



MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 1 2024/2025

WEEK 3B: SERIAL COMMUNICATION

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, 23 October 2024

Date of Submission: Wednesday, 30 October 2024

ABSTRACT

This experiment explores serial communication between a Python script and an Arduino to control a servo motor's angle. By sending angle data from Python to the Arduino, the servo motor adjusts accordingly, allowing for remote, programmable control. The setup aims to demonstrate fundamental concepts in microcontroller interfacing and serial communication, utilising simple input-output (I/O) operations and providing a foundation for more complex sensor-actuator systems.

TABLE OF CONTENTS

NO	CONTENT	PAGE
1	INTRODUCTION	4
2	MATERIALS AND EQUIPMENTS	5
3	EXPERIMENTAL SETUP	5
4	METHODOLOGY	6
5	DATA COLLECTION	8
6	DATA ANALYSIS	8
7	RESULTS	9
8	DISCUSSION	9
9	CONCLUSION	9
10	RECOMMENDATIONS	10
11	REFERENCES	10
12	ACKNOWLEDGEMENT	10
13	DECLARATION	11

INTRODUCTION

The purpose of this experiment is to demonstrate serial communication between an Arduino and a computer using Python to control a servo motor's position. The experiment introduces students to basic serial communication principles, allowing real-time angle input from a Python script to the Arduino, which actuates a servo motor. This interaction offers a practical understanding of controlling hardware through software commands, a foundational skill in mechatronics and embedded systems. The hypothesis is that the Arduino will accurately receive angle data from Python and adjust the servo to the specified angle, demonstrating successful serial communication and control.

MATERIALS AND EQUIPMENTS

1. ARDUINO MEGA 2560
2. SERVO MOTOR
3. JUMPER WIRES

EXPERIMENTAL SETUP

1. Connect the potentiometer to the Arduino with one leg to 5V, the other to GND, and the middle (wiper) to the analog input pin A0.
2. Connect the servo motor to the Arduino with the red one to the 5V, the orange one to pin D9, and the brown one to the GND.
3. Connect the Arduino to the computer via USB, as shown in Figure 1, and run the Python script to begin reading potentiometer values in the terminal.
4. Adjust the potentiometer knob to observe the values displayed in real-time.
5. To view real-time data on the Arduino Serial Plotter, close the Python script.
6. Open the Serial Plotter from “Tools” and “Serial Plotter” in the Arduino IDE.
7. Ensure the correct COM port and baud rate (9600) are selected.
8. Observe and customise the data graph as the potentiometer is turned.

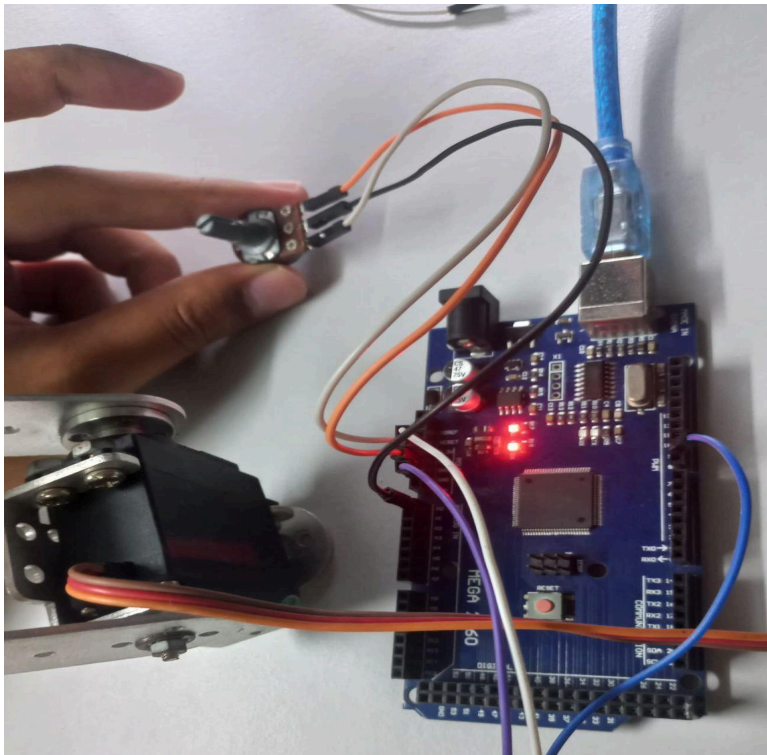


Figure 1

METHODOLOGY

1. Setup the Arduino Mega 2560
2. IDE code implementation
3. Python code implementation
4. Run the python script
5. Experiment the servo angle
6. Read the angle in python terminal
7. Code snippet

