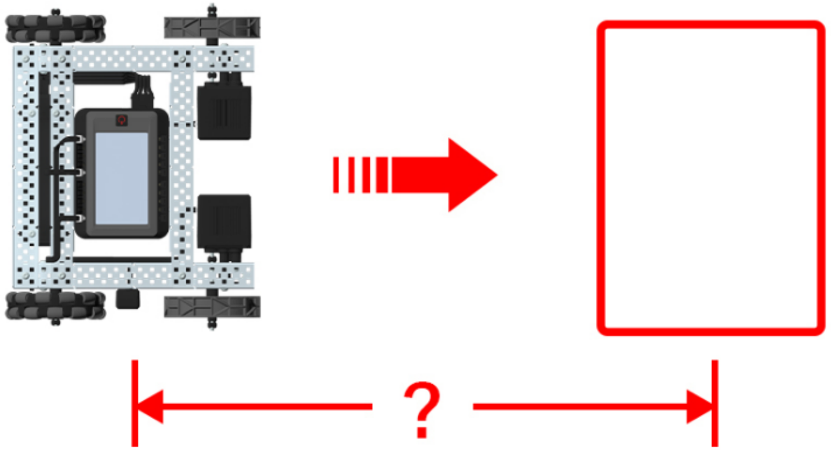
VexIQ Robotics Platform

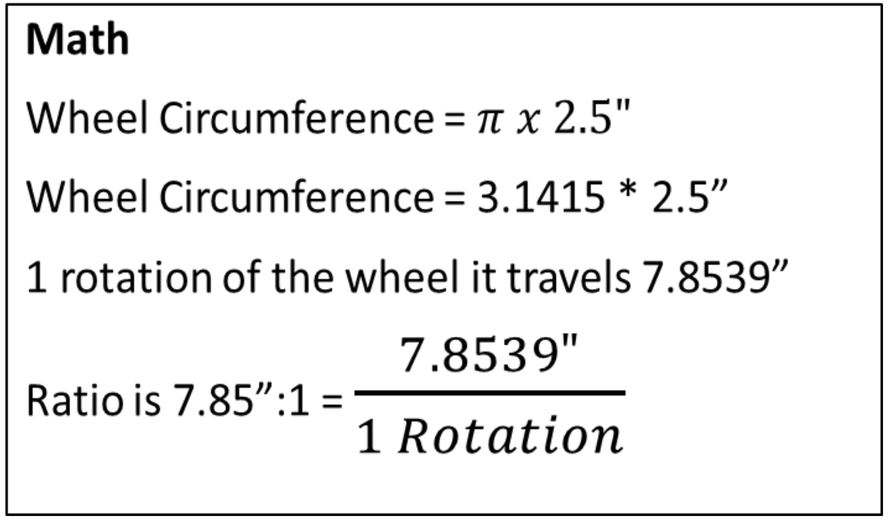
Travelling a set Distance….
Ratios & Proportions



Before you start (Need to knows)

- Circumference of Wheel = πD
- How far you will travel in one rotation of the wheel

Travelling a set Distance….
Ratios & Proportions



Math

Wheel Circumference = $\pi\ x\ 2.5"$

Wheel Circumference = 3.1415 * 2.5”

1 rotation of the wheel it travels 7.8539”

Ratio is 7.85”:1 = $\dfrac{7.8539"}{1\ Rotation}$

Math

Wheel Circumference = $\pi\ x\ 2.5"$

Wheel Circumference = 3.1415 * 2.5”

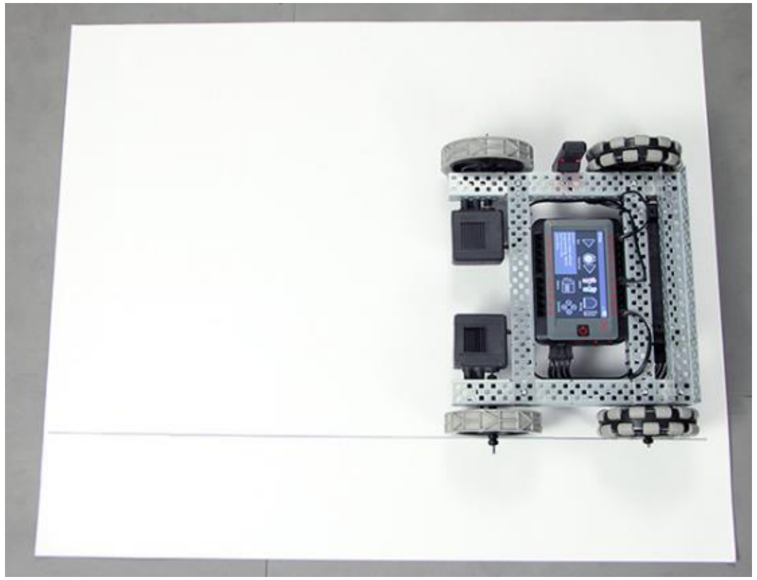
1 rotation of the wheel it travels 7.8539”

