KULLIYYAH OF ENGINEERING

DEPARTMENT OF MECHATRONICS ENGINEERING

Mechatronics System Integration (MCTA3203)

WEEK 4: Serial Communication

Section 1

(Group E)

Adam Mikhail Bin Khairul Effendy	2117613
Muhammad Afiq bin Mazhalimi	2110491
Muhammad Fahmi Ikhwan bin Ahmad Suparjo	2111167
Eiman Bin Azzam	2110565
Zulhilmi bin Abdul Rahman	2117679

Date of submission: 8/11/2023

Prepared for: Dr. Zulkifli bin Zainal Abidin

Abstract

An Inertial Measurement Unit (IMU), is an electronic device that measures and reports acceleration, orientation, angular rates, and other gravitational forces. It combines 3 accelerometers, 3 gyroscopes, and depending on the heading requirement, 3 magnetometers.

Radio Frequency Identification (RFID) is a form of wireless communication that uses electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to identify an object, animal or person uniquely.

Table of Contents

1.0 Introduction

PART A
The MPU6050's small size, low cost, and ease of integration make it a versatile sensor for a variety of applications that require motion and orientation data. Its ability to combine accelerometer and gyroscope measurements provides a valuable source of information for a wide range of projects and devices.

Objectives:
1. To learn how to use an IMU (Inertial Measurement Unit), specifically the MPU6050..
2. To learn to take advantage of MPU6050's ability to combine accelerometer and gyroscope measurements.

PART B
To link a computer to a RFID card reader through a USB cable, a distinct method is required for the Python code to engage with it. Most USB-connected RFID card readers act as USB Human Interface Devices (HID), a library or module is needed to handle USB HID communication.

Objectives:
1. To learn how to use RFID technology.

1.2 Materials & Equipments

PART A

1) Arduino board MPU6050 sensor
2) Computer with Arduino IDE and Python installed
3) Connecting wires: Jumper wires or breadboard wires to establish the connections between the Arduino, MPU6050, and the power source.
4) USB cable: A USB cable to connect the Arduino board to a personal computer. This will be used for uploading the Arduino code and serial communication.
5) Power supply: Arduino board and MPU6050 require an external power source, make sure to have the appropriate power supply.

PART B

1. Arduino board
2. RFID card reader MFRC522
3. RFID tags or cards that can be used for authentication
4. Servo Motor: A standard servo motor to control the angle
5. Jumper wires
6. Breadboard
7. LEDs of various colors
8. USB cables to connect the Arduino board and the RFID reader to your computer.
9. Computer with Arduino IDE and Python installed
10. Power Supply (optional): If the servo motor requires a power supply other than what the Arduino can provide, you'll need the appropriate power supply.

1.3 Experimental Setup

PART A

Circuit setup:

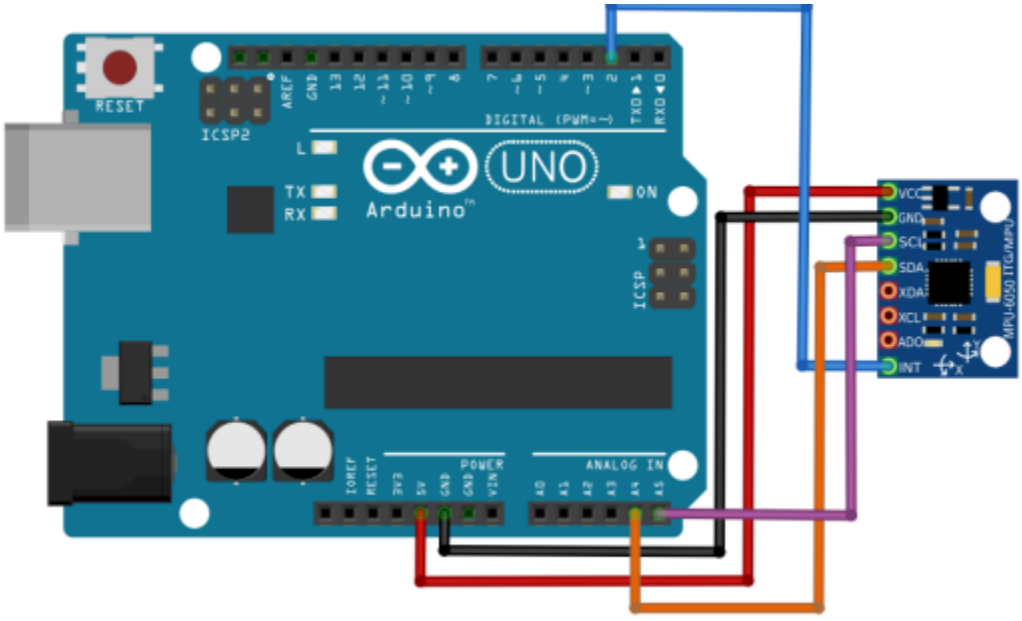


Figure 1: Arduino UNO connection to MPU6050

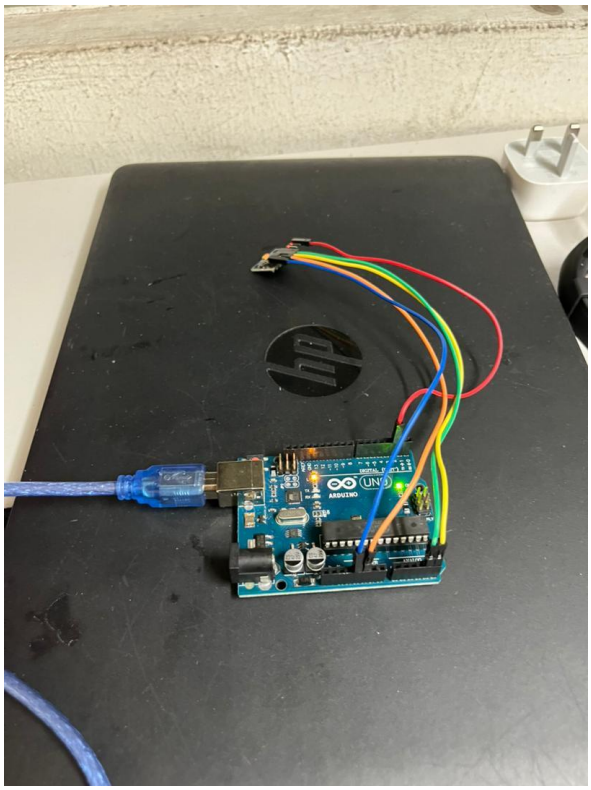


Figure 2: Full setup of MPU6050 to Arduino Uno

Experiment steps:

1. Connected the MPU6050 sensor to the Arduino board using the appropriate pins. The MPU6050 typically uses I2C communication, so connect the SDA and SCL pins of the MPU6050 to the corresponding pins on the Arduino (usually A4 and A5 for most Arduino boards).
2. Connected the power supply and ground of the MPU6050 to the Arduino's 5V and GND pins.
3. Ensured that the Arduino board is connected to PC via USB.

PART B

Circuit setup:

