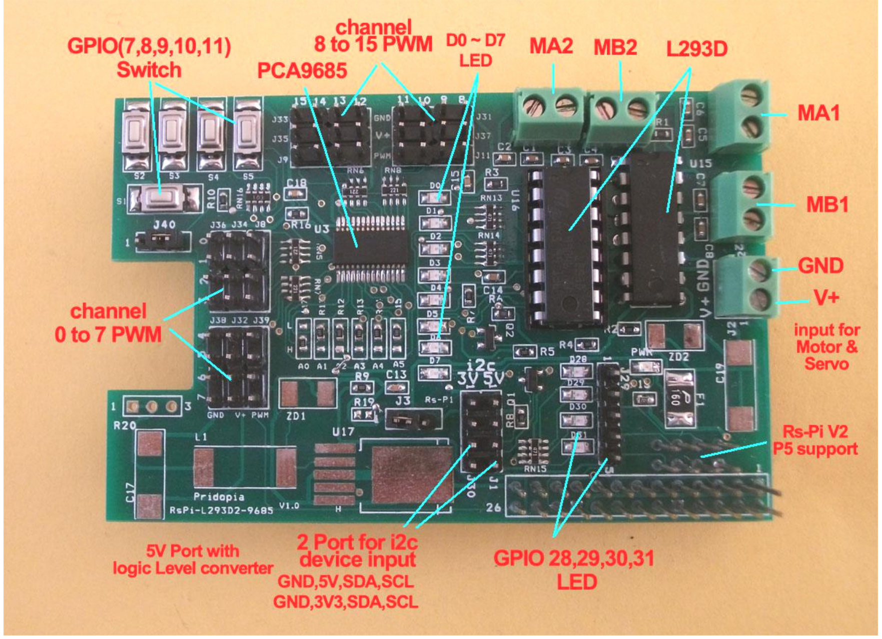
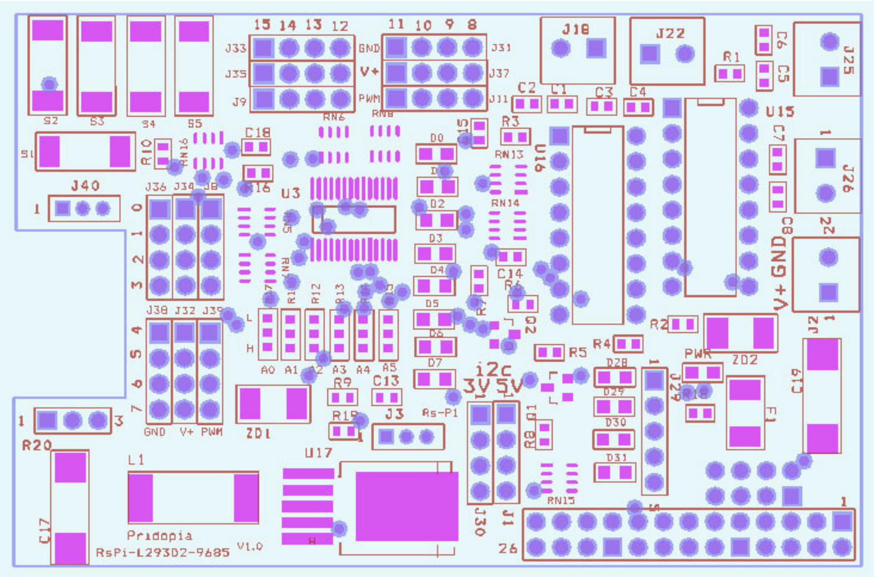
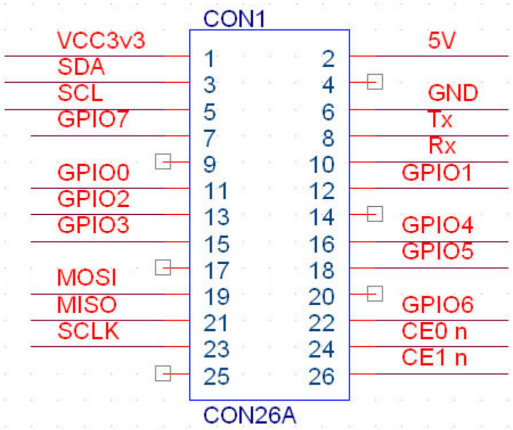
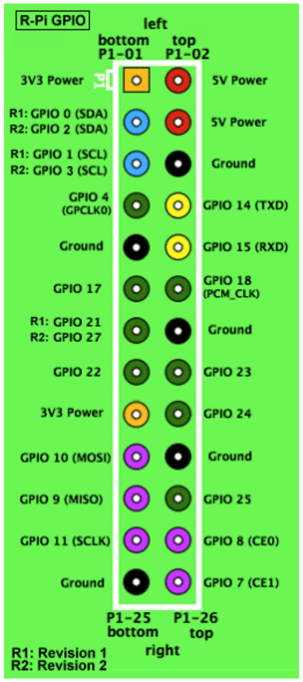
RsPi-9685-16PWM & L293D-2 4 Motor Board User Manual

