INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

Garden of Knowledge and Virtue

REPORT ON MECHATRONICS SYSTEM INTEGRATION

4B- SERIAL COMMUNICATION RFID READER
GROUP 7

SECTION 2, SEMESTER 1, 24/25

TEAM MEMBERS

NAME	MATRIC NO.
MUHAMMAD ZAMIR FIKRI BIN MOHD ZAMRI	2212515
MUHD AKMAL HAKIM BIN SAIFUDDIN	2216093
NUR SHADATUL BALQISH BINTI SAHRUNIZAM	2212064
NORHEZRY HAKIMIE BIN NOOR FAHMY	2110061
NUR AMIRA NAZIRA BINTI MOHD NASIR	2110026

DATE OF EXPERIMENT: 30 OCTOBER 2024
DATE OF SUBMISSION:6 NOVEMBER 2024

ABSTRACT

The experiment explores how a microcontroller and computer-based system may integrate USB and Serial interfaces to accomplish servo motor control and RFID card identification. The experiment demonstrates real-time identification and actuation using an Arduino microcontroller, a USB-connected RFID reader, and Python programming. The method determines if an RFID card is authorized after reading it. When access is allowed to authorized cards, a command is transmitted from Python to the Arduino, which causes a servo motor to rotate; when access is denied, the servo stays in its default position. By showing how USB HID devices may be efficiently controlled with microcontroller-based systems for automation and security applications, the configuration offers an advantageous way of safe access control.

TABLE OF CONTENT

INTRODUCTION

The purpose of this project is to apply Python and an Arduino microcontroller to operate a servo motor and authenticate RFID cards by connecting an RFID card reader to a computer via USB. When trying to build systems that integrate sensors and actuators for automation and security applications, it is necessary to explore the Serial and USB connection between a microcontroller and a system powered by a computer.

MATERIALS AND EQUIPMENT

- Arduino board
- RFID card reader with USB connectivity
- RFID tags or cards that can be used for authentication
- Servo Motor: A standard servo motor to control the angle
- Jumper wires
- Breadboard
- LEDs of various colors
- USB cables to connect the Arduino board and the RFID reader to your computer.
- Computer with Arduino IDE and Python installed
- Datasheets and Manuals: Make sure you have the datasheets or manuals for the RFID reader, servo motor, and any other components you are using. Most of them can be downloaded from the internet. Before starting the experiment, carefully read the documentation for each component and understand the electrical and mechanical requirements. Also, consider safety protocols and guidelines to ensure a safe working environment in the lab.
- 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.
- Mounting Hardware (for the servo): If you want to mount the servo in a specific orientation or location, you might need screws, brackets, or other mounting hardware

EXPERIMENTAL SETUP

1. Attach the servo's power wire (commonly red) to the Arduino’s 5V output.

2. Connect the servo's ground wire (typically brown or black) to a ground (GND) pin on the Arduino.
3. Link the servo’s signal wire (usually orange or yellow) to one of the Arduino's PWM pins, such as pin 9.
4. Make sure there is a shared ground connection between the Arduino and the servo motor to complete the circuit.

METHODOLOGY

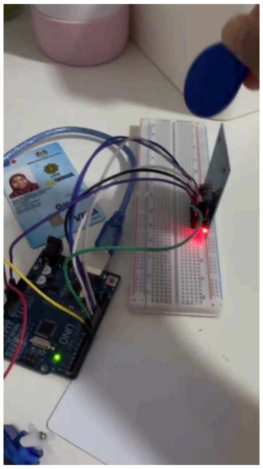
1. Attach the servo motor to the Arduino.
2. Connect the RFID reader to the computer using a USB cable.
3. Ensure the Arduino and the servo motor share a common ground connection.
4. Open the Arduino IDE and upload the program code.
5. Install the `pyusb` library.
6. Create a Python script for handling RFID authentication and controlling the servo motor, and save it with a `.py` extension.
7. Execute the Python script. When an RFID card is presented, the script will read the card's unique identifier (UID).
8. If the UID is authorized, the script will signal the Arduino to move the servo motor. If unauthorized, the servo will remain in its default position.

