INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

Garden of Knowledge and Virtue

MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2 2024/2025

WEEK 4a: SERIAL COMMUNICATION IMU

SECTION 1

GROUP 3

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO

NO.	GROUP MEMBERS	MATRIC NO.
1.	AHMAD DARWISH BIN AHMAD TERMIZI	2212089
2.	MUHAMMAD AQIF BIN ROSZAHAN	2210345
3.	AKMAL ZARIF BIN NAJIB	2211971
4.	SIDDIQUI HAMMAD AHMED	2221201

Date of Experiment: Monday, 24 March 2025

Date of Submission: Monday, 7 April 2025

ABSTRACT

This experiment focuses on integrating the MPU6050 sensor with an Arduino for hand gesture recognition. The sensor's accelerometer and gyroscope data are sent to a PC via serial communication, where a Python script processes the data to detect and classify hand movements. The system also visualizes the hand movement paths in an x-y coordinate system. This setup demonstrates the use of sensors, microcontrollers, and gesture recognition for simple human-computer interaction applications.

TABLE OF CONTENTS

INTRODUCTION	**4**
MATERIALS AND EQUIPMENTS	**5**
EXPERIMENTAL SETUP	**5**
METHODOLOGY	**6**
DATA COLLECTION	**8**
DATA ANALYSIS	**8**
RESULT	**8**
DISCUSSION	**9**
CONCLUSION	**9**
RECOMMENDATIONS	**9**
REFERENCES	**10**
ACKNOWLEDGEMENTS	**11**

INTRODUCTION

In this experiment, we connect the MPU6050 sensor to an Arduino to capture motion and orientation data for hand gesture recognition. Using I2C communication, the Arduino sends sensor data to a PC, where a Python script identifies specific gestures. The project also includes visualizing the gestures' movement paths. This exercise provides hands-on experience with sensor integration, serial communication, and gesture recognition, laying the foundation for more complex mechatronic systems.

MATERIALS AND EQUIPMENTS

1. Arduino Mega 2560
2. MPU6050 Sensor
3. Servo Motor
4. Jumpers
5. Breadboard

EXPERIMENTAL SETUP

1. As the MPU6050 communicates via I2C, its SDA and SCL pins should be connected to the corresponding Arduino pins, usually A4 and A5 on most boards.
2. Connect the VCC and GND pins of the MPU6050 to the Arduino's 5V and GND pins to provide power.
3. Finally, use a USB cable to link the Arduino to your computer for programming and data transmission.

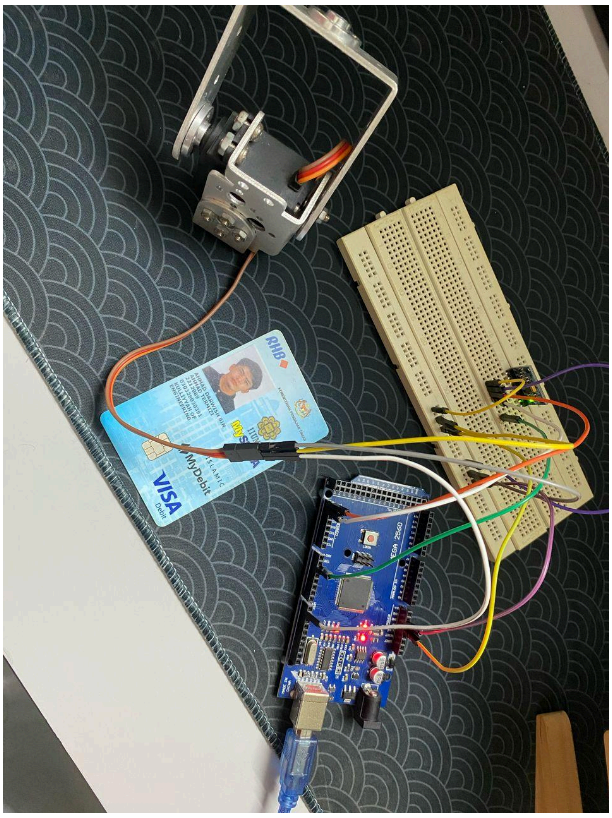

Figure 1: Hardware Setup

METHODOLOGY

1. Setup the Arduino Mega 2560
2. Implement the codes
3. Servo angle moves when pitch changes
4. Code snippet

Arduino

```
#include <Wire.h>
#include <MPU6050.h>
#include <Servo.h>
Servo servo;
MPU6050 mpu;

const int threshold = 122;
int previousGesture = -1;

void setup() {
Serial.begin(9600);
servo.attach(9);
pinMode(9, OUTPUT);
Wire.begin();
mpu.initialize();
}
void loop() {

int gesture = detectGesture();

    if (gesture != previousGesture) {

      if (gesture == 1) {
        Serial.println("Detected Gesture: Gesture 1");
        servo.write(45);
      // Perform an action for Gesture 1
      } else if (gesture == 2) {
        Serial.println("Detected Gesture: Gesture 2");
        servo.write(115);
      // Perform an action for Gesture 2
      }
    }
// Add more gesture cases as needed
previousGesture = gesture;
```

```
}

int detectGesture() {
      int16_t ax, ay, az;
      int16_t gx, gy, gz;
      int pitch;
      mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
      pitch = map(ay, -17000, 17000, 0, 255);
      // Perform gesture recognition here based on sensor data
      // Define conditions to recognize specific gestures
      if (pitch >= threshold) {
      return 1; // Gesture 1
      } else if (pitch < threshold) {
      return 2; // Gesture 2
      }
      // Add more gesture conditions as needed
      return 0; // No gesture detected
}
```

