Interfacing DC motor and servo motor using PWM Driver

e-Yantra Team

June 11, 2016

Contents

1 Objective

In this tutorial we will learn how to control speed of DC motor and servo motor using

- GPIO pins on Raspberry Pi
- PWM driver IC PCA9685.

2 Prerequisites

- Python programming skills
- Servo and dc motor basics

3 Hardware Requirement

1. Raspberry Pi (I will be using Version 2 Model B+)
2. Power adapter
3. Connecting wires
4. L293D IC
5. DC motor
6. Servo Motor
7. PCA9685 IC
8. Capacitor 0.1 uF (Two nos.)
9. Bread board

4 Software Requirement

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

5 Theory and Description

DC Motor
Direct current(DC) motor is a device which converts eletrical energy in mechanical energy. Since long time dc motors are used in variable speed drives for its versatile characteristics of providing high starting torque. High starting torque is required for traction drives. Speed of DC motor is varied by PWM technique.

Servo Motor
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]

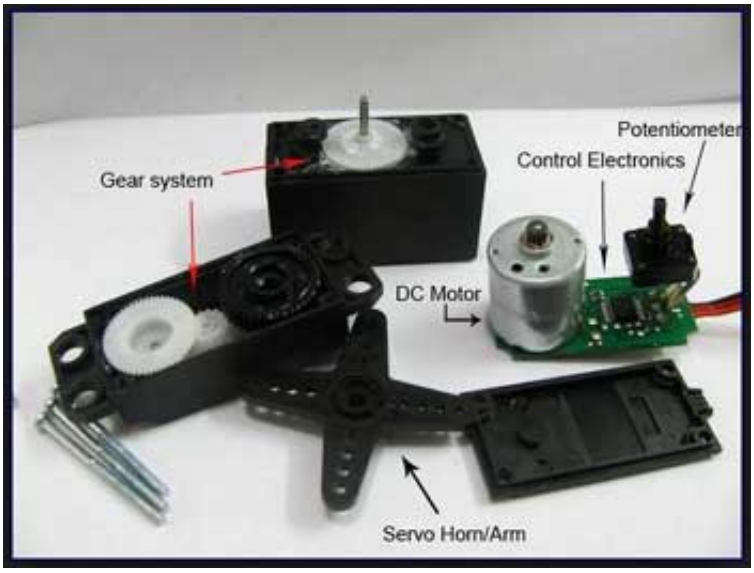


Figure 1: [1]

PWM Principle
It is a method to generate binary signals, which has two signal periods high and low. Depending upon the duration signal duration, power is delivered to the device.The term duty cycle is defined as:

$$DutyCycle = T_{ON}/(T_{ON} + T_{OFF}) \tag{1}$$

T_{ON} = On Time in one cycle
T_{OFF} = Off Time in one cycle

We use it here to control the amount of power going to the motor and hence how fast it spins.
The diagram below shows the signal from the PWM pin of the Raspberry Pi.

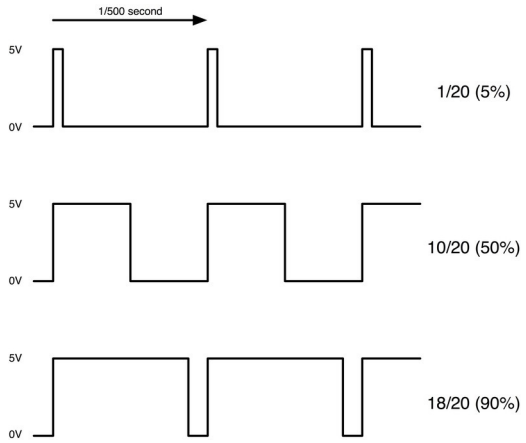


Figure 2: [1]

Every 1/500 of a second, the PWM output will produce a pulse. The length of this pulse controls the amount of energy that the motor gets. No pulse at all and the motor will not turn, a short pulse and it will turn slowly. If the pulse is active for half the time, then the motor will receive half the power it would if the pulse stayed high until the next pulse came along.

