Servo Motor Control Using Raspberry Pi for Precision Motion

Introduction:

Servo motors are widely used in robotics, automation, and control systems due to their ability to provide precise motion control. A Raspberry Pi can efficiently control a servo motor using Pulse Width Modulation (PWM). This manual provides an easy guide on how to connect and operate a servo motor with a Raspberry Pi for accurate and smooth movement.

Components Required:

- Raspberry Pi (any model with GPIO support)
- Servo Motor (e.g., SG90 or MG995)
- External Power Supply (if required)
- Jumper Wires
- Breadboard (optional)

Circuit Connection:

1. Servo Motor Pins:
    - Connect the VCC (Red wire) to 5V on Raspberry Pi (or external power supply if needed).
    - Connect the GND (Black/Brown wire) to GND on Raspberry Pi.
    - Connect the Signal (Yellow/Orange wire) to a PWM-supported GPIO pin on Raspberry Pi.
2. If an external power source is used, ensure the ground of the power source and Raspberry Pi are connected together.
3. Use software PWM to control the servo position by adjusting the duty cycle.

Applications:

- Robotics: Controlling robotic arms and legs.
- Automation Systems: Precision control in industrial machines.
- Cameras & Surveillance: Positioning camera gimbals for tracking.
- RC Projects: Steering and throttle control in remote-controlled vehicles.

Learnings:

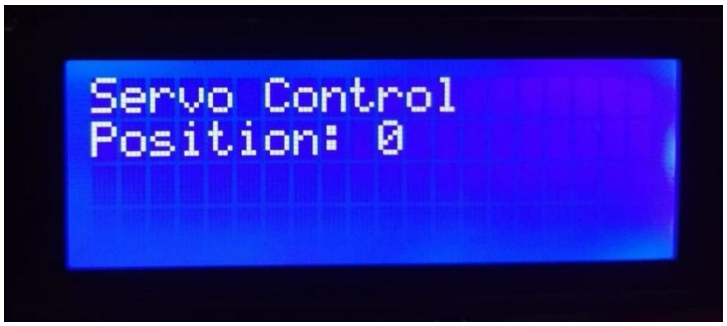
- Understanding how servo motors work and their applications.
- Learning to use PWM for motion control.
- Gaining experience in interfacing hardware with Raspberry Pi.

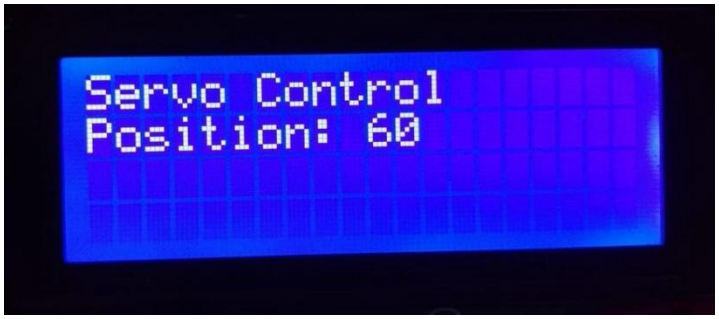
- Developing problem-solving skills for troubleshooting circuit connections.

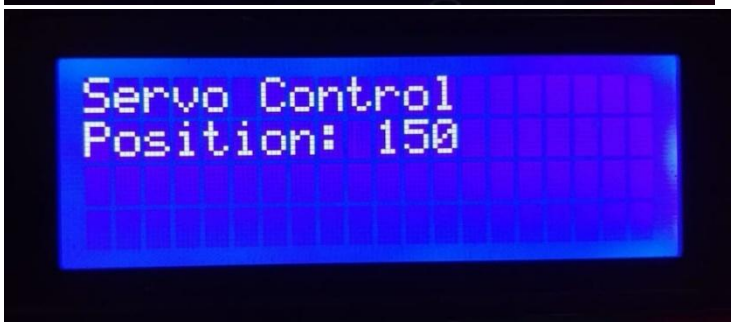
Conclusion:

Controlling a servo motor with Raspberry Pi is a fundamental yet essential task in automation and robotics. This project helps in understanding precise motion control, which is useful for various applications like robotics and industrial automation. With further advancements, multiple servos can be controlled simultaneously for complex movements and tasks.

Outcome:







Program:

```
import RPi.GPIO as GPIO
import time
from RPLCD.i2c import CharLCD

# Setup GPIO for Servo
SERVO_PIN = 18  # PWM Pin
GPIO.setmode(GPIO.BCM)
GPIO.setup(SERVO_PIN, GPIO.OUT)

# Setup PWM for Servo (50Hz)
pwm = GPIO.PWM(SERVO_PIN, 50)
```

```
pwm.start(0)  # Start with 0 duty cycle

# Initialize LCD
lcd = CharLCD('PCF8574', 0x27)

def set_servo_angle(angle):
    """Convert angle (0-180) to duty cycle and move servo."""
    duty = (angle / 18) + 2  # Convert angle to duty cycle
    GPIO.output(SERVO_PIN, True)
    pwm.ChangeDutyCycle(duty)
    time.sleep(0.5)  # Give time to move
    GPIO.output(SERVO_PIN, False)
    pwm.ChangeDutyCycle(0)

    # Display the position on LCD
    lcd.clear()
    lcd.cursor_pos = (0, 0)
    lcd.write_string("Servo Control")
    lcd.cursor_pos = (1, 0)
    lcd.write_string(f"Position: {angle}°")

try:
    while True:
        for angle in range(0, 181, 30):  # Move from 0° to 180° in steps
            set_servo_angle(angle)
            time.sleep(1)

        for angle in range(180, -1, -30):  # Move back from 180° to 0°
            set_servo_angle(angle)
            time.sleep(1)
```

```
except KeyboardInterrupt:
    pwm.stop()
    GPIO.cleanup()
    lcd.clear()
    print("Program Stopped")
```