Python Environment Setup

Python Setup

- Download Python installation file for your OS and Install

  https://www.python.org/downloads/

- Install pip module on your python

  - Open your web browser and download
    https://bootstrap.pypa.io/get-pip.py
  - Open terminal
    Install pip “python get-pip.py”

- Install pyserial module using pip on python
  “python -m pip install pyserial”

  https://pyserial.readthedocs.io/en/latest/pyserial.html#installation

- Install OpenCV for Python
  “python -m pip opencv-python”

  https://pypi.org/project/opencv-python/

Simple Drawing and pySerial

Python Code

```
from time import sleep
import serial
import numpy as np
import cv2

img = np.zeros((512,512,3), np.uint8)

ser = serial.Serial('/dev/ttyACM0', 115200) # Establish the connection on a specific port
counter = 32 # Below 32 everything in ASCII is gibberish

xPnt = 0
yPnt = 0
while True:
    msg = ser.readline() # Read the newest output from the Arduino
    print msg
    try:
                        yPnt = int(msg)
            img = cv2.line(img,(xPnt,256),(xPnt,yPnt),(255,0,0),5)
            cv2.imshow('image',img)
            cv2.waitKey(3)
            #cv2.destroyAllWindows()
#           sleep(0.1) # Delay for one tenth of a second
            xPnt +=1
            if xPnt > 512:
                        xPnt = 0
                        img = np.zeros((512,512,3), np.uint8)

    except ValueError:
                        pass

#           counter +=1
#           ser.write(str(chr(counter))) # Convert the decimal number to ASCII then send it to the Arduino
```

Simple Drawing and pySerial

Arduino Code

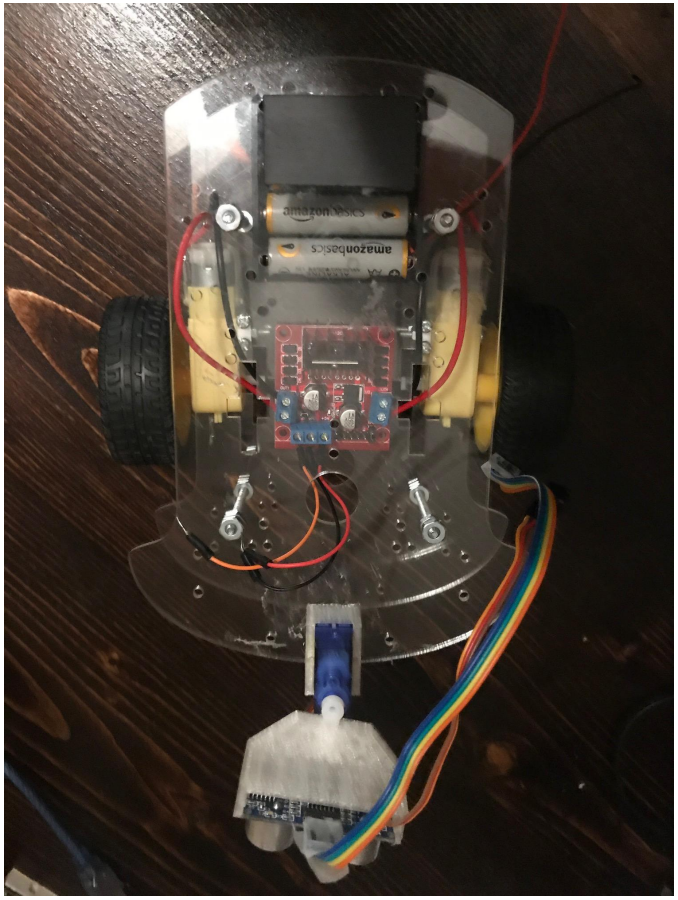
```
const int analogInPin = A1;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 13; // Analog output pin that the LED is attached to

int sensorValue = 0;         // value read from the pot
int outputValue = 0;         // value output to the PWM (analog out)

void setup() {
  Serial.begin(115200);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 512);
  analogWrite(analogOutPin, outputValue);

  // print the results to the Serial Monitor:
  Serial.println(outputValue);

  delay(20);
}
```

