MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 1 2024/2025

WEEK 4A: SERIAL COMMUNICATION RFID

SECTION 2

GROUP 8

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO

NO.	GROUP MEMBERS	MATRIC NO.
1.	AHMAD DARWISH BIN AHMAD TERMIZI	2212089
2.	SYABAB ALHAQQI BIN JAAFAR	2211117
3.	WAN NAIMULLAH BIN MOHD AZRUDDIN	2214837
4.	ZAIMUDIN ZAKWAN BIN NORZAMRI	2217513
5.	ZAMIR MUTTAQIN BIN MUHAMMAD BALYAN	2212985

Date of Experiment: Wednesday, 30 October 2024

Date of Submission: Wednesday, 6 November 2024

ABSTRACT

The experiment aims to create an RFID authentication system that controls a servo motor using an Arduino and Python. The setup includes an RFID card reader connected to the computer via USB, which reads the unique identifiers (UIDs) of RFID cards. If an authorized card was scanned, a green LED lit up and the servo motor moved to a user-defined angle. If the card was unrecognised, a red LED activates and the motor remains stationary. The system utilises the pyusb library for communication and includes structured JSON data handling for better organisation. This experiment demonstrates how RFID technology can be integrated with microcontrollers for access control applications.

TABLE OF CONTENTS

INTRODUCTION

In this experiment, we aim to create an access control system by integrating an RFID card reader with a servo motor using Arduino and Python. This setup mimics real-world authentication systems, where RFID cards are used to grant or restrict access. An RFID card reader will detect and identify RFID tags, sending data to the computer via USB, which in turn will process the input through a Python script. When an authorized RFID tag is detected, the system will signal the Arduino to control a servo motor, simulating an action (such as unlocking a door) by setting the motor to a specific angle. Unauthorized tags will trigger a different response, illuminating LEDs of various colors to indicate access status.

This experiment not only demonstrates fundamental principles of serial and USB interfacing but also provides practical experience in handling USB HID devices, controlling actuators, and processing structured data with JSON for improved flexibility and data management.

MATERIALS AND EQUIPMENTS

1. ARDUINO MEGA 2560
2. RFID Card Reader
3. RFID cards
4. Servo Motor
5. Breadboard
6. Jumper Wires
7. LED

EXPERIMENTAL SETUP

1. Wiring the Servo Motor

- Connect the servo motor's Power (Red) wire to the 5V pin on the Arduino.
- Connect the Ground (Brown) wire to a GND pin on the Arduino.
- Connect the Signal (Yellow) wire to PWM pin D9 on the Arduino.
- Ensure a common ground connection between the Arduino and the servo motor.

2. USB RFID Reader Setup

- Connect the RFID reader to the computer via USB. The USB connection provides both power and data communication.
- Identify the COM port assigned to the RFID reader by checking the device in a serial port manager (such as Device Manager on Windows). This COM port will be used in the Python code to communicate with the reader.

3. LED Indicators

- Attach two LEDs (one green and one red) to the breadboard.
- Connect each LED's positive leg to a digital output pin on the Arduino. D3 for green and D6 for red.
- Connect the negative legs of the LEDs to a common ground on the Arduino, using appropriate resistors to prevent excess current.