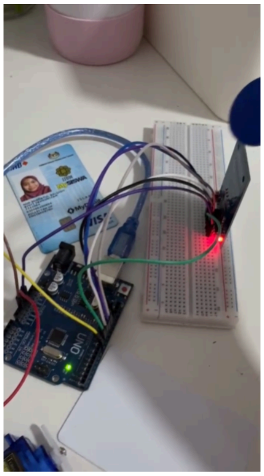
DATA COLLECTION

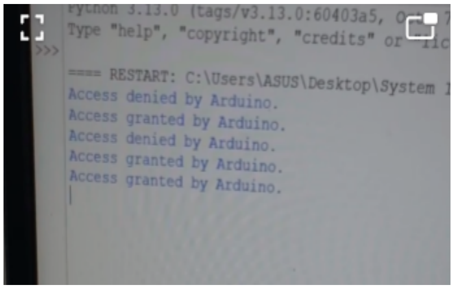
Card	Servo	Access
Authorized	Move	Granted
Unauthorized	Does not move	Denied

RESULT

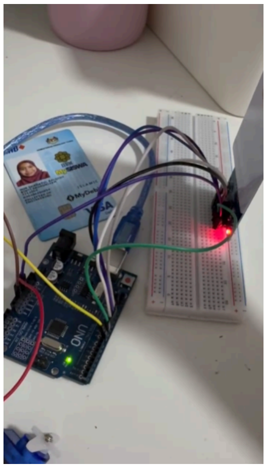
The result of the experiment shows us that authorized cards are granted access by arduino when the card is tapped to RFID. It leads to the movement of servo. The unauthorized card cannot move the servo because its access is denied by the arduino. Python scripted informed us the access status of the card that they receive from arduino code.

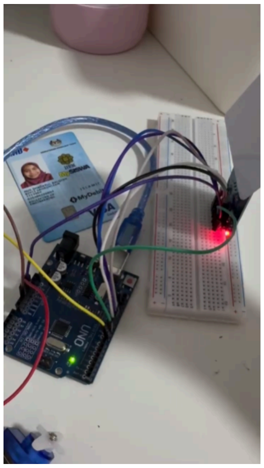


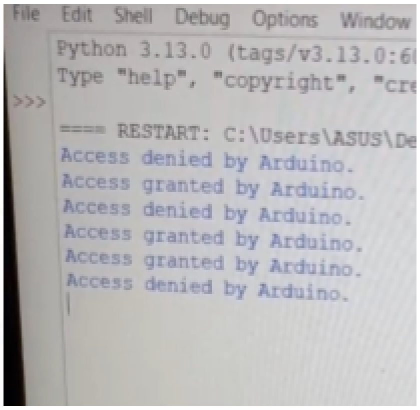




```
Python 3.13.0 (tags/v3.13.0:60403a5, Oct 7
Type "help", "copyright", "credits" or "lic
>>>
==== RESTART: C:\Users\ASUS\Desktop\System I
Access denied by Arduino.
Access granted by Arduino.
Access denied by Arduino.
Access granted by Arduino.
Access granted by Arduino.
```







```
Python 3.13.0 (tags/v3.13.0:6
Type "help", "copyright", "cre
>>>
==== RESTART: C:\Users\ASUS\De
Access denied by Arduino.
Access granted by Arduino.
Access denied by Arduino.
Access granted by Arduino.
Access granted by Arduino.
Access denied by Arduino.
```

DISCUSSION

1. Interpretation of Results
   - RFID Tag Scanning:
     1) The Arduino continuously checks for new RFID tags. When a tag is presented, it reads the UID (Unique Identifier) of the tag and compares it against an authorized list (authorizedUID).
     2) If the UID matches, the system sends a signal to the servo motor (to unlock or open a door, for example) and transmits an 'A' (for authorized) to the serial connection.
     3) If the UID does not match, it sends a 'D' (for denied).

   - Python Output:
     1) The Python script listens for data from the Arduino via a serial connection. It interprets the received characters:
        'A': The Python script prints "Access granted by Arduino."
        'D': The Python script prints "Access denied by Arduino."

   This feedback loop confirms whether the tag was recognized as authorized or not.