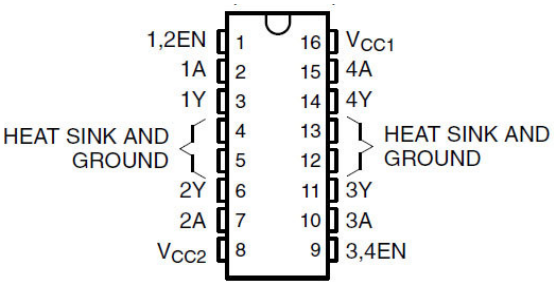




Rs-Pi P1Connector







Pin 1	Pin 2	Pin 7	Function
High	Low	High	Turn clockwise
High	High	Low	Turn anti-clockwise
High	Low	Low	Stop
High	High	High	Stop
Low	Not applicable	Not applicable	Stop

1. This board use RS-Pi GPIO 17,18,27,22,23,24,25,4
2.GPIO 17,18,27,22 for MA1, MB1
3.GPIO 23,24,25,4 for MA2, MB2
4. GPIO 28 ~31 (need V2 - P5 connector) for ultrasonic sensor connector (2GPIO , one for Trig, one for Echo) or other devices

2 extra 12c input for sensors (5V or 3v3) (5V port with Level converter)

5. U15,U16, L293D

6. J17 RS-Pi-V2 GPIO28,GPIO29,GPIO30,GPIO31 from V2 GPIO

7.switch S1 TO S5 (use GPIO 7,8,9,10,11 signal) (SPI signal)

J40 1-2 switch pull-high 10K ohm off

J40 2-3 switch pull-high 10k ohm

8. J2 2PIN BLOCK TERMINAL for DC 5V INPUT (Motor & Servo Motor)
9. J29 Rs-Pi V2 GPIO (28,29,30,31,GND) output
10. R11,R12,R13,R14,R15,R17 (for U3 Address select A0,A1,A2,A3,A4,A5)
11. U3 PCA9685 (PWM Port 0 ~ 15)
12. R10,R11,R12,R13,R14,R15(for U3 Address select A0,A1,A2,A3,A4,A5)
13. 1.6A PolySwitch Fuse for V+ input protected.