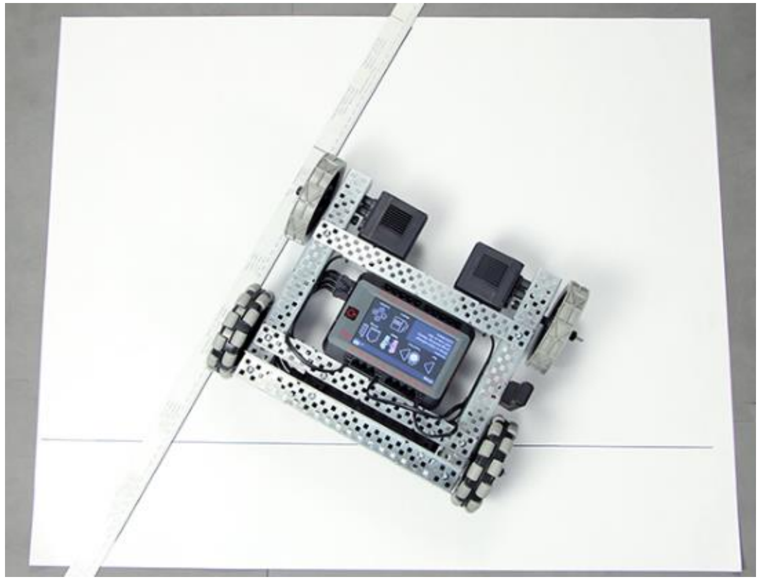
Ratio is 7.85”:1 = $\dfrac{7.8539"}{1\ Rotation}$

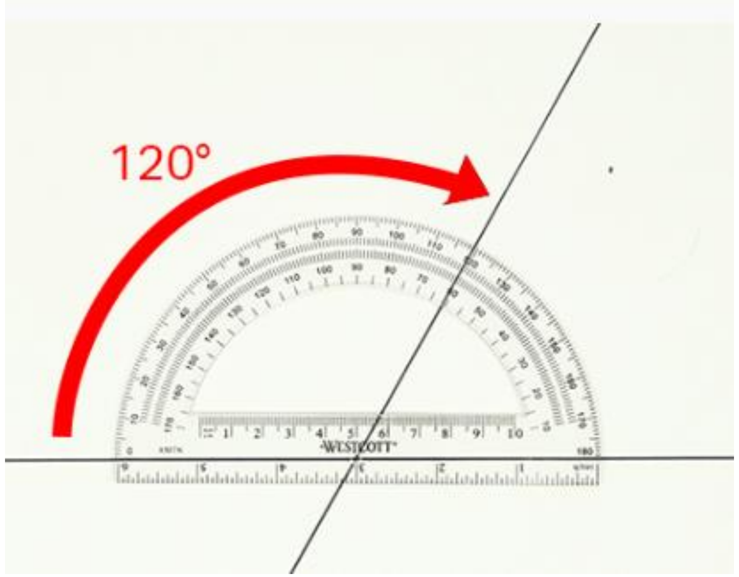
Proportion

1 rotation = 7.8539”	**Ratio is 7.85:1”**
Distance Travelled in 5 rotations	**5 x 7.8539” = 39.2699”**
Rotations need if travelling 24”	**24” ÷ 7.8539” = 3.0558”**

Turning Precisely….
Ratios & Proportions







Proportions for turning

1 rotation = 132^0 **Ratio is 120^0 :1 or** $\frac{\mathbf{120^0}}{\mathbf{1\ Rotation}}$

Wanting to Rotate 180^0 $\mathbf{180^0 \div \frac{120^0}{1\ Rotation} = 180^0\ X\ \frac{1 Rotation}{120^0}}$

Rotations need to travel 180^0 = $\frac{180^0}{120^0}$ = **1.50 rotations**

90^0 Turn = .75 rotations

45^0 Turn = .375 rotations

Ratios & Proportions
Mathematical Equations

Distance Travelled

$$\frac{7.8539''}{1\ Rotation} = \frac{Distance\ to\ Travel}{\#\ rotations}$$

I want to go 5 rotations

$$\frac{7.8539''}{1\ Rotation} = \frac{Distance\ to\ Travel}{\#\ rotations}$$

$$\frac{7.8539''}{1\ Rotation} = \frac{x}{5}$$

X = 5 * 7.8539

X = 39.2695”

I want to travel a distance of 24”

