REPORT 4A : SERIAL COMMUNICATION

GROUP 4

MCTA 3203

SEMESTER 2 2023/2024

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION:
10 APRIL 2024

	NAME	MATRIC NUMBER
1.	NUR SHAHEERA BINTI KAMIS	2214040
2.	NORAMIRA HUSNA BT NORKHAIRANI	2214496
3.	MUHAMMAD EIQBAL BIN HASBOLLAH	2216911

TABLE OF CONTENT

ABSTRACT

This experiment investigates a practical experiment in hand gesture recognition with an Arduino UNO and an MPU6050 sensor. The setup consists of attaching the sensor to the Arduino, uploading code to both the Arduino and a PC running Python, and analysing sensor data to detect hand motions. The findings show that inexpensive electronics may be used for gesture recognition, with an emphasis on calibration, noise reduction, and algorithm selection to ensure reliable detection. Calibration of the sensor, noise reduction in data, sample rate optimisation, and well-defined gesture parameters are all recommended for good identification. The project demonstrates the potential of the Arduino UNO for gesture detection and proposes areas for future investigation and enhancement.

Keywords: Arduino UNO, MPU6050 sensor, hand gesture identification, calibration, noise reduction, sample rate, gesture definition.

MATERIALS AND EQUIPMENT

- Arduino board
- MPU6050 sensor
- Computer with Arduino IDE and Python installed
- Connecting wires: Jumper wires or breadboard wires to establish the connections between the Arduino, MPU6050, and the power source.
- USB cable: A USB cable to connect the Arduino board to your personal computer. This will be used for uploading the Arduino code and serial communication.

- Power supply: If your Arduino board and MPU6050 require an external power source, make sure to have the appropriate power supply.
- LEDs of different colours.

EXPERIMENTAL SETUP

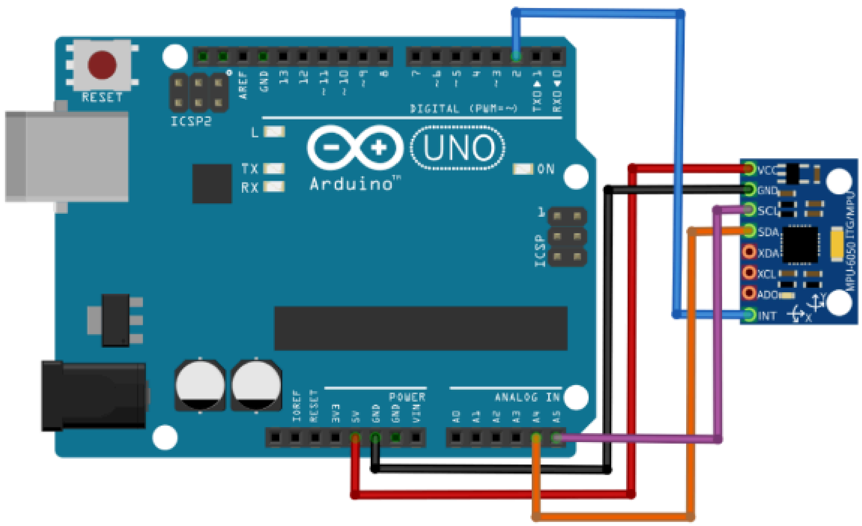


Fig. 1: Arduino-MPU6050 Connections

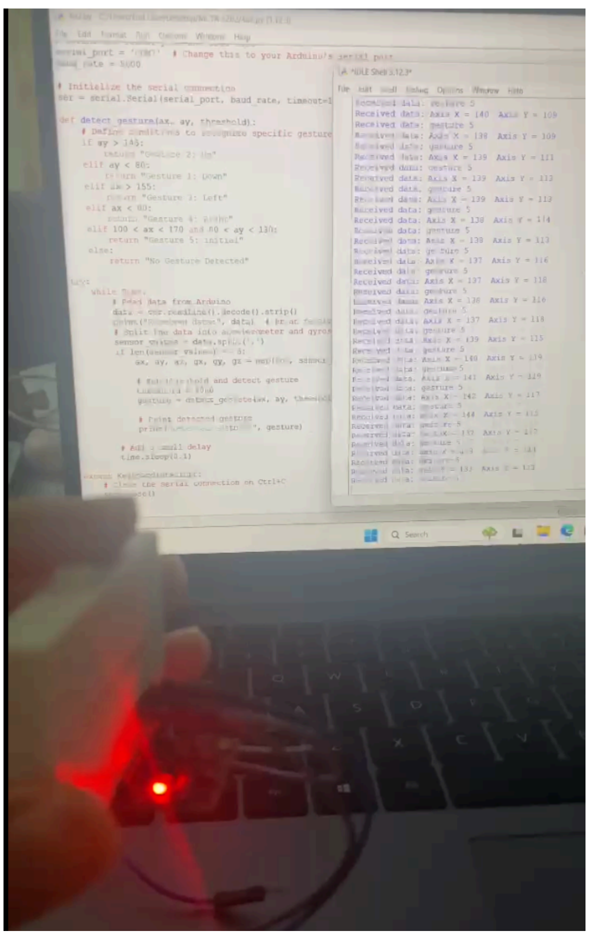
METHODOLOGY

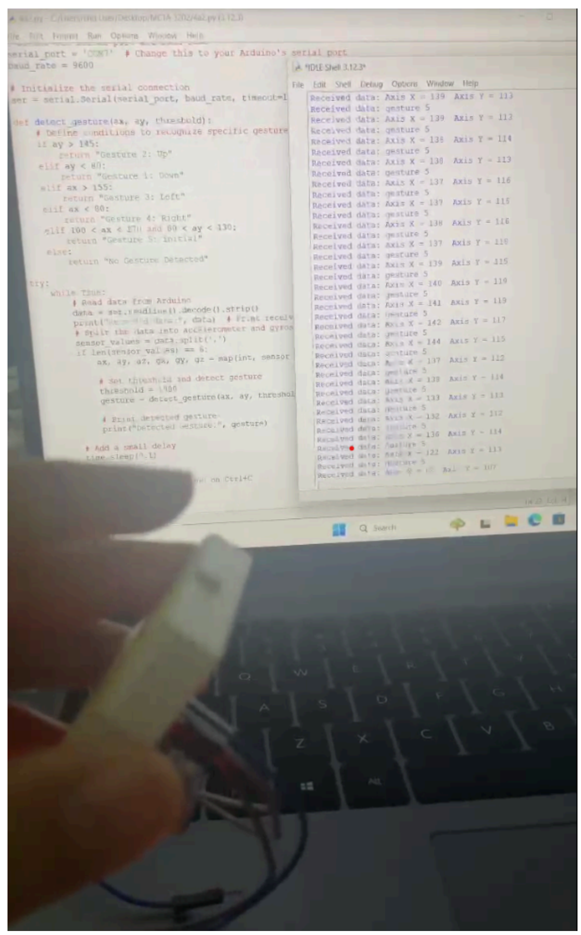
- Connect the MPU5060 the Arduino board.
- Connect the VCC to 5v in Arduino
- Connect the GND to GND in Arduino
- Connect the SCL to A5 pin Arduino.
- Connect the SDA to A4 pin Arduino.
- Connect the INT to pin3 Arduino.

