Generate PWM for velocity control of a Servo Motor

e-Yantra Team

July 7, 2015

Contents

1 Objective

In this tutorial we shall learn to control the angle of rotation of a servo motor(using PWM) interfaced with Raspberry Pi.

2 Prerequisites

- Python programming skills

3 Hardware Requirement

1. Raspberry Pi (I will be using Version 2 Model B)
2. MCP23017
3. Power adapter
4. Servo motor (I will be using the Futuba make)
5. Connecting wires
6. Bread board

4 Software Requirement

1. PyScripter (version 2.7 or above)
2. Mobaxterm (for windows user)

5 Theory and Description

Servo motor is normally a simple DC motor which is controlled for specific angular rotation with help of additional servomechanism (a typical closed loop feedback control system).
In short Servo motor is a special type of motor which is automatically operated up to certain limit for a given command with help of error-sensing feedback to correct the performance.[1]

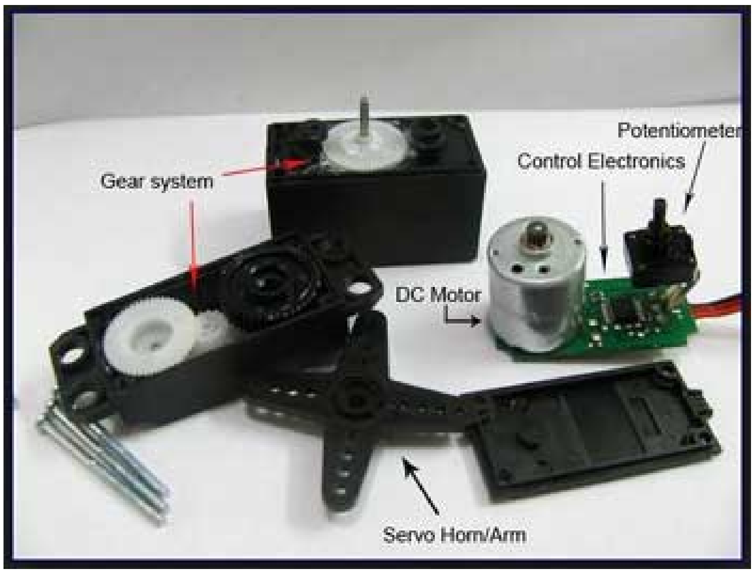

Figure 1: [1]

Angle of Rotation
The angle (of mechanical rotation) is determined by the width of an electrical pulse that is applied to the control wire. This is a form of pulse-width modulation, however servo position is not defined by the PWM duty cycle (i.e., ON vs OFF time) but only by the width of the pulse. The servo expects to see a pulse every 20 ms, however this can vary within a wide range that differs from servo to servo. The width of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position). [2] The minimal width and the maximum width of pulse that will command the servo to turn to a valid position are functions of each servo. Different brands, and even different servos of the same brand, will have different maximum and minimums. Generally the minimum pulse will be about 1 ms wide and the maximum pulse will be 2 ms wide.[3]

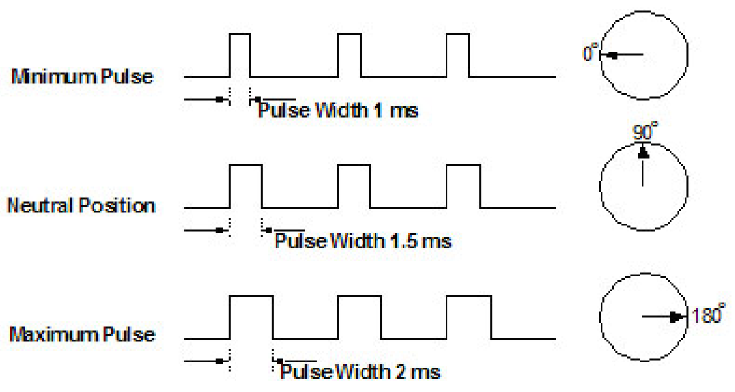

Figure 2: [3]

Another parameter that varies from servo to servo is the turn rate. This is the time it takes from the servo to change from one position to another. The worst case turning time is when the servo is holding at the minimum rotation and it is commanded to go to maximum rotation. This can take several seconds on very high torque servos.[3]

Pin Diagram

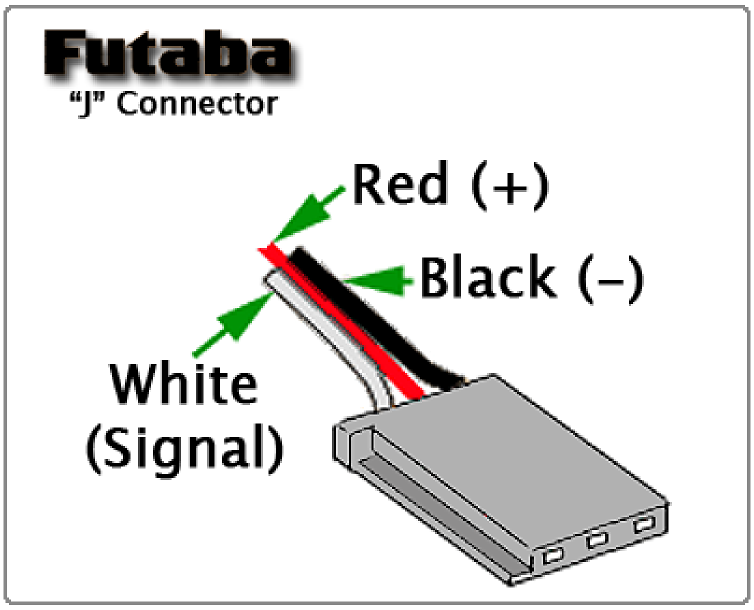

Figure 3: [3]

6 Experiment

PWM generation for controlling the angle of rotation of a Servo motor interfaced with R-Pi

In this experiment we will be programming a servo motor using software pwm generation technique.