Arduino code

```
const int potPin = A0;
#include <Servo.h>
Servo servo;
int value = 0;
int angle = 0;

void setup() {
  // put your setup code here, to run once:
  servo.attach(9);
  pinMode(9, OUTPUT);
  pinMode(potPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  value = analogRead(potPin);
  angle = map(value, 0, 1023, 0, 180);
  Serial.println(angle);
  servo.write(angle);
  delay(200) ;
}
```

Python code

```
import serial
import time
import keyboard # Requires 'keyboard' library for keypress detection

arduino_port = "COM7"
baud_rate = 9600
```

```

# Establish serial communication with Arduino
ser = serial.Serial(arduino_port, baud_rate)
time.sleep(2) # Allow time for connection to establish

print("Press 's' to stop the servo control.")

try:
    while True:
        # Check for user input to send stop command to Arduino
        if keyboard.is_pressed('s'):
            ser.write(b's') # Send 's' to Arduino to stop the loop
            print("Stop command sent to Arduino.")
            break

        # Read and display the servo angle (0-180 degrees) sent from Arduino
        if ser.in_waiting > 0:
            angle = ser.readline().decode().strip()
            print(f"Servo Angle: {angle}°")

        time.sleep(0.1) # Small delay to prevent high CPU usage

except KeyboardInterrupt:
    print("Process interrupted by user.")

finally:
    ser.close()
    print("Serial connection closed.")

```

DATA COLLECTION

Potentiometer Reading	Servo Angle (°)
0	0
128	23
256	45
384	68
512	90
640	113
768	135
896	158
1023	180

DATA ANALYSIS

From table 1, the servo angle varies for each potentiometer input due to the potentiometer voltage. The angle given is rounded up because the python script does not give any decimal points. Based on this data, we can make use of the precise angle of the servo that reacts to potentiometer value for any mechanism that requires precise movement. It can vary in angles from 0 to 180 degrees that makes up for most of the mechanisms that require precise control movement. For example, a robotic arm needs to do precise movements in various angles. Thus, this is a perfect applications using a servo motor and a potentiometer.

RESULT

The experiment successfully demonstrated serial communication between a Python script and Arduino, enabling accurate control of the servo motor's location using angle data received via a potentiometer. The servo motor precisely adjusted to the received angle values, showcasing effective control and integration. Real-time feedback displayed the servo angle within the Python terminal, allowing for continuous monitoring and verification of the system's accuracy and responsiveness. Observations of system reaction time were recorded, with any possible delays in communication or motor response identified as places to improve. Figures and tables would help to elucidate these findings by graphically exhibiting angle values, servo locations, and reaction times, providing a thorough summary of the data.

DISCUSSION

The results confirm that serial communication between Python and Arduino is reliable for controlling a servo motor, demonstrating the feasibility of using serial commands for real-time control. By sending angle commands from Python, the Arduino precisely actuates the servo, showing that software can effectively manage hardware operations remotely.

Minor sources of error include occasional data transmission delays and potential COM port selection issues, both of which can be minimized with proper setup and configuration. Ensuring accurate baud rate matching and error handling in both the Arduino and Python code reinforces data integrity and smooth communication. This experiment's success highlights serial communication's potential for broader applications in robotics and IoT systems.

CONCLUSION

The report that we can conclude that the experiment was successfully showed serial communication between Python and Arduino, allowing the control of the servo motor through input commands by potentiometer. This confirms the hypothesis and provides valuable insights into the principles of mechatronics system integration, emphasizing the reliability and capability of microcontroller interfacing.

RECOMMENDATIONS

1. Code Optimization: I recommend optimizing both Arduino and Python scripts for smoother serial communication and enhanced response time.
2. Additional Components: Proposes use of more advanced components for better accuracy and functionality when controlling the servo motor.

REFERENCES

Serial communication between python and arduino :
<https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756>

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*

Name: AHMAD DARWISH BIN AHMAD TERMIZI

Matric Number: 2212089

Contribution: Data Analysis

Read [/]

Understand [/]

Agree [/]

Signature: *syabab*

Name: SYABAB ALHAQQI BIN JAAFAR

Matric Number: 2211117

Contribution: Methodology, Data Collection

Read [/]

Understand [/]

Agree [/]

Signature: *naim*

Name: WAN NAIMULLAH BIN MOHD AZRUDDIN

Matric Number: 2214837

Contribution: Materials and Equipments, Reference

Read [/]

Understand [/]

Agree [/]

Signature: *zaim*

Name: ZAIMUDIN ZAKWAN BIN NORZAMRI

Matric Number: 2217513

Contribution: Result, Conclusion, Recommendation, Experimental Setup

Read [/]

Understand [/]

Agree [/]

Signature: *zamin*

Name: ZAMIR MUTTAQIN BIN MUHAMMAD BALYAN

Matric Number: 2212985

Contribution: Abstract, Introduction, Discussion

Read [/]

Understand [/]

Agree [/]