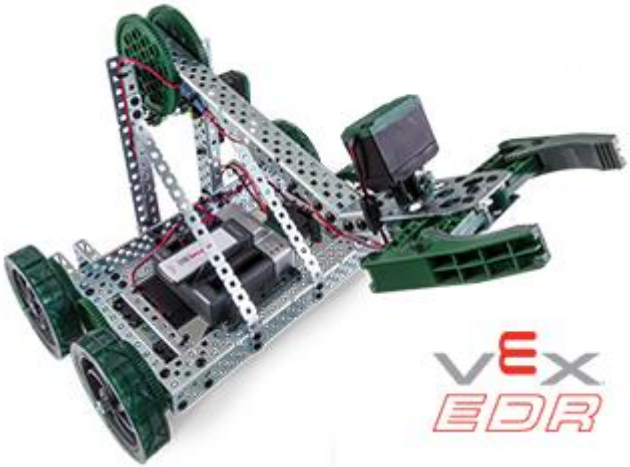
Vex Robotics Summer Camp Day 4



Sensors – Basics

- What are sensors?
  - Robotic sensors are used to estimate a robot's condition and environment. These signals are passed to a controller to enable appropriate behavior

Sensors – Basics

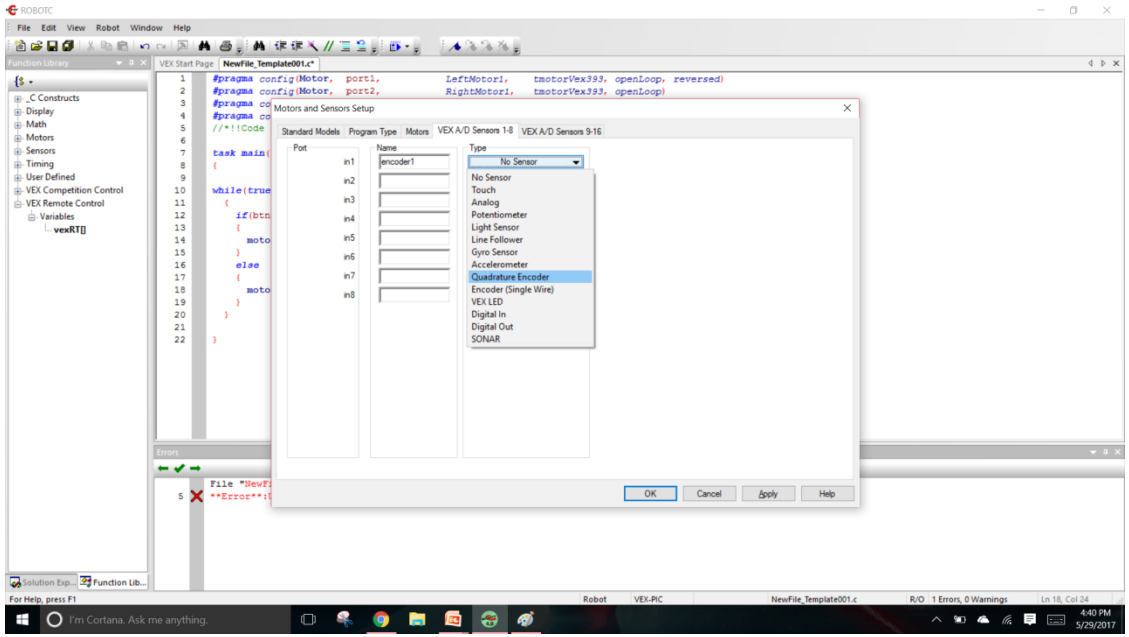
- Can you name a few sensors in your everyday life?

Sensors – Basics

- Examples
  - Heart Rate monitor
  - Car lights
  - Microphone
  - Etc

Sensors – Quadrature encoder

- What is a Quadrature?
  - It is a sensor that measures ticks
  - Needs an axle through it
  - One degree spun is equal to one ticks
- Go to “robot”→”motors and sensors” and make a new sensor in “A/D sensors 1-8”



