LAB 4 Report

I. **Task 1:** Testing MCU

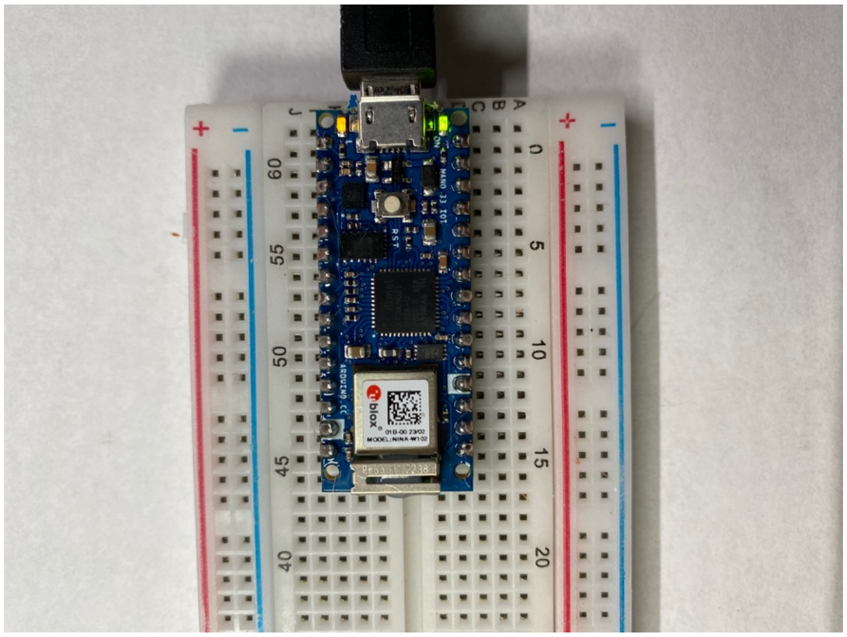
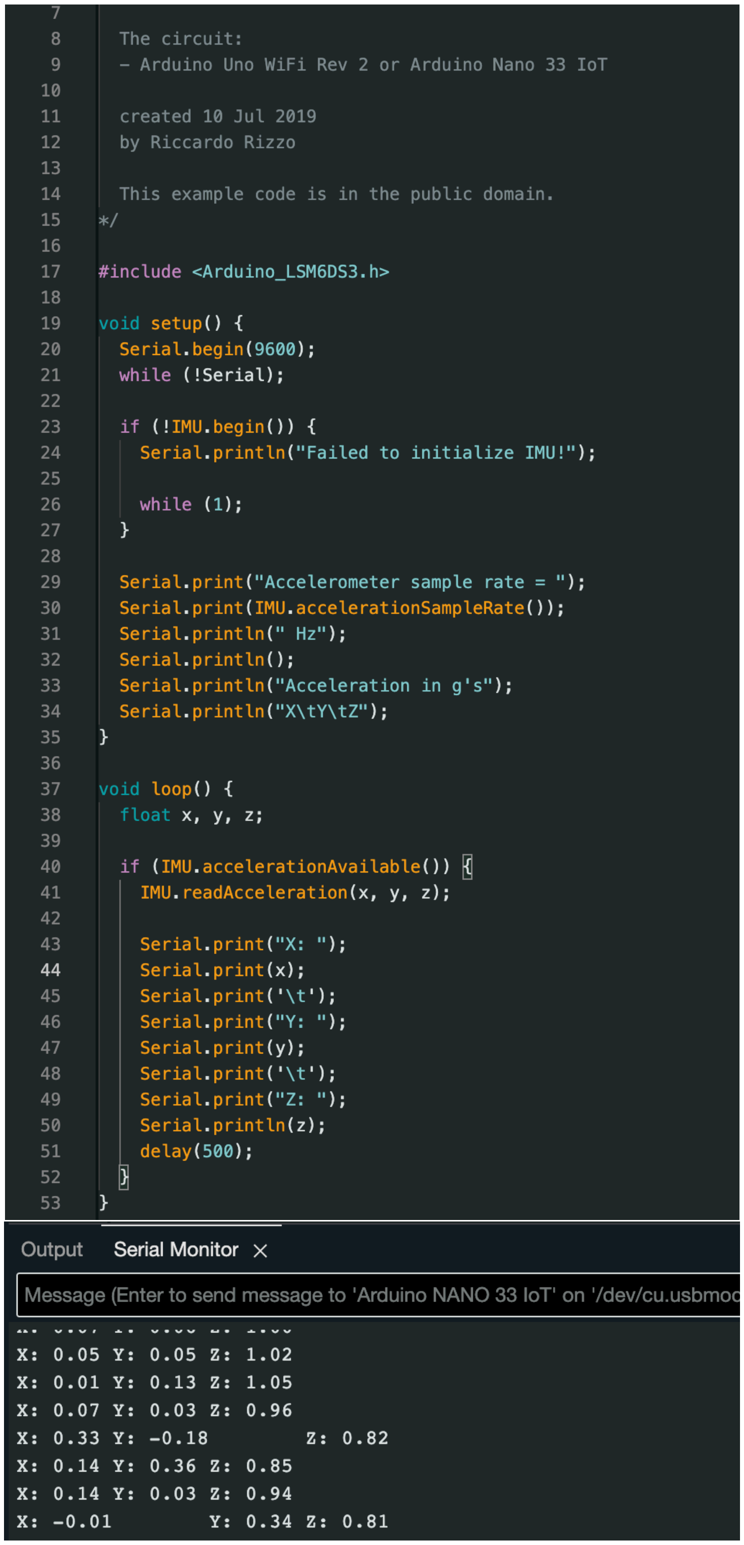