2. Implications of Results
   - Security Functionality:
     1) The successful identification of authorized and unauthorized tags suggests that the system can effectively manage access control. This has significant implications for security applications, such as securing entry points to buildings, restricted areas, or sensitive information.
     2) The servo motor's action (e.g., unlocking a door) is an effective way to provide physical access based on RFID verification.

3. Discrepancies
    - UID Mismatch: Ensure that the actual UID of the RFID card matches the one defined in the authorizedUID array.
    - Servo Movement: The servo does not behave as expected (e.g., it may not reach 180 degrees if it's physically restricted).
    - Delay Timing: The delay(2000) at the end of the loop may cause it to miss multiple card reads in quick succession.

4. Sources of Error and Limitations of Experiment
    - Hardware Limitations
        1) RFID Reader Range: The RFID reader has a limited range, which result in inconsistent reads if the tag is not positioned correctly.
        2) Servo Limitations: The servo has mechanical restrictions or insufficient torque, preventing it from moving to the desired angles.
        3) Connection Issues: Loose or faulty connections lead to intermittent failures in communication between the Arduino and the RFID reader or the servo.
    - Environmental Factors
        1) Interference: Electromagnetic interference from other devices affect the performance of the RFID reader.
        2) Distance and Alignment: The distance and angle at which the RFID tag is presented to the reader can significantly impact the read success rate. If the card is not held properly, it might not be detected.
    - Software Limitations
        1) UID Comparison Logic: If there are leading/trailing spaces or case sensitivity issues in UID comparison, it may lead to false rejections. Ensure that the UID is formatted consistently.
        2) Error Handling: The current code lacks robust error handling for unexpected scenarios (e.g., no card present, read failures), which can cause the system to behave unpredictably.
        3) Limited Feedback: There may be a lack of feedback to the user about the status of the system (e.g., whether a card was not detected at all).
    -

RECOMMENDATION

Recommendations for future Iterations of the experiment

1) Clear Instructions on Python and Arduino Integration
    a) Improve the clarity of instructions on how Python code interfaces with the Arduino, particularly for controlling the servo motor based on RFID authentication.
2) Circuit Diagram and Hardware Setup Guidance
    a) Include a detailed circuit diagram for connecting the RFID reader, Arduino, servo motor, and LEDs. Clearly label the connections and explain any power requirements, especially if an external power source is used.
3) Servo Motor Power Considerations
    a) Specify when an external power supply might be necessary for the servo motor, as the Arduino’s onboard power supply may not suffice for certain servo requirements.
4) Introduce Feedback Mechanism with LEDs
    a) Use LEDs to give students visual feedback on the state of the system
5) Error Handling and Troubleshooting Guide
    a) Include a section on common errors, such as connectivity issues between the RFID reader and computer or failed communication between Python and Arduino. Explain solutions or ways to troubleshoot these issues.

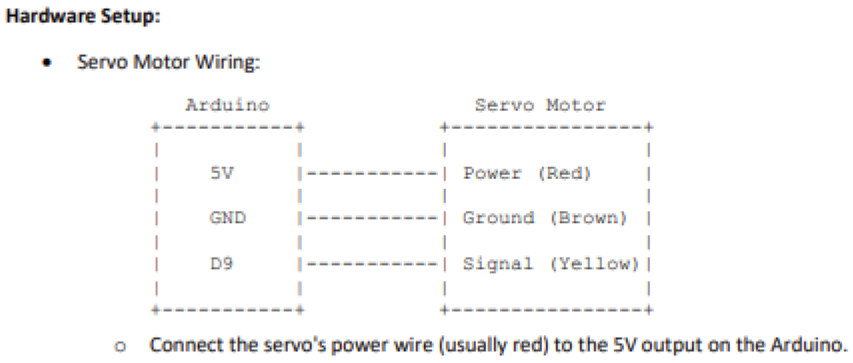
Insights and lessons learned for future students

1) Power Management is Key for Stability
    a) Managing power supply to the servo motor is essential, as underpowered or overpowered setups can cause malfunctions. Learning about voltage requirements and possible surge protection is helpful for future projects.
2) Error Handling is Essential
    a) Building in error handling for connections and data transmission is crucial. Troubleshooting failed communication between Python and Arduino will help students develop patience and systematic debugging skills.
3) Practical Applications of RFID and Servors
    a) Building in error handling for connections and data transmission is crucial. Troubleshooting failed communication between Python and Arduino will help students develop patience and systematic debugging skills.
4) Importance of Feedback Systems
    a) LEDs as indicators provide immediate feedback on the system's state, making it easier to follow the experiment's progress. This experience teaches the

importance of adding feedback mechanisms in real-world applications for better user experience and easier troubleshooting.