Download GPIO library

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

GPIO library - RPi.GPIO-0.5.3a.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-0.5.3a.tar.gz
# gunzip RPi.GPIO-0.5.3a.tar.gz
# tar –xvf RPi.GPIO-0.5.3a.tar
# cd   RPi.GPIO-0.5.3a
# sudo python setup.py install

# sudo python 6motor.py
```

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.

**** for DC Motor & Servo Motor DC Power, you need DC5V plug in to J2 2P Terminal block**

J3 pin 1-2 PSU output to DC Motor & Servo Motor

J3 pin 2-3 (Rs-Pi) PSU power extra output to P1 pin2

(if your DC Motor & Servo Motor use the voltage 5V, then you can choose pin 2-3, but not recommend , only for test purpose)

Test Program can be download from our web site

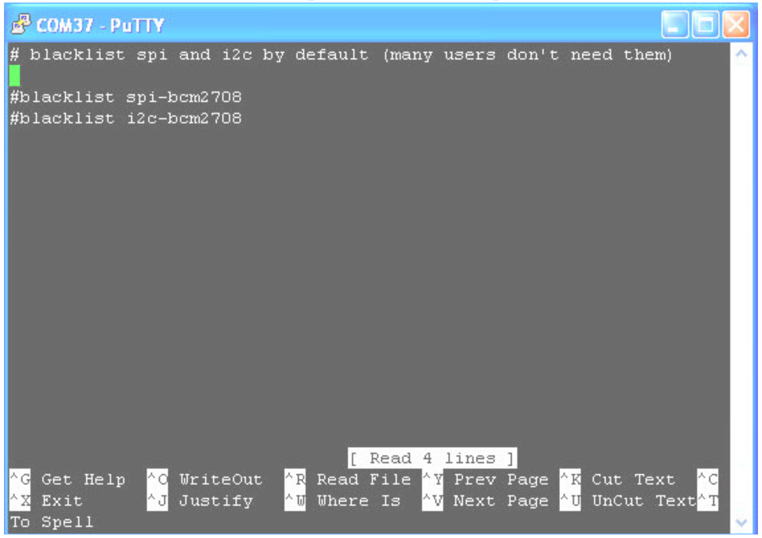
http://www.pridopia.co.uk/pi-9685-l293d-pw.html

2motor.py 4motor .py servo-41.py

1.Make sure you I2C driver are enable

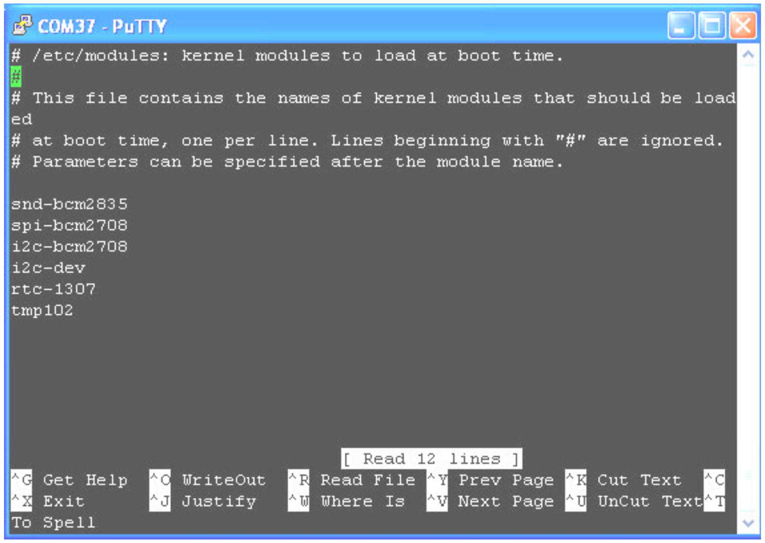
To enable it all you need to do is comment out a line by putting # in front

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



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

#blacklist spi-bcm2708
#blacklist i2c-bcm2708

                                [ Read 4 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T
To Spell
```