$$\frac{7.8539''}{1\ Rotation} = \frac{Distance\ to\ Travel}{\#\ rotations}$$

$$\frac{7.8539''}{1\ Rotation} = \frac{24''}{x}$$

7.8539x = 24

$$X = \frac{24''}{7.8539}$$

X = 3.0558 rotations

Amount of Degrees of Rotation

$$\frac{120^0}{1\ Rotation} = \frac{Distance\ to\ Rotate}{\#\ rotations}$$

Degrees after 5 rotation

$$\frac{120^0}{1\ Rotation} = \frac{Degrees\ to\ Rotate}{\#\ rotations}$$

$$\frac{120^0}{1\ Rotation} = \frac{x}{5}$$

X = 5 * 120

X = 600^0

I want Rotate 180^0

$$\frac{120^0}{1\ Rotation} = \frac{Degrees\ to\ Rotate}{\#\ rotations}$$

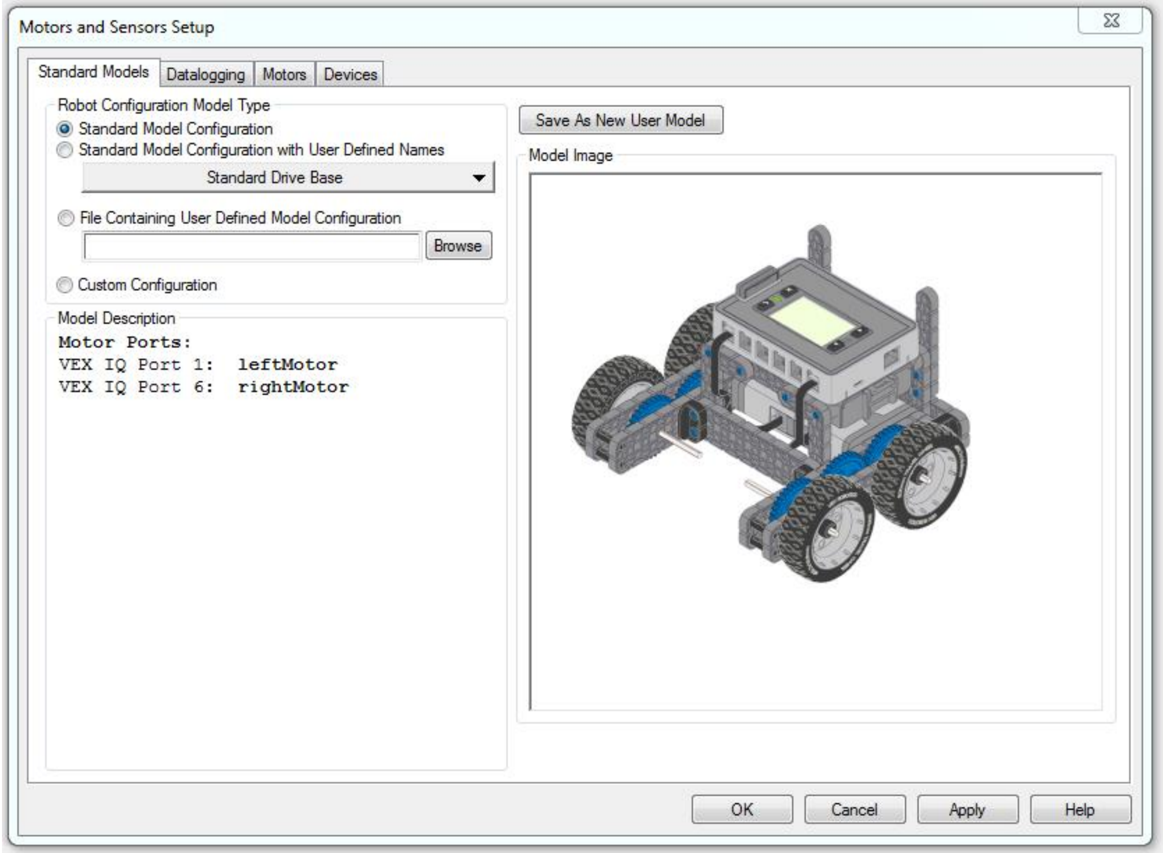
$$\frac{120^0}{1\ Rotation} = \frac{180^0}{x}$$

