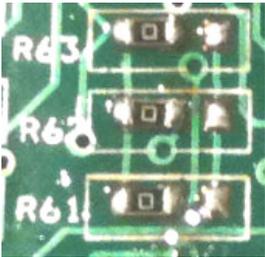
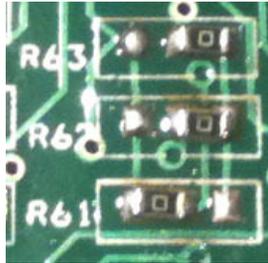


A0, A1, A2 address * right side GND low - 0 * left side Vcc High - 1

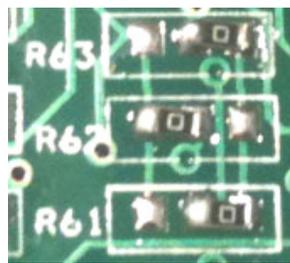
000 -



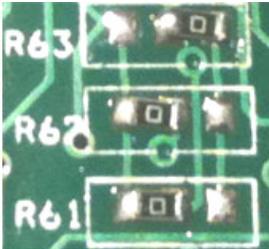
001 -



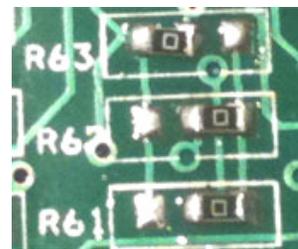
010 -



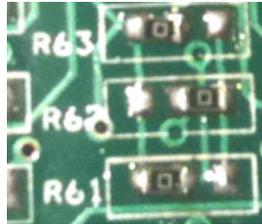
011



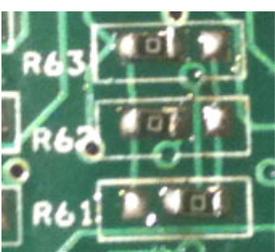
100



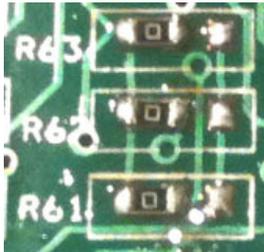
101



110



111



For I2C 16 channel PWM/servo

1. Make sure you I2C & SPI 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`

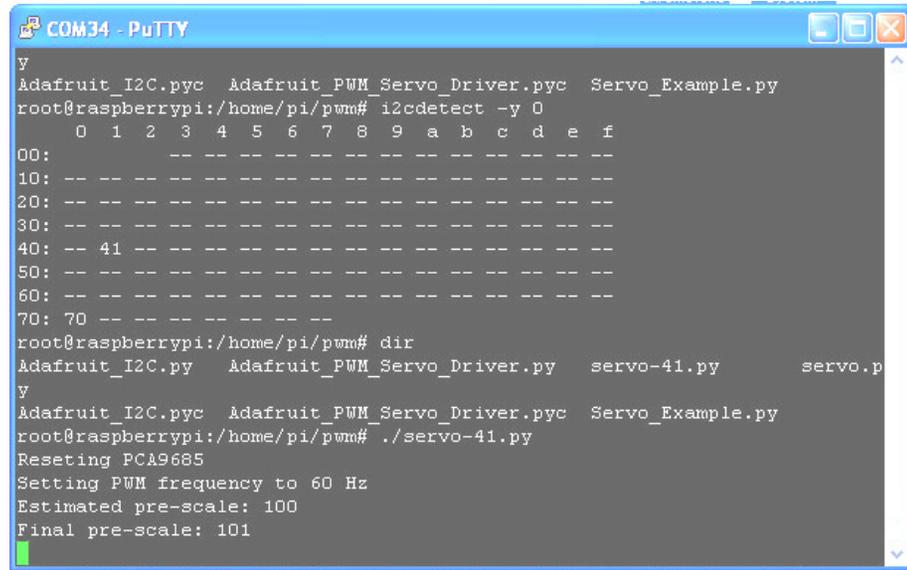
```
COM37 - PuTTY
# blacklist spi and i2c by default (many users don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

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
y
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:  --  41  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:/home/pi/pwm# dir
Adafruit_I2C.py  Adafruit_PWM_Servo_Driver.py  servo-41.py  servo.p
y
Adafruit_I2C.py  Adafruit_PWM_Servo_Driver.py  Servo_Example.py
root@raspberrypi:/home/pi/pwm# ./servo-41.py
Reseting PCA9685
Setting PWM frequency to 60 Hz
Estimated pre-scale: 100
Final pre-scale: 101
```

in `i2cdetect` you can found 1 device in system (41)
41 - 9685

Next install the `python-smbus` python module:

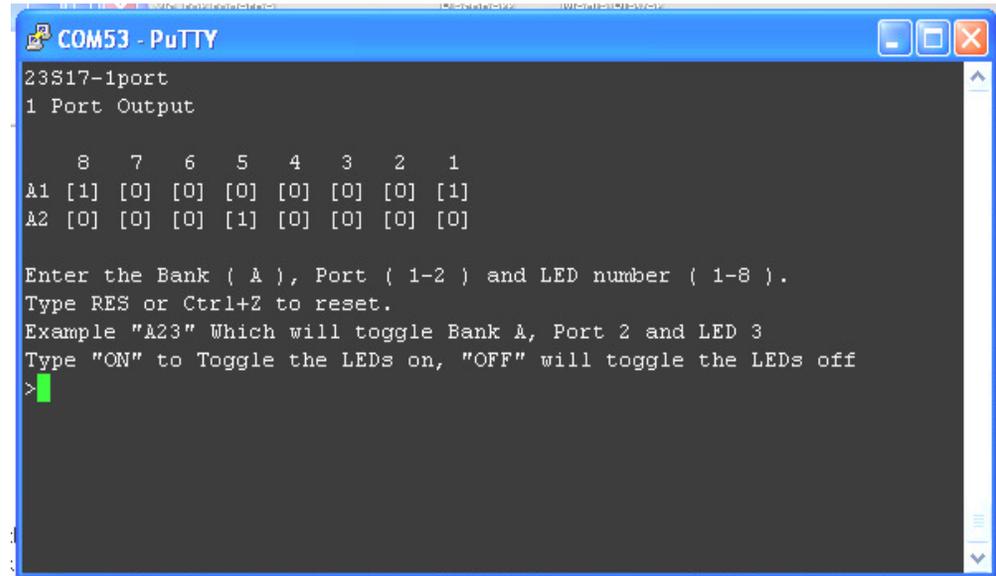
```
sudo apt-get install python-smbus
sudo apt-get install i2c-tools
```

Now you are ready to use the `i2c` with python.

9685 test code information

<http://www.pridopia.co.uk/pi-9685-23s17-lp.html>

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

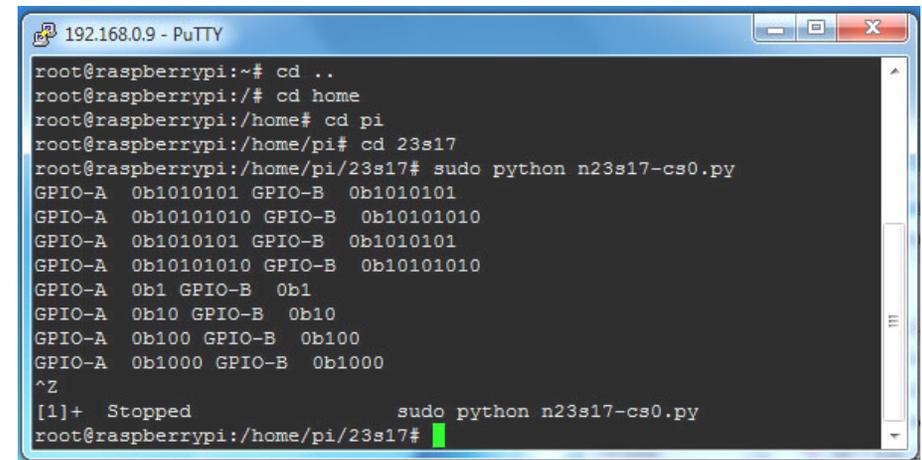


```
COM53 - PuTTY
23S17-1port
1 Port Output

      8  7  6  5  4  3  2  1
A1 [1] [0] [0] [0] [0] [0] [0] [1]
A2 [0] [0] [0] [1] [0] [0] [0] [0]

Enter the Bank ( A ), Port ( 1-2 ) and LED number ( 1-8 ).
Type RES or Ctrl+Z to reset.
Example "A23" Which will toggle Bank A, Port 2 and LED 3
Type "ON" to Toggle the LEDs on, "OFF" will toggle the LEDs off
>
```

