LAB REPORT 4a :

SERIAL AND USB INTERFACING WITH MICROCONTROLLER AND COMPUTER BASED SYSTEM (2): SENSORS AND ACTUATORS)

GROUP 5

MCTA 3203

SEMESTER 1 2024/2025

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION: 6 NOVEMBER 2024

NO	NAME	MATRIC NUMBER
1.	MUHAMMAD AFIQ ADHAM BIN MOHD NADZRI	2227531
2.	MUHAMMAD AMIN BIN MOHAMAD RIZAL	2217535
3.	MUHAMMAD AFIQ BIN MOHD ASRI	2212541
4.	MUHAMMAD AKMAL BIN MOHD FAUZI	2214077
5.	MUHAMMAD AFIQ IKHWAN BIN NOR SHAHRIZAL	2215897

TABLE OF CONTENTS

INTRODUCTION

In the mechatronics field, sensors and microcontrollers are used to create systems that can communicate with their surroundings. The MPU6050 sensor, which is the subject of this manual, is suitable for a variety of applications requiring motion and orientation data because of its small size, affordability, and ease of interface. In summary, hardware setup and coding are the first major steps in connecting an Arduino board to a personal computer with an MPU 6050. It is possible to efficiently read and process data from an MPU 6050 for innovative projects like gesture recognition systems using both Arduino and Python.

ABSTRACT

This experiment examines how the MPU6050 sensor can be integrated with microcontrollers, with a focus on its use in capturing motion and orientation data.. The MPU6050 is a great option for a variety of mechatronics projects because of its small size, low cost, and simplicity of interface. The study describes how to use an Arduino board to connect a personal computer to the MPU6050, including the hardware configuration and coding steps required. The system efficiently reads and processes data from the MPU6050 by combining Python and Arduino, allowing it to be used in creative projects like gesture recognition systems.

MATERIALS AND EQUIPMENT

- Arduino Board
- MPU6050 Sensor
- Jumper Wires
- USB Cable for Arduino
- LED
- Computer with Arduino IDE and Python installed
- Power Source

EXPERIMENTAL SETUP

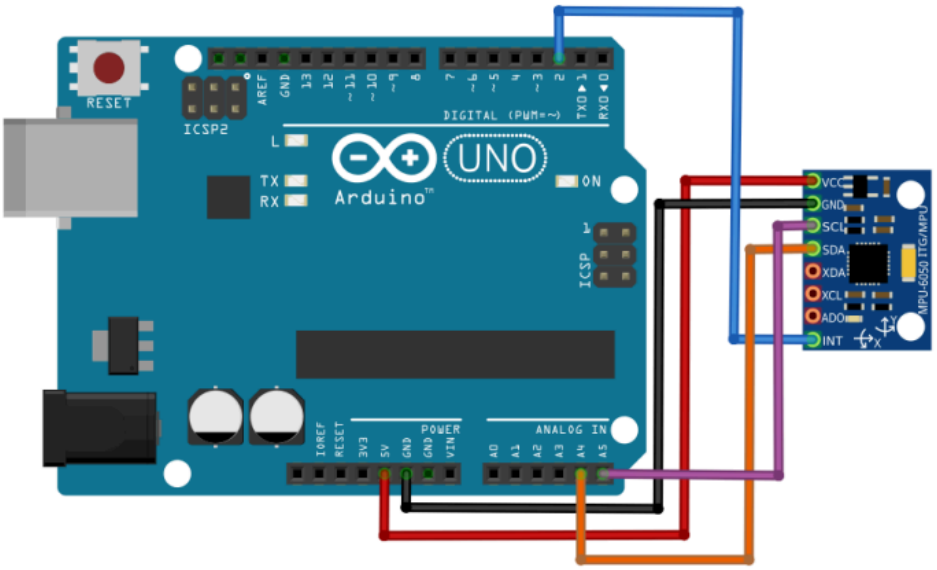


Fig. 1: Arduino-MPU6050 Connections

1. Connect the MPU6050 sensor to the Arduino board.

2. Connect the SDA and SCL pins of the MPU6050 to the corresponding pins on the Arduino (usually A4 and A5 for most Arduino boards).