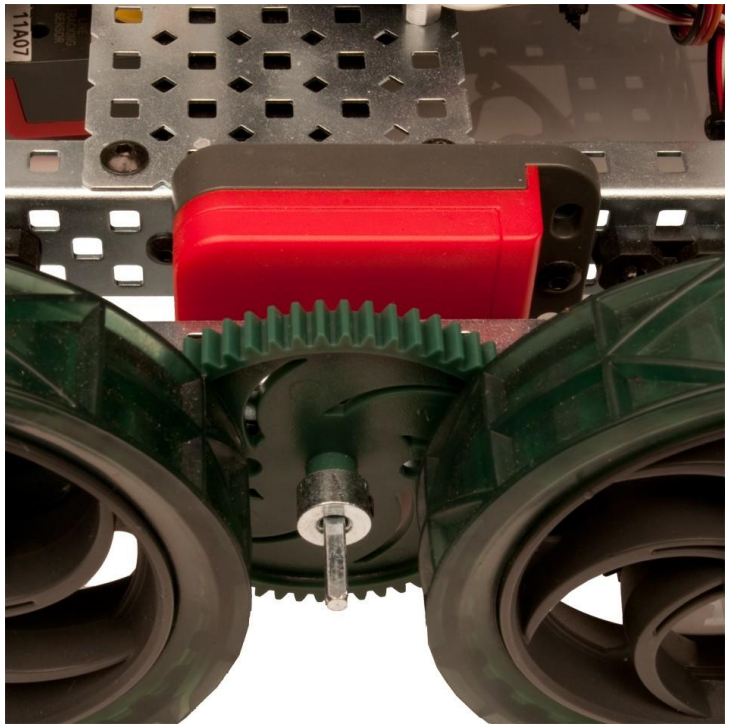
Sensors – Quadrature encoder

- The tern “SensorValue” calls on the value the Quadrature is giving
- We first use “SensorValue” to set the Quadrature to zero then we use it for the conditional statement

```
#pragma config(Sensor, dgtl1,  rightEncoder,        sensorQuadEncoder)
#pragma config(Sensor, dgtl3,  leftEncoder,         sensorQuadEncoder)
#pragma config(Motor,  port2,           rightMotor,    tmotorNormal, openLoop, reversed)
#pragma config(Motor,  port3,           leftMotor,     tmotorNormal, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

task main()
{
  wait1Msec(2000);  // 2 Second Delay

  //Clear Encoders
  SensorValue[rightEncoder] = 0;
  SensorValue[leftEncoder] = 0;

  while(SensorValue[leftEncoder] < 1800)  // While less than 5 rotations on the leftEncoder...
  {
    //...Move Forward
    motor[rightMotor] = 63;
    motor[leftMotor] = 63;
  }

}
```

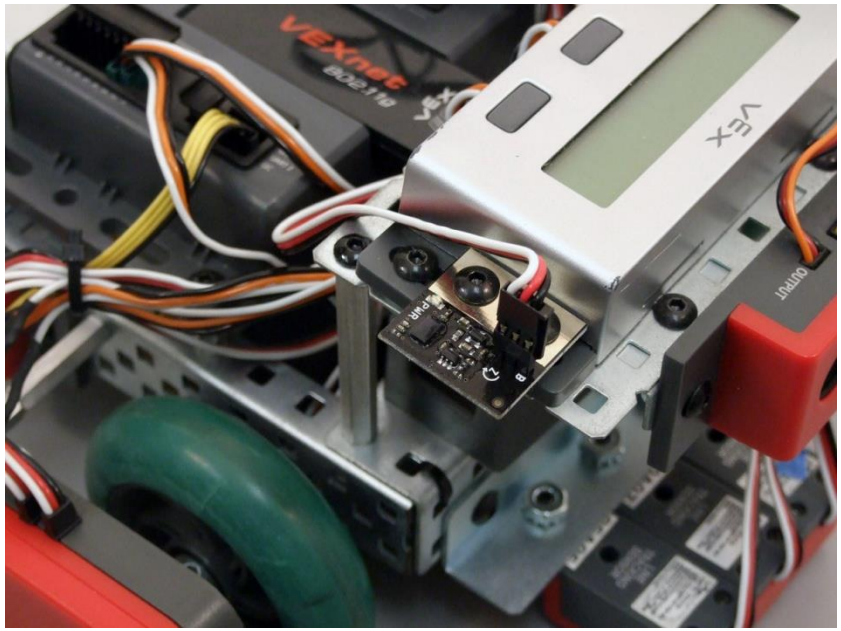
Sensors – Practice

- Attach two Quadratures to the robot. Make the robot move forward for 5 rotations, point turn for 2 rotations, the move backwards for 5 rotations



Sensors – Gyroscope

- The Gyroscope
  - Sensor that tracks degree of rotation on whole robot
  - One tick is equal to ten degrees