Figure 1. Running the Blink example code for Arduino Nano

II. **Task 1:** Running Accelerometer script



```

  The circuit:
  - Arduino Uno WiFi Rev 2 or Arduino Nano 33 IoT

  created 10 Jul 2019
  by Riccardo Rizzo

  This example code is in the public domain.
*/

#include <Arduino_LSM6DS3.h>

void setup() {
  Serial.begin(9600);
  while (!Serial);

  if (!IMU.begin()) {
    Serial.println("Failed to initialize IMU!");

    while (1);
  }

  Serial.print("Accelerometer sample rate = ");
  Serial.print(IMU.accelerationSampleRate());
  Serial.println(" Hz");
  Serial.println();
  Serial.println("Acceleration in g's");
  Serial.println("X\tY\tZ");
}

void loop() {
  float x, y, z;

  if (IMU.accelerationAvailable()) {
    IMU.readAcceleration(x, y, z);

    Serial.print("X: ");
    Serial.print(x);
    Serial.print('\t');
    Serial.print("Y: ");
    Serial.print(y);
    Serial.print('\t');
    Serial.print("Z: ");
    Serial.println(z);
    delay(500);
  }
}
```

Output
Serial Monitor
Message (Enter to send message to 'Arduino NANO 33 IoT' on '/dev/cu.usbmod

```
X: 0.05 Y: 0.05 Z: 1.02
X: 0.01 Y: 0.13 Z: 1.05
X: 0.07 Y: 0.03 Z: 0.96
X: 0.33 Y: -0.18        Z: 0.82
X: 0.14 Y: 0.36 Z: 0.85
X: 0.14 Y: 0.03 Z: 0.94
X: -0.01        Y: 0.34 Z: 0.81
```

Figure 2. This figure shows the example code for reading IMU accelerometer data and displaying it to the serial monitor