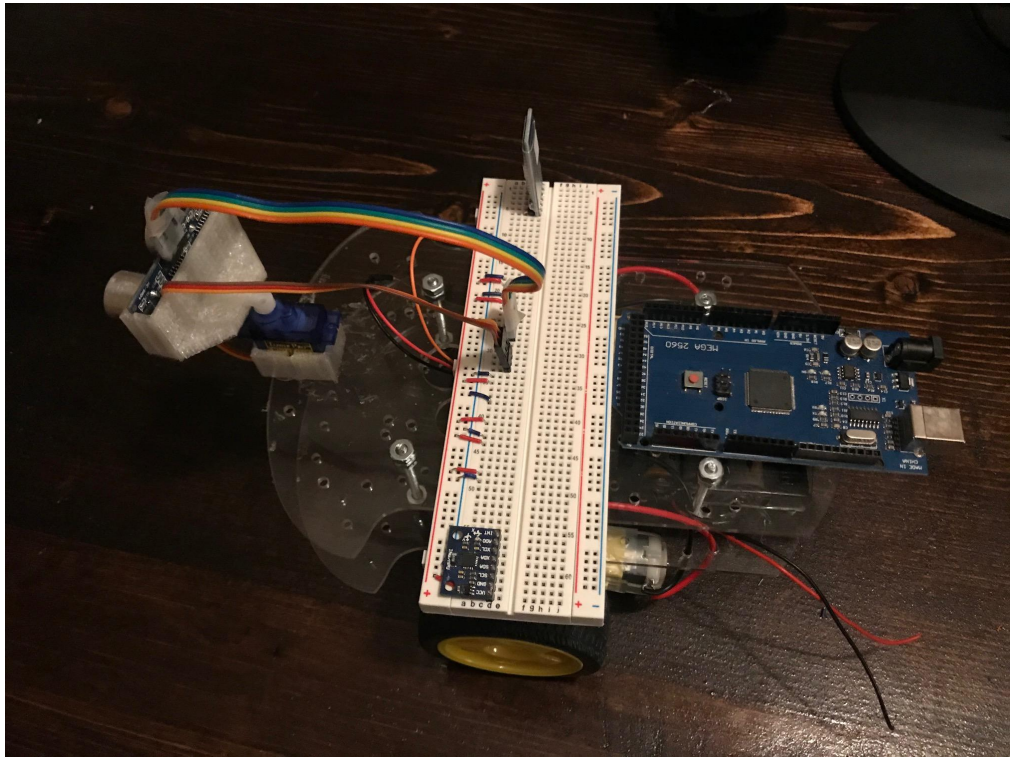
Read IMU data and Drawing Compass

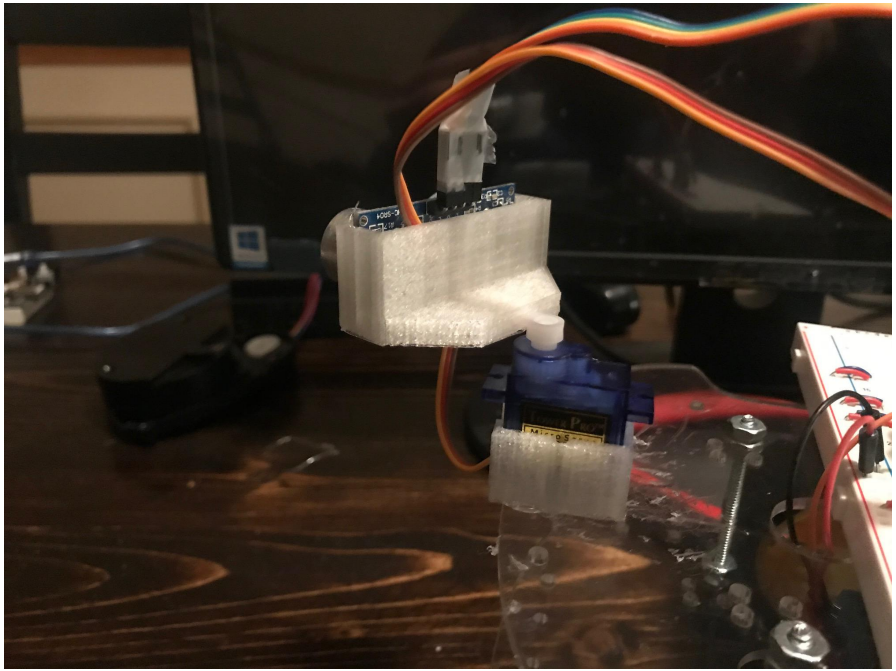
Read z value for mapping on Compass Drawing

