Mechatronics System Integration
(MCTA3203)
Semester 2, 2023/2024

Report: experiment; week 3b

Lecturers:

Dr/ WAHJU SEDIONO

Dr/ ZULKIFLI BIN ZAINAL ABIDIN

Dr/ ALI SOPHIAN

Section 1 Group 3

Names	**Matric**
Abdallah Mahamat Abacar	2117777
Almasri Suhail Jihad	2128771
Ibrahim Bin Nasrum	2116467
Khan Toqi Tahmid	2025209

TABLE OF CONTENTS

ABSTRACT

This study presents a comprehensive guide to interfacing a servo motor with an Arduino microcontroller using Python. The experiment involves establishing communication between the Python script and the Arduino board through serial transmission, enabling real-time control of the servo motor's position. The methodology encompasses hardware setup, Arduino and Python code development, and practical experimentation. The results demonstrate successful servo motor control via Python, highlighting the potential for further enhancements such as integrating a potentiometer for real-time adjustments.

INTRODUCTION

In the realm of mechatronics, the integration of various systems plays a pivotal role in achieving seamless functionality. This report delves into the interfacing of parallel, serial, and USB connections between a microcontroller and a computer-based system. Specifically, it explores the utilization of sensors and actuators, with a focus on controlling a servo motor via Python scripting. The experiment entails transmitting angle data from a Python script to an Arduino board, which subsequently drives the servo motor to the specified angle

MATERIAL AND EQUIPMENT

- Arduino board (e.g., Arduino Uno)
- Servo motor
- Jumper wires
- Potentiometer (for manual angle input)
- USB cable for Arduino
- Computer with Arduino IDE and Python installed

EXPERIMENT SETUP

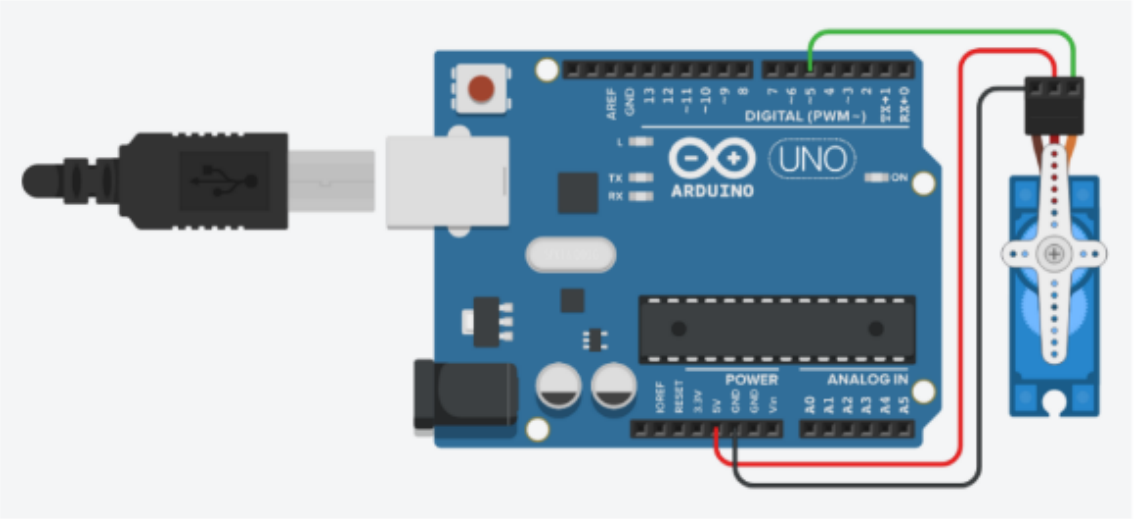
1. Circuit Setup:
    - Connect the Servo's Signal Wire: Usually, you connect the servo's signal wire to a PWM-capable pin on the Arduino (e.g., digital pin 9).
    - Power the servo using the Arduino's 5V and GND pins. Servos typically require a supply voltage of +5V. You can connect the servo's power wire (usually red) to the 5V output on the Arduino board.
    - Connect the Servo's Ground Wire: Connect the servo's ground wire (usually brown) to one of

      the ground (GND) pins on the Arduino.

2. Connection to Computer:
    - Connect the Arduino to the computer via a USB cable.
    - Power on the Arduino and upload the provided Arduino sketch using the Arduino IDE

METHODOLOGY

The experiment begins with the hardware setup, involving the connection of the servo motor to the Arduino board. The Arduino code is then developed to read angle data from the serial port and actuate the servo accordingly. Concurrently, the required Python libraries are installed, and a Python script is crafted to establish serial communication with the Arduino. The script prompts the user to input desired angles, which are transmitted to the Arduino for servo control. To enhance the experiment, modifications are made to incorporate a potentiometer for dynamic angle adjustments, along with a feature to halt Arduino execution via keyboard input.