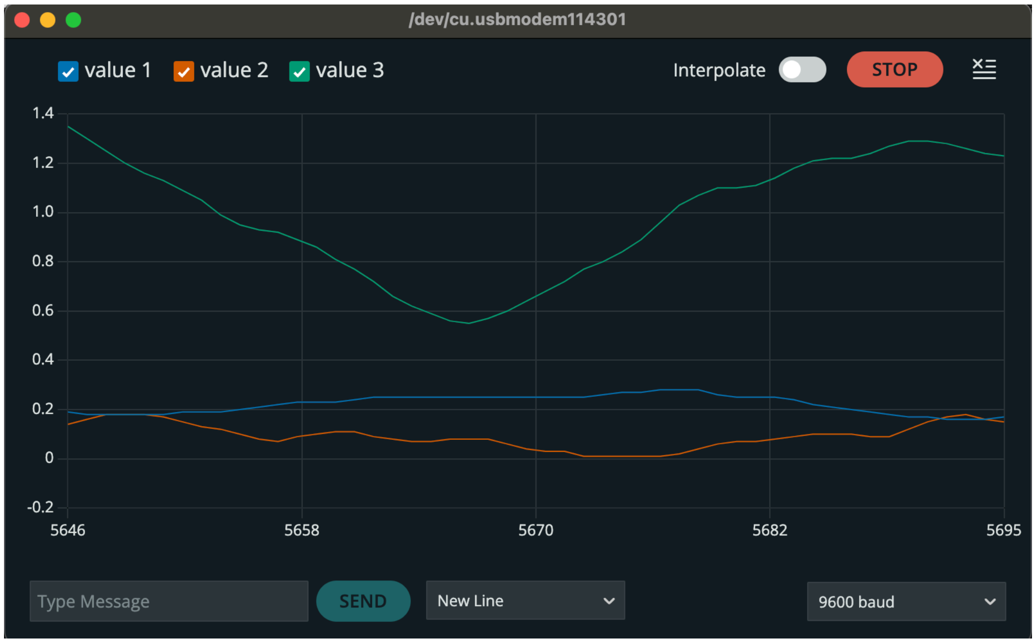
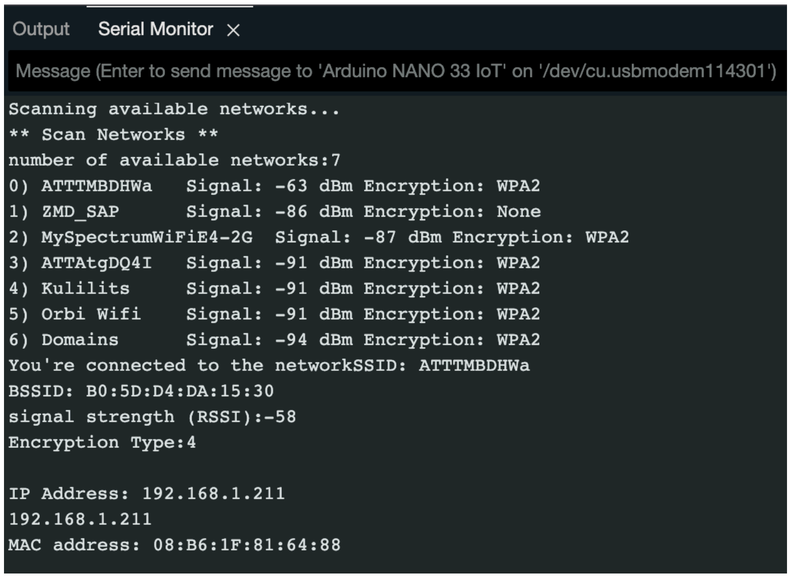


Figure 3. This is the Serial Plotter which is displaying the values of the acceleration

