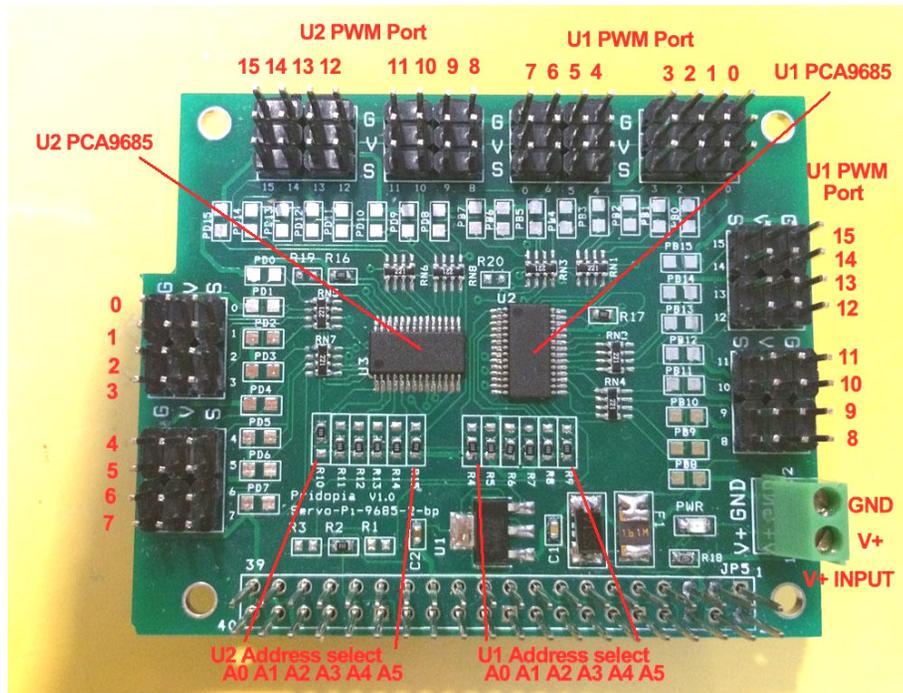


Raspberry Pi - I2C 32 Channel PWM / Servo Board



The PCA9685 is an I2C-bus controlled 16-channel LED controller optimized for LCD 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 40 Hz to 1000 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.

PCA9685 also has a built-in oscillator for the PWM control. However, the frequency used for PWM control in the PCA9685 is adjustable from about 40 Hz to 1000 Hz as compared to the typical 97.6 kHz frequency of the PCA9635. This allows the use of PCA9685 with external power supply controllers. All bits are set at the same frequency.

3. U2 PCA9685 (PWM Port 0 ~ 15)
4. R4,R5,R6,R7R8,R9(for U2 Address select A0,A1,A2,A3,A4,A5)
5. U3 PCA9685 (PWM Port 0 ~ 15)
6. R10,R11,R12,R13,R14,R15(for U3 Address select A0,A1,A2,A3,A4,A5)
- 7.Red power-good V+ LED
8. 1.6A PolySwitch Fuse for V+ input protect.

<https://pypi.python.org/pypi/RPi.GPIO> GPIO library

GPIO library - RPi.GPIO-0.5.6.tar.gz

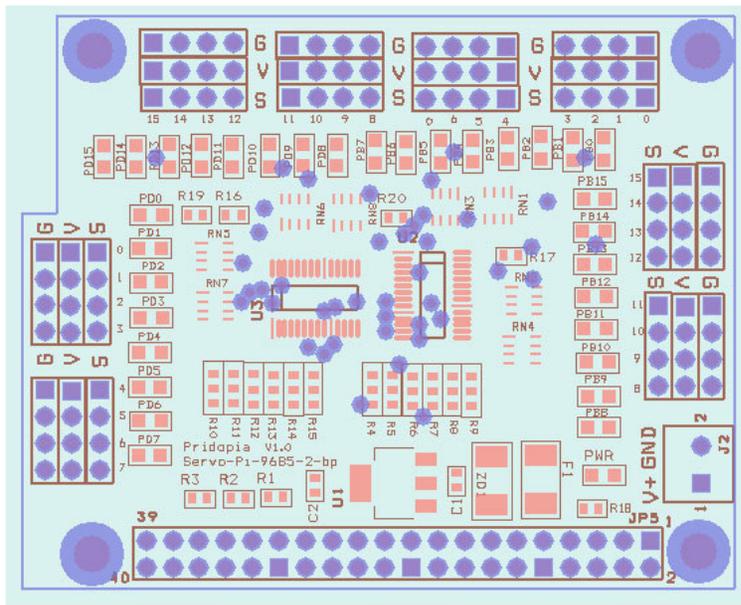
Install python , library and run the test program

```
# sudo apt-get install python-dev
# wget http://www.pridopia.co.uk/pi-pgm/RPi.GPIO-0.5.6.tar.gz
# gunzip RPi.GPIO-0.5.6.tar.gz
# cd RPi.GPIO-0.5.6
# sudo python setup.py install
```

1.Make sure you I2C driver are enable

To enable it all you need to do is comment out a line by putting # in front

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```



```

COM37 - PuTTY
# blacklist spi and i2c by default (many users don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708