120x = 180

$$X = \frac{180}{120}$$

X = 1.50 rotations

Motor and Sensor Setup



Robot Configuration Model Type

● Standard Model Configuration

○ Standard Model Configuration with User Defined Names

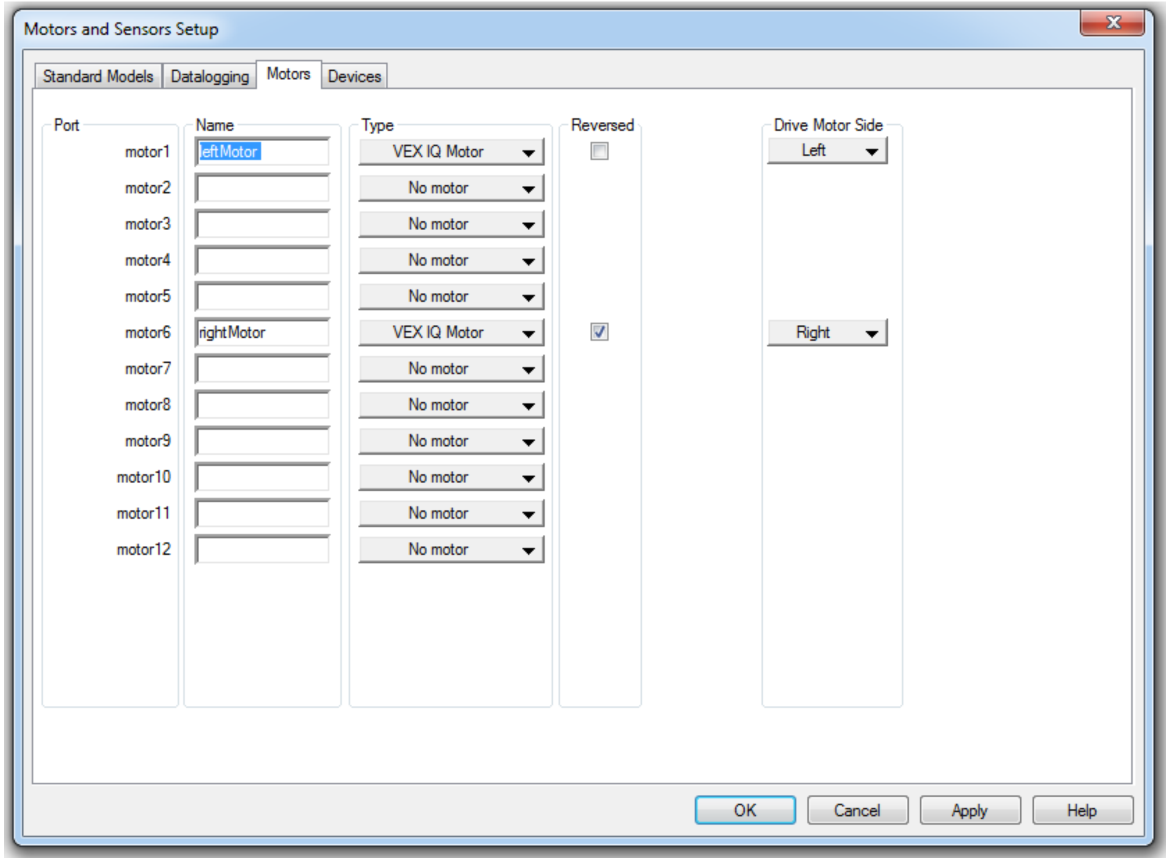
Standard Drive Base

○ File Containing User Defined Model Configuration

○ Custom Configuration

Model Description

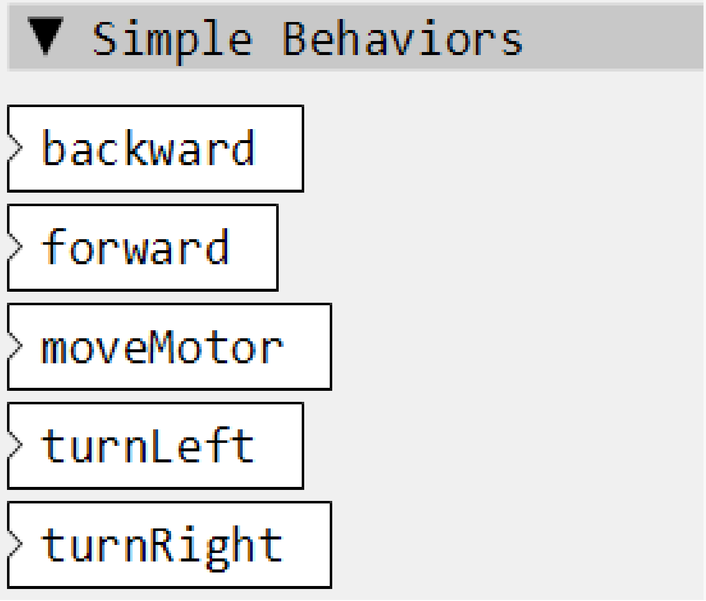
```
Motor Ports:
VEX IQ Port 1:  leftMotor
VEX IQ Port 6:  rightMotor
```



Port	Name	Type	Reversed	Drive Motor Side
motor1	leftMotor	VEX IQ Motor	□	Left
motor2		No motor		
motor3		No motor		
motor4		No motor		
motor5		No motor		
motor6	rightMotor	VEX IQ Motor	☑	Right
motor7		No motor		
motor8		No motor		
motor9		No motor		
motor10		No motor		
motor11		No motor		
motor12		No motor		

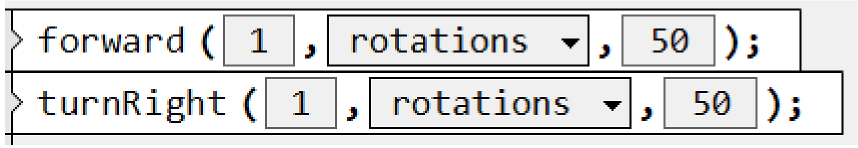
Always make sure your motor ports are setup correctly. The Standard Drive base as shown above uses Port 1 and Port 6. If you are in different Ports your will have to change them in the Motors Tab.

Basic Code
Using Simple Behaviors

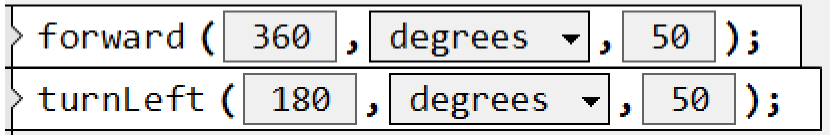


```
▼ Simple Behaviors
backward
forward
moveMotor
turnLeft
turnRight
```

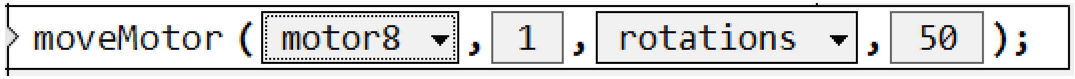
Simple Behaviors is the easiest way to get your robot driving. As long as you define the motors correctly in the “Motor and Sensor Setup” based on the previous slide.



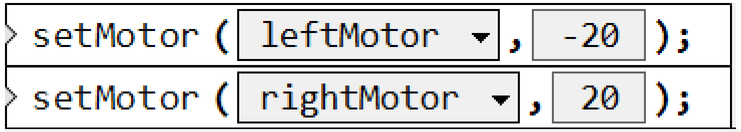
```
forward ( 1 , rotations , 50 );
turnRight ( 1 , rotations , 50 );
```



```
forward ( 360 , degrees , 50 );
turnLeft ( 180 , degrees , 50 );
```

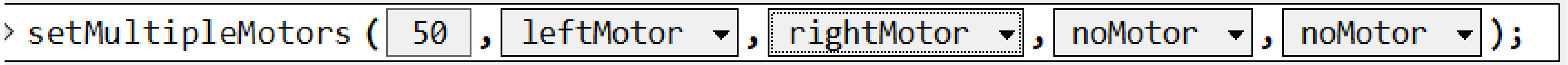


```
moveMotor ( motor8 , 1 , rotations , 50 );
```



```
setMotor ( leftMotor , -20 );
setMotor ( rightMotor , 20 );
```