III. **Task 2:** Scanning and connecting to local WIFI networks

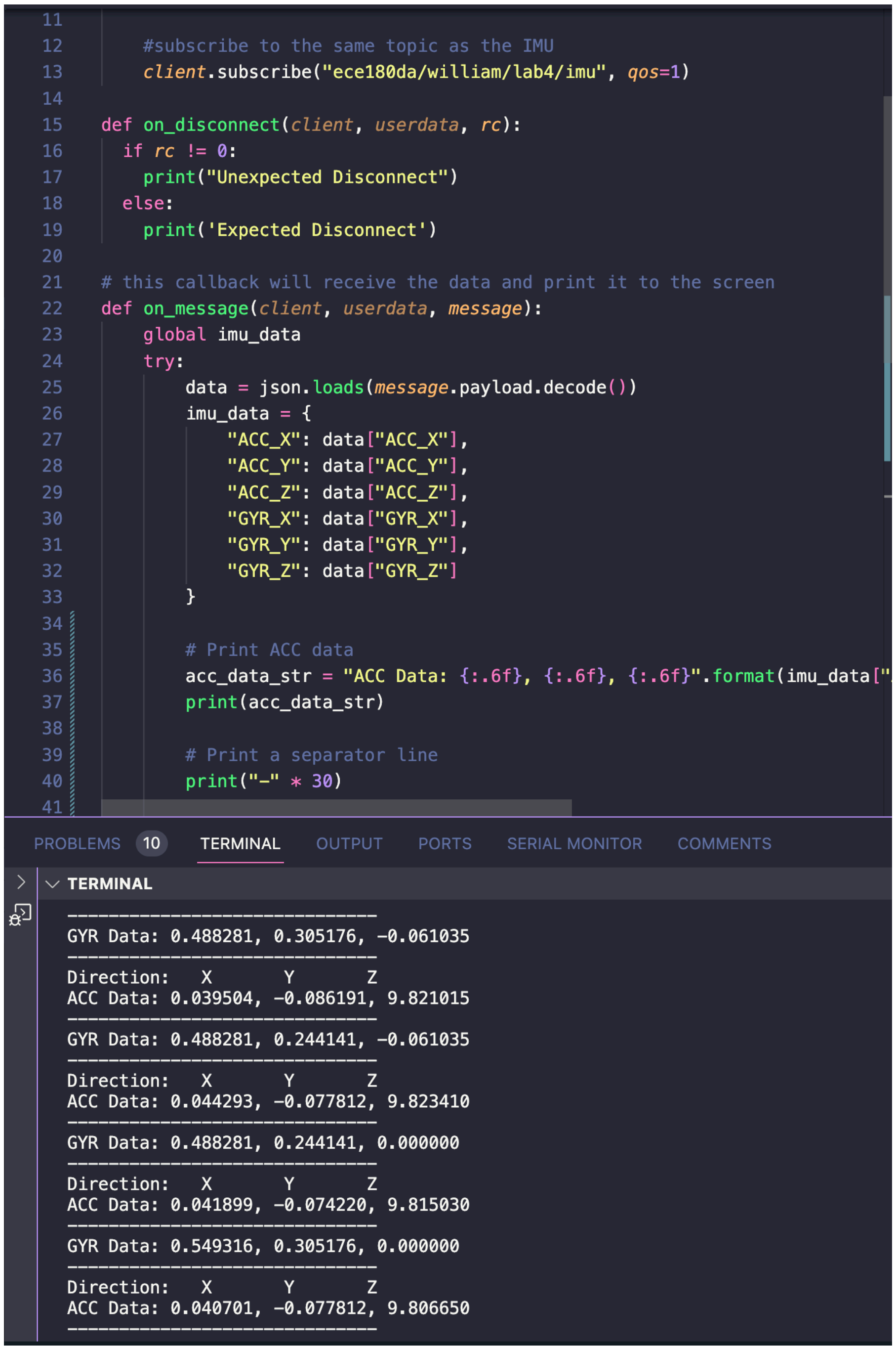

Output
Serial Monitor
Message (Enter to send message to 'Arduino NANO 33 IoT' on '/dev/cu.usbmodem114301')

```
Scanning available networks...
** Scan Networks **
number of available networks:7
0) ATTTMBDHWa   Signal: -63 dBm Encryption: WPA2
1) ZMD_SAP      Signal: -86 dBm Encryption: None
2) MySpectrumWiFiE4-2G  Signal: -87 dBm Encryption: WPA2
3) ATTAtgDQ4I   Signal: -91 dBm Encryption: WPA2
4) Kulilits     Signal: -91 dBm Encryption: WPA2
5) Orbi Wifi    Signal: -91 dBm Encryption: WPA2
6) Domains      Signal: -94 dBm Encryption: WPA2
You're connected to the networkSSID: ATTTMBDHWa
BSSID: B0:5D:D4:DA:15:30
signal strength (RSSI):-58
Encryption Type:4

IP Address: 192.168.1.211
192.168.1.211
MAC address: 08:B6:1F:81:64:88
```

Figure 4. Output after scanning and connecting to local network

IV. Task 2: MQTT code to communicate with Nano 33 & computer



```

    #subscribe to the same topic as the IMU
    client.subscribe("ece180da/william/lab4/imu", qos=1)

def on_disconnect(client, userdata, rc):
  if rc != 0:
    print("Unexpected Disconnect")
  else:
    print('Expected Disconnect')

# this callback will receive the data and print it to the screen
def on_message(client, userdata, message):
    global imu_data
    try:
        data = json.loads(message.payload.decode())
        imu_data = {
            "ACC_X": data["ACC_X"],
            "ACC_Y": data["ACC_Y"],
            "ACC_Z": data["ACC_Z"],
            "GYR_X": data["GYR_X"],
            "GYR_Y": data["GYR_Y"],
            "GYR_Z": data["GYR_Z"]
        }

        # Print ACC data
