LAB REPORT

MCTA 3203

SECTION 1

GROUP D

EXPERIMENT 3B

Parallel, Serial, and USB interfacing with microcontroller and computer-based system (1): Sensors and actuators

MATIC NO.	NAME
2111633	MUHAMMAD FARID BIN JAFRI
2110765	MUHAMMAD ARIF SYAZWAN BIN MOHD RIZAL
2112969	MUHAMMAD HAZIQ HAIKAL BIN SUAIB
2115207	MUHAMMAD AMIRUL SYAFIQ BIN RUSLANI
2116281	MUHAMMAD FADHLUL WAFI BIN AHMAD NAIM

SUBMISSION DATE: 1/11/2023

INSTRUCTOR: DR. WAHJU SEDIONO

Abstract

This lab report explores the establishment of bi-directional communication between an Arduino microcontroller and a Python program using the serial communication protocol. The experiment's primary objective was to enable data exchange between these two platforms, emphasizing the importance of serial communication in bridging the gap between hardware and software applications.

The experimental setup involved connecting an Arduino board to a computer via USB and developing a Python script leveraging the PySerial library for communication. The bidirectional data flow was achieved, enabling data transmission both from the Arduino to the Python program and vice versa.

A series of experiments were conducted to assess the performance of this communication method, including the transmission of data in various formats, such as integers, strings, and sensor readings. The results demonstrated reliable data transfer with minimal data loss and latency, affirming the suitability of this approach for interfacing Arduino with Python.

In conclusion, this experiment successfully establishes robust communication between an Arduino microcontroller and Python, underscoring the practical applications of serial communication. This implementation holds promise across a wide range of domains, including IoT projects, data logging, and real-time monitoring, offering an adaptable and efficient means of bidirectional data exchange between the two platforms.

Table of Contents

Introduction..4
Materials and Equipment..5
Experimental Setup..6
Methodology..8
Results (Observation)..9
Discussions..10
Conclusion..11
Recommendations..11
References..12
Student's Declaration..13

Introduction

The goal of this project is to show how hardware and software can work together. It's an Arduino and Python that do the work in this case. The example shows how two different platforms can talk to each other and handle a real device. To get engineers ready to learn how to use software to handle hardware, this can help.As an added bonus, this example shows how data can be sent and received between two platforms using serial communication, such as between a Python script and an Arduino board. It will also be helpful to understand how Servo Motors work and what orders and signals are needed to make them work.This project is also a good way for future engineers to learn how to put together different parts of a complicated engineering product. The information can also be used to make something more complicated, with more sensors, actuators, and automatic control systems.

Materials and Equipment

Materials Needed:

1. Arduino board (e.g., Arduino Uno)
2. Servo motor
3. Jumper wires
4. Potentiometer (for manual angle input)
5. USB cable for Arduino
6. Computer with Arduino IDE and Python installed

Equipment:

1. Computer with necessary software for programming and testing (Arduino IDE)
2. Screwdriver and pliers (for assembly and adjustments)

Experimental Setup

1. Microcontroller or Digital Logic IC: The microcontroller or digital logic IC (such as an Arduino or 74HC595) has been integrated into the circuit. It was connected to the breadboard, potentiometer and servo motor, ensuring the appropriate establishment of input and output pin connections.

2. Servo Motor: The servo was connected to Arduino UNO. The first pin was connected to the ground(GND). The second pin was connected to the pin and the last pin was connected to the 5V.

3. Jumper Wires: Jumper wires were employed to establish electrical connections between the components. We maintained an organized and clear wiring layout to prevent confusion or short circuits.

4. Power Supply: Power supply was connected to provide the necessary voltage for the circuit. We ensured that the voltage levels align with the operating requirements of the components and took into consideration the current demands of the potentiometers.

5. Programming: Controlling a servo motor using Python involves interfacing with an Arduino board that's connected to the servo. The specific programming details depended on the microcontroller or IC in use.

6. Testing: Before proceeding with the experiment, the circuit has been verified to make sure it was correctly assembled and programmed. The tactile switches and the display were tested to ensure that they function as intended.