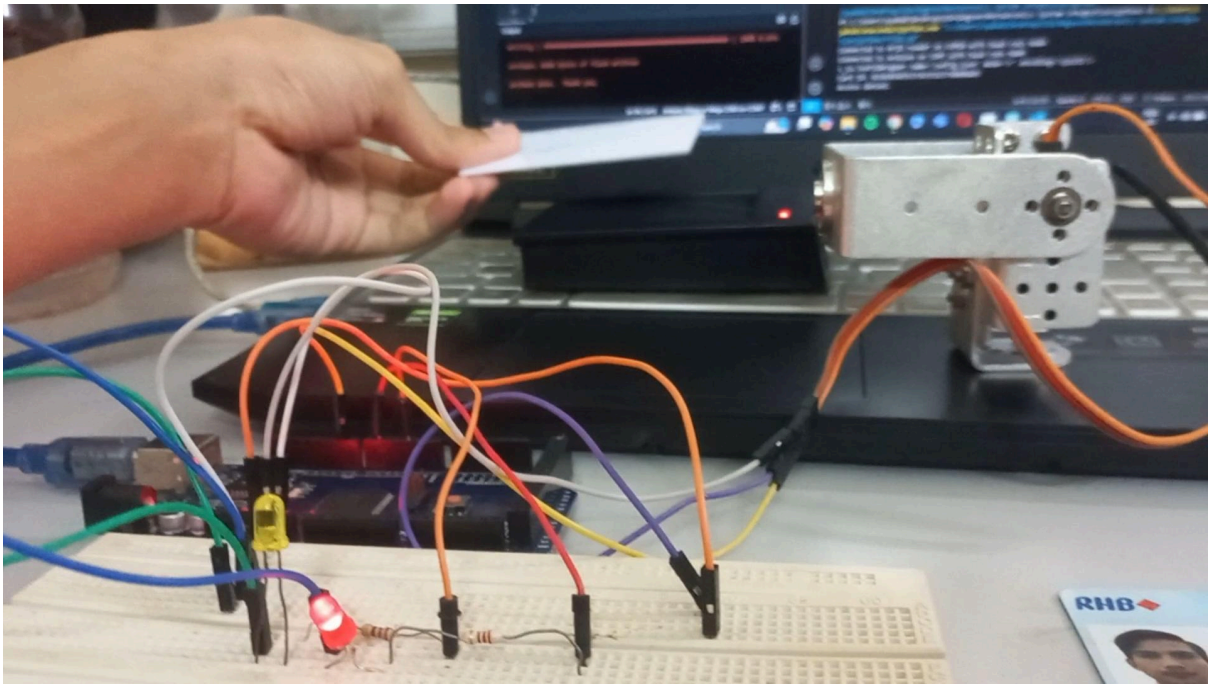

Figure 1 : Experimental Setup

METHODOLOGY

1. Setup the Arduino Mega 2560, servo motor, LEDs and RFID scanner
2. Code implementation
3. Testing
4. Authorized card will raise the servo and light the yellow LED
5. Unauthorized card will not raise the servo and light the red LED
6. Code snippet

Arduino Code

```
#include <Servo.h>
const int servoPin = 9;
const int greenLEDPin = 3;
const int redLEDPin = 6;
Servo myServo;

void setup() {
  Serial.begin(9600);       // Begin serial communication
  myServo.attach(servoPin);
  pinMode(greenLEDPin, OUTPUT);
  pinMode(redLEDPin, OUTPUT);
  Serial.println("Arduino ready.");
}

void loop() {
  if (Serial.available() > 0) {
    char accessStatus = Serial.read(); // Read access status ('G' for granted, 'D' for denied)

    // Wait for the angle to be sent next
    delay(10); // Slight delay for data availability
    if (Serial.available() > 0) {
      int angle = Serial.parseInt(); // Read the servo angle sent from Python

      if (accessStatus == 'G') {
        digitalWrite(greenLEDPin, HIGH);  // Turn on green LED
        digitalWrite(redLEDPin, LOW);     // Turn off red LED
        myServo.write(angle);             // Move servo to granted angle
      } else if (accessStatus == 'D') {
        digitalWrite(greenLEDPin, LOW);   // Turn off green LED
        digitalWrite(redLEDPin, HIGH);    // Turn on red LED
        myServo.write(angle);             // Move servo to denied angle
      }
  }
```

```
 }
}
```

Python Code

```
import serial
import time
import json

# Serial port for RFID reader and Arduino
RFID_PORT = "COM10"
ARDUINO_PORT = "COM7"
BAUD_RATE_RFID = 9600
BAUD_RATE_ARDUINO = 9600

# Load configuration
def load_config():

    with open("config.json", "r") as file:
        print(file)
        return json.load(file)

# Initialize RFID and Arduino connections
try:
    # Set up RFID serial connection
    rfid_serial = serial.Serial(RFID_PORT, BAUD_RATE_RFID, timeout=1)
    print(f"Connected to RFID reader on {RFID_PORT} with baud rate
{BAUD_RATE_RFID}")

    # Set up Arduino serial connection
    arduino_serial = serial.Serial(ARDUINO_PORT, BAUD_RATE_ARDUINO, timeout=1)
    print(f"Connected to Arduino on {ARDUINO_PORT} with baud rate
{BAUD_RATE_ARDUINO}")

    # Load config settings
    config = load_config()
    authorized_cards = config["authorized_cards"]
    servo_angle_granted = config["servo_angle_granted"]
    servo_angle_denied = config["servo_angle_denied"]

    last_card_id = None
    last_read_time = 0
```

```
    debounce_time = 3  # Seconds to wait before reading the next card

    # Main loop
    while True:
        if rfid_serial.in_waiting > 0:  # Check if data is available
            card_data = rfid_serial.read(rfid_serial.in_waiting).hex()
            current_time = time.time()

            # Check if the card ID is different or enough time has passed since the last read
            if (card_data != last_card_id or (current_time - last_read_time) >= debounce_time):
                last_card_id = card_data
                last_read_time = current_time
                print(f"Card ID: {card_data}")

                if card_data in authorized_cards:
                    print("Access granted.")
                    arduino_serial.write(b"G")  # Send 'G' for granted access
                    arduino_serial.write(str(servo_angle_granted).encode())
                else:
                    print("Access denied.")
                    arduino_serial.write(b"D")  # Send 'D' for denied access
                    arduino_serial.write(str(servo_angle_denied).encode())

            time.sleep(0.5)  # Short delay to avoid continuous reads

except Exception as e:
    print(f"Error: {e}")

finally:
    if 'rfid_serial' in locals():
        rfid_serial.close()
    if 'arduino_serial' in locals():
        arduino_serial.close()
    print("Serial connections closed.")
```

