LABORATORY PROJECT REPORT
SERIAL COMMUNICATION (POTENTIOMETER) 3A

MCTA 3203

SEMESTER 2, 2024/2025

Date of submission : 23 March 2025
Group : 10
Section : 1

NO.	NAME	MATRIC NO.
1.	MUHAMMAD FAIZ BIN MD RIZAM	2229695
2.		
3.		

<u>**Table of Contents**</u>

Abstract 2
Introduction 2
Materials and Equipment 3
Experimental Setup 3
Figure 1 3
Photograph of the arduino setup 4
Circuit setup 4
Methodology 5
Result 6
Discussion 7
Conclusion 8
Recommendation 9
Acknowledgment 9
References 10
Student’s Declaration 10

Abstract

This project explores serial communication between an Arduino microcontroller and a computer using Python. The primary objective is to read potentiometer values from an Arduino and transmit the data via a USB connection to a Python script for real-time monitoring and visualization. The experiment involves setting up a hardware circuit with a potentiometer connected to the Arduino, writing an Arduino sketch to send sensor readings over the serial port, and developing a Python script using the pyserial library to receive and display the data. Additionally, the use of the Arduino Serial Plotter and Python's matplotlib library for graphical representation of the data is examined. This project demonstrates the fundamental concepts of microcontroller-based data acquisition and serial communication, providing a foundation for more advanced sensor integration and control applications.

Introduction

The objective of this experiment is to transfer data between a computer and a microcontroller using a potentiometer and an LED. By send the readings from the potentiometer connected to an arduino to a python script via a USB connection, we aim to learn how data is exchange with code uploaded to the microcontroller

Materials and Equipment

- Arduino Uno Board
- Potentiometer
- Jumper Wires
- LED
- 220 ohm resistor
- Breadboard

Experimental Setup

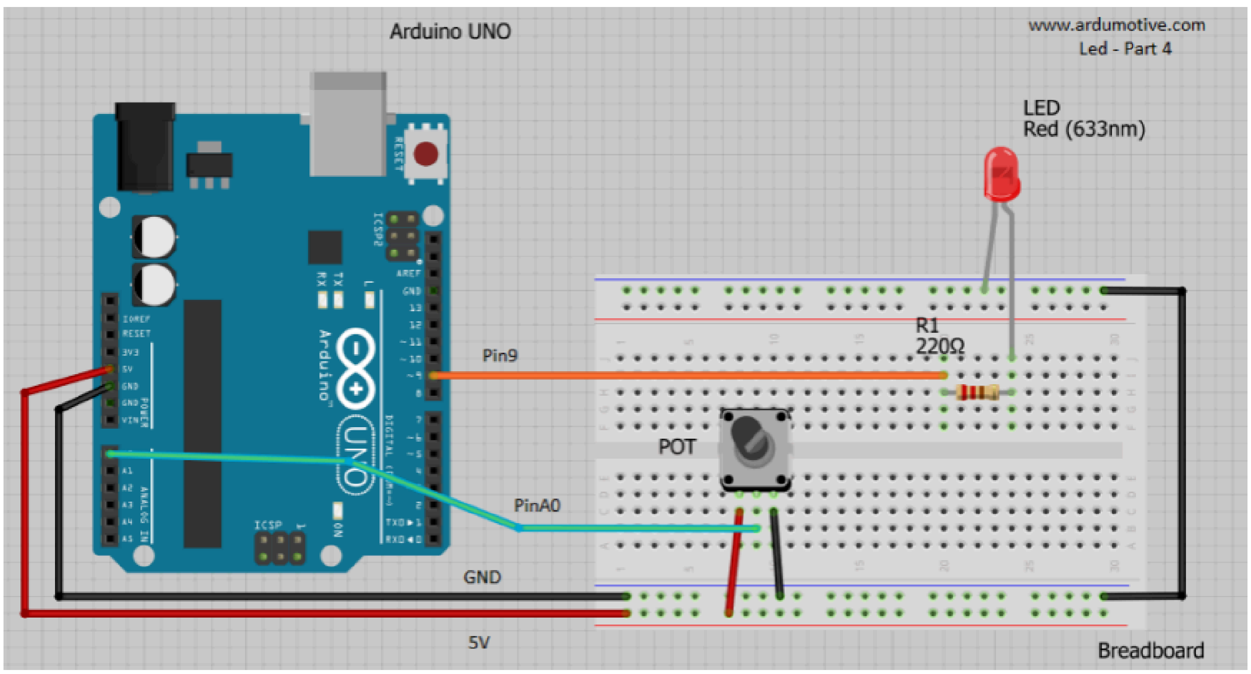
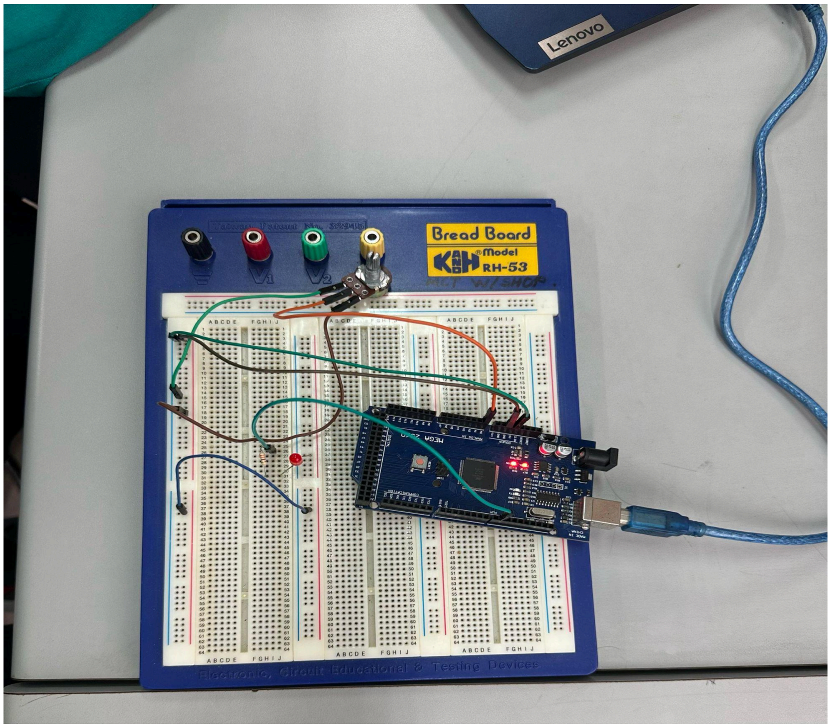


