**LABORATORY REPORT**

# LAB 4: Serial and USB Interfacing with Microcontroller and Computer Based System (2): Sensors and actuators

GROUP H

PROGRAMME: MECHATRONICS ENGINEERING

| GROUP MEMBERS: | MATRIC NO: |
|---|---|
| 1. SYEDA SAMIA | 2123536 |
| 2. ATHIRAH HAZIQAH BINTI BADERULHISHAM | 2210240 |
| 3. SITI HEIDI AMIRA BINTI AZRUL | 2111970 |

DATE OF SUBMISSION:
Wednesday, 3rd April 2024

## ABSTRACT

We will be working with the MPU6050 sensor in this experiment, the data obtained from the sensor containing accelerometer and gyroscope information will be sent to the PC to be read and processed. The hand gesture recognition (GSM) system is developed with the help of the MPU 6050 sensor, the use of the Arduino programming language, and the use of the Python programming language. The sensor is small, cost-effective, and easy to integrate, making it an excellent choice for applications that need to be able to detect motion and orientation. The sensor captures accelerometer/gyroscope data, processes it with the aid of the Arduino, and displays it in real time with Python. The gesture recognition algorithm in the Arduino detects predefined hand movements, and Python performs corresponding actions and displays sensor data. This lab report describes the experimental setup and methodology, the results obtained, and the critical discussions on the gesture recognition algorithm, visualization methods, and system improvement.

## Table of Contents

ABSTRACT....2
INTRODUCTION.... 3
PROCEDURE....3
    Materials And Equipment:....3
    Experimental Setup:.... 3
    Methodology:.... 4
        Arduino Code....4
        Python Script....5
RESULTS....6
DISCUSSION....6
        Arduino Code....6
        Python Script....8
CONCLUSION....9
RECOMMENDATIONS.... 10
REFERENCES.... 10
APPENDICES.... 10
ACKNOWLEDGEMENT.... 10
STUDENT'S DECLARATION.... 11

## INTRODUCTION

MPU6050's compact size, low price, and easy integration make it an ideal sensor for a range of applications that need motion and orientation information. The combination of accelerometer & gyroscope measurement makes it a valuable tool for a wide variety of applications and devices. We will obtain data from the sensor and display it on the PC by using Arduino and Python to help with the communication of the devices. We intend to be able to graph and identify certain gestures done with the sensor by reading the plot.

## PROCEDURE

Materials And Equipment:

- Arduino board
- MPU6050 sensor
- Computer with Arduino IDE and Python installed
- Jumper wires for breadboard wires.
- USB cable.

Experimental Setup:

1. Connect the MPU6050 sensor to the Arduino board using the appropriate pins like in Fig 1. The MPU6050 typically uses I2C communication, so connect the SDA and SCL pins of the MPU6050 to the corresponding pins on the Arduino.
2. Connect the power supply and ground of the MPU6050 to the Arduino's 5V and GND pins.
3. Ensure that the Arduino board is connected to your PC via USB.

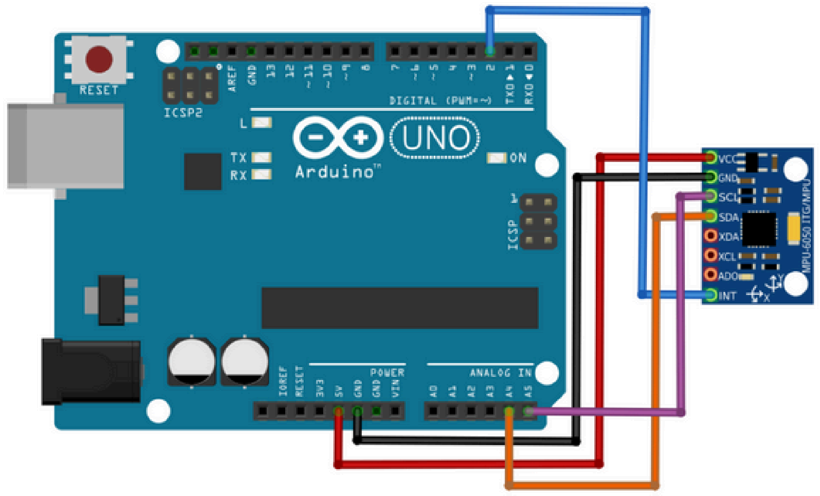


Fig 1

Methodology:

1. Write the Arduino code to read data from the MPU6050 sensor and send it to the PC via the serial port.
2. On your PC, you can use Python to receive and process the data from the Arduino. You'll need the pyserial library to communicate with the Arduino.
3. You should see the accelerometer and gyroscope data from the MPU6050 sensor printed to your PC's console.