JSON Code

```
{
    "authorized_cards":["023030303639393938323 10d0a03"],
    "servo_angle_granted": 90,
    "servo_angle_denied": 0
}
```

DATA COLLECTION

RFID Card ID	Servo Angle (°)	LED Yellow	LED Red
023030303639393938323 10d0a03	90	On	Off
023030303537363332373 00d0a03	0	Off	On

Table 1

DATA ANALYSIS

The data in table 1 above shows an access control mechanism using RFID cards, a servo motor, and two LEDs to indicate the status. When the RFID system detects the first card ID (023030303639393938323 10d0a03) which represents authorized access, the servo rotates to 90°, unlocking the system and the yellow LED lights up to indicate successful access while the red LED remains off. In contrast, when the second card ID (023030303537363332373 00d0a03) is scanned which is an unauthorized card, the servo stays at 0° and the yellow LED is off but the red LED turns on to signal denial of access. This setup visually and mechanically differentiates between access-granted and access-denied states that makes it an ideal mechanism for security applications by providing a clear indicator of each status. It is great for a guarded residential area to have an ease of access to their house while keeping unauthorized people off.

RESULT

The experiment was successful in implementing an RFID authentication system that controlled a servo motor using an Arduino. When an authorised RFID card was scanned, a green LED illuminated, and the servo moved to a user-defined angle, while an unrecognised card triggered a red LED and kept the servo in its default position. This functionality confirmed the system's effectiveness in real-time processing of RFID data. Additionally, the structured JSON handling in the Python script allowed for easy updates to the authorised card list, demonstrating the project's flexibility. Overall, this setup highlights the practical application of RFID technology in access control systems.

DISCUSSION

This experiment demonstrated the effective integration of RFID technology with an Arduino-based system for access control. By utilising a USB-connected RFID card reader, we successfully read unique identifiers from RFID cards, allowing the system to respond in real-time. The inclusion of visual indicators, such as green and red LEDs, enhanced user interaction by clearly signalling access status. This feedback mechanism not only improved the user experience but also served as a useful debugging tool during the development process.

Additionally, the use of structured JSON data handling in the Python script facilitated better organisation and scalability, making it easier to modify the system. While the project achieved its objectives, there are opportunities for improvement, such as optimising USB communication for faster response times and exploring more advanced security measures. Overall, this experiment highlights the potential of RFID technology in automation and security applications, providing a solid foundation for future enhancements in microcontroller-based systems.

CONCLUSION

In conclusion, this experiment successfully demonstrated the integration of RFID technology with an Arduino-based system to create an effective authentication mechanism for controlling a servo motor. By utilising a USB-connected RFID card reader, we were able to read card UIDs and distinguish between authorised and unauthorised cards. The visual feedback provided by the LEDs enhanced the user experience, clearly indicating access status. The use of structured JSON data handling improved code organisation and flexibility, allowing for easy modifications to the servo motor's angle settings. Overall, this project showcases the potential of combining RFID and microcontrollers in access control systems, paving the way for future applications in automation and security.

RECOMMENDATIONS

We recommend the use of hash RFID card data for a more secure authentication system. Using encrypted or hashed RFID card data we can reduce the risk of unauthorised access by making it harder to replicate card IDs.

REFERENCES

- RFID READER RFID-IDR-232N USER'S MANUAL

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: Abstract, Result, Discussion, Conclusion

Signature: *syabab* Read [/]
Name: SYABAB ALHAQQI BIN JAAFAR Understand [/]
Matric Number: 2211117 Agree [/]
Contribution: Methodology, References

Signature: *naim* Read [/]
Name: WAN NAIMULLAH BIN MOHD AZRUDDIN Understand [/]
Matric Number: 2214837 Agree [/]
Contribution: Data Analysis, Recommendation

Signature: *zaim* Read [/]
Name: ZAIMUDIN ZAKWAN BIN NORZAMRI Understand [/]
Matric Number: 2217513 Agree [/]
Contribution: Experimental Setup, Data Collection

Signature: *zamir* Read [/]
Name: ZAMIR MUTTAQIN BIN MUHAMMAD BALYAN Understand [/]
Matric Number: 2212985 Agree [/]
Contribution: Introduction, Materials and Equipment