Python

```
import serial
ser = serial.Serial('COM6', 9600)


while True:
    try:
        data = ser.readline().decode('utf-8').strip()
        print(f"{data}")
        u[]
        if data == "Detected Gesture: Gesture 1":
            # Perform an action for Gesture 1
                print("'Servo Left'")
        elif data == "Detected Gesture: Gesture 2":
            # Perform an action for Gesture 2
                print("'Servo Right'")
            # Add more gesture actions as needed
    except Exception as e:
        print(f"Error reading data: {e}")
```

DATA COLLECTION

Pitch	Servo Angle (°)
0	45
122	45
123	115
255	115

Table 1: Pitch and Servo Angle

DATA ANALYSIS

The data in Table 1 shows a clear relationship between pitch values and corresponding servo angles, indicating a control system that responds to changes in orientation. When the pitch ranges from 0 to 122, the servo consistently holds at 45°. However, once the pitch reaches 123 or higher, the servo angle abruptly shifts to 115°, maintaining that value up to a pitch of 255. This reveals a threshold-based control approach where the servo only changes position once a specific pitch value is crossed. Such behavior is useful in applications that require automatic adjustments based on tilt—like stabilizing platforms or systems that react to certain angles. The simple two-stage response also suggests the system is designed for efficient and straightforward orientation-based control.

RESULT

The experiment successfully showcased basic hand gesture recognition using the MPU6050 sensor connected to an Arduino. By tracking accelerometer and gyroscope readings, specific hand motions were identified and matched to predefined gesture categories. The sensor data was sent to a computer via serial communication and processed in real time with a Python script, which detected gestures based on threshold values defined in the Arduino code. Recognized gestures were printed in the console, and their movements were plotted on an x-y coordinate system, clearly highlighting the differences between each gesture. These results demonstrate the MPU6050's ability to accurately detect and distinguish hand gestures, offering a solid base for gesture-controlled systems.

DISCUSSION

This experiment successfully demonstrated a basic hand gesture recognition system using the MPU6050 sensor and Arduino. By reading accelerometer and gyroscope data through I2C communication, the Arduino detected motion patterns and transmitted them to a PC via serial communication. A Python script processed this data in real time and identified gestures based on simple threshold conditions set in the Arduino code. The system reliably recognized predefined gestures and displayed them in the console.

The visualization of movement in an x-y coordinate system further confirmed accurate detection and differentiation of gestures. While the system used a simple threshold-based method, it effectively showcased the potential of the MPU6050 for motion-based control. This setup offers a solid starting point for more advanced applications involving gesture recognition or orientation-based automation.

CONCLUSION

The experiment successfully demonstrated a simple yet effective hand gesture recognition system using the MPU6050 sensor and Arduino. It highlighted the sensor's ability to detect motion and orientation, and showed how real-time data processing can be used for gesture-based control. This setup provides a strong foundation for developing more advanced human-computer interaction systems.

RECOMMENDATIONS

To enhance the accuracy and flexibility of the hand gesture recognition system, it is recommended to implement machine learning algorithms for gesture classification. While threshold-based methods are suitable for basic gestures, they may not scale well for more complex or subtle motion patterns. Additionally, incorporating sensor fusion techniques with complementary sensors can enhance orientation estimation and reduce noise. Future work could also explore wireless data transmission to increase system portability and real-world applicability in wearable devices or smart control systems.

REFERENCES

Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). *Estimation of IMU and MARG orientation using a gradient descent algorithm*. IEEE International Conference on Rehabilitation Robotics. https://doi.org/10.1109/ICORR.2011.5975346

ACKNOWLEDGEMENTS

Special thanks to ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO for their guidance and support during this experiment.

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 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.**

Signature: darwish Read [/]
Name: AHMAD DARWISH BIN AHMAD TERMIZI Understand [/]
Matric Number: 2212089 Agree [/]
Contribution: Introduction, Experimental Setup, Methodology

Signature: aqif Read [/]
Name: MUHAMMAD AQIF BIN ROSZAHAN Understand [/]
Matric Number: 2210345 Agree [/]
Contribution: Result, Discussion, Conclusion

Signature: Read [/]
Name: AKMAL ZARIF BIN NAJIB Understand [/]
Matric Number: 2211971 Agree [/]
Contribution: Recommendation, Reference

Signature: HAMMAD Read [/]
Name: SIDDIQUI HAMMAD AHMED Understand [/]
Matric Number: 2221201 Agree [/]
Contribution: