LABORATORY PROJECT REPORT

SERIAL AND USB INTERFACING WITH MICROCONTROLLER

EXPERIMENT 4B

DATE : 31TH MARCH 2025

SECTION : 1

GROUP : 1

SEMESTER 2, 2024/2025

NO	NAME	MATRIC NO
1.	AKMAL FAUZAN BIN AZHAR	2319531
2.	AIMAN SAIFULLAH BIN AMINNURLLAH	2319185
3.	SYAFIQ HELMIE BIN SAIFULLAH ADHA	2316049

DATE OF SUBMISSION :

Monday, 7TH April 2025

Abstract

This experiment explores the integration of an RFID card reader with an Arduino microcontroller and a Python-based computer interface using USB communication. The objective is to implement an authentication system that enables or restricts servo motor control based on RFID card authorization. The methodology involves interfacing a USB-connected RFID reader with a Python script using the pyusb library to identify card IDs, while the Arduino receives serial commands to actuate a servo motor. Authorized card detection prompts Python to transmit a command to rotate the servo, while unauthorized cards revert it to a neutral position. Visual indicators, such as LEDs, are incorporated to signal authentication status. The Arduino is programmed to interpret control signals over serial communication, and Python handles USB HID interactions and decision logic. Key findings highlight the importance of proper USB HID configuration, effective serial command parsing, and user feedback mechanisms through LEDs. The results demonstrate a functional framework for secure access systems and provide practical experience in USB interfacing, device control, and cross-platform integration. This experiment serves as a hands-on introduction to real-time embedded control systems and authentication-based automation in mechatronics.

Table of Contents

1.0 Introduction

This experiment aims to demonstrate the principles of USB-based serial communication by linking an Arduino microcontroller, an RFID card reader, and a computer to create an authentication-based control system. The objective is to detect RFID card IDs using a USB-connected RFID reader and use Python to process the identification data, which then communicates with the Arduino to control a servo motor. Through this setup, students will gain hands-on experience in embedded system integration, actuator control, and real-time data exchange between multiple devices.

Serial and USB communication are foundational to modern embedded systems, enabling reliable data transfer between microcontrollers, sensors, and computers. This experiment involves detecting RFID card data using Python's pyusb library and sending control signals to the Arduino over a serial USB connection. The Arduino, in turn, receives these commands to actuate a servo motor and control LED indicators, providing visual feedback based on authentication status. This experiment reinforces important concepts such as USB HID interfacing, serial command interpretation, and microcontroller-based decision execution.

The hypothesis for this experiment is that, with proper configuration and coding of both the Arduino and the Python script, the RFID card reader will accurately identify authorized and unauthorized card IDs, leading to correct servo actuation and LED response. A successful outcome will validate the system's ability to integrate sensor input with actuator response through synchronized serial and USB communication, highlighting its potential for real-world access control applications.

2.0 Materials and Equipment

The following materials and components were used in the experiment:

2.1 Electronic Components

- Arduino Board – Serves as the main microcontroller unit to receive serial commands and control the servo motor.
- RFID Card Reader (USB) – Used to scan RFID tags and send identification data to the computer.
- RFID Tags or Cards – Contain unique identifiers used for authentication.
  Servo Motor – Actuator that changes position based on authorization results.
- LEDs (Red and Green) – Visual indicators to show access status (granted or denied).
- Jumper Wires – For making electrical connections between the Arduino and components.
- Breadboard – A reusable platform for building the circuit without soldering.

2.2 Equipment and Tools

- Arduino IDE – Development environment for writing and uploading code to the Arduino board.
- USB Cables – For connecting both the Arduino and the RFID reader to the computer.
  Computer with Python Installed – Runs the Python script to process RFID data and communicate with the Arduino.
- PyUSB Library – Python library for USB HID communication with the RFID reader.
- Power Supply (Optional) – May be required for powering the servo motor if the Arduino's 5V output is insufficient.
- Datasheets and Manuals – Technical references used to understand wiring, protocols, and component limitations.
- Mounting Hardware (Optional) – Used to position the servo motor securely for controlled movement.

3.0 Experimental Setup

```
       Arduino                      Servo Motor
    +-----------+             +----------------+
    |           |             |                |
    |    5V     |-------------| Power (Red)    |
    |           |             |                |
    |    GND    |-------------| Ground (Brown) |
    |           |             |                |
    |    D9     |-------------| Signal (Yellow)|
    |           |             |                |
    +-----------+             +----------------+
```

3.1 Circuit Setup

The servo motor and RFID card reader were connected to the Arduino to enable access control functionality as follows:

- The servo motor's power wire (typically red) was connected to the 5V pin on the Arduino to supply voltage.
- The ground wire (typically brown or black) was connected to a GND pin on the Arduino to complete the power circuit.
- The signal wire (typically yellow or orange) was connected to digital pin 9, which supports PWM control for positioning the servo.
- The USB RFID card reader was connected directly to the computer via a USB port. Since most RFID readers operate as USB HID devices, no additional wiring to the Arduino was required for the reader.
- A green LED with a 220-ohm resistor was optionally connected to a digital output pin (e.g., D7) to indicate authorized access.
- A red LED with a 220-ohm resistor was optionally connected to another digital pin (e.g., D8) to indicate access denial.
- All components were assembled on a breadboard using jumper wires, ensuring a secure and solderless configuration.
- A common ground was maintained between the Arduino and the servo motor to ensure stable operation.

4.0 Methodology

4.1 Implementation and Testing

The circuit was assembled using a breadboard, Arduino board, servo motor, jumper wires, USB RFID reader, and optional LEDs with 220-ohm resistors. The setup involved the following:

1. The servo motor was connected with:

    a. Power wire to 5V on the Arduino
    b. Ground wire to GND
    c. Signal wire to digital pin 9 for PWM control
    d.
2. The green and red LEDs were connected with 220-ohm resistors to digital pins D7 and D8, respectively, for visual feedback.
3. The RFID reader was connected directly to the computer via USB and powered through the connection.
4. The Arduino code was uploaded using the Arduino IDE to enable servo actuation based on serial input commands.
5. A Python script, using the pyusb library, was executed to detect RFID card scans and send corresponding commands ('A' for access granted, 'D' for denied) to the Arduino via serial communication.
6. Authorized and unauthorized cards were scanned to test the system.
7. Servo movement and LED activation were observed to confirm appropriate responses based on card recognition.

4.2 Control Algorithm

The Arduino board was programmed using the Arduino IDE to respond to serial commands from the Python script and control the servo motor accordingly. The Arduino control algorithm followed these steps:

- Initialize **serial communication** at a baud rate of **9600**.
- Attach the servo to **pin 9** and set its initial position.
- Continuously listen for incoming serial characters.
- If the received character is **'A'**, set the servo to **180°** (access granted).
- If the character is **'D'**, return the servo to **90°** (default position, access denied).
- Optionally, turn on the **green LED** for access granted or **red LED** for access denied.

The Python script executed the following logic:

- Use the pyusb library to **connect to the RFID reader** by specifying its **vendor and product IDs**.
- Continuously **read RFID card data** from the device endpoint.
- Check if the scanned **card ID matches an authorized list**.
- If matched, print a confirmation message and send the **'A' command** to the Arduino over serial.
- If unmatched, print an access denied message and send the **'D' command**.
- Optionally, structure card data using **JSON** for future extensibility.
- Maintain the USB and serial connections during execution and safely handle termination when needed.

5.0 Data Collection

This experiment utilized an RFID reader as the primary sensor for data acquisition. The reader, connected via USB to a computer, was used to scan RFID cards and transmit unique identifiers (UIDs) to a Python script. The script processed the data and sent appropriate serial commands to the Arduino. The Arduino then actuated a servo motor and illuminated either a red or green LED to indicate access status.

The following components were used as data acquisition instruments:

- RFID Reader (USB HID device) – Captured the UID of each scanned RFID card.
- Python Script – Logged the UID and determined whether the card was authorized or unauthorized.
- Servo Motor – Provided a physical output by rotating to a designated angle based on access status.
- LEDs – Green LED indicated successful authentication; red LED indicated denied access.

6.0 Data Analysis

The purpose of this experiment was to validate whether the integration of RFID-based input and Arduino-controlled output could reliably enforce an access control system.

Analysis Overview:

- Success Rate: 100% recognition accuracy was achieved. All 3 matric cards triggered the intended response, and all unauthorized cards were denied access.
- Servo Actuation: The servo consistently moved to 180° when access was granted and 90° when denied, aligning perfectly with the intended logic.
- Response Time: From card scan to servo movement, the response time was observed to be less than 1 second, indicating a responsive system.
- Error Rate: No false positives or false negatives occurred.

Significance of Results:

The data confirms the hypothesis that a properly configured system can accurately detect RFID card IDs and perform actions based on authorization status. The consistent servo angles and LED responses support the reliability of the control logic and hardware-software integration. These results also demonstrate that such a system could be extended to more advanced applications such as smart locks, attendance tracking, or entry control systems.

No complex mathematical modeling was necessary for this experiment, but the clear logical flow from input (RFID scan) to output (servo and LED) illustrates a successful application of control logic, sensor interfacing, and serial communication in embedded systems.

7.0 Results

The experiment successfully demonstrated real-time USB and serial communication between a USB RFID reader, a Python-based control script, and an Arduino microcontroller. When RFID cards were scanned, the system accurately detected and processed the card UID, determined the card's authorization status, and triggered appropriate responses through the Arduino.

For authorized cards, the Python script sent a signal ('A') to the Arduino, resulting in the servo rotating to 180° and the green LED turning on. For unauthorized cards, the script sent a 'D' signal, prompting the servo to return to its default 90° position and activating the red LED. These outcomes occurred consistently across all trials.

Data transmission between the devices was smooth and responsive, with no observable lag or communication errors. The USB RFID reader functioned effectively using the pyusb library to interface with the computer, and the Arduino received and processed serial data promptly due to the synchronized baud rate of 9600.

8.0 Discussion

The experiment successfully demonstrated USB and serial communication between a Python-based control interface and an Arduino microcontroller, using an RFID reader as the input device. The correct transmission of card identification data from the RFID reader to the Python script, followed by real-time serial communication to the Arduino, validated the accuracy of the system's hardware and software integration.

Authorized RFID cards reliably triggered the servo motor to rotate to 180° and activated the green LED, while unauthorized cards resulted in the servo resetting to 90° and the red LED turning on. These outcomes confirmed that both the Arduino and Python scripts were correctly implemented, and that command signals were consistently interpreted and executed.

The system's responsiveness and stability during repeated trials demonstrated that:

- The pyusb library effectively handled USB HID communication,
- The serial port was properly managed in both the Arduino and Python environments,
- And the control logic accurately translated RFID data into actuator output.

A minor challenge found was ensuring that the servo motor received sufficient and stable power, especially during rapid directional changes. This could be improved in future iterations by using an external power source for the servo rather than relying solely on the Arduino's 5V output.

No false positives or missed card readings were observed, indicating high reliability of the system when components were properly connected and configured. However, additional robustness could be added by implementing timeout handling and UID validation checks in the Python script.

Overall, the experiment effectively illustrated the principles of HID and serial communication, actuator control, and sensor-based decision logic. It reinforced key practices in microcontroller interfacing, including matching baud rates, handling USB devices, and providing immediate visual feedback through indicators. These findings demonstrate the potential for expanding the setup into more complex mechatronic systems, such as secured access control, automated gates, or networked identity verification systems.

9.0 <u>Conclusion</u>

In this experiment, we successfully demonstrated real-time communication between an RFID card reader, a Python-based control system, and an Arduino microcontroller using USB and serial protocols. The system accurately read RFID card UIDs, determined access status, and triggered appropriate responses through a servo motor and visual LED indicators. This confirmed the correct wiring, synchronization, and implementation of both the Python and Arduino code.

The experiment reinforced key concepts in mechatronic system integration, including USB HID interfacing, serial communication, actuator control, and real-time decision-making based on sensor input. The results validate the effectiveness of the control logic and highlight the reliability of combining software-based authentication with hardware actuation.

This project provides a solid foundation for more advanced applications, such as network-based access control, integration with databases, or adding biometric sensors. Future improvements may include enhanced user feedback, secure data handling, and modular code design for scalability.

10.0 <u>Recommendations</u>

To improve the reliability and performance of the RFID-based authentication system, it is recommended to modularize both the Arduino and Python code by using functions to simplify repetitive tasks and enhance code readability. Incorporating serial debugging through the Arduino Serial Monitor and terminal print statements in Python can help identify and resolve communication issues more efficiently. Ensuring that all hardware components, particularly jumper wires, connectors, and the servo motor, are in good condition will minimize erratic behavior and improve data consistency. Replacing any worn or faulty parts is essential for maintaining accuracy. For future improvements, integrating real-time data visualization using Python plotting libraries or an OLED/I2C display can offer more user-friendly feedback. Additionally, structuring RFID card data using JSON can support cleaner code logic and facilitate potential expansion into database integration. Finally, exploring wireless communication methods such as Bluetooth or WiFi can increase system flexibility and enable remote control or monitoring capabilities in more advanced applications.

11.0 References

```
   Arduino                  Servo Motor
+-----------+            +----------------+
|           |            |                |
|    5V     |------------| Power  (Red)   |
|           |            |                |
|    GND    |------------| Ground  (Brown)|
|           |            |                |
|    D9     |------------| Signal (Yellow)|
|           |            |                |
+-----------+            +----------------+
```

Weekly Module - Google Drive. (n.d.). Google Drive. https://drive.google.com/drive/folders/1rq0wLF6mA7jEoNsPWyAbWR9n0X3SQsQz?usp=sharing

12.0 Acknowledgement

A special thank you to Dr. Wahju Sediono and Dr. Zulkifli Bin Zainal Abidin, my teaching assistant, and peers, for their outstanding assistance and support in completing this report. Their advice, input, and expertise have had a significant impact on the quality and comprehension of this work. Their time, patience, and dedication to my academic progress are deeply appreciated.

13.0 <u>Student's Declaration</u>

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.

Name: Aiman Saifullah bin Aminnurllah	Read	/
Matric Number: 2319185	Understand	/
Signature:	Agree	/

Name: Akmal Fauzan Bin Azhar	Read	/
Matric Number:2319531	Understand	/
Signature:	Agree	/

Name: Syafiq Helmie bin Saifullah Adha	Read	/
Matric Number: 2316049	Understand	/
Signature:	Agree	/