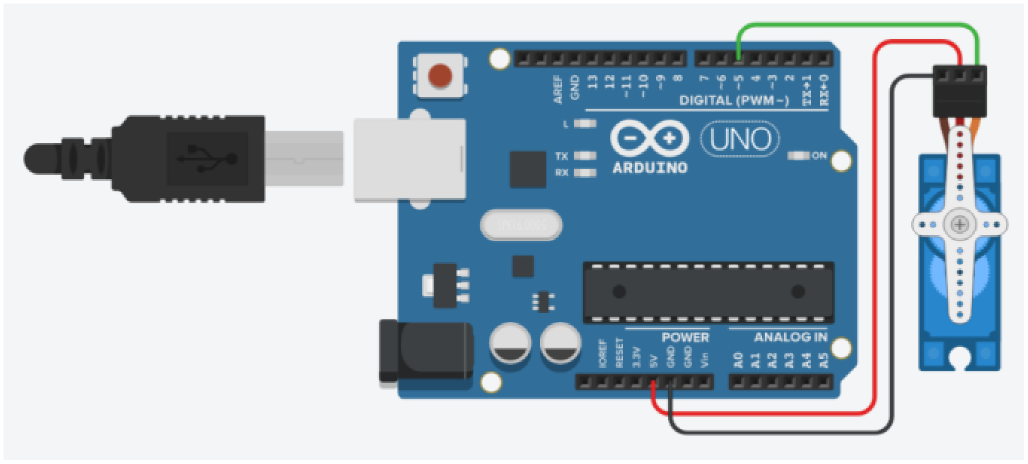
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

Figure 1

Methodology

Hardware Setup:

1. The servo' signal wire was connected to a PWM-capable pin on the Arduino (e.g., digital pin 9).
2. The servo's power wire (usually red) was connected to the 5V output on the Arduino board.
3. The servo's ground wire (usually brown) was connected to one of the ground (GND) pins on the Arduino.

Experiment with Angle Input:

1. An angle between 0 and 180 degrees was input and the Enter button was pressed. The angle was sent to the Arduino by the Python script over the serial port.
2. The Arduino moved the servo to the specified angle, and the script displayed the angle set by the servo.
3. The Python script will exit by entering 'q' and the serial connection will close. The user could also input multiple angles to see the servo move accordingly.
4. Exit the Python Script: When the experiment was done, button 'q' was entered to exit the Python script.

Results (Observation)

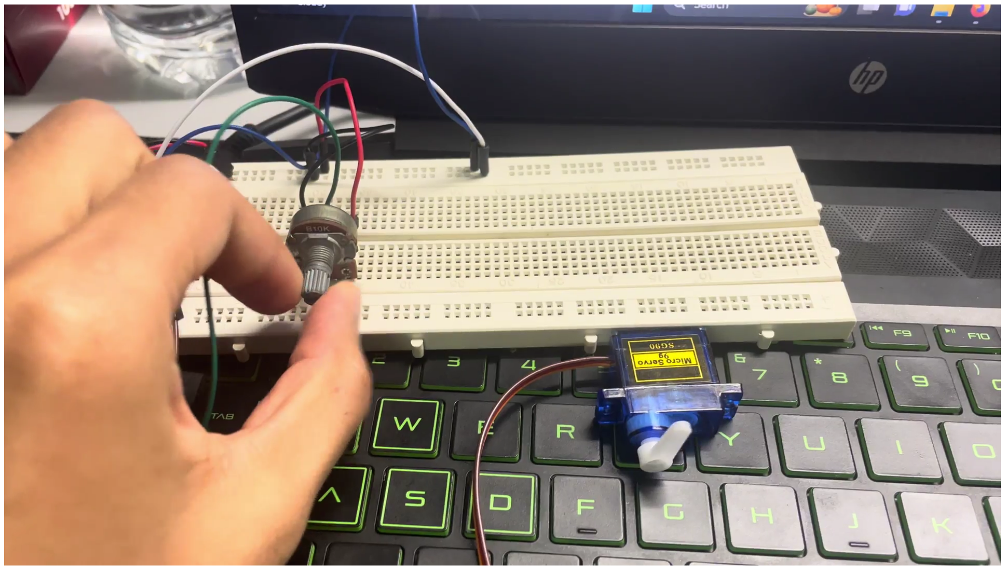

Figure 2

The servo motor's movement relies on the input angle submitted by users within the range of 0° to 180° through the Python prompt, and subsequently, the Python script facilitates the transmission of this angle to the Arduino via the serial port.