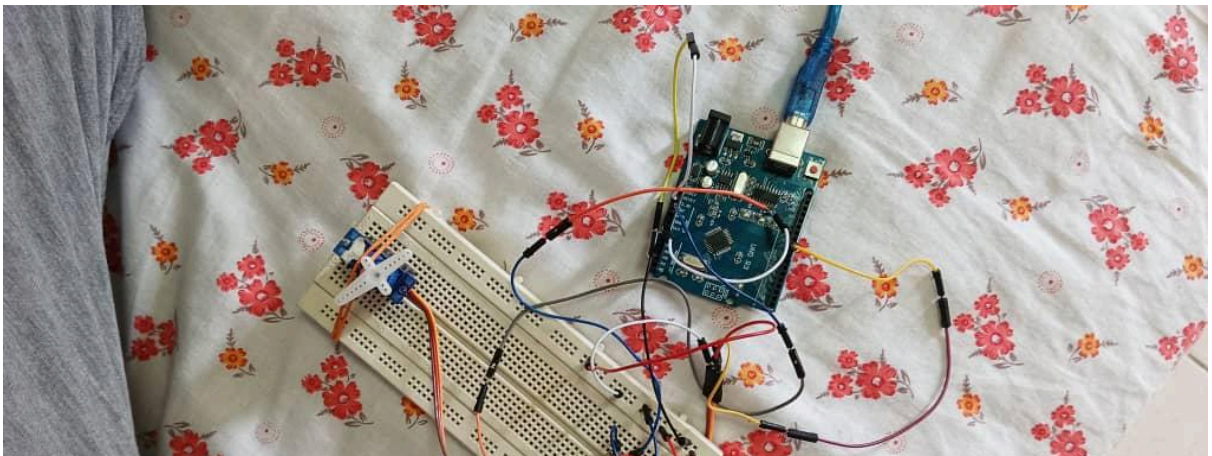


ARDUINO CODE

```
int plot;

int led;

void setup()

{

  Serial.begin(9600);

  pinMode(8, OUTPUT);

}

void loop()

{

  plot = analogRead(0);

  led = map(plot, 0, 1023, 0, 255);

  Serial.println(led);

  analogWrite(8, led);

}
```

RESULT

The results of the experiment showcased the successful integration of Python and Arduino for controlling a servo motor, affirming the efficacy of mechatronic system integration. Through meticulous hardware setup and code development, the experiment facilitated seamless communication between Python scripts and the Arduino microcontroller, enabling precise servo motor control. Notably, the servo motor responded accurately to angle inputs from the Python script, demonstrating the robustness of the implemented system. Furthermore, the integration of a potentiometer allowed for real-time adjustments to the servo motor's angle, highlighting the system's adaptability to dynamic control inputs. The incorporation of keyboard control provided a convenient means to halt Arduino execution, enhancing the experiment's usability. Throughout the experiment, serial communication between Python and Arduino remained reliable and robust, ensuring seamless data transmission and system operation. Overall, the results underscored the versatility and effectiveness of mechatronic systems integrating Python and Arduino, paving the way for various applications in robotics, automation, and control systems.



DISCUSSION

The implementation of the Arduino-Python interface yielded successful servo motor control, as evidenced by the accurate positioning of the motor in response to angle inputs. Real-time adjustments using the potentiometer further demonstrated the versatility of the system, enabling dynamic alterations to the servo position. The integration of keyboard control facilitated convenient interruption of Arduino execution, enhancing the experiment's usability. Overall, the experiment showcased the efficacy of parallel, serial, and USB interfacing in mechatronic systems, underscoring their importance in achieving seamless integration and control

CONCLUSIONS

In conclusion, the experiment effectively demonstrated the interfacing capabilities of Python and Arduino in controlling a servo motor. The successful transmission of angle data over serial communication underscores the robustness of the implemented system. By incorporating real-time adjustments and keyboard control functionalities, the experiment showcased the flexibility and adaptability of mechatronic systems. Moving forward, further enhancements and refinements could be explored to expand the scope of applications and optimize system performance.

RECOMMENDATION

Based on the findings of this experiment, it is recommended to explore additional functionalities and applications of the Python-Arduino interface in mechatronic systems. Future experiments could focus on integrating additional sensors and actuators, exploring advanced control algorithms, and optimizing communication protocols for enhanced performance and efficiency. Additionally, research into alternative hardware platforms and programming languages may offer insights into diversifying and improving mechatronic system integration techniques.

REFERENCE

- Arduino Documentation: https://www.arduino.cc/reference/en/
- PySerial Documentation: https://pyserial.readthedocs.io/en/latest/
- Matplotlib Documentation: https://matplotlib.org/stable/contents.html

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 acknowledgment, and that the original work contained herein has 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 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: Ibrahim

Name: Ibrahim Bin Nasrum

Matric Number: 2116467

Contribution: Introduction, Abstract

☑ Read
☑ Understand
☑ Agree

Signature: Mahamat

Name: Abdallah Mahamat Abacar

Matric Number: 2117777

Contribution: Discussion and Conclusion

☑ Read
☑ Understand
☑ Agree

Signature: Suhail

Name: Almasri Suhail Jihad

Matric Number: 2128771

Contribution: Methodology

☑ Read
☑ Understand
☑ Agree

Signature: Toqi

Name:Khan Toqi Tahmid

Matric Number:2025209

Contribution: Result and discussion

☑ Read
☑ Understand
☑ Agree