3. Connect the power supply and ground of the MPU6050 to the Arduino's 5V and GND pins.

4. Connect the Arduino Board to the computer using a USB cable.

METHODOLOGY

1. Hardware Setup

- The MPU6050 sensor was connected to an Arduino board via the I2C interface, using the SDA and SCL pins (commonly pin 20 and 21 on the Arduino).

- Power and ground were also connected, with the MPU6050 receiving 5V and GND from the Arduino.

-The Arduino itself was connected to a PC via a USB cable, which allowed for both code uploading and serial communication.

2. Software Programming and Execution

- **Arduino Code Development:**

  -Using the Arduino IDE, code was written to initialize the MPU6050, read data from the accelerometer and gyroscope, and send this data to the PC over serial communication at a baud rate of 9600.

  - An additional version of the code included gesture detection by setting threshold values for the accelerometer data, enabling the system to identify specific hand movements (gestures).
- **Running the Python Script:**

  -To run the data collection script on the PC, the necessary libraries (like `pyserial`) were installed to read and display serial data from the Arduino.

  - This data, including accelerometer and gyroscope values, was continuously printed to the console, allowing for real-time observation.

-For gesture recognition, the Python script was modified to interpret specific gestures based on patterns in the incoming data and to display actions associated with each recognized gesture.

- **Gesture Recognition Task:**

-In the Arduino code, threshold values were set to classify different hand movements into gestures.

- These gestures were then sent to the Python script, which processed the data and provided feedback or triggered actions in response to each detected gesture.