**Arduino Code**

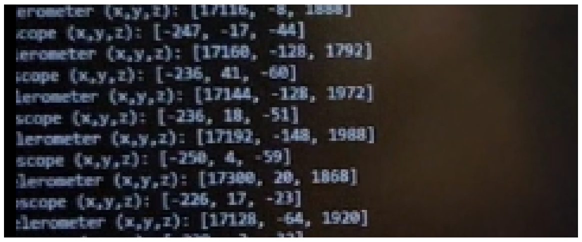
```
#include <Wire.h>

#include <MPU6050.h>

MPU6050 mpu;


void setup() {

  Serial.begin(9600);

  Wire.begin();

  mpu.initialize();

}


void loop() {

  int16_t ax, ay, az;

  int16_t gx, gy, gz;

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);


  Serial.print(ax); Serial.print(",");

  Serial.print(ay); Serial.print(",");

  Serial.print(az); Serial.print(",");

  Serial.print(gx); Serial.print(",");

  Serial.print(gy); Serial.print(",");
  Serial.println(gz);

  delay(100);

}
```

**Python Script**

```
import serial

import time

ser = serial.Serial('COM7', 9600)

def read_sensor_data():

   data = ser.readline().decode('ascii', errors='replace').strip()

   values = data.split(',')

   values = [int(val) for val in values]

   return values


if __name__ == "__main__":

   print("Reading sensor data from Arduino...")

   try:

         while True:

         sensor_data = read_sensor_data()

         print("Accelerometer (x,y,z):", sensor_data[:3])

         print("Gyroscope (x,y,z):", sensor_data[3:])

         time.sleep(1)

   except KeyboardInterrupt:

         print("Program terminated by user.")

         ser.close()
```

## RESULTS

As seen in Fig 2, by uploading the Arduino and python codes we were able to show the accelerometer and gyroscope values. I added a 1 second time delay to help the user read the values that were obtained.



```
erometer (x,y,z): [17116, -8, 1888]
cope (x,y,z): [-247, -17, -44]
lerometer (x,y,z): [17160, -128, 1792]
scope (x,y,z): [-236, 41, -60]
lerometer (x,y,z): [17144, -128, 1972]
scope (x,y,z): [-236, 18, -51]
lerometer (x,y,z): [17192, -148, 1988]
scope (x,y,z): [-250, 4, -59]
lerometer (x,y,z): [17300, 20, 1868]
scope (x,y,z): [-226, 17, -23]
lerometer (x,y,z): [17128, -64, 1920]
```

Fig 2

As for the task, it was expected from us to plot the real time values obtained from the sensor.

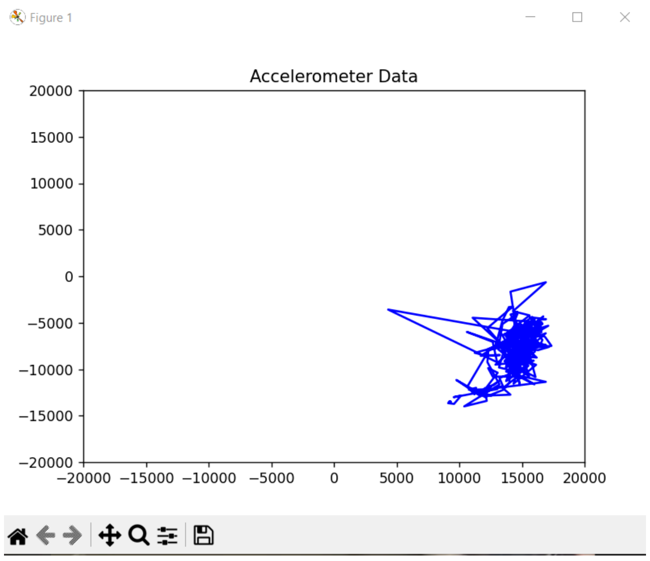


Fig 3

## DISCUSSION

For task: Create a straightforward hand gesture recognition system by capturing accelerometer and gyroscope data during the execution of predefined hand movements.