APPENDICES

1) Circuit Diagram



Hardware Setup:

- Servo Motor Wiring:

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

  - Connect the servo's power wire (usually red) to the 5V output on the Arduino.

   a)

2) Code Snippets
   a) From Arduino:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

#define SS_PIN    10
#define RST_PIN   9
#define SERVO_PIN 8

MFRC522 rfid(SS_PIN, RST_PIN);
Servo servo;

byte authorizedUID[4] = {0x33, 0xEF, 0xD6, 0x2C}; // Update with actual
authorized UID

void setup() {
  Serial.begin(9600);
  SPI.begin();          // Initialize SPI bus
  rfid.PCD_Init();      // Initialize RFID reader
  servo.attach(SERVO_PIN); // Attach the servo to pin 9
  servo.write(90);       // Start servo at 90°
  Serial.println("Tap RFID/NFC Tag on reader");
}

void loop() {
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    bool isAuthorized = (rfid.uid.uidByte[0] == authorizedUID[0] &&
                         rfid.uid.uidByte[1] == authorizedUID[1] &&
                         rfid.uid.uidByte[2] == authorizedUID[2] &&
                         rfid.uid.uidByte[3] == authorizedUID[3]);

    if (isAuthorized) {
      Serial.write('A');  // Send 'A' for authorized tag
      servo.write(180);   // Move servo to 180°
      delay(1000);        // Optional delay for servo action
      servo.write(90);    // Return servo to 90°
    } else {
      Serial.write('D');  // Send 'D' for unauthorized tag
    }

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
    delay(2000); // Optional delay for stability
  }
}
```

b) From Python:

```
import serial
import time

# Initialize Arduino serial connection
ser = serial.Serial('COM4', 9600)  # Adjust 'COM4' to your port
time.sleep(2)  # Allow time for connection

# Loop to check for RFID data
while True:
    try:
        # Check if data is available from Arduino
        if ser.in_waiting > 0:
            data = ser.read(1)  # Read a byte from the serial port
            card_id = data.decode('utf-8')

            # Check if the card is authorized
            if card_id == 'A':
                print("Access granted by Arduino.")
            elif card_id == 'D':
                print("Access denied by Arduino.")

    except serial.SerialException as e:
        print(f"Serial error: {e}")
        break
```

3) Troubleshooting Tips and Error Codes
   a) The RFID reader not initializing or connecting via USB
   b) Power related issues with the servo motor
   c) Python library installation issues for USB HID communication
   d) Sample error codes

REFERENCES

1. Arduino Get Started. (n.d.). *Arduino RFID NFC Servo Motor*. Retrieved November 5, 2024, from https://arduinogetstarted.com/tutorials/arduino-rfid-nfc-servo-motor

2. Arduino Get Started. (n.d.). *Arduino Servo Motor*. Retrieved November 5, 2024, from https://arduinogetstarted.com/tutorials/arduino-servo-motor

STUDENT DECLARATION

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 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.**

Name : Muhammad Zamir Fikri Bin Mohd Zamri	**Read**	/
Matric No. : 2212515	**Understand**	/
Signatures :	**Agree**	/

Name : Muhd Akmal Hakim Bin Saifuddin	**Read**	/
Matric No. : 2216093	**Understand**	/
Signatures :akmal	**Agree**	/

Name : Nur Shadatul Balqish Binti Sahrunizam	**Read**	/
Matric No. : 2212064	**Understand**	/
Signatures : shadatul	**Agree**	/

Name :NORHEZRY HAKIMIE BIN NOOR FAHMY	**Read**	/
Matric No. :2110061	**Understand**	/
Signatures :hezry	**Agree**	/

Name : Nur Amira Nazira Binti Mohd Nasir	**Read**	/
Matric No. :2110026	**Understand**	/
Signatures :amira	**Agree**	/