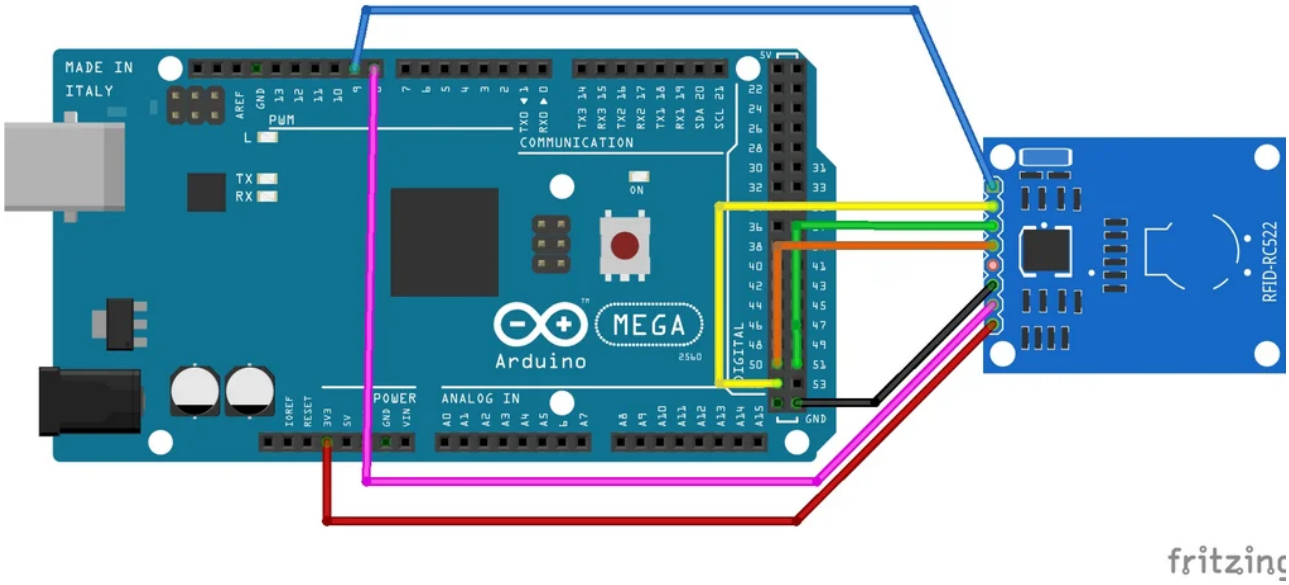


Figure 3: Arduino Mega connection to RFID-RC522

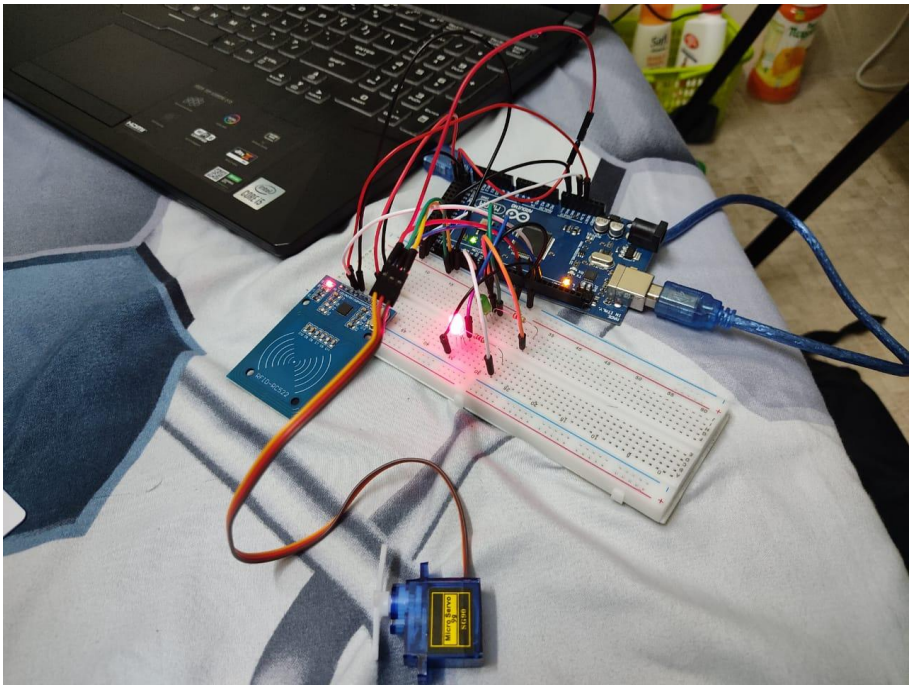


Figure 4: Full Setup

Experiment steps:

1. Connected the servo's power wire (usually red) to the 5V output on the Arduino.
2. Connected the servo's ground wire (usually brown or black) to one of the ground (GND) pins on the Arduino.
3. Connected the servo's signal wire (usually orange or yellow) to one of the PWM pins on the Arduino (e.g., pin 9).
4. Ensured that we have a common ground connection between the Arduino and the servo motor to complete the circuit.
5. RFID-RC522 is connected to the arduino based on the pinout at circuit setup..
6. Connected two LEDs, red and green and resistors as pull up resistors controlled by digital inputs from the Arduino Mega.

1.4 Methodology

PART A

1. Connect the Arduino to your computer via a USB cable.
2. Powered on the Arduino (uploaded the sketch to our Arduino using the Arduino IDE).
3. Run the Python script on your computer.
4. Move the sensor in one axis to get the readings in the python through the coding in Arduino IDE
5. Selected the Correct COM Port: In the Serial Plotter, selected the correct COM port to which our Arduino was connected.
6. Set the Baud Rate: Ensured that the baud rate in the Serial Plotter matched the one set in our Arduino code (e.g., 9600).
7. Read Real-Time Data: Read the data we got from Arduino through python by installing a pyserial library in the command prompt.

PART B

1. Connected the Arduino to the computer via USB Cable.
2. Powered on the Arduino and uploaded sketch “Week4b " using PlatformIO through Visual Studio Code. (refer to GitHub)

3. Ran the Python script “Exp4b.py” on the computer. (refer to GitHub)
4. Scanned the RFID tag onto the RFID reader and collected the ID of the tag.
5. Inserted the ID of the tag in the .json file.
6. Scanned the RFID tag again, and observed the LEDs and Servo movement.

