left velocity to 40 %
    set Right velocity to 30 %
    spin Left forward
    spin Right forward
  else
    set Left velocity to 30 %
    set Right velocity to 40 %
    spin Left forward
    spin Right forward
stop Left
stop Right
```

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

Task 5

Turning the robot

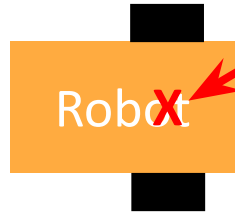
Turning The Robot

- For our example here, we wish to turn the robot 90 degrees
- There are several methods for turning a tripod robot. We will focus on two methods
 - “Spin” turn
 - “Swing” turn

90 Degree Spin

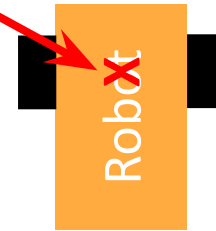
- Let's have the robot spin 90 degrees CCW
- The robot will rotate about center of the drive wheels

Starting Position



Center of
drive
wheels

Final Position



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

Spin 90 Degrees

- We can use the code below to spin the robot



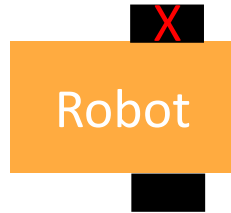
- For our sample robot, it takes 310 degrees to spin the robot 90 degrees