Figure 1



Photograph of the arduino setup

Circuit setup

1. Connect one leg of the potentiometer to 5V on the Arduino
2. Connect the other leg of potentiometer to GND on the Arduino
3. Connect the middle leg (wiper) of the potentiometer to an analog input pin on the Arduino, such as A0. An example of the circuit setup is shown in figure 1

Methodology

1. Arduino board setup
   - Connect the potentiometer to the Arduino. One wire of the potentiometer connect to 5V, the other one to GND and the centre wire to an analog input on the Arduino A0
   - Connect the LED to Arduino, the negative part to GND and the positive part to the resistor that connect to pin 9
   - Connect the Arduino to computer via USB cable

2. Write a program in Arduino IDE to :
   - Define the pins that connected to the potentiometer and LED
   - Initialize serial communication at pin 9 to read the potentiometer values and send it over the serial port
   - Include a potentiometer function to control the LED using a potentiometer
   - Upload the Arduino sketch to the Arduino UNO

3. Arduino Execution
   - Control the LED brightness using the potentiometer
   - Open Serial Plotter in Arduino IDE to monitor the data received from Arduino board using 9600 baud rate
   - Record the graph of the Serial Plotter

4. Python execution
   - Write a program using python to read the potentiometer data from the Arduino via the serial plot
   - Turn the potentiometer knob and record the values

