

08a Servo Motors Intro

Nicholas Bruzzese

January 5, 2025

Understanding PWM and Servo Control

Pulse Width Modulation (PWM) is a technique where the width of a pulse is varied while keeping the frequency constant. This method is essential for controlling servo motors, as the duration of the pulse determines the position of the servo's shaft.

In a typical setup, a 50Hz PWM signal (period of 20ms) is used:

- **0° Position:** A 1ms pulse width (5% duty cycle) positions the servo at 0 degrees.
- **90° Position:** A 1.5ms pulse width (7.5% duty cycle) positions the servo at 90 degrees.
- **180° Position:** A 2ms pulse width (10% duty cycle) positions the servo at 180 degrees.

By adjusting the pulse width within this range, you can set the servo to any desired angle between 0° and 180°.

Setting Up Your Raspberry Pi with a Servo Motor

Hardware Connections

Servo Motor Wires:

- **Red (VCC):** Connect to the 3.3V pin on the Raspberry Pi.
- **Brown/Black (GND):** Connect to a ground (GND) pin on the Raspberry Pi.
- **Yellow/Orange (Signal):** Connect to a GPIO pin (e.g., GPIO17, pin 11).

Optional: For added safety, place a $\sim 1\text{k}\Omega$ resistor between the signal wire and the GPIO pin.

Note: If the servo doesn't operate correctly, it might be due to insufficient power. In such cases, consider using an external power source (4–6V) for the servo.

Software Configuration

Install the RPi.GPIO Library:

```
sudo apt-get update
sudo apt-get install python3-rpi.gpio
```

Wiring Diagram

This is a simple setup:

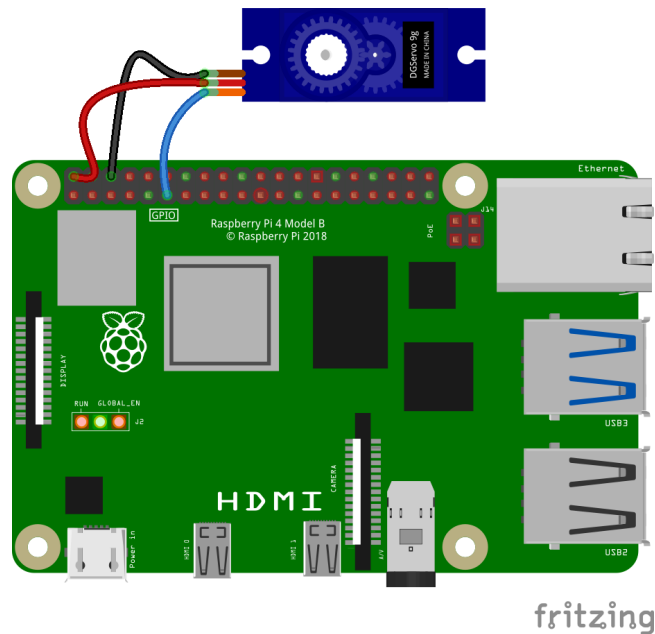


Figure 1: Wiring Diagram

Python Script to Control the Servo:

Create a Python script (e.g., `servomotor.py`) with the following content:

```
import RPi.GPIO as GPIO
import time

servoPIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
p.start(2.5) # Initialization

try:
    while True:
        p.ChangeDutyCycle(5) # Move to 0 degrees
        time.sleep(0.5)
        p.ChangeDutyCycle(7.5) # Move to 90 degrees
        time.sleep(0.5)
        p.ChangeDutyCycle(10) # Move to 180 degrees
        time.sleep(0.5)
        p.ChangeDutyCycle(7.5) # Move back to 90 degrees
        time.sleep(0.5)
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

Running the Script:

Execute the script using Python:

```
python3 servomotor.py
```

This script will cycle the servo motor through 0°, 90°, and 180° positions. Adjust the `ChangeDutyCycle` values to set specific angles as needed.