our new output test program `23s17-1port-v100.py` display all 16 GPIO status

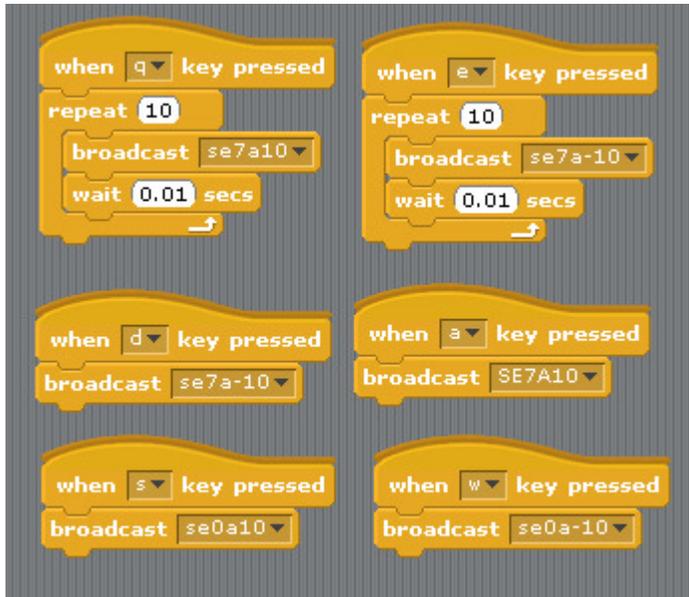


```
192.168.0.9 - PuTTY
root@raspberrypi:~# cd ..
root@raspberrypi:/# cd home
root@raspberrypi:/home# cd pi
root@raspberrypi:/home/pi# cd 23s17
root@raspberrypi:/home/pi/23s17# sudo python n23s17-cs0.py
GPIO-A 0b1010101 GPIO-B 0b1010101
GPIO-A 0b10101010 GPIO-B 0b10101010
GPIO-A 0b1010101 GPIO-B 0b1010101
GPIO-A 0b10101010 GPIO-B 0b10101010
GPIO-A 0b1 GPIO-B 0b1
GPIO-A 0b10 GPIO-B 0b10
GPIO-A 0b100 GPIO-B 0b100
GPIO-A 0b1000 GPIO-B 0b1000
^Z
[1]+  Stopped                  sudo python n23s17-cs0.py
root@raspberrypi:/home/pi/23s17#
```

`n23s17-cs0.py` demo

New Pridopia scratch interface software you can download from our web site

<http://www.pridopia.co.uk/rs-pi-set-scratch.html>

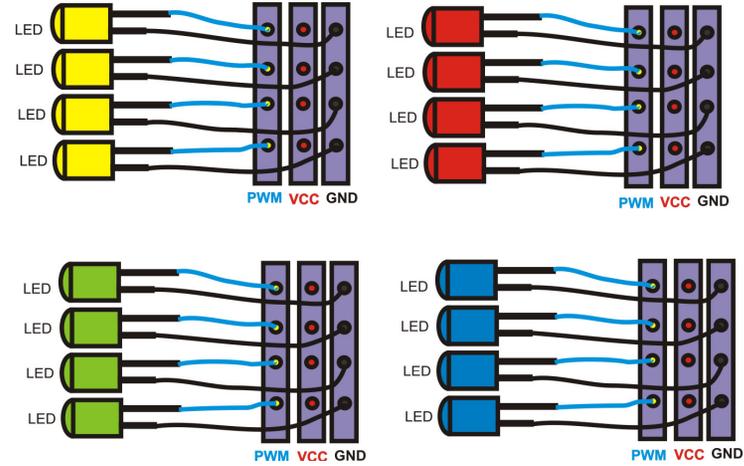


2 Servo in channel 0 & channel 7

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

se7a10 --> channel 7 servo move 10 angle address 41
 se7a-10 -> channel 7 servo move -10 angle address 41
 se0a10 --> channel 0 servo move 10 angle address 41
 se0a-10 --> channel 0 servo move -10 angle address 41