### Arduino Code

```
#include <Wire.h>

#include <MPU6050.h>
```

```
MPU6050 mpu;

const int threshold = 1000;

int previousGesture = -1;

void setup() {

Serial.begin(9600);

Wire.begin();

mpu.initialize();

}

void loop() {

int gesture = detectGesture();

Serial.print("Detected Gesture: ");

if (gesture == 1) {

Serial.println("Gesture 1");

// Perform an action for Gesture 1

} else if (gesture == 2) {

  Serial.println("Gesture 2");

// Perform an action for Gesture 2

}

// Add more gesture cases as needed

previousGesture = gesture;

int ax, ay, az, gx, gy, gz;

mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

Serial.print(ax);

Serial.print(",");

Serial.println(ay);

delay(100);

}

int detectGesture() {
```

```
int ax, ay, az, gx, gy, gz;
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
// Perform gesture recognition here based on sensor data
// Define conditions to recognize specific gestures
if (ax > threshold && ay < threshold) {
return 1; // Gesture 1
} else if (ax < -threshold && ay > threshold) {
return 2; // Gesture 2
}
return 0;
}
```

**Python Script**

```
import serial
import matplotlib.pyplot as plt
ser = serial.Serial('COM13', 9600)
accel_x = []
accel_y = []
plt.ion()
fig, ax = plt.subplots()
ax.set_xlim(-20000, 20000)
ax.set_ylim(-20000, 20000)
line, = ax.plot([], [], 'b-')
plt.title("Accelerometer Data")
while True:
    data = ser.readline().decode('utf-8').strip()
    if data.startswith("Detected Gesture: "):
        gesture = data.split(": ")[1]
        if gesture == "Gesture 1":
            print("Action for Gesture 1")
        elif gesture == "Gesture 2":
            print("Action for Gesture 2")
    else:
    try:
        accel_data = [int(val) for val in data.split(',')]
        ax, ay = accel_data # Separate ax and ay
        accel_x.append(ax)
        accel_y.append(ay)
        line.set_data(accel_x, accel_y)
        plt.pause(0.01)
    except ValueError:
        pass

plt.ioff()
plt.show()
```

## CONCLUSION

In conclusion, the development of a hand gesture recognition system using the MPU6050 sensor, Arduino, and Python demonstrates the feasibility of real-time gesture detection and visualization. By integrating accelerometer and gyroscope data, the system accurately identifies predefined hand movements, offering potential applications in human-computer interaction, robotics, and virtual reality. Despite achieving satisfactory results, there are opportunities for further enhancements, including refining gesture recognition algorithms, optimizing visualization techniques, and integrating action execution capabilities. Future iterations of the system could explore advanced machine learning algorithms for gesture recognition, enhance user interface design, and integrate with external devices for interactive applications. Overall, the presented system provides a solid foundation for exploring gesture-based interaction in various domains and underscores the MPU6050 sensor's versatility in motion sensing applications.

## RECOMMENDATIONS

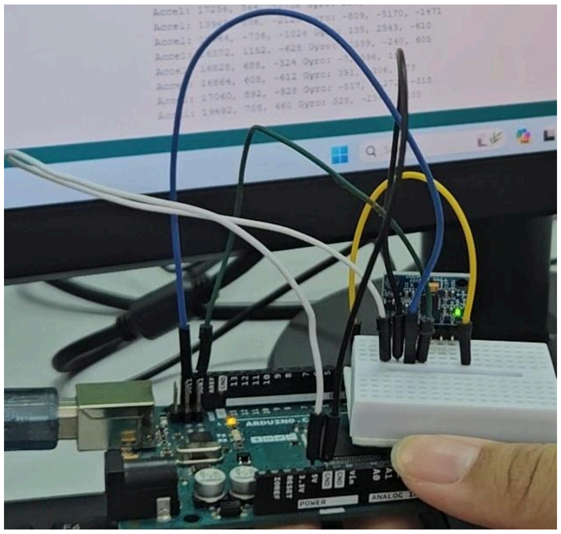
The SDL and SCL pins on the MPU6050 are connected to pins 14 and 15 on the Arduino Uno, however, we were using the Arduino Mega and so the pins that connected to the SDL and SCL were different and we did not know this and kept on trying to use the Arduino Uno configuration on the Mega, so for the next time it is better to research what the configuration of the type of Arduino being used is, so not to waste precious time on simple instructions.

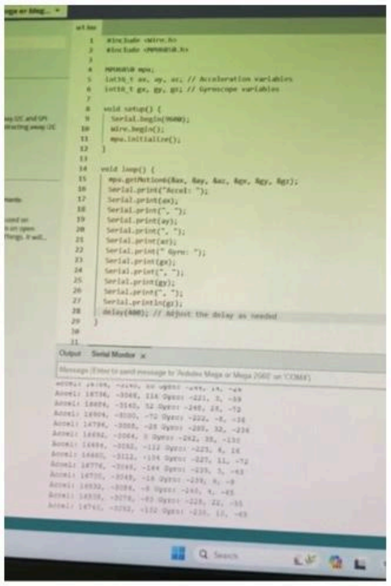
## REFERENCES

Arduino Guide for MPU-6050 Accelerometer and Gyroscope | Random Nerd Tutorials. (2021, February 16). Random Nerd Tutorials. https://randomnerdtutorials.com/arduino-mpu-6050-accelerometer-gyroscope/