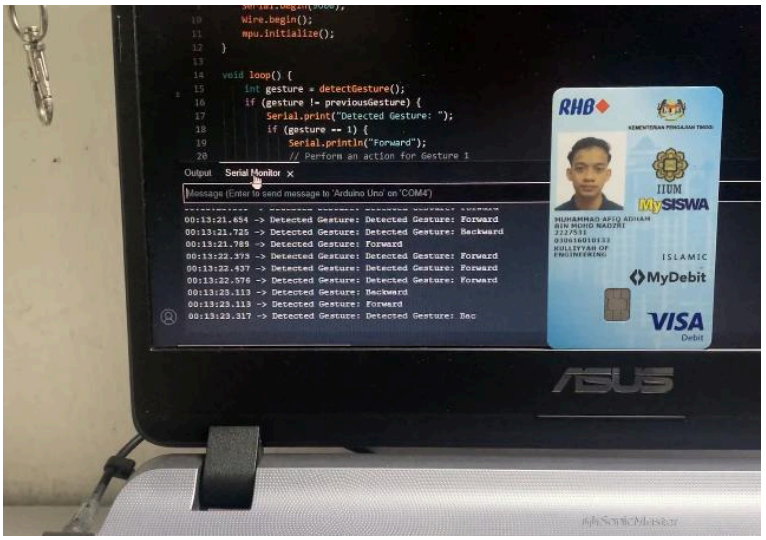
- **Execution and Visualization:**

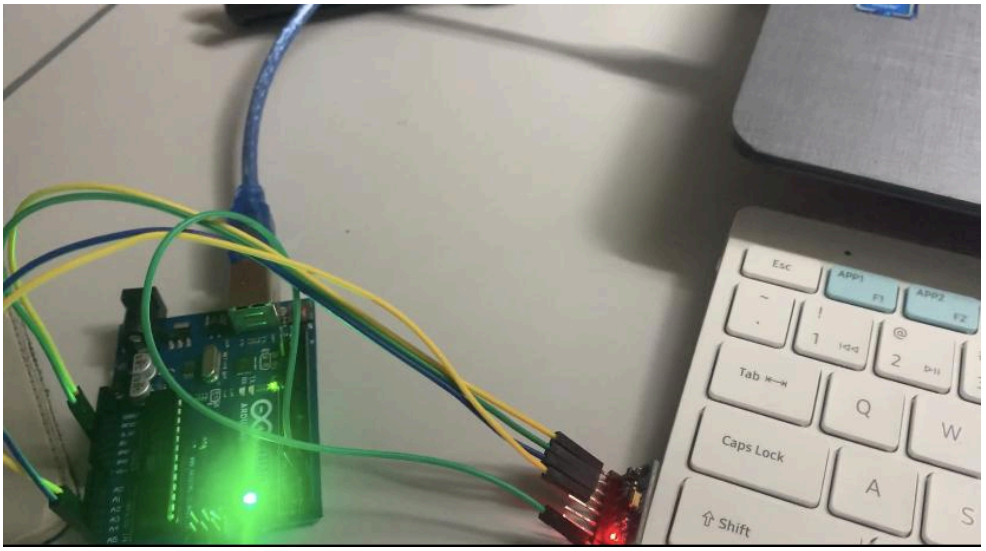
-The system was tested with predefined hand movements to ensure gesture recognition accuracy.

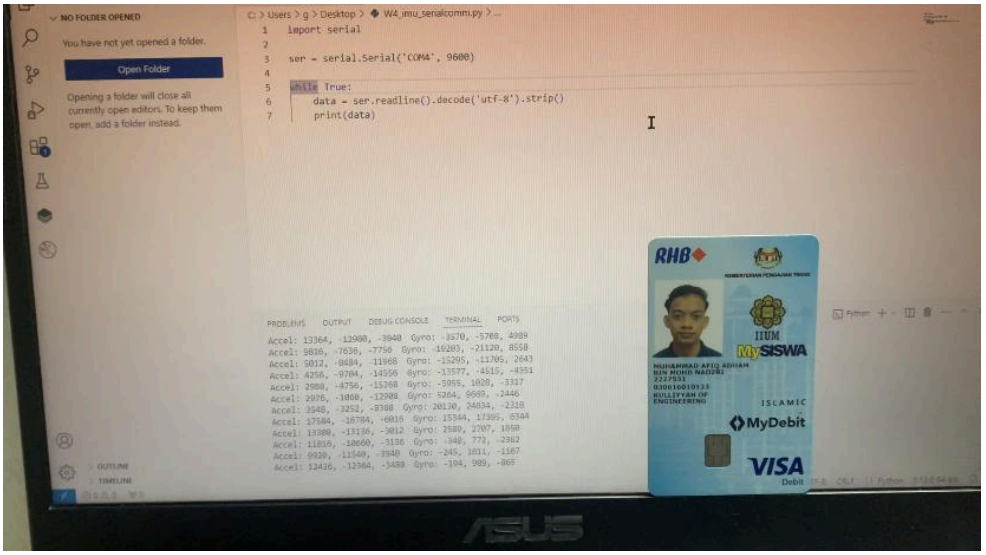
- The collected accelerometer and gyroscope data were processed and displayed on the PC console.

- As a suggested improvement, a visualisation feature could be added to track gesture movement paths on an x-y coordinate system, providing a clearer view of hand movement patterns.

