Lesson 7 Servo

Introduction

In this project, we will learn about Servo Motors, Raspberry Pi Servo Motor Interface and How to Control a Servo Motor using Raspberry Pi and Python. For this project, we will be using the Raspberry Pi 3 and Tower Pro SG90 Servo Motor.

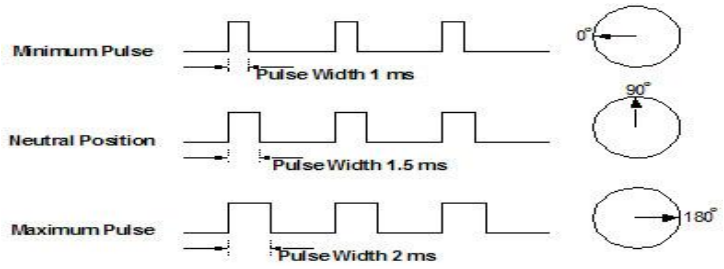
Principle

A Servo Motor is a simple device that consists of a DC Motor, Gears and a Feed Back based Position Control System. The main advantage of a Servo Motor is its ability to hold the angular position of its shaft.

There are several varieties of Servo Motors you can choose from. Two of the most common Servo Motors are Tower Pro SG90 and Tower Pro MG90S.

SG90 is a plastic gear motor whereas MG90S is a metal gear motor. In this project, I'll be using the SG90 Servo Motor.

The Tower Pro SG90 Servo Motor Consists of three Pins: PWM (Orange or Yellow), V_{CC} (Red) and GND (Brown). The V_{CC} and GND pins must be connected to +5V and GND of the power supply.



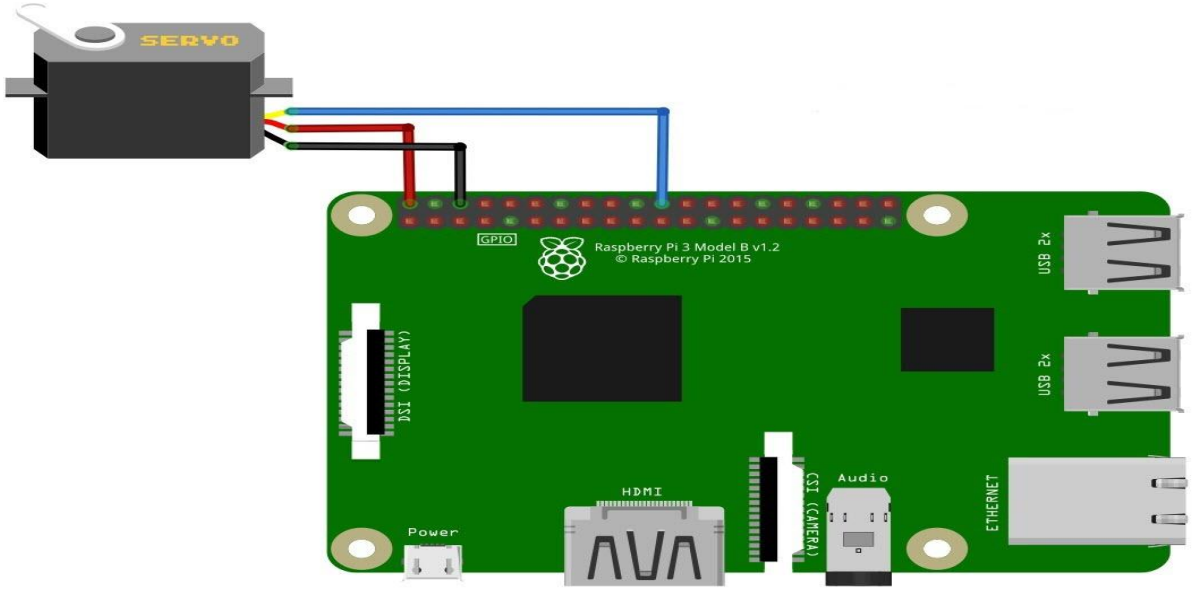
PWM or Signal Pin of the Servo Motor must be connected to the PWM Output of the Controller (Raspberry Pi). Based on the width of the Pulses from the PWM signal, the angle of the Servo Motor's shaft will vary.

Hardware Required

- Raspberry Pi
- Tower Pro SG90 Servo Motor (any Servo Motor can be used)
- Connecting Wires
- Power Supply
- Computer

Hardware Setup

Connect the V_{CC} and GND of the Tower Pro SG90 Servo Motor to +5V and GND pins of the power supply. Then connect the PWM Pin of the Servo Motor to Physical Pin 22 of Raspberry Pi i.e. GPIO25.



Python Coding

```
import RPi.GPIO as IO   # calling for header file for GPIO's of PI
import time             # calling for time to provide delays in program
IO.setwarnings(False)   # do not show any warnings
IO.setmode (IO.BCM)  # programming the GPIO by BCM pin numbers.(like PIN29 as'GPIO5')
IO.setup(25,IO.OUT)     # initialize GPIO19 as an output
p = IO.PWM(25,50)       # GPIO19 as PWM output, with 50Hz frequency
p.start(7.5)            # generate PWM signal with 7.5% duty cycle
while 1:                # execute loop forever
   p.ChangeDutyCycle(7.5)# change duty cycle for getting the servo position to 90º
   time.sleep(1)         # sleep for 1 second
   p.ChangeDutyCycle(12.5) # change duty cycle for getting the servo position to 180º
   time.sleep(1)             # sleep for 1 second
   p.ChangeDutyCycle(2.5)  # change duty cycle for getting the servo position to 0º
   time.sleep(1)        # sleep for 1 second
```

Output

we are going to use the PWM feature of the Raspberry Pi. As mentioned, earlies, based on the Duty Cycle of the PWM Signal from the Raspberry Pi, the position of the Servo Motor will vary. Since, 5% Duty Cycle of the PWM signal corresponds to extreme left position and 10% Duty Cycle corresponds to extreme right position, we need to vary the Duty Cycle between 5 and 10% to get a sweeping effect from the Servo Motor.
If you observe in the code, the duty cycle gradually rises from 5 to 10% with an increment of 0.5% in every step. Once it reaches the 10% mark, the reverse action will begin.

Applications

Interfacing a Servo Motor with Raspberry Pi and controlling the angle of rotation of the Servo can be helpful in several applications like:

- RC Car
- RC Plane
- Robot
- Quadcopter