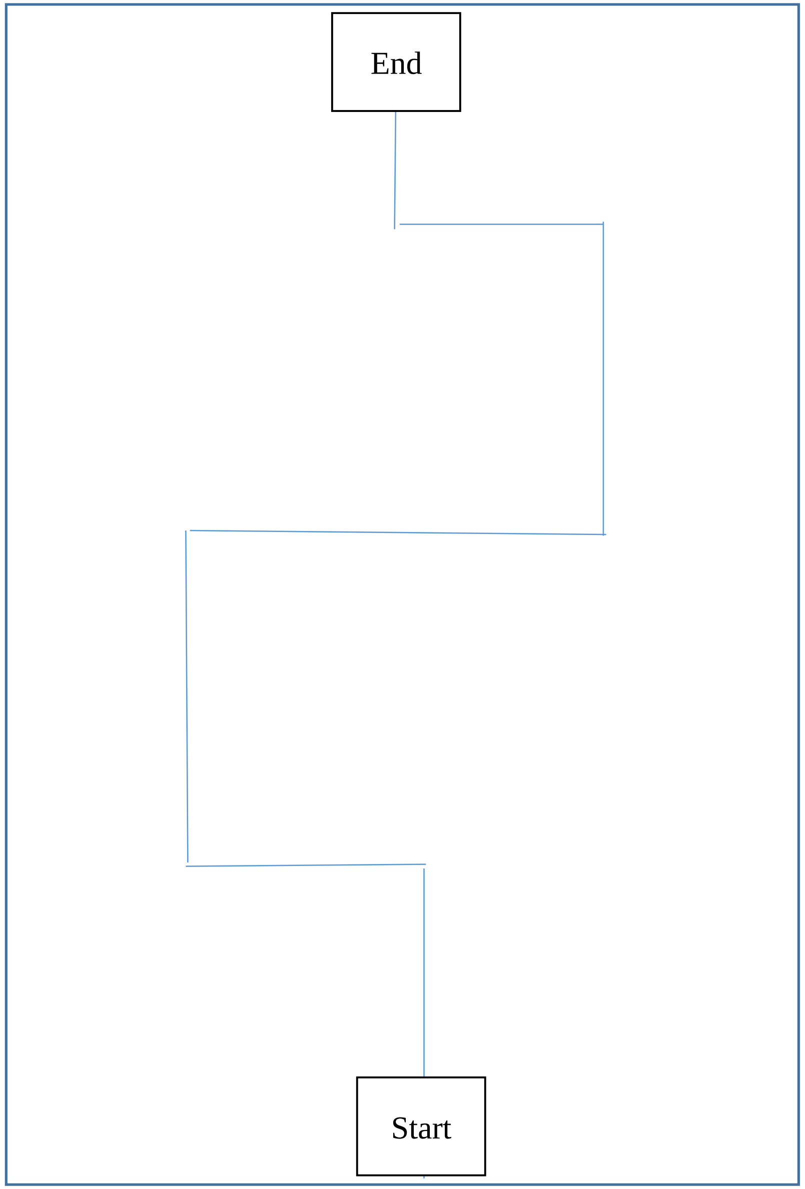
Not using simple behaviors is more difficult. These lines just “Set” the motors but don’t even move them



```
setMultipleMotors ( 50 , leftMotor , rightMotor , noMotor , noMotor );
```

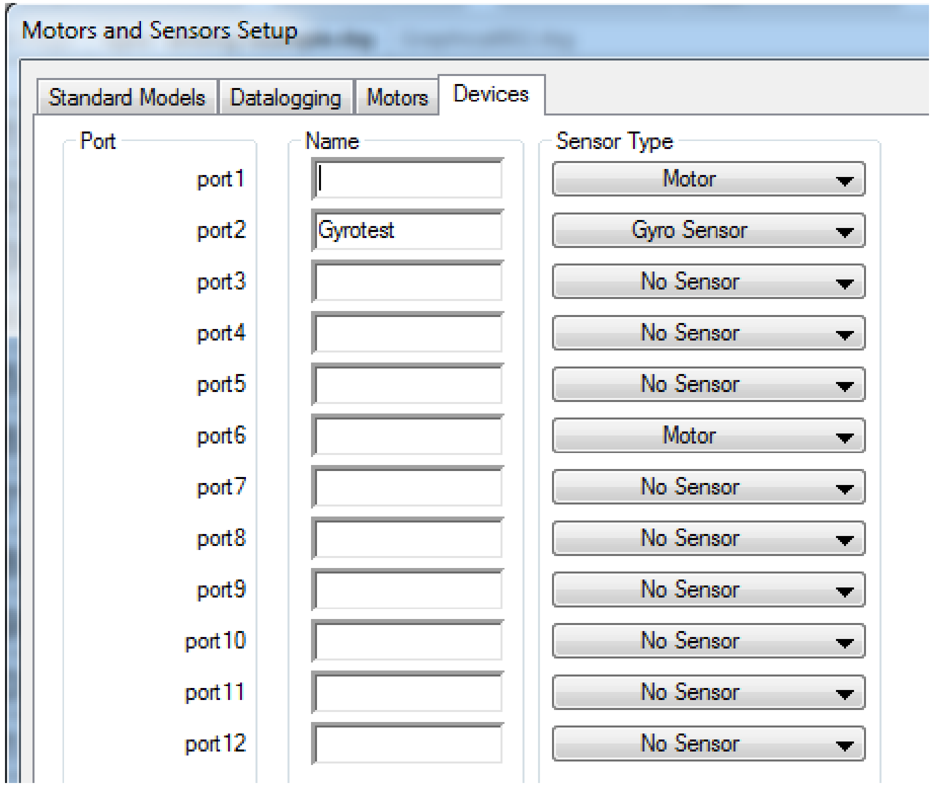
Maze Assignment

STEP #1	Measure the distance CENTRE to CENTRE of each leg of the trip and record it on a piece of paper
STEP #2	Convert ALL the distances to ROTATIONS using the equations you have learned
STEP #3	Calculate your Turning ROTATION precisely
STEP #4	Program the robot
STEP #5	Complete the challenge in the FIRST TRY!!!