Final Individual Project

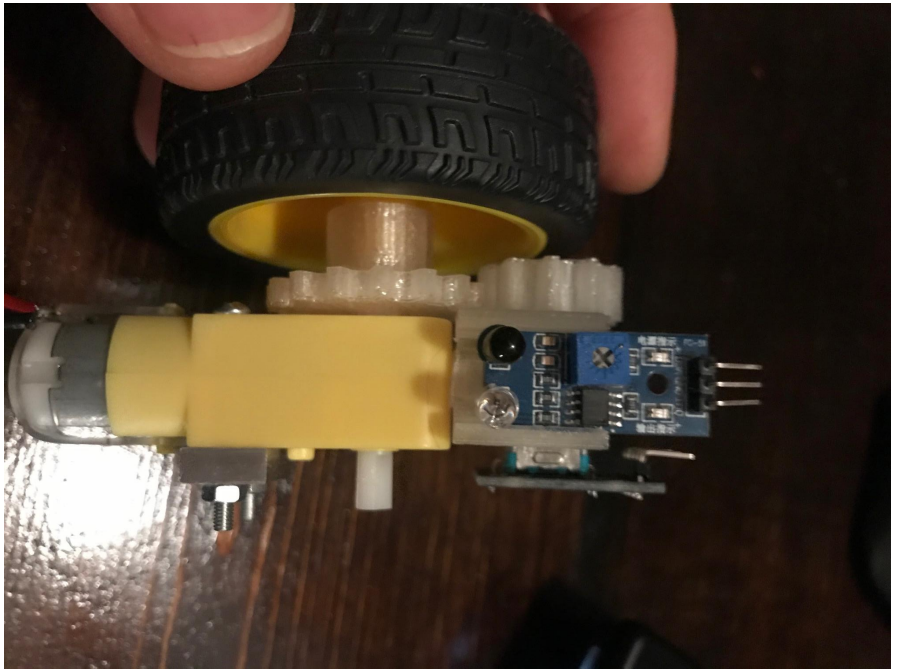
Simple Rescue Robot





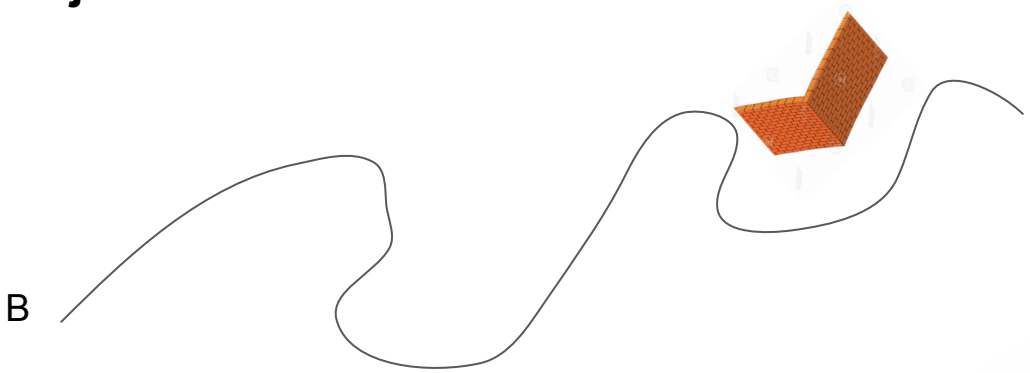


Sonar Scanner Module

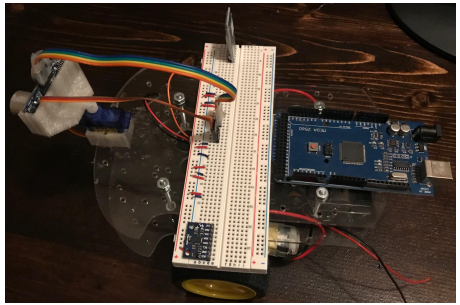


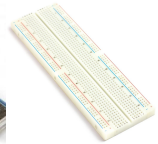
Wheel Encoder and Line Detector Module

Project Scenario



A





Simple Rescue Robot

- **Simple Rescue Robot**
  - **Sensing**
    - **Scanning using a Sonar (2 feet range)**
    - **Line Sensing (two IR sensors)**
    - **Odometer (two Rotary Encoders)**
    - **Orientation Sensing (one IMU sensor through I^2C)**
  - **Thinking**
    - **Communication with Remote Controller for mission/control update**
    - **Localization (Final Goal)**
    - **Mission**
      - **Manual Control**
      - **Line Following**
      - **Path Recording and Replay (Final goal)**
  - **Acting**
    - **DC Motor Control (one dual channel H-bridge motor driver)**
    - **Scanning Position Control (one servo motor : 30 ~ 150 degree)**

Simple Rescue Robot

- **Simple Remote Controller**
  - **Sensing**
    - **Joystick Status**
      - **X,Y axis and swidth**
  - **Thinking**
    - **Communication with your Simple Turtlebot for mission/control update**
      - **Manual Control**
      - **Path Recording and Replay**
  - **Acting**
    - **LEDs Display on your breadboard (Your design)**
      - **Line detect indication (two LEDs)**
    - **Display Sonar Data and IMU data on your pc through Serial communication**
    - **Drawing current location and path on your pc through Serial communication**


UNL
1.23 9
DSL
2.36 9