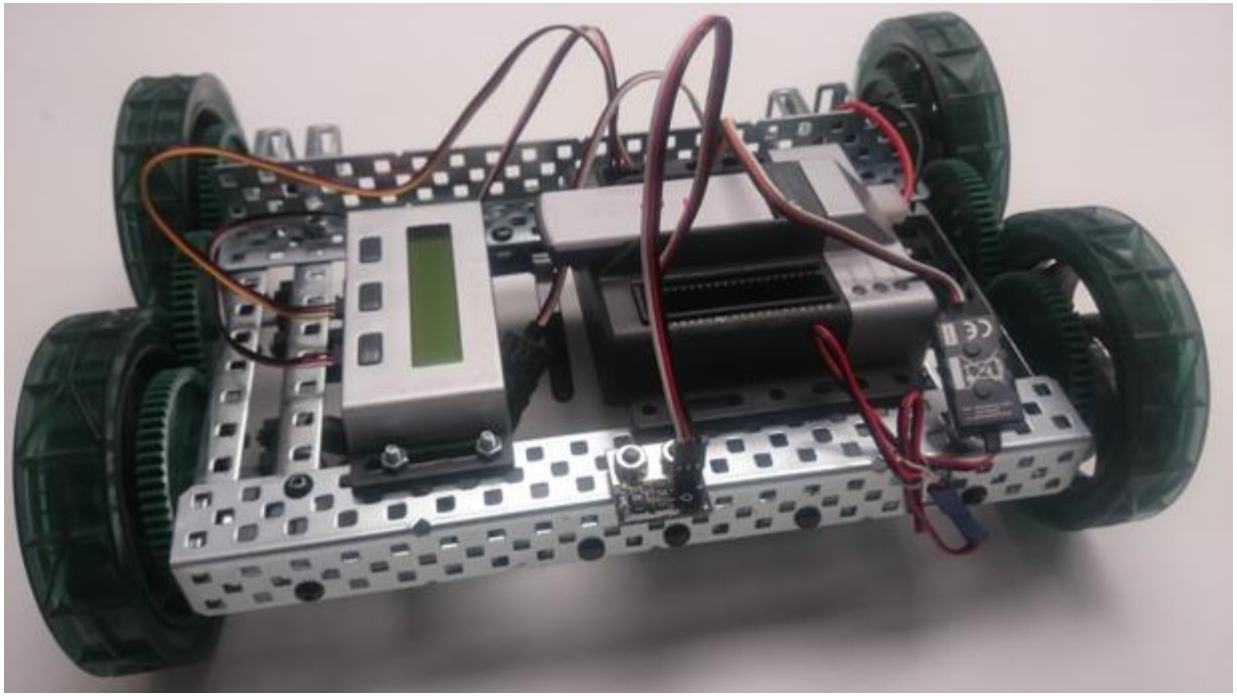
Sensors – Gyroscope

- Below, we use the Gyroscope to rotate the robot 90 degrees

```
#pragma config(Motor,  port2,           LeftMotor2,    tmotorVex393, openLoop)
#pragma config(Motor,  port3,           RightMotor1,   tmotorVex393, openLoop, r
#pragma config(Motor,  port4,           RightMotor2,   tmotorVex393, openLoop, r
//*!!Code automatically generated by 'ROBOTC' configuration wizard

task main()
{
//clears our any value the previous port had of the sensor
SensorType[in8]= sensorNone;
wait1Msec(1000);

//Sets the port equal to the gyroscope sensor
SensorType[in8]=sensorGyro;
wait1Msec(2000);

int degrees = 900;
//90 degrees because 1 tick is equal to ten degrees

//turns robot 90 degrees
while (abs(sensorValue[in8]) < degrees )
{
motor[LeftMotor1]=25;
motor[LeftMotor2]=25;
motor[RightMotor1]=-25;
motor[RightMotor2]=-25;
}


}
```

Sensors – Practice

- Use gyroscope to move robot forward for 5 seconds, turn 180 degrees, then move forward for 10 seconds.



Sensors – Buttons

- Buttons on robots take only two values, 0 or 1
- When pressed, button takes value of 1. When it is not pressed, it takes value of 0.

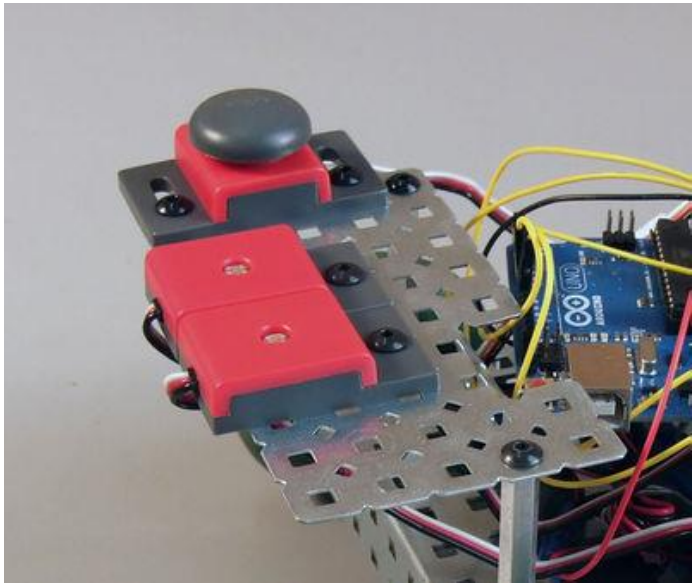
Sensors – touch sensors

- In the program below, the left and right motor will continuously spin until the button is pressed.

```
#pragma config(Sensor, dgtl6,  touchSensor,         sensorTouch)
#pragma config(Motor,  port2,           rightMotor,    tmotorNormal, openLoop, reversed)
#pragma config(Motor,  port3,           leftMotor,     tmotorNormal, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

task main()
{
  wait1Msec(2000);  // Robot waits for 2000 milliseconds before executing program

  while(SensorValue(touchSensor) == 0)  // Loop while robot's bumper/touch sensor isn't pressed in
  {
    motor[rightMotor] = 63;      // Motor on port2 is run at half (63) power forward
    motor[leftMotor]  = 63;      // Motor on port3 is run at half (63) power forward
  }
}
```

Sensors – Practice

- Program the robot to move until it hits a wall
- When it hits a wall, make it stop for 5 seconds, move back for 2 seconds, turn right, then continue moving forward.



Maze Challenge

- Using all three sensors we learn, program your robot to move through the maze. The fastest one wins!