        acc_data_str = "ACC Data: {:.6f}, {:.6f}, {:.6f}".format(imu_data["
        print(acc_data_str)

        # Print a separator line
        print("-" * 30)

```

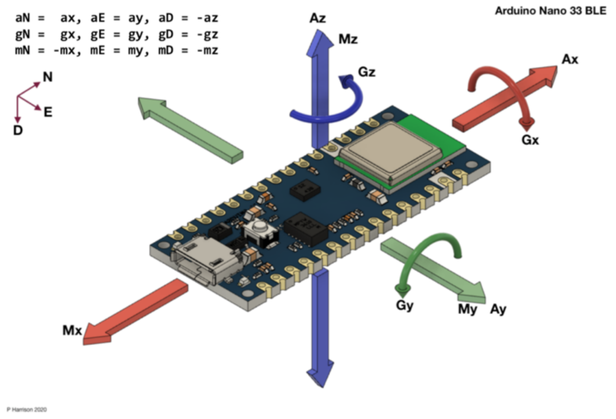
```
------------------------------
GYR Data: 0.488281, 0.305176, -0.061035
------------------------------
Direction:    X       Y       Z
ACC Data: 0.039504, -0.086191, 9.821015
------------------------------
GYR Data: 0.488281, 0.244141, -0.061035
------------------------------
Direction:    X       Y       Z
ACC Data: 0.044293, -0.077812, 9.823410
------------------------------
GYR Data: 0.488281, 0.244141, 0.000000
------------------------------
Direction:    X       Y       Z
ACC Data: 0.041899, -0.074220, 9.815030
------------------------------
GYR Data: 0.549316, 0.305176, 0.000000
------------------------------
Direction:    X       Y       Z
ACC Data: 0.040701, -0.077812, 9.806650
------------------------------
```

Figure 5. Showing the successful transmission of IMU data through MQTT (IMU was stationary)

I do notice a bit of lag when the values are changing rapidly. I think that the IMU is trying to send so much data that it can't be processed quickly enough. This could be adjusted by using data buffers.

What we could do is have a variable on the local device that holds a larger amount of data than what is being sent. This would act like a queue where we store in sensor values and have that data outputted as soon as we can. This means that we can work at a lower frequency but not lose data. The issue with this is that we would expect a delay in the data if fast actions are encountered. This delay would have to be considered in the processing of our actions and outcomes.