PCA9685 PWM Driver IC

The PCA9685 is an I2C-bus controlled 16-channel LED controller optimized for Red/Green/Blue/Amber (RGBA) color backlighting applications. Each LED output has its own 12-bit resolution (4096 steps) fixed frequency individual PWM controller that operates at a programmable frequency from a typical of 24 Hz to 1526 Hz with a duty cycle that is adjustable from 0% to 100% to allow the LED to be set to a specific brightness value. All outputs are set to the same PWM frequency. We have used PCA9685 PWM servo shield given below:

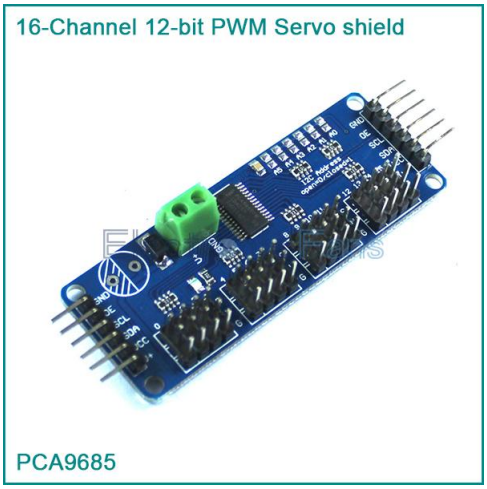


Figure 3: [1]

Pins of PCA9685

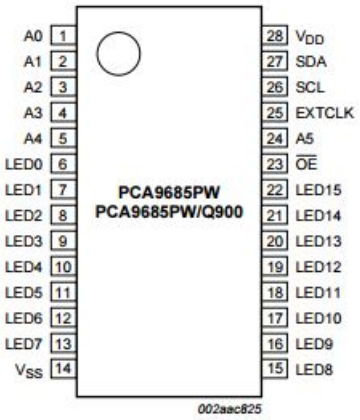


Figure 4: [1]

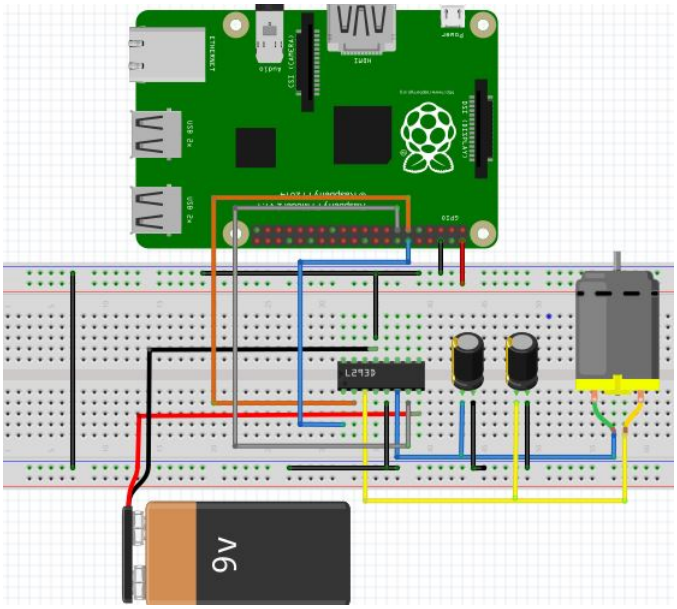
6 Experiment

6.1 Controlling the speed of dc motor through L293D using PWM from the GPIO pins of RPi

There is only one pin on RPi for PWM generation i.e. IC pin 12(GPIO 18). In this experiment we will be rotating the DC motor in both direction using L293D(Motor Driver IC) and GPIO pins of RPi.

L293D Connections

Pins	Connections
1	Rpi IC pin 12
2	RPi IC pin 11
3	motor terminal 1
4	–
5	–
6	motor terminal 2
7	Rpi IC pin 13
8	Battery
13	Battery ground
16	5V



Code

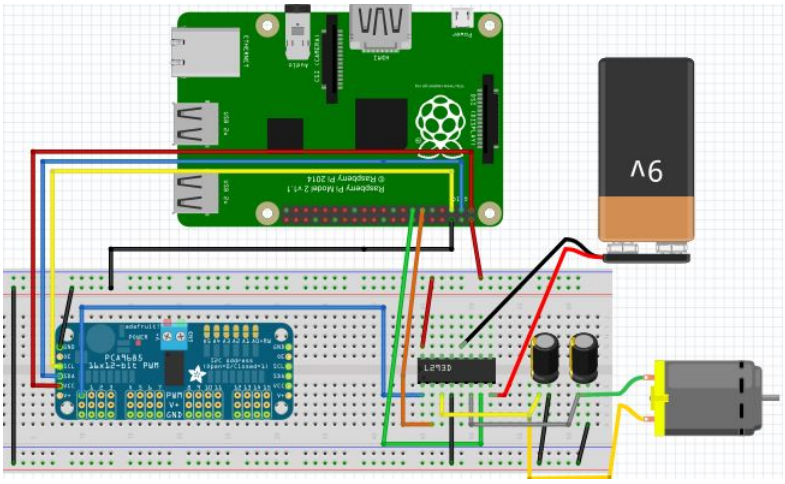
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD) #set the RPi mode as Board mode
GPIO.setup(11,GPIO.OUT)  #set pin 11 as output
GPIO.setup(13,GPIO.OUT)  #set pin 13 as output
GPIO.setup(12,GPIO.OUT)  #set pin 12 as output
GPIO.setwarnings(False)
en=GPIO.PWM(12,50) #set pin 12 as PWM pin