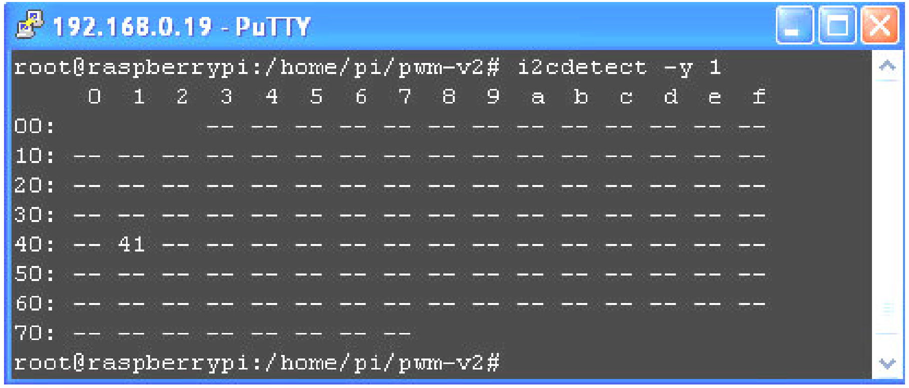
2. Add i2c-dev in /etc/modules by use
sudo nano /etc/modules



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



                              [ Read 12 lines ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T
To Spell
```

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



192.168.0.19 - PuTTY

```
root@raspberrypi:/home/pi/pwm-v2# i2cdetect -y 1
     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: -- -- -- -- -- -- -- --
root@raspberrypi:/home/pi/pwm-v2#
```

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

Pi_Scratch interface software download from our web site

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

(1) Motor On Off + Speed control

A - GPIO 17,18 Motor A   B - GPIO 27,22   Motor B

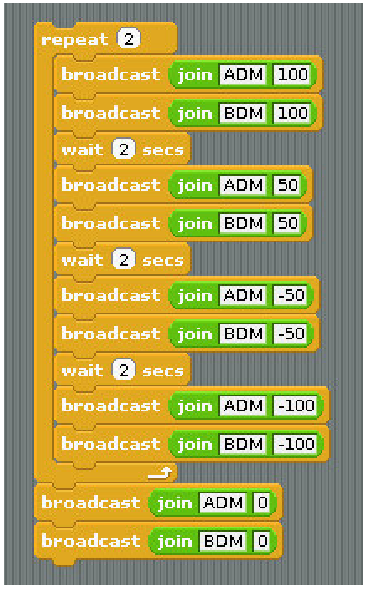
C - GPIO 23,24 Motor C   D - GPIO 25,4 Motor D

command "Motor Name"+ "DM"+"speed"

speed (10 ~100)   clockwise

speed (-10 ~ -100) anticlockwise

ADM100   DC Motor A full speed 100   ADM50   DC Motor A speed 50
BDM100   DC Motor B full speed 100   BDM50   DC Motor B speed 50
ADM-100   DC Motor A anticlockwise full speed 100
BDM-100   DC Motor B anticlockwise full speed 100
ADM0   DC Motor A stop   BDM0   DC Motor B stop