PWM LED control 16 channel LED control



LED Scrolling Command **PWMLED41S0E15D4True**
PWMLED"Address" **S**[Start channel] **E**[End channel] **D**[Delay / Timing] [True/False]

Address 41, 42,43,44

Start channel & End channel 0 ~ 15 16 channel

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

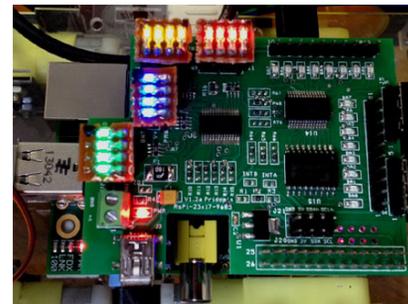
LED Brightness control Command

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

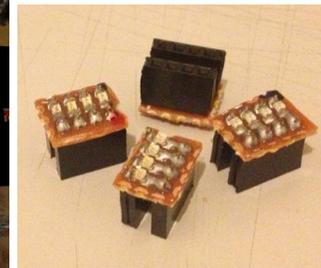
PWMLED41B1000

Stop command

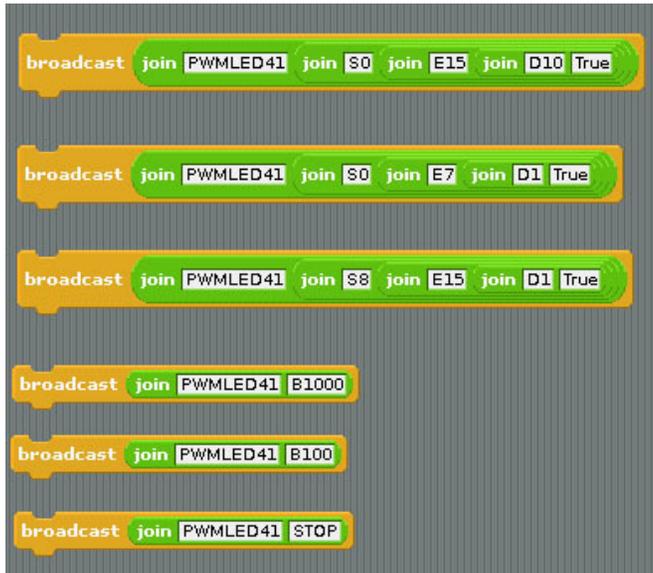
"**PWMLED41STOP**"



16 channel PWM board



LED module (Blue, Green, Yellow, Red)



```

40 --> 1  42 --> 2  44 --> 3  46 --> 44
48 --> 5  4a --> 6  4c --> 7  4e --> 8
Command "sp"+ "address(1-8)" + "a" +"bit(1 to 8)" Port A
Command "sp"+ "address(1-8)" + "b" +"bit(1 to 8)" Port B
Command "bits"+ "address(1-8)"+ "a" +"bit(8 to 1)" Port A
Command "bits"+ "address(1-8)"+ "b" +"bit(8 to 1)" Port B

```

```

sp2b7 --> spi address 2 Port B bit 7 ON/OFF
sp3b4 --> spi address 3 Port B bit 4 ON/OFF
bits2b01010101 --> address 2 port B from bit 8 to 11
output --> 01010101
bits2a01010101 --> address 2 port A from bit 8 to 1
output --> 01010101
bits2aoff --> address 2 Port A all OFF/clear
bits2aclr --> address 2 Port A all OFF/clear

```

(23s17 Scratch control)



U1 to U4 spi 23s17 address 40,42,44,46
U1 to U2 spi 23s17 address 40,42