en.start(0) # start PWM cycles

# Function name :forward
# Input : None
# Logic : give pin 11 logic 1 connected to IN1 of L293D
#         and give pin 13 logic 0 connected to IN2 of L293D
#         to rotate the motor in clockwise direction.
# Output : Motor moves in clockwise direction i.e.,forward
# Example call: forward()
def forward():
    GPIO.output(11,True)
    GPIO.output(13,False)
    return

# Function name :backward
# Input : None
# Logic : give pin 11 logic 0 connected to IN1 of L293D
#         and give pin 13 logic 1 connected to IN2 of L293D
#         to rotate the motor in anticlockwise direction.
# Output : Motor moves in anticlockwise direction i.e.,backward
# Example call: backward()
def backward():
    GPIO.outpur(11,False)
    GPIO.output(13,True)
    return

try:
    while True:
        forward()
        for i in range (100):
            en.ChangeDutyCycle(i) # speed of motor goes on increasing
                                  #in clockwise direction
            time.sleep(0.02)
```

```
        for i in range(100):
            en.ChangeDutyCycle(100-i) # speed of motor goes on decreasing
            time.sleep(0.02)

        en.ChangeDutyCycle(0)

        backward()
        for i in range (100):
            en.ChangeDutyCycle(i) # speed of motor goes on increasing
                                  #in anticlockwise direction.
            time.sleep(0.02)
        for i in range(100):
            en.ChangeDutyCycle(100-i) # speed of motor goes on decreasing
            time.sleep(0.02)

        en.ChangeDutyCycle(0)

except KeyboardInterrupt:
    pass
GPIO.cleanup()
en.stop()
```

6.2 Controlling the speed of dc motor through L293D using PWM Driver IC

In this experiment, we are using PWM driver IC (PCA9685) to generate PWM signal to control the velocity of dc motor.

- Connections to L293D is same as above except few changes listed below.
- Pin 1 to channel 0 of PCA9685 for PWM
- Pin 2 to RPi pin 11
- Pin 7 to Rpi pin 13



Note: Download Adafruit_PCA9685 library from Github.
Code

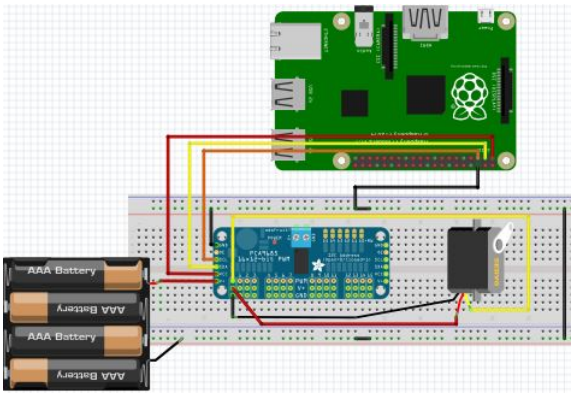
```
import Adafruit_PCA9685
import smbus
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD) # to use Raspberry Pi board pin numbers
GPIO.setup(11,GPIO.OUT)  # to set board pin 11 as output
GPIO.setup(13,GPIO.OUT)  # to set board pin 13 as output
GPIO.setwarnings(False)
pwm = Adafruit_PCA9685.PCA9685()

pwm.set_pwm_freq(60)    # set the frequency of pwm to 60Hz

# Function name :forward
# Input : None
# Logic : give pin 11 logic 1 connected to IN1 of L293D
#         and give pin 13 logic 0 connected to IN2 of L293D
#         to rotate the motor in clockwise direction.
# Output : Motor moves in clockwise direction i.e.,forward
# Example call: forward()
def forward():   # for clockwise motion of motor
    GPIO.output(11,True)
    GPIO.output(13,False)
    return

