International Center for Free and Open Source Software

SERVO MOTOR CONTROL USING ARDUINO

Azwa Harshad

Internship

Open IoT

FEB 2023

Contents

List of Figures

Chapter 1

Introduction

1.1 General Background

Servo motors are widely used in various applications for their precise control of angular position, making them suitable for tasks such as robotics, automation, and model control. By interfacing the servo motor with an Arduino board, we can leverage the board's computational power and flexibility to precisely manipulate the motor's position and speed.

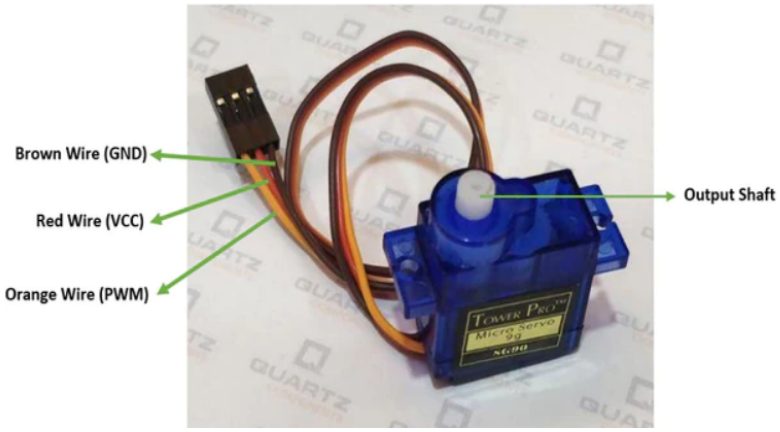

Figure 1.1: Servo motor

1.2 Working Principle

A servo motor operates on a closed-loop control system, where its precise positioning is achieved through continuous feedback between its actual position and the desired

position. Inside the servo motor, a DC motor, gear train, and a feedback device, typically a potentiometer or encoder, work together. The control circuit interprets incoming PWM signals, comparing the desired position with the feedback from the potentiometer or encoder. Based on this comparison, the control circuit generates an error signal that drives the DC motor in the appropriate direction to minimize the error. This feedback mechanism ensures accurate and stable positioning of the servo motor's output shaft, making it suitable for various applications requiring precise control over angular position.

1.3 Components Needed

1. Arduino Board (e.g., Arduino Uno)
2. Servo Motor
    - Operating Voltage: 3V to 7.2V
    - Stall torque @4.8V: 1.2 kg-cm
    - Stall torque @6.6V: 1.6 kg-cm
    - Gear Type Plastic
    - Weight of Motor 9gms
3. LED
4. Breadboard
5. Jumper Wires

Chapter 2

Circuit Diagram

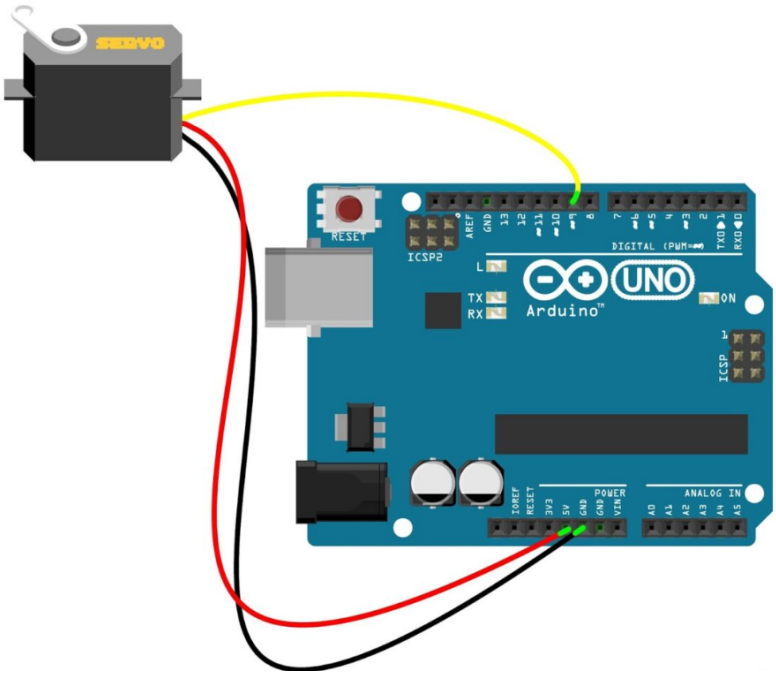

Figure 2.1: Circuit Diagram

2.1 Circuit Connections

1. Red wire (Power): Connect to Arduino's 5V pin.
2. Brown or Black wire (Ground): Connect to Arduino's GND pin.
3. Yellow or Orange wire (Control): Connect to any digital pin on the Arduino (e.g., Pin 9).

2.2 Block Diagram

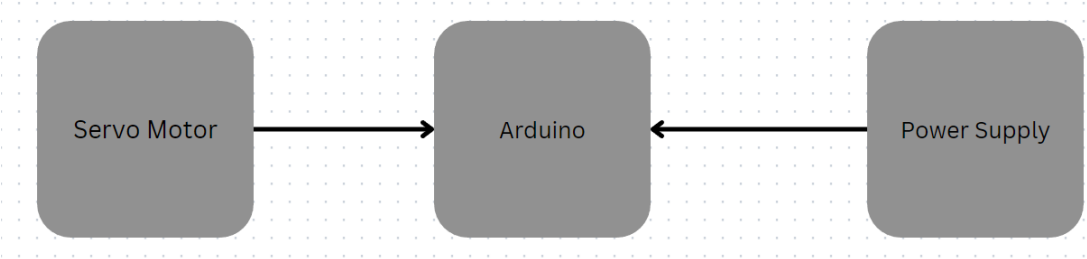

Figure 2.2: Block Diagram

Chapter 3

Implementation

3.1 Procedure

1. Setup Hardware: Gather all necessary components. Connect them according to the circuit diagram.

2. Install Arduino IDE: Download and install Arduino IDE from the official website.

3. Open Arduino IDE: Launch the Arduino IDE software.

4. Write Code: Compose your program using Arduino programming language (based on C/C++).
   Write setup and loop functions.

5. Verify Code: Click on the Verify button (checkmark icon) to check for any errors in the code.

6. Upload Code: Connect Arduino board to the computer via USB. Select the correct board and port from Tools menu.
   Click on the Upload button (right arrow icon) to upload the code to the Arduino board.

7. Test: Make sure the hardware is powered on.

Chapter 4

Code

4.1 Code for control of servo motor using arduinio

It is done through ArduinoIDE

Code

```
#include <Servo.h>
int pin = 9;
Servo servo; // analog pin used to connect the potentiometer
int val=0;    // variable to read the value from the analog pin
void setup() {
  servo.attach(pin);  // attaches the servo on pin 9 to the servo object
}
void loop() {
  for(val=0;val<180;val++);{
    servo.write(val);
    delay(15);
  }
  for(val=180;val>0;val--);{
     servo.write(val);
     delay(15);
  }
}
```

Chapter 5

Result

5.1 Result

This demonstrates successful control of the servo motor using the Arduino microcontroller. Through the implemented circuit connection and programmed code, precise manipulation of the servo motor's position was achieved. By sending PWM signals from the Arduino to the servo motor, we were able to command it to move to specific angular positions within its range of motion. The servo motor responded accurately to the commands, smoothly transitioning between positions without any observable jitter or deviation. This result confirms the functionality and effectiveness of the servo motor control system implemented using the Arduino platform.