Gyro

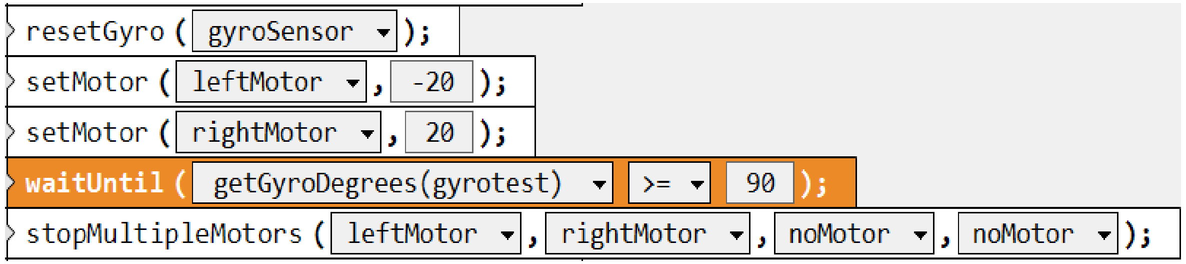
Gyro measures turn rate and calculates direction based on your robots movement



Motors and Sensors Setup

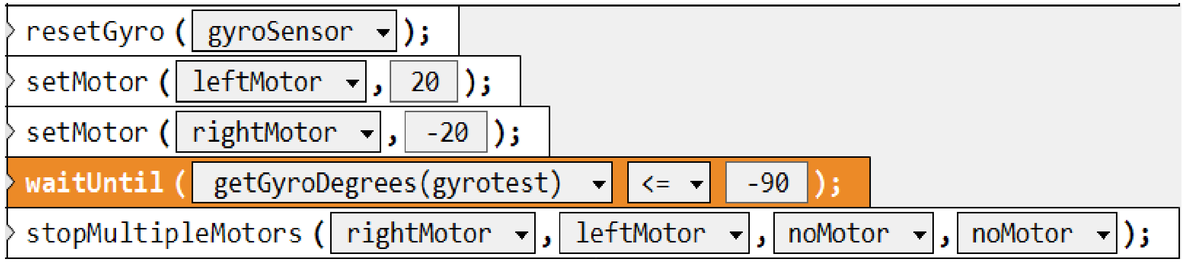
Standard Models | Datalogging | Motors | Devices

Port	Name	Sensor Type
port1		Motor
port2	Gyrotest	Gyro Sensor
port3		No Sensor
port4		No Sensor
port5		No Sensor
port6		Motor
port7		No Sensor
port8		No Sensor
port9		No Sensor
port10		No Sensor
port11		No Sensor
port12		No Sensor



```
resetGyro ( gyroSensor );
setMotor ( leftMotor , -20 );
setMotor ( rightMotor , 20 );
waitUntil ( getGyroDegrees(gyrotest) >= 90 );
stopMultipleMotors ( leftMotor , rightMotor , noMotor , noMotor );
```

Left Turn

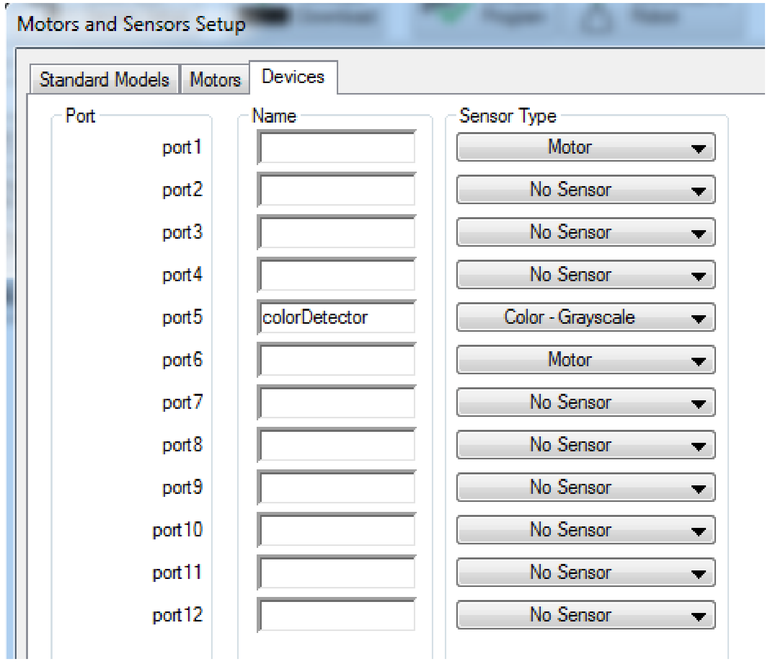


```
resetGyro ( gyroSensor );
setMotor ( leftMotor , 20 );
setMotor ( rightMotor , -20 );
waitUntil ( getGyroDegrees(gyrotest) <= -90 );
stopMultipleMotors ( rightMotor , leftMotor , noMotor , noMotor );
```