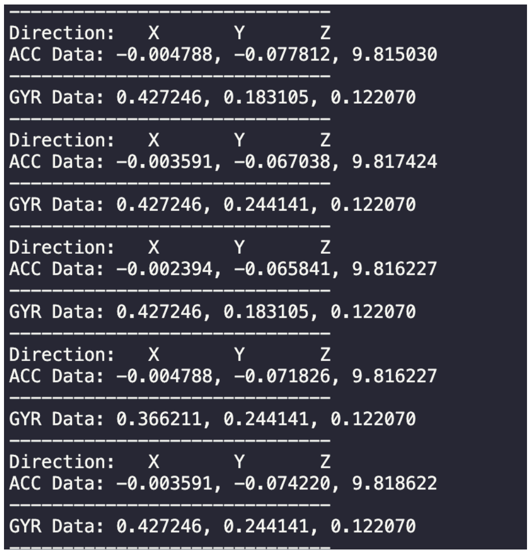
V. **Task 4:** IMU Classification



aN = ax, aE = ay, aD = -az
gN = gx, gE = gy, gD = -gz
mN = -mx, mE = my, mD = -mz
N
E
D
Az
Mz
Gz
Arduino Nano 33 BLE
Ax
Gx
Mx
Gy
My
Ay
P Harrison 2020

Figure 6. Directions for Nano 33 IOT MCU

- From testing, I found that the direction is correspond with the graphic that is shown above.
- The gravity acceleration that is seen is of *9.81m/s^2* in the z-direction when the board is lying flat like in the image above.
- From analyzing the data, when the IMU is idle there is not much fluctuation above 0.1 (tenth of a measurement) value in any direction for acceleration readings. Past this decimal point there is considerable fluctuations.
- Thus our accuracy looks be within a single decimal place, this can be seen in this chunk of data:



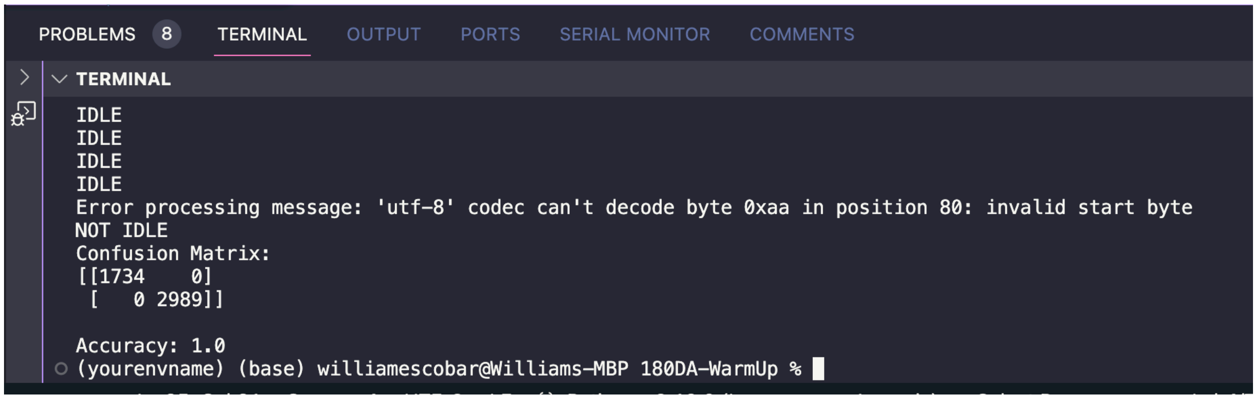
```
------------------------------
Direction:   X        Y        Z
ACC Data: -0.004788, -0.077812, 9.815030
------------------------------
GYR Data: 0.427246, 0.183105, 0.122070
------------------------------
Direction:   X        Y        Z
ACC Data: -0.003591, -0.067038, 9.817424
------------------------------
GYR Data: 0.427246, 0.244141, 0.122070
------------------------------
Direction:   X        Y        Z
ACC Data: -0.002394, -0.065841, 9.816227
------------------------------
GYR Data: 0.427246, 0.183105, 0.122070
------------------------------
Direction:   X        Y        Z
ACC Data: -0.004788, -0.071826, 9.816227
------------------------------
GYR Data: 0.366211, 0.244141, 0.122070
------------------------------
Direction:   X        Y        Z
ACC Data: -0.003591, -0.074220, 9.818622
------------------------------
GYR Data: 0.427246, 0.244141, 0.122070
------------------------------
```