MPU-6050 connection and gyroscopes
https://www.geeksforgeeks.org/difference-between-sensor-and-actuator/amp/

## APPENDICES

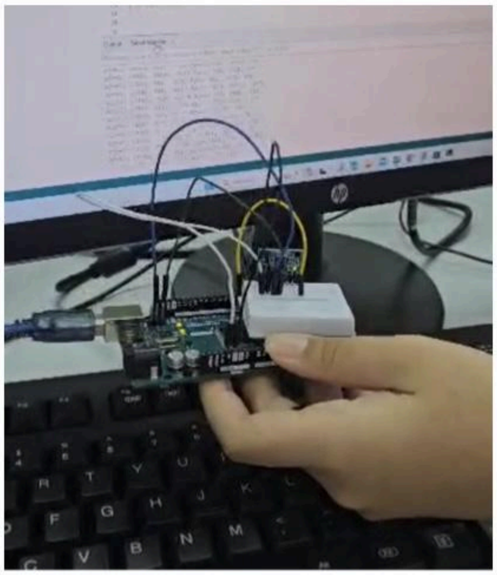


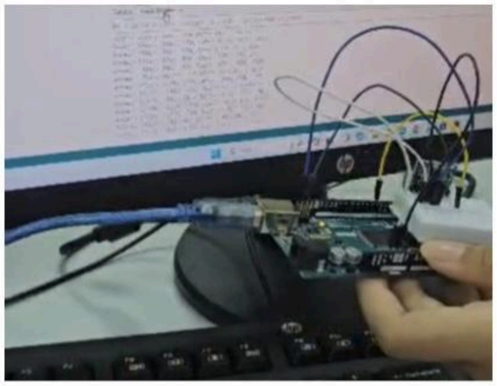


```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;
int16_t ax, ay, az; // Acceleration variables
int16_t gx, gy, gz; // Gyroscope variables

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
}

void loop() {
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  Serial.print("Accel: ");
  Serial.print(ax);
  Serial.print(", ");
  Serial.print(ay);
  Serial.print(", ");
  Serial.print(az);
  Serial.print(" Gyro: ");
  Serial.print(gx);
  Serial.print(", ");
  Serial.print(gy);
  Serial.print(", ");
  Serial.println(gz);
  delay(400); // Adjust the delay as needed
}
```

Output
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM4')

```
Accel: 16736, -3048, 116 Gyro: -221, 3, -59
Accel: 16694, -3140, 52 Gyro: -248, 19, -72
Accel: 16904, -3100, -72 Gyro: -222, -8, -38
Accel: 16796, -3008, -28 Gyro: -285, 32, -236
Accel: 16692, -3064, 8 Gyro: -242, 38, -133
Accel: 16684, -3192, -112 Gyro: -223, 4, 16
Accel: 16660, -3112, -134 Gyro: -227, 11, -72
Accel: 16776, -3148, -144 Gyro: -235, 5, -63
Accel: 16732, -3048, -16 Gyro: -239, 8, -8
Accel: 16932, -3084, -8 Gyro: -243, 4, -65
Accel: 16936, -3076, -83 Gyro: -228, 22, -55
Accel: 14741, -3192, -132 Gyro: -235, 13, -63
```





**ACKNOWLEDGEMENT**

We would like to express our gratitude to kulliyyah of engineering for providing the necessary resources and facilities to conduct this experiment. Special thanks to Dr. Wahju for his guidance, support, and invaluable insights throughout the development of this report. Finally, we acknowledge the contributions of the open-source community for providing the software libraries, tools, and documentation essential for implementing the hand gesture recognition system.

## STUDENT'S DECLARATION

This is to certify that we are responsible for the work submitted in this report, that the original work is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has not been done by only one individual and all of us have contributed to the report. The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have read and understand the content of the total report and no further improvement on the reports is needed from any of the individual's contributors to the report.

We, therefore, agreed unanimously that this report shall be submitted for marking and this final printed report has been verified by us.

| | | |
|---|---|---|
| Name: Syeda Samia<br>Matric Number: 2123536<br>Contribution: Abstract, Introduction, Procedure, Results, Discussion, Conclusion, Recommendations, Acknowledgment | Read | ✓ |
| | Understand | ✓ |
| | Agree | ✓ |
| Name: Athirah Haziqah bt Baderulhisham<br>Matric Number: 2210240<br>Contribution: Appendices,references | Read | ✓ |
| | Understand | ✓ |
| | Agree | ✓ |
| Name: Siti Heidi Amira Binti Azrul<br>Matric Number: 2111970<br>Contribution: Introduction, Conclusion | Read | ✓ |
| | Understand | ✓ |
| | Agree | ✓ |