Right Turn

Line Following

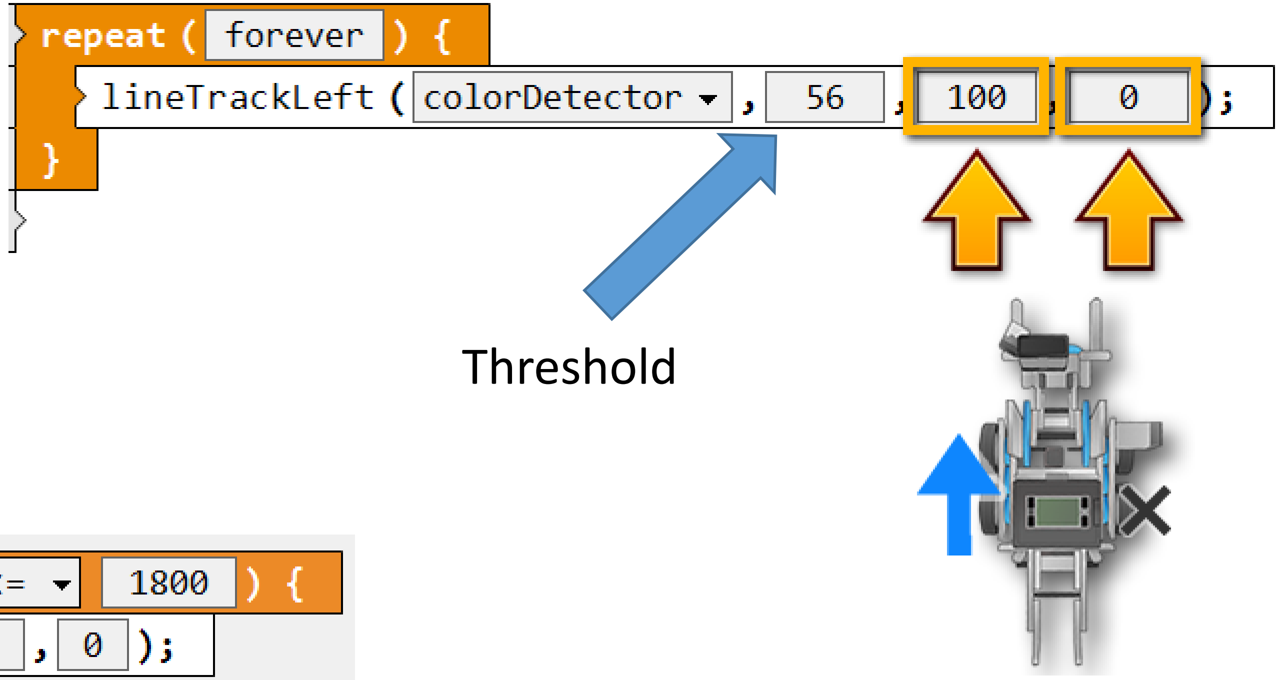
The Color Sensor detects the color of objects. When following a line it goes into a "Greyscale Mode" that provides the amount of reflected light from the surface under it. Dark surfaces absorb light, while bright surfaces reflect it.



Motors and Sensors Setup

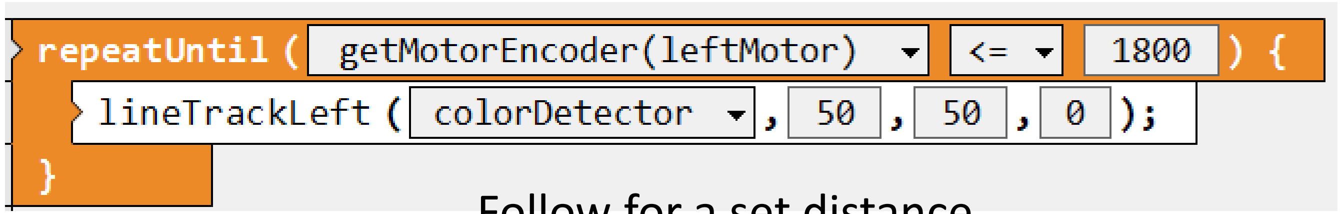
Standard Models | Motors | Devices

Port	Name	Sensor Type
port1		Motor
port2		No Sensor
port3		No Sensor
port4		No Sensor
port5	colorDetector	Color - Grayscale
port6		Motor
port7		No Sensor
port8		No Sensor
port9		No Sensor
port10		No Sensor
port11		No Sensor
port12		No Sensor



```
repeat ( forever ) {
    lineTrackLeft ( colorDetector , 56 , 100 , 0 );
}
```