Figure 7. Chuck of IMU idle data

- However, for the gyroscope, the values seem to fluctuate heavily at the single decimal place at idle
- Thus, the accuracy of our gyroscope will be within the whole number range (i.e. up to the ones place.)

VI. Task 4: Testing accuracy with confusion matrix

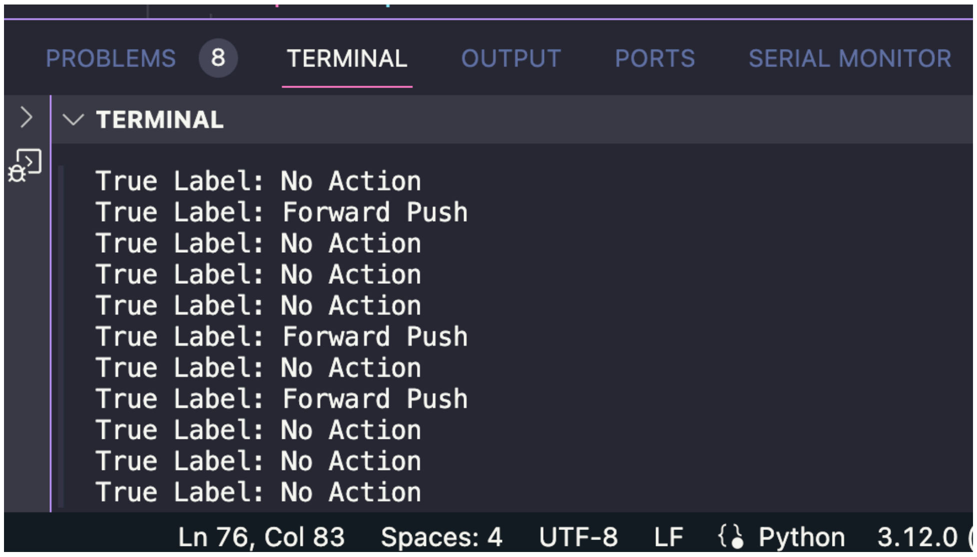
I used the “sklearn.metrics” header file which allowed me to easily create a confusion matrix to analyze the accuracy of the chosen thresholds. I chose a threshold of 0.1 to register if there was a deviation from the idle and non-idle states. From testing over a 60s period, it found that the accuracy is 1. Personally, I know that this value isn’t entirely accurate and is only so accurate because the threshold is so small. However, for our game application where we will likely have a larger threshold to register movements, we will need to tune this in the future.



```
IDLE
IDLE
IDLE
IDLE
Error processing message: 'utf-8' codec can't decode byte 0xaa in position 80: invalid start byte
NOT IDLE
Confusion Matrix:
[[1734    0]
 [   0 2989]]

Accuracy: 1.0
(yourenvname) (base) williamescobar@Williams-MBP 180DA-WarmUp %
```

Figure 8. Output after using confusion matrix while determining if the IMU was idle



```
True Label: No Action
True Label: Forward Push
True Label: No Action
True Label: No Action
True Label: No Action
True Label: Forward Push
True Label: No Action
True Label: Forward Push
True Label: No Action
True Label: No Action
True Label: No Action
```

Figure 9. Output after using thresholding to determine if an action was a push