YouTube: <https://youtu.be/AUqXzsnf0Nl>

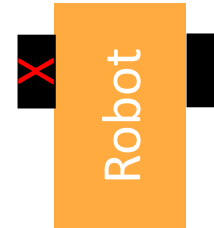
90 Degree Swing

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

Starting Position

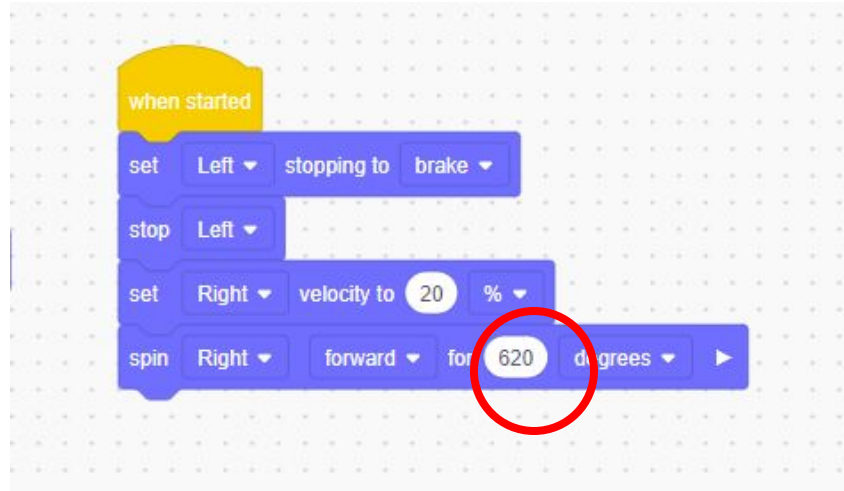


Final Position



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 620 degrees to swing the robot 90 degrees

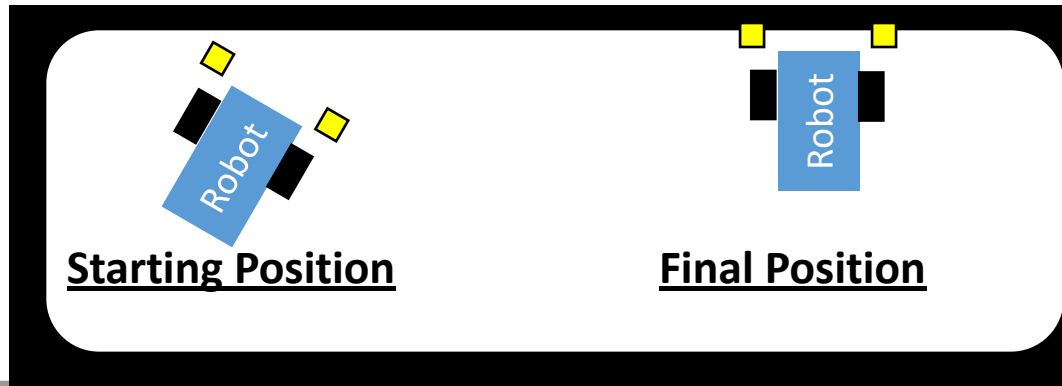


Task 6

Aligning the robot to an edge

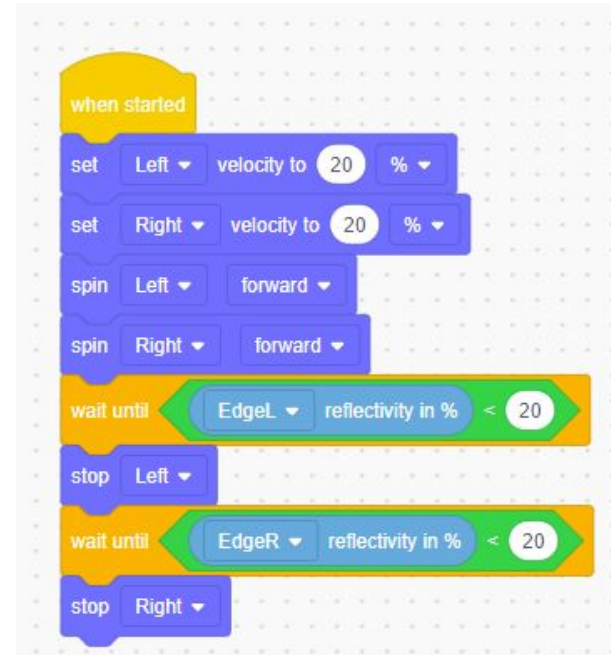
Aligning the robot to an edge

- 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

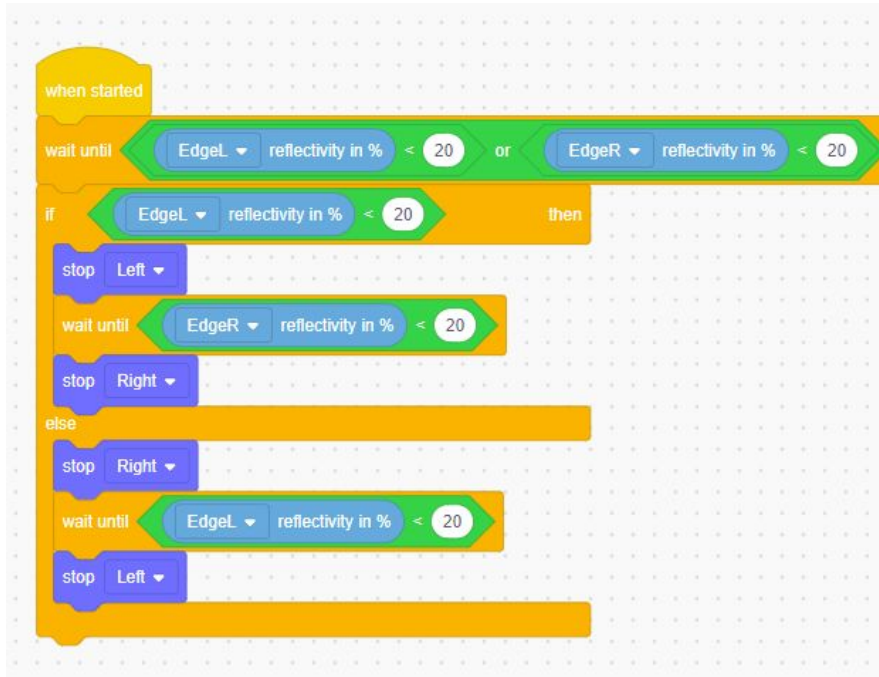
- Travel until LH color sensor reaches the edge, swing robot until it is aligned with the edge



Copyright © 2021 Robofest

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



Copyright © 2021 Robofest

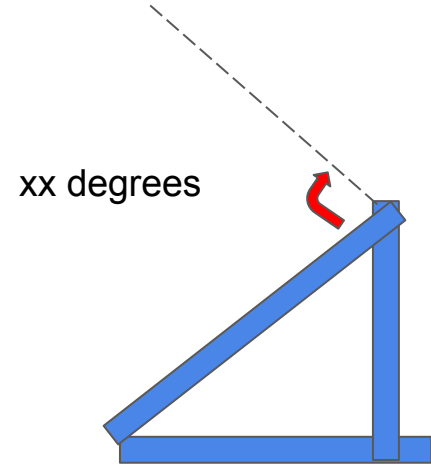
YouTube: <https://youtu.be/Jq0L2ekQARk>

Task 7

Manipulating Paper Rolls
Moving the Arm and Claw

How to Control the Arm and Claw

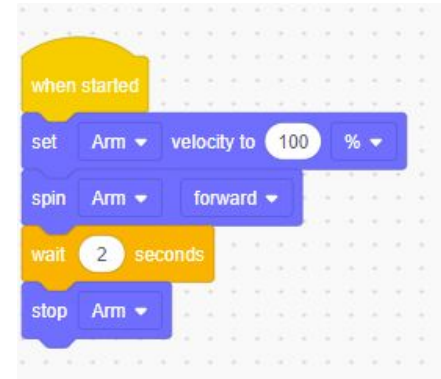
- Time
- Rotation(encoder degrees)