```

2. Add i2c-dev in /etc/modules by use
 sudo nano /etc/modules

```

COM37 - PuTTY
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be load
ed
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
spi-bcm2708
i2c-bcm2708
i2c-dev
rtc-1307
tmp102

```

If you already install I2c driver , then
 i2cdetect -y 0 i2cdetect -y 1
 if Rs-Pi-v2 you need change 0 to 1

```

COM34 - PuTTY
Adafruit_I2C.py  Adafruit_PWM_Servo_Driver.py  Servo_Example.py
root@raspberrypi:/home/pi/pwm# i2cdetect -y 0
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  40  41  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:/home/pi/pwm# dir
Adafruit_I2C.py  Adafruit_PWM_Servo_Driver.py  servo-41.py  serv
o.py
Adafruit_I2C.py  Adafruit_PWM_Servo_Driver.py  Servo_Example.py
root@raspberrypi:/home/pi/pwm# ./servo-41.py
Resetting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101

```

in i2cdetect you can found 2 device in system (40,41)

Next install the python-smbus python module:

```

sudo apt-get install python-smbus
sudo apt-get install i2c-tools

```

Now you are ready to use the i2c with python.

Some 9685 test code information

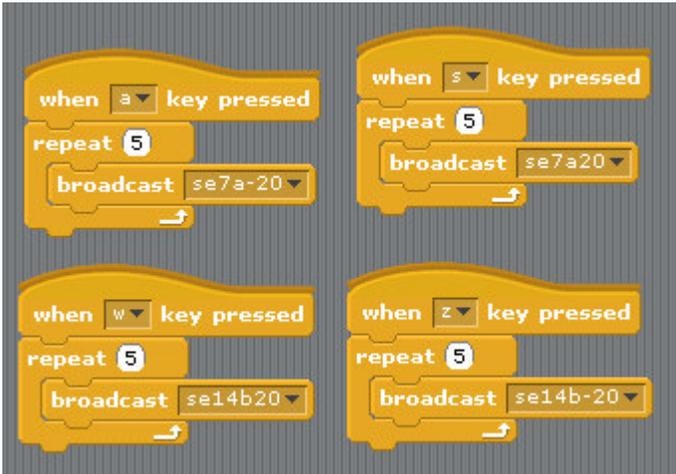
<http://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/using-the-adafruit-library>

Download test code from our web site
<http://www.pridopia.co.uk/pi-9685-2-bp.html>

- Test Program
- [servo-40.py](#) [servo-41.py](#) [pwm](#)
- [servo-4041.py](#)

New Scratch GPIO control program
<http://www.pridopia.co.uk/rs-pi-set-scratch.html>

(1) PWM Servo Motor demo



Command "SE"+ "PWM (0-15)" + "a" +"angle" for Address 41

Command "SE"+ "PWM (0-15)" + "b" +"angle" for Address 40
 se7a20 --> channel 7 servo move 20 angle address 41
 se7a-20 --> channel 7 servo move -20 angle address 41
 se14b20 --> channel 14 servo move 20 angle address 40
 se14b-20 --> channel 14 servo move -20 angle address 40

LED Scrolling Command PWMLED41S0E15D4True

PWMLED"Address" S[Start channel] E[End channel] D[Delay / Timing] [True/False]

Address 40, 41, 42, 43, 44

Start channel & End channel 0 ~ 15 16 channel

D 1,2,3,4,5 (1 ~ 100) Delay Timing

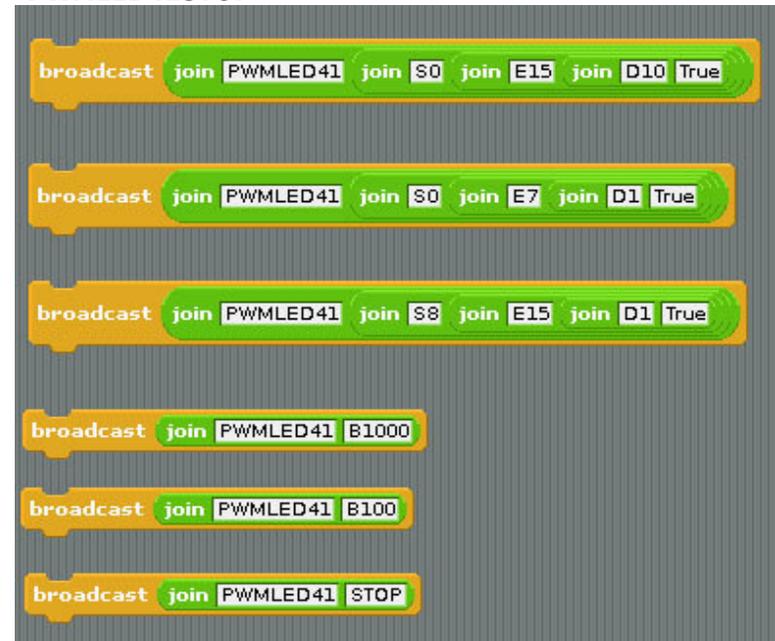
LED Brightness control Command

PWMLED "Address" "B" "0 ~ 1000"

PWMLED41B1000

Stop command

"PWMLED41STOP"



(2) PWM LED output demo