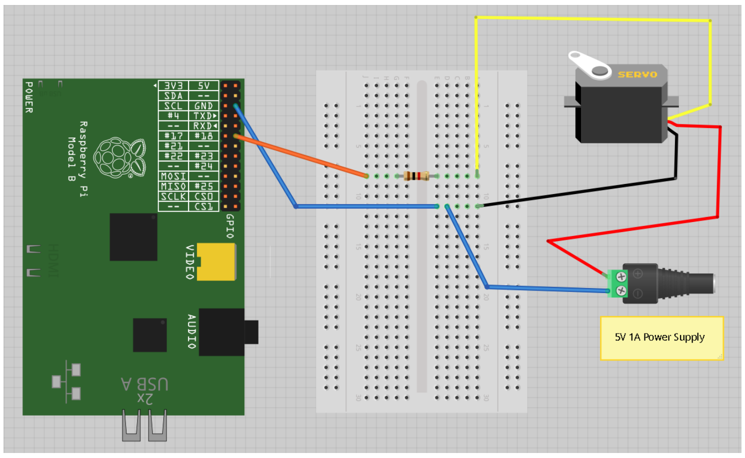

Figure 4: [4]

In this figure the enable pin of the motor is connected to GPIO 11 of R-Pi.

Code

```
import RPi.GPIO as GPIO # module to control Pi GPIO channels
import time

GPIO.setmode(GPIO.BOARD) # to use Raspberry Pi board pin numbers
i1= 11 # servo motor enable
GPIO.setup(i1,GPIO.OUT,initial=GPIO.LOW) # set up GPIO output channel
# i.e. on pin 11 and initially output is low

# Function name : rotation
# Input : Angle in degrees
# Output : On time and off time of a pwm pulse for angle = x degrees
# Example call: rotation(x)
def rotation(x):
    m = (2.2 - 0.6)/(180 - 0) # 0.6ms corresponds to 0 degree
                              # and 2.2ms corresponds to 180 degree
     on_time = 0.6 + m*x
         off_time = 20 - on_time # time period of a pwm pulse for a
                                 # servo motor is 20 ms
```

```
        return on_time, off_time


try:
    while 1:
                angle = 90 # it can range from 0 to 360 degrees
                on_time, off_time = rotation(angle)
                GPIO.output(i1, True)
            time.sleep(on_time/1000) # time in ms
                GPIO.output(i1, False)
                time.sleep(off_time/1000)


except KeyboardInterrupt:
        pass
GPIO.cleanup() # all GPIO pins are cleared
```

7 Appendix

7.1 Raspberry Pi 2 Pin-out Diagram

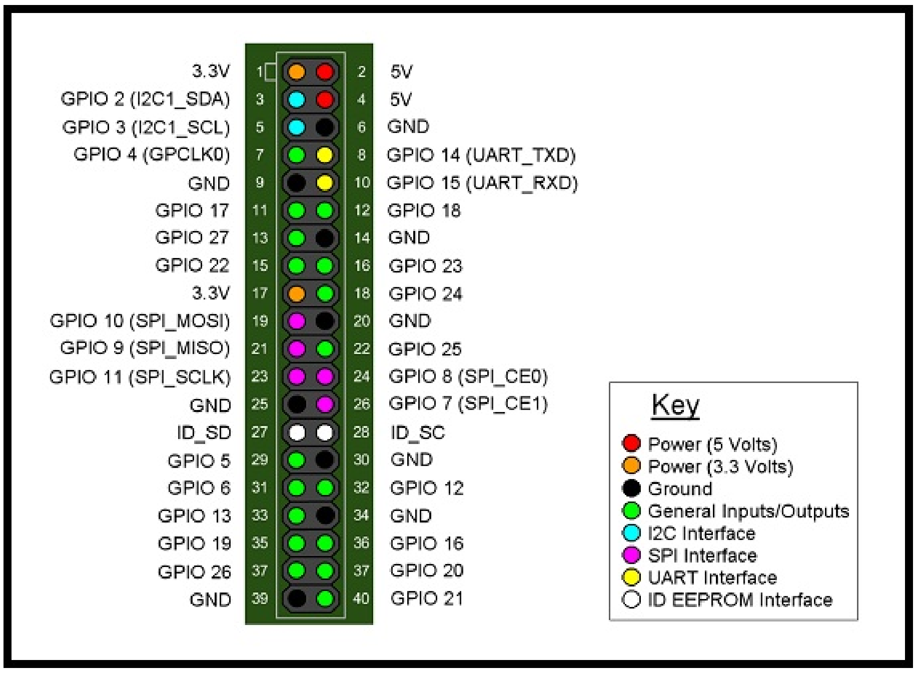

Figure 5: [4]

7.2 MCP23017 datasheet

`http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf`

7.3 PWM

In order to learn more about PWM kindly refer
`https://github.com/eyantrainternship/eYSIP_2015_`
`RaspberryPi-Development-Board/tree/master/Task%206`

8 References

1. http://www.electrical4u.com/servo-motor-servo-mechanism-theory-and-working-principle/
2. https://en.wikipedia.org/wiki/Servo_control
3. https://www.servocity.com/html/how_do_servos_work_.html#.VZsqx_mqqkp
4. http://razzpisampler.oreilly.com/images/rpck_1001.png
5. http://data.designspark.info/uploads/images/53bc258dc6c0425cb44870b50ab30621