# Function name :backward
# Input : None
```

```
# Logic : give pin 11 logic 0 connected to IN1 of L293D
#         and give pin 13 logic 1 connected to IN2 of L293D
#         to rotate the motor in anticlockwise direction.
# Output : Motor moves in anticlockwise direction i.e.,backward
# Example call: backward()
def backward():  #for anticlockwise motion of motor
    GPIO.output(11,False)
    GPIO.output(13,True)
    return

try:
    while True:
        forward()
        for value in range (0,4095,409):
            pwm.set_pwm(0, 0, value) #motor rotates in forward direction
            time.sleep(2)            #with increasing speed
        for value in range (0,4095,409):
            pwm.set_pwm(0, 0 ,4095-value)#motor rotates in backward direc
            time.sleep(2)              # with decreasing speed

        backward()
        for value in range (0,4095,409):
            pwm.set_pwm(0,0,value)   #motor rotates in backward direction
            time.sleep(2)           #with increasing speed
        for value in range (0,4095,409):
            pwm.set_pwm(0,0,4095-value)   #motor rotates in backward dire
            time.sleep(2)             # with decreasing speed
except KeyboardInterrupt:
    pass
GPIO.cleanup()     # clears GPIO pins which stalls the motor
```

6.3 Controlling the velocity of servo motor using PWM Driver IC

In this experiment we are using PWM driver IC(PCA9685) to generate PWM signal.
Connections

- Control pin of servo motor is connected to S pin of channel 0.
- Ground and Vcc of servo motor is connected to ground and Vcc of channel 0 respectively.



Note: Download Adafruit_PCA9685 library from Github.
Code

```
from __future__ import division
import time

# Import the PCA9685 module.
import Adafruit_PCA9685


# Uncomment to enable debug output.
#import logging
#logging.basicConfig(level=logging.DEBUG)

# Initialise the PCA9685 using the default address (0x40).
pwm = Adafruit_PCA9685.PCA9685()

# Alternatively specify a different address and/or bus:
#pwm = Adafruit_PCA9685.PCA9685(address=0x41, busnum=2)
```

```
# Configure min and max servo pulse lengths
servo_min = 150  # Min pulse length out of 4096
servo_max = 600  # Max pulse length out of 4096

# Set frequency to 60hz, good for servos.
pwm.set_pwm_freq(60)

print('Moving servo on channel 0, press Ctrl-C to quit...')
try:
    while True:
        angle=input('Enter angle:')
        value=angle*5/2+150
        value=int(value)
        print value
        pwm.set_pwm(0,0,value)
        time.sleep(1)
except KeyboardInterrupt:
    pass
pwm.set_pwm(0, 0, 0)
    #time.sleep(1)
```

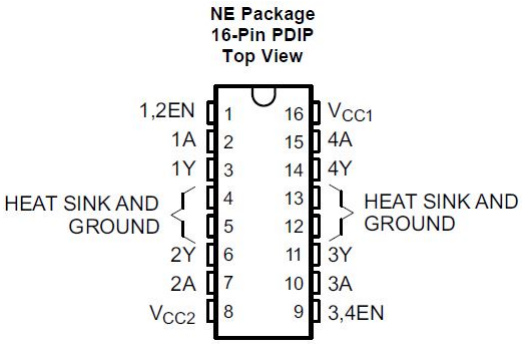
7 Exercise

1. Interface a dc motor through L293D motor driver IC using hardware PWM pins(GPIO 18 or IC pin 12) on RPi.

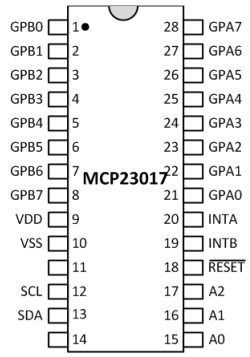
8 Appendix

8.1 PIN Diagram

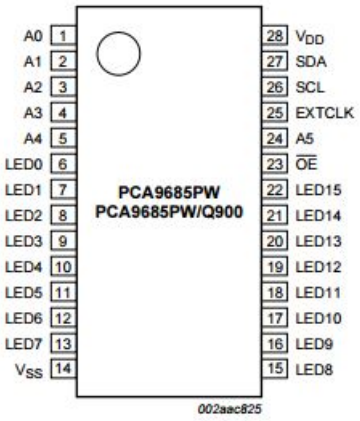
L293D



MCP23017



PCA9685



8.2 Datasheets

PCA9685
`https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf`
L293D
`http://www.ti.com/lit/ds/symlink/l293.pdf`

9 References

1. https://projects.drogon.net/software-pwm-on-the-raspberry-pi/
2. http://electronut.in/controlling-two-servos-with-hardware-pwm-on-the-raspberry-pi-model-a/
3. http://raspi.tv/2013/how-to-use-soft-pwm-in-rpi-gpio-pt-2-led-dimming-and-motor-speed-control

4. http://raspi.tv/2013/rpi-gpio-0-5-2a-now-has-software-pwm-how-to-use-it