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



```
when started
set Arm velocity to 100 %
spin Arm forward
wait 2 seconds
stop Arm
```

Raise
Arm

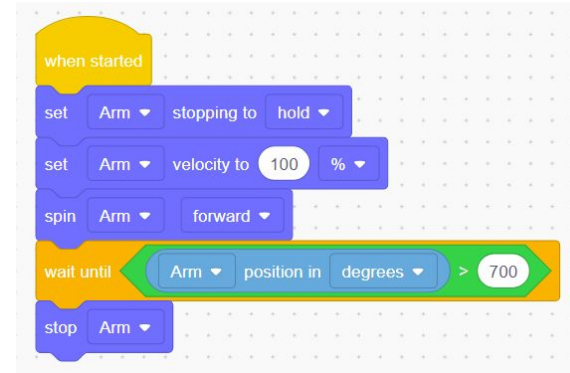


```
when started
set Arm velocity to 100 %
spin Arm reverse
wait 2 seconds
stop Arm
```

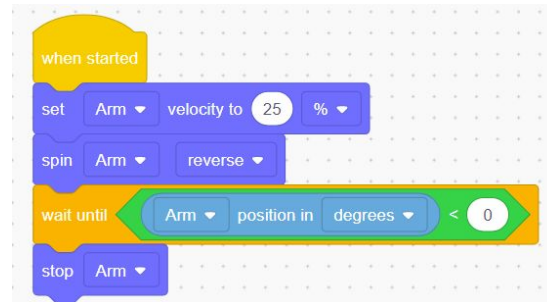
Lower
Arm

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



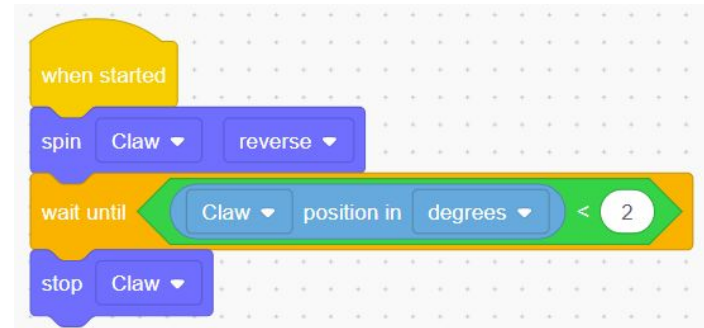
Lower
Arm

Moving the Claw

- As with the arm, may use time or rotation
- May need to overdrive motor to get enough grip
- May use encoder to open back to zero position



Use time to close



Use rotation to open

Combine Arm and Claw Movement

```
when started
  set Claw velocity to 100 %
  set Claw stopping to hold
  spin Claw forward
  wait 1 seconds
  stop Claw
  set Arm stopping to hold
  set Arm velocity to 100 %
  spin Arm forward
  wait until Arm position in degrees > 700
  stop Arm
  wait 2 seconds
```

```
set Arm velocity to 25 %
spin Arm reverse
wait until Arm position in degrees < 0