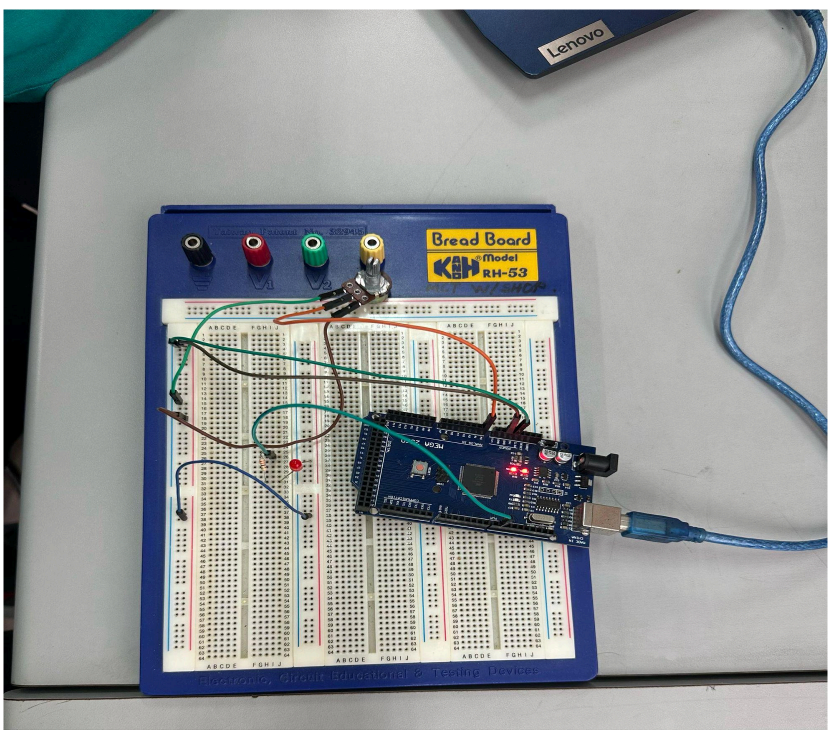
Result

In this experiment, we built a simple circuit to control the brightness of an LED using a potentiometer. The potentiometer acts as a variable resistor, allowing us to adjust the voltage and, in turn, the brightness of the LED. When the potentiometer is turned, it reads a value that is used to determine how bright the LED should be. This value is then divided by four to scale it appropriately before being sent as an input to the function that controls the LED's brightness.

To monitor and debug the potentiometer's readings, we printed the values to the serial monitor, allowing us to observe changes in real time. This helped verify that the potentiometer was functioning correctly and that the brightness adjustments were responding as expected.

Overall, the experiment successfully demonstrated how a potentiometer can be used to regulate the brightness of an LED, showcasing the basics of analog input and pulse-width modulation (PWM) in microcontroller programming.

The following diagrams illustrate the connections of the LED and potentiometer circuit and the Serial Plotter output from the Arduino IDE.



Discussion

In this experiment, several key hardware components were used to build and test the system. The foundation of the setup was a breadboard, which made it easy to connect and rearrange various electronic components. It provided a flexible way to wire the circuit while working with the Arduino Mega 2560. As the central microcontroller, the Arduino Mega 2560 played a crucial role in processing data and controlling the system. With its multiple input and output pins, it could handle complex tasks efficiently.

To link the components, male-to-male jumper wires were used to create secure electrical connections between the breadboard and the Arduino. One of the main components in the circuit was a potentiometer, a variable resistor with three terminals. By turning its knob, users could adjust the resistance, which affected the voltage output. This changing voltage was then used to control the brightness of an LED—an indicator light that glows when current flows through it. To prevent excessive current from damaging the LED, a 220-ohm resistor was included in the circuit.

The potentiometer was wired to the Arduino in a straightforward way: one terminal connected to the 5V power supply, another to the ground (GND), and the middle terminal (wiper) was linked to the analog input pin A0. This setup created a voltage divider circuit, generating a variable voltage signal based on the potentiometer's position. The Arduino's built-in analog-to-digital converter (ADC) measured this voltage and converted it into a digital value between 0 and 1023 using the 'analogRead ( )' function.

To monitor the potentiometer's readings in real-time, the Arduino communicated with a computer using serial communication via USB. A Python script on the computer received the data and displayed it on the screen. For smooth communication, both the Arduino and the Python script were set to a baud rate of 9600 to avoid data transmission errors.

The Arduino code followed a simple workflow: it initialized the serial connection, read the potentiometer's value, and adjusted the LED's brightness accordingly. The 'analogRead (Pot_pin)' function collected the potentiometer data, while the 'analogWrite (Led_pin, brightness)' function used Pulse Width Modulation (PWM) to control the LED's brightness.

Meanwhile, the Python script continuously received and displayed the potentiometer values, ensuring clear and accurate real-time monitoring. With this setup, the system worked efficiently, allowing users to observe changes in brightness based on the potentiometer's position.

Conclusion

In summary, these experiments helped us understand how a computer and a microcontroller communicate using Python and Arduino through serial communication. By working with a potentiometer, we explored key concepts related to sensors and actuators in embedded systems.

In this experiment, we set up a potentiometer to send analog readings to a Python script via a USB connection. This process gave us hands-on experience with microcontroller programming, data transfer protocols, and the crossover between electronics and computer science. It also allowed us to monitor real-time data, showing how easily analog inputs can be integrated into software applications for tasks like data logging or control systems.

Recommendation

To make the experiment more reliable and effective, a few improvements can be made. First, always double-check the wiring before turning on the circuit. Incorrect connections can cause malfunctions or even damage the components.

Second, using a multimeter to measure voltage at different points in the circuit can help identify issues like power supply problems or incorrect resistor values. It's also crucial to ensure the resistors have the right values—using the wrong ones could result in a dim LED display or even damage the components. The correct resistor will prevent the LED from burning out.

Once the basic setup is working, testing different scenarios can help catch potential issues early. By following these steps, the experiment will run more smoothly, be more effective, and provide a better learning experience.

Acknowledgment

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.

References

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

Student's 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 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 : Muhammad Faiz bin Md Rizam

Matric Numbers : 2229695

Signature :

Read	Understand	Agree
/	/	/