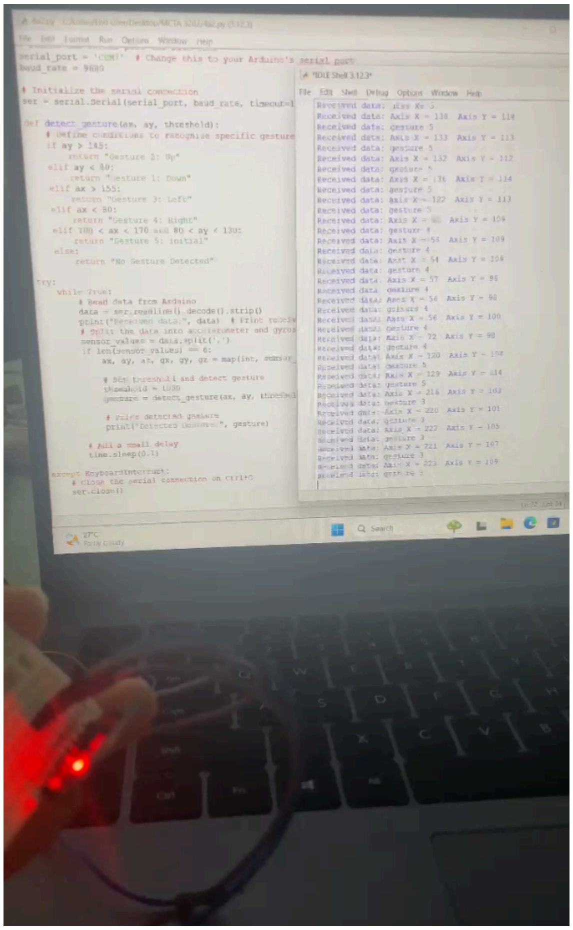
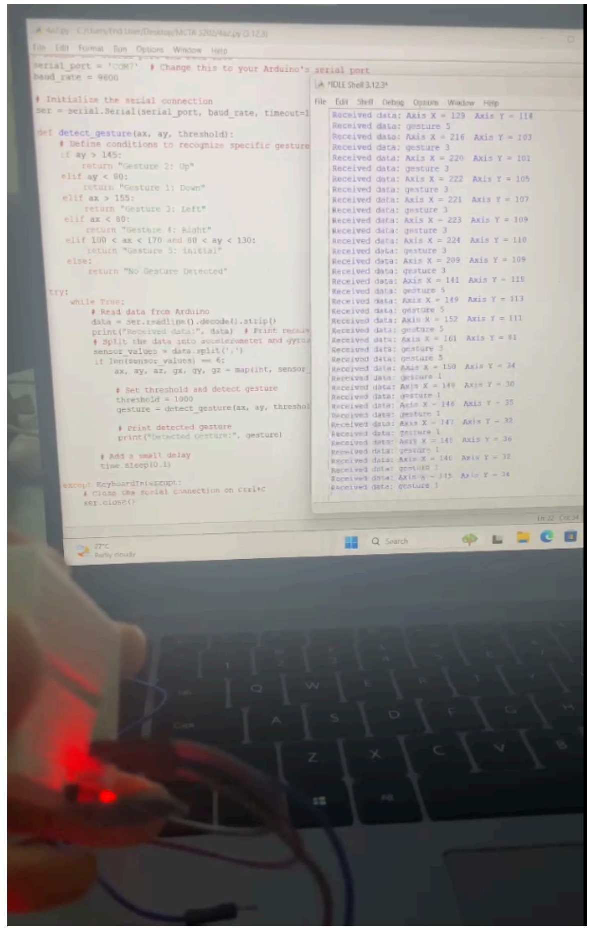
PROCEDURE

1. Built the circuit per the setup instructions provided.
2. The provided Arduino code to your Arduino Uno and Python.
3. The sensor will show the our gesture hand in python if the sensor moved according to our hand gestures.

RESULTS





DISCUSSIONS

- **Hardware**

  In the serial communication experiment, the input devices include the MPU6050 sensor, which detects motion and orientation and sends data like as acceleration and rotation to the Arduino board. Furthermore, the computer running Arduino IDE and Python acts as a platform for receiving data from the Arduino board via serial connection. Python programmes running on the computer can analyse this information for analysis or presentation.

On the output side, the Arduino board functions as both a control unit and an output device. It provides data to the computer via serial connection, including sensor readings and other acquired information. Furthermore, LEDs function as visual output devices, providing feedback by showing data transmission or receipt or graphically depicting sensor values. These components work together to create a system in which data is sent between the input devices, Arduino board, and computer, allowing for the serial communication experiment.

- **Electrical**

  To set up the experiment, the MPU6050 sensor is linked to the Arduino board by connecting its power and ground wires to the Arduino, and its Serial Clock (SCL) and Serial Data (SDA) lines to the Arduino's analogue pins. In addition, the sensor's Interrupt (INT) pin is linked to a digital pin on the Arduino to allow for interrupt-based communication. This physical setup guarantees that the sensor can efficiently interact with the Arduino board, delivering information on motion and orientation.