1.5 Data Collection

- No data collection

1.6 Data Analysis

- No data analysis

1.7 Results

PART A

As the accelerometer moves, the reading was shown in the Python

Figure 4:

The axis that users enter into the Python prompt, which is axis-y, determines the reading of the movement. The Python script then transfers this movement to the Arduino using the serial connection.

Video link:
https://drive.google.com/file/d/1WMvx0aGfG4riokGU7sF7ahG50YIf55fA/view?usp=sharing

PART B

When scanning the RFID tag that has the UID authorized in the .json file, the servo moves to 180 degrees before returning back to 90 degrees and lights up the green LED. The python script also displays the name of the scanned UID.

When an unknown tag is entered, the LED stays red, the servo does not move and the python script will print “Hello?” to indicate unauthorized access.

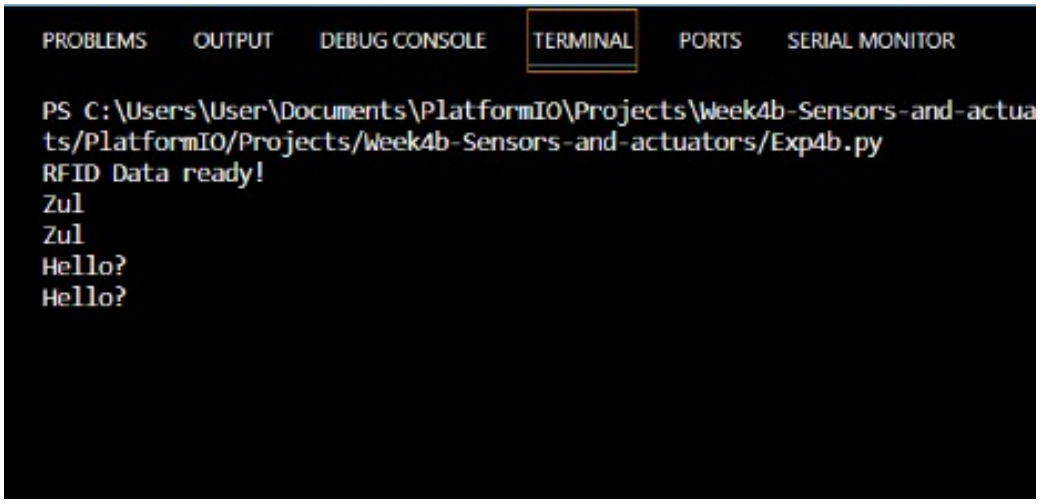


Figure 5: Output of Python Script for RFID reading

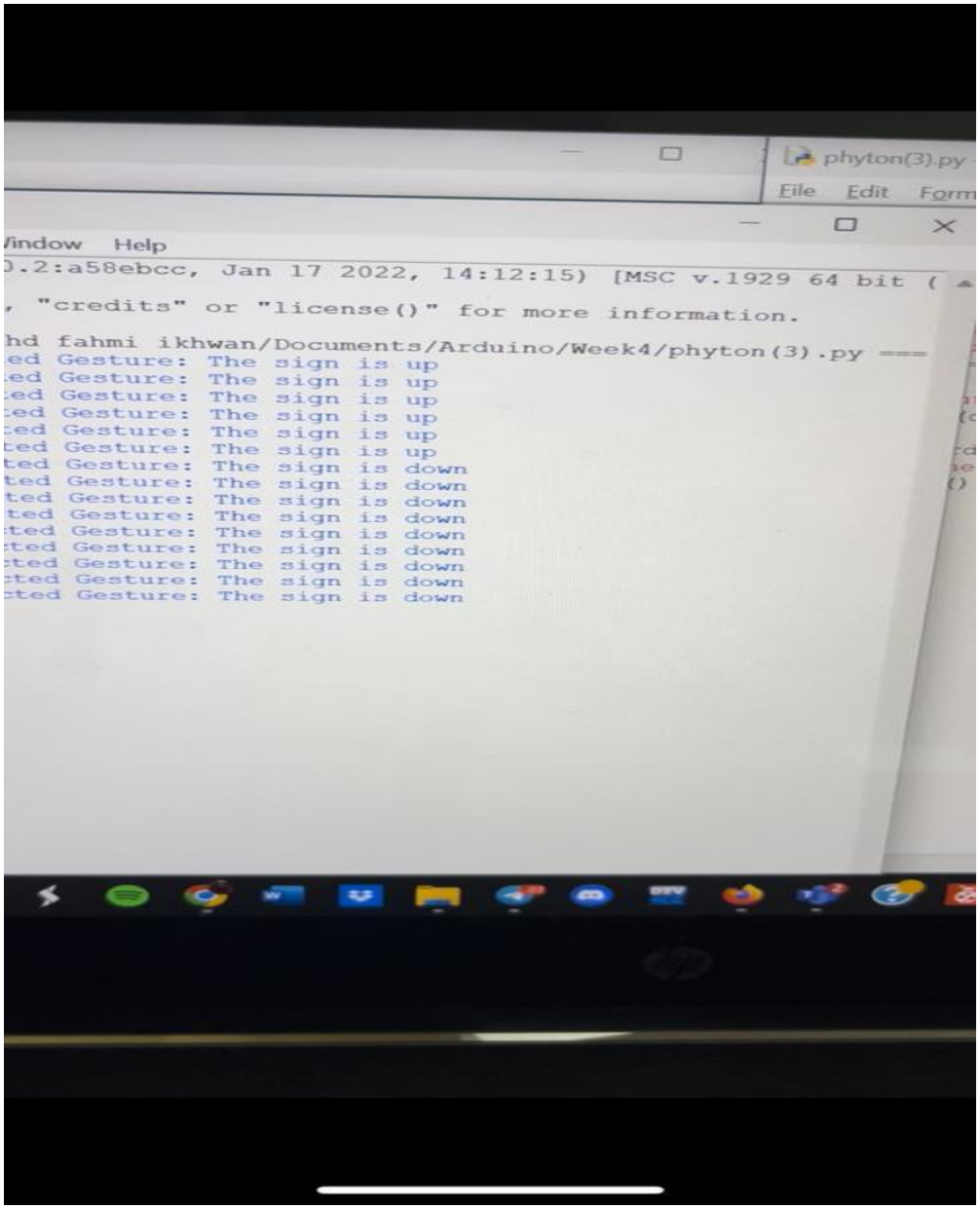
Video Link (can also refer to GitHub titled “RFID-Demo”) :