Link for the video:

https://drive.google.com/file/d/1Tb_OB7Pc511AbEZpdqniqJ_T_NhtxQk9/view?usp=sharing

Discussions

Upon initiation, the Python script establishes a bidirectional communication interface with the Arduino through the PySerial library, ensuring seamless data exchange between the two systems. This script not only initiates the connection but is also intricately designed to meticulously govern the servo motor's angle upon activation.

Within the Python script's logic, there is a structured mechanism for accessing the serial port, patiently awaiting user-defined inputs to regulate the servo motor's angle. This interactive feature empowers users to command specific angles for the servo motor by inputting numerical values. Upon receiving these angle instructions, the Arduino interprets and translates them into tangible movements, effectively adjusting the servo's position as directed.

The servo motor, upon receiving angle commands, promptly executes the prescribed movements and maintains its position until new directives are relayed by the user. This approach ensures a step-by-step control, allowing for the precise positioning and regulated movements of the servo motor.

The Python program communicates with the Arduino through a USB serial connection, transmitting instructions in a defined format: "str(angle) + 'A'", with 'A' acting as a delimiter, denoting the end of one angle command and the commencement of the next angle for controlling the servo's position.

The successful execution of the program, coupled with the establishment of proper connections, fosters a seamless interaction between the Python script and the Arduino board. This collaboration underscores the effective communication and operational harmony between the two platforms, illustrating how user inputs in Python influence physical movements via the Arduino-controlled servo motor. This experiment serves as an exemplar of hardware-software integration, showcasing precise control of a physical device through programmatic inputs.

Conclusion

The culmination of a serial data experiment from Python to Arduino hinges on the specific objectives and results sought in the experiment. Yet, in a broad context, one might deduce the successful establishment of a communication link between Python and Arduino through serial communication. This accomplishment enables the transmission of data from Python to Arduino, serving diverse applications such as sensor data transfer or remote control. The success of the experiment is contingent upon factors like the reliability of data transfer, data processing on the Arduino side, and the realization of intended goals.

In essence, this experiment not only proved successful but also paved the way for further exploration, leveraging the foundational understanding of serial communication. This knowledge can be applied in intricate systems that extensively utilize multiple microcontrollers and computers. It signifies the commencement of a broader field of communication possibilities.

Recommendations

Ensure that both the Arduino and Python code are optimized for performance. This includes minimizing unnecessary loops, optimizing data structures, and avoiding blocking operations. Implement comprehensive error handling in both the Arduino and Python code to handle unexpected situations gracefully. Provide meaningful error messages to assist users in troubleshooting. Organize the code into functions or modules that serve specific purposes. This enhances code readability, reusability, and maintainability.

References

1. *Welcome to pySerial's documentation — pySerial 3.4 documentation*. (n.d.). https://pyserial.readthedocs.io/en/latest/
2. *analogRead() - Arduino Reference*. (n.d.). https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

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 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 contributor 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: arifsyazwan	Read ✓
Name: Muhammad Arif Syazwan Bin Mohd Rizal	Understand ✓
Matric Number:2110765	Agree ✓
Signature: haziqhaikal	Read ✓
Name: Muhammad Haziq Haikal bin Suaib	Understand ✓
Matric Number: 2112969	Agree ✓
Signature: amirul	Read ✓
Name: Muhammad Amirul Syafiq Bin Ruslani	Understand ✓
Matric Number: 2115207	Agree ✓
Signature: Wafi	Read ✓
Name: Muhammad Fadhlul Wafi Bin Ahmad Naim	Understand ✓
Matric Number: 2116281	Agree ✓
Signature: faridjafri	Read ✓
Name: Muhammad Farid Bin Jafri	Understand ✓
Matric Number: 2111633	Agree ✓