  Once the hardware is configured, the accompanying Arduino code is uploaded to the Arduino Uno. This code most likely contains instructions for reading data from the MPU6050 sensor and processing it appropriately. Next, Python programmes are used to interact with the sensor data. These programmes receive data sent from the Arduino board to the computer using serial connection. By analysing this data, Python programmes can understand motions observed by the sensor, providing a practical way to engage with the experiment. Together, these stages enable the experiment's setup and execution, allowing hand motions to be detected and interpreted using the MPU6050 sensor.

- **Software**
  - **First code (Python):**

    This Python script sets up serial communication with the Arduino board to receive sensor data. It establishes the serial connection using the provided port and baud rate. Within the main loop, data is continually read from the serial port, separated into accelerometer and gyroscope measurements, and then sent to the detect_gesture function to determine the gesture based on predetermined circumstances. When a gesture is detected, it prints the matching message. There's also a KeyboardInterrupt exception handler that gently closes the serial connection when the user stops the script (for example, by pressing Ctrl + C).
  - **Second code (Arduino):**

    The Arduino code begins by adding the libraries required for I2C communication and interacting with the MPU6050 sensor. The setup function configures serial connectivity and the MPU6050 sensor. The loop function constantly receives accelerometer data from the sensor, translates the values to a range appropriate for gesture recognition, and displays the mapped values to the serial port. Based on the mapped values, it recognises particular movements such as up, down, left, right, and beginning gestures and transmits relevant messages via the serial port.

CONCLUSION

The experiment utilising Arduino UNO for hand gesture identification based on accelerometer and gyroscope data emphasises the viability of adopting low-cost electronics for this purpose. By following to safeguards such sensor calibration and noise reduction procedures, adequate accuracy levels can be obtained. The choice of categorization algorithm strongly effects system performance, underscoring the necessity of algorithm selection and testing. Additionally, the quality and diversity of training data play key roles in developing effective recognition models. Real-time validation of the system is vital for assessing practical applicability and suggesting opportunities for development. Despite various restrictions, such as complexity limits and processing limitations, the experiment highlights the potential of Arduino UNO for hand gesture detection and proposes routes for further study to expand system capabilities.

REFERENCE

[1]“4. Serial Communications - Arduino Cookbook [Book],” www.oreilly.com. https://www.oreilly.com/library/view/arduino-cookbook/9781449399368/ch04.html

[2]“Cross Platform serial communication using Python (PySerial) and Arduino | xanthium enterprises,” *www.xanthium.in*. https://www.xanthium.in/Cross-Platform-serial-communication-using-Python-and-PySerial

[3]“Serial Communication between Python and Arduino,” projecthub.arduino.cc. https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756

RECOMMENDATION

Calibration: Prior to commencing the experiment, it is imperative to verify that your accelerometer and gyroscope sensors are accurately calibrated. Calibration aids in mitigating systematic mistakes and guarantees precise results.

Noise Reduction: Apply filtering techniques to mitigate noise in the sensor data. One such approach is to employ low-pass filtering to eliminate high-frequency noise and utilise averaging techniques to mitigate volatility.

Sample Rate: Select an optimal sample rate for acquiring sensor data. Increasing the sample rate can yield more intricate data on hand movements, however it may necessitate additional processing resources.

Gesture Definition: Precisely specify the hand motions that you intend to identify and verify that they are consistently performed throughout the experiment. This facilitates the precise training and testing of the recognition algorithm.

ACKNOWLEDGEMENT

A special thanks goes out to Dr. Wahju Sediono, Dr. Ali Sophian, 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

<u>Certificate of Originality and Authenticity</u>

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 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:	**Read**	/
Name: Noramira Husna Bt Norkhairani	**Understand**	/
Matric No: 2214496	**Agree**	/

Signature:	**Read**	/
Name: Nur Shaheera Bt Kamis	**Understand**	/
Matric No:2214040	**Agree**	/

Signature:	**Read**	/
Name: Muhammad Eiqbal bin Hasbollah	**Understand**	/
Matric No: 2216911	**Agree**	/