1.8 Discussion

PART A

Question:

Create a straightforward hand gesture recognition system by capturing accelerometer and gyroscope data during the execution of predefined hand movements. Employ an algorithm to identify and categorize these gestures using the collected sensor data. Additionally, visualize the paths of hand movement in an x-y coordinate system.



Video link (can also refer to GitHub as “MPU-Gesture-Demo”):
https://drive.google.com/file/d/1-OaRCdeto2i_GwI5OrvBlyRHnpbBeR_V/view?usp=sharing

PART B

Question:

Enhance the existing code to introduce a visual indicator, such as illuminating a green LED, when a recognized UID is detected by the RFID reader, and conversely, activate a red LED when an unrecognized card is read. Incorporate structured JSON data handling within your code for better organization and flexibility. Add some options for the user to freely set the angle position of the servo.

We have already incorporated a visual indicator and separated authorized UIDs in a separate .json file. Refer to 1.7 Results: PART B.

1.9 Conclusion

The objectives for both part A and part B have been achieved. We learned how to interface the MPU6050 and use its data to create gesture tasks. We also learned how to use RFID technology and code in UID to create an authorization system.

1.10 Recommendations

Use an external power supply to power the servo, as directly supplying current to the servo using the computer may end up damaging the computer due to the servo’s high current draw. We should also reduce the amount of info dumped from the RFID reader to increase the speed at which the other commands are run as we are using serial commands to turn the servo.

1.11 References

1. Interfacing RFID-RC522 With Arduino MEGA a Simple Sketch — Instructables.com
https://www.instructables.com/Interfacing-RFID-RC522-With-Arduino-MEGA-a-Simple-/

1.12 Appendices

Refer to GitHub for coding and videos.

1.13 Acknowledgements

We would like to acknowledge our instructor, Dr. Wahju Sediono, for teaching us how to interface the MPU6050 and RFID card reader and answering all our tedious questions due to the difficulties we faced during the experiment. We hope to learn more in the future from such a talented instructor if the opportunity arises.

We would also like to give thanks to Allah S.W.T for giving us guidance in demonstrating this experiment successfully.

1.14 Student's Declaration

Declaration:

We certify that this project/assignment is entirely our own work, except where we have given fully documented references to the work of others, and that the material in this assignment has not previously been submitted for assessment in any formal course of study.

_adam_____ Date: 8/11/2023

Name: Adam Mikhail Bin Khairul Effendy

fahmi_____

Name: Muhammad Fahmi Ikhwan Bin Ahmad Suparjo

_____

Name: Eiman Bin Azzam

afiq

Name: Muhammad Afiq bin Mazhalimi

_Zul_____

Name: Zulhilmi bin Abdul Rahman