Threshold

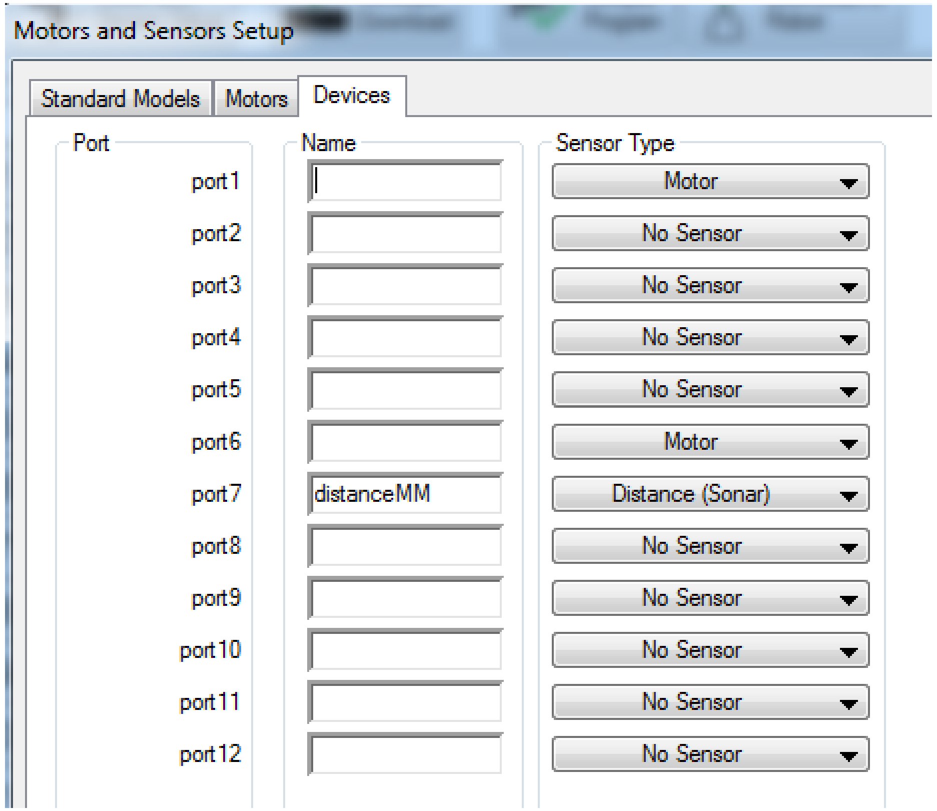


```
repeatUntil ( getMotorEncoder(leftMotor) <= 1800 ) {
    lineTrackLeft ( colorDetector , 50 , 50 , 0 );
}
```

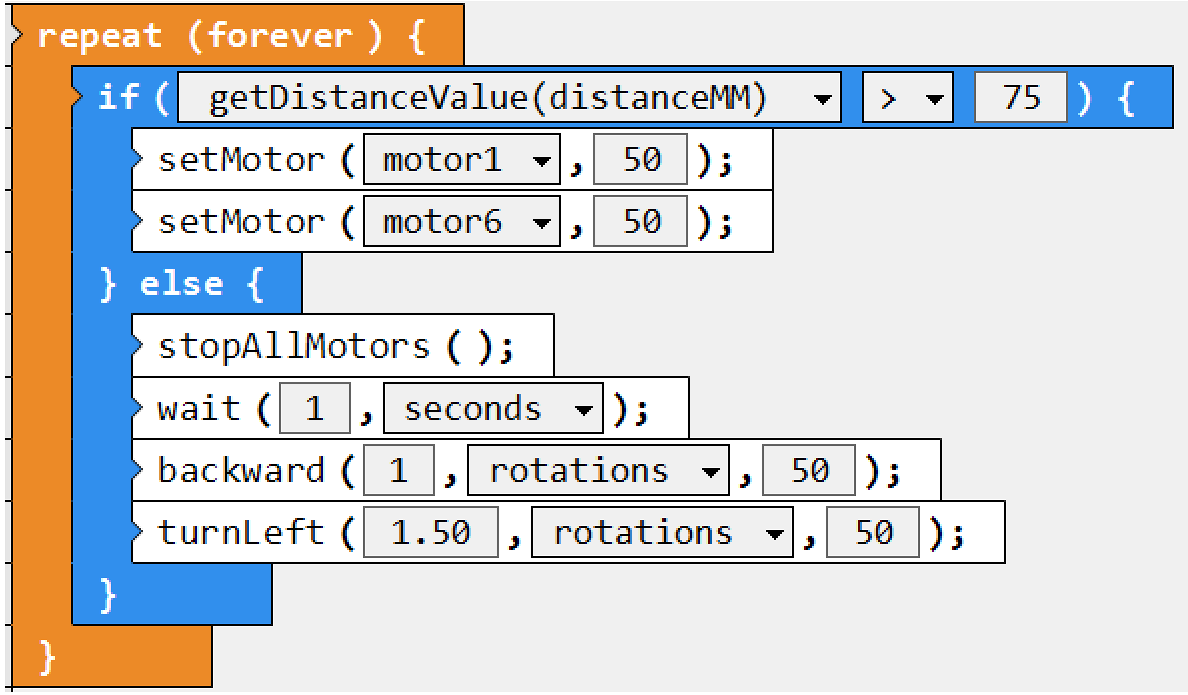
Follow for a set distance

Distance Sensor

The Distance Sensor uses ultrasonic sound waves to measure distances from 50mm to 1m



Port	Name	Sensor Type
port1		Motor
port2		No Sensor
port3		No Sensor
port4		No Sensor
port5		No Sensor
port6		Motor
port7	distanceMM	Distance (Sonar)
port8		No Sensor
port9		No Sensor
port10		No Sensor
port11		No Sensor
port12		No Sensor



```
repeat (forever ) {
  if ( getDistanceValue(distanceMM) > 75 ) {
    setMotor ( motor1 , 50 );
    setMotor ( motor6 , 50 );
  } else {
    stopAllMotors ( );
    wait ( 1 , seconds );
    backward ( 1 , rotations , 50 );
    turnLeft ( 1.50 , rotations , 50 );
  }
}
```

Code that is used to avoid a wall.....