RESULT


Wire.begin();
mpu.initialize();
}
void loop() {
int gesture = detectGesture();
if (gesture != previousGesture) {
Serial.print("Detected Gesture: ");
if (gesture == 1) {
Serial.println("Forward");
// Perform an action for Gesture 1
Output
Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM4')
00:13:21.654 -> Detected Gesture: Detected Gesture: Forward
00:13:21.725 -> Detected Gesture: Detected Gesture: Backward
00:13:21.789 -> Detected Gesture: Forward
00:13:22.373 -> Detected Gesture: Detected Gesture: Forward
00:13:22.437 -> Detected Gesture: Detected Gesture: Forward
00:13:22.576 -> Detected Gesture: Detected Gesture: Forward
00:13:23.113 -> Detected Gesture: Backward
00:13:23.113 -> Detected Gesture: Forward
00:13:23.317 -> Detected Gesture: Detected Gesture: Bac
RHB
IIUM
MySISWA
ISLAMIC
MyDebit
VISA
Debit
ASUS


Esc
Tab
Caps Lock
Shift
Q
W
A
S


NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
C: > Users > g > Desktop > W4_imu_serialcomm.py > ...
import serial
ser = serial.Serial('COM4', 9600)
while True:
data = ser.readline().decode('utf-8').strip()
print(data)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
RHB
IIUM
MySISWA
ISLAMIC
MyDebit
VISA
Debit
ASUS

Video link:
https://github.com/GROUP5-MSI/WEEK-4_Serial-Communication/blob/main/Video%20W4%20partA%20arduino.mov

https://github.com/GROUP5-MSI/WEEK-4_Serial-Communication/blob/main/Video%20W4%20partA%20python.mov

https://github.com/GROUP5-MSI/WEEK-4_Serial-Communication/blob/main/Video%20W4%20partA%20task%20arduino.mov

CONCLUSION

In the final analysis, the experiment has successfully shown the application of the MPU6050 IMU sensor in hand gesture recognition. The interfacing of the MPU6050 with the Arduino enabled us to capture real-time data from the accelerometer and gyroscope for the classification of specific hand gestures. This small-sized, easy-to-integrate MPU6050 is quite fitting for such applications, allowing effective tracking of motion and orientation. Through the processing and analysis of the sensor data, the algorithm was able to recognize the gestures accurately according to the predefinition. Also, the visualization of hand paths in the x-y coordinate system represents each movement, proving the capability of the sensor in motion-tracking applications. In this paper, this experiment proves that MPU6050 has the potential for developing gesture-based systems and goes further to show its viability in real-world applications extending into human-computer interaction and wearable technology.

RECOMMENDATION

Firstly, the student should know the procedure used for sensor calibration, the methods adopted to acquire data and give an analysis of the accelerometer and gyroscope readings. To provide a clearer view of hand movement, enhance the x-y coordinate visualization with labeled axes, grids, or real-time updates to visualize motion paths dynamically. This can help in better interpreting the gestures and movement patterns. To enhance the smoothness and effectiveness of the experiments, students should discuss the interfacing experiment of the MPU6050 IMU, covering all the details necessary for serial communication between the PC and Arduino.

ACKNOWLEDGEMENTS

A special thanks goes out to Dr. Wahju Sediono and Dr. Zulkifli Bin Zainal Abidin, my teaching assistant, and my peers for their invaluable help and support in finishing this report. Their advice, feedback, and experience have greatly influenced the level of quality and understanding of this work. Their time, patience, and commitment to supporting my academic success are greatly appreciated.

STUDENT'S DECLARATION

Certificate of Originality and Authenticity

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 acknowledgment, and that the original work contained herein has 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 that no further improvement on the reports is needed from any of the individual 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**.

Signature:
Name: MUHAMMAD AFIQ ADHAM BIN MOHD NADZRI
Matric Number: 2227531

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MUHAMMAD AMIN BIN MOHAMAD RIZAL
Matric Number: 2217535

Read ☑
Understand ☑
Agree ☑

Signature: afiq
Name: MUHAMMAD AFIQ BIN MOHD ASRI
Matric Number: 2212541

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MUHAMMAD AKMAL BIN MOHD FAUZI
Matric Number: 2214077

Read ☑
Understand ☑
Agree ☑

Signature: Afiq
Name: MUHAMMAD AFIQ IKHWAN BIN NOR SHAHRIZAL
Matric Number: 2215897

Read ☑
Understand ☑
Agree ☑