stop Arm
spin Claw reverse
wait until Claw position in degrees < 2
stop Claw
```

Task 8

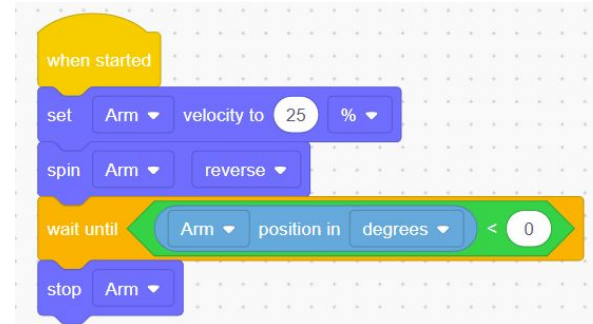
Building MyBlocks

MyBlocks

- 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 “MyBlocks” to create custom blocks that can replace sections of your program
- Variables can be used to make the MyBlock more flexible

MyBlocks

- 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
- The code may look like this

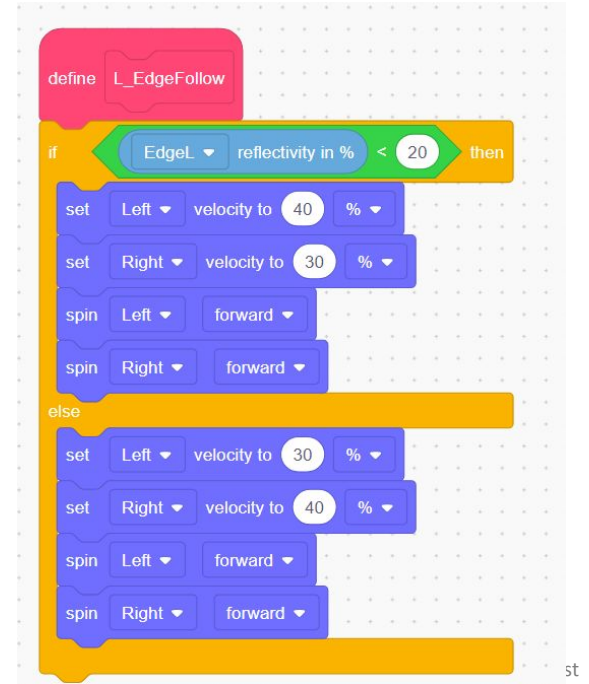


- My blocks will allow us to convert this to a single block

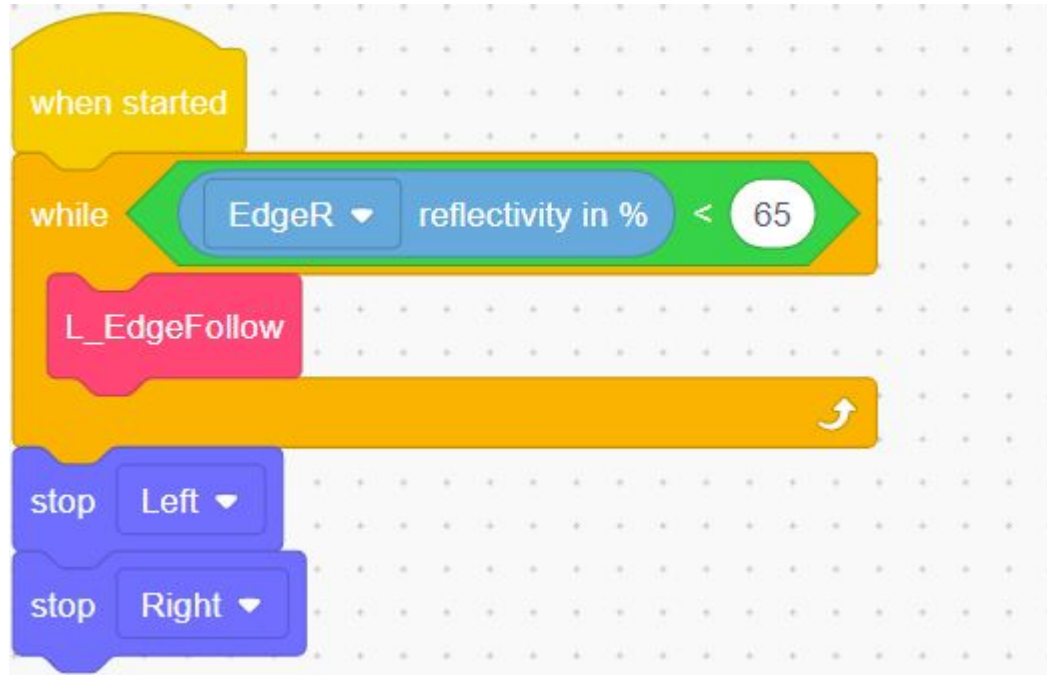


Create a Edge Following MyBlock

- Go to MyBlocks Menu
- Click “MyBlock”
- Name it “followLine”
- Drag the line following blocks and attach to the define MyBlock block



Use the MyBlock in the Program

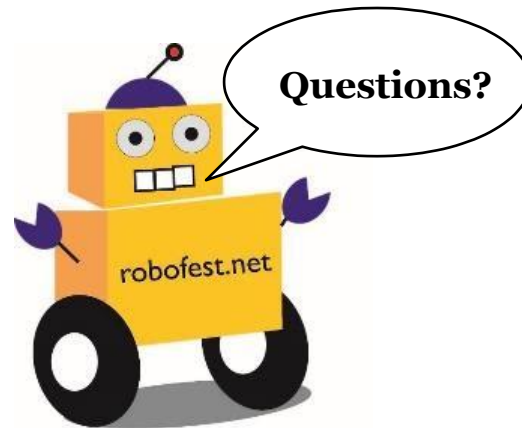


Copyright © 2021 Robofest

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

Little Robots, Big Missions



robofest@LTU.edu
LTU Computer Science

Copyright © 2021 Robofest