```
repeat 2
    broadcast join ADM 100
    broadcast join BDM 100
    wait 2 secs
    broadcast join ADM 50
    broadcast join BDM 50
    wait 2 secs
    broadcast join ADM -50
    broadcast join BDM -50
    wait 2 secs
    broadcast join ADM -100
    broadcast join BDM -100
broadcast join ADM 0
broadcast join BDM 0
```

(2) BASIC GPIO Motor on off control

define GPIO 17,18,27,22, as output

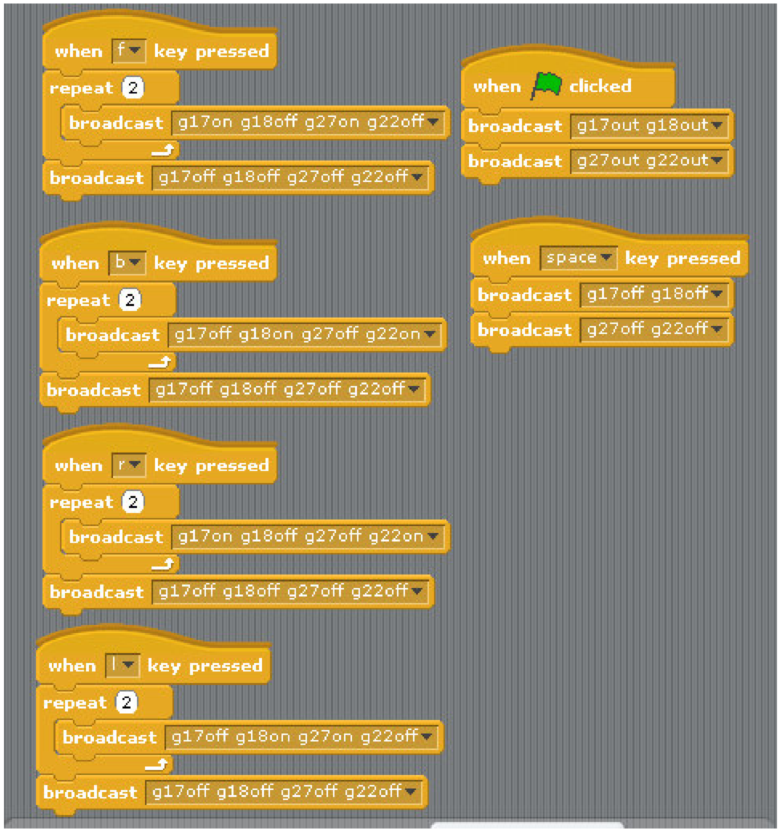
1.press keyboard " f " GPIO 17,27 LED "ON" & 18,22 LED "OFF" car move forward

2.press keyboard " b " GPIO 18,22 LED "ON" & 17,27 LED "OFF" car move backward

3.press keyboard " l " GPIO 18,27 LED "ON" & 17,22 LED "OFF" car turn right

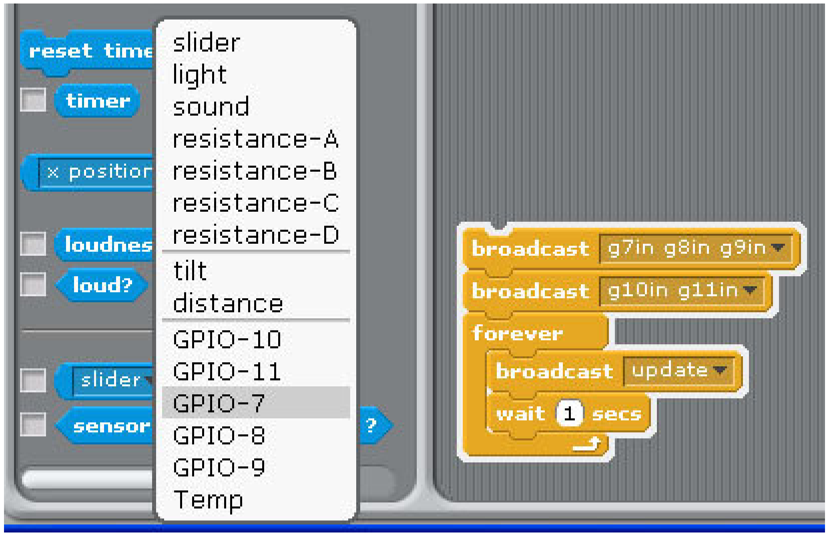
4.press keyboard " r " GPIO 17,22 LED "ON" & 18,27 LED "OFF" car turn left

5.press keyboard " space " GPIO 17,27 LED "OFF" & 18,22 LED "OFF" car STOP



```
when f key pressed
repeat 2
    broadcast g17on g18off g27on g22off
broadcast g17off g18off g27off g22off

when b key pressed
repeat 2
    broadcast g17off g18on g27off g22on
broadcast g17off g18off g27off g22off

when r key pressed
repeat 2
    broadcast g17on g18off g27off g22on
broadcast g17off g18off g27off g22off

when l key pressed
repeat 2
    broadcast g17off g18on g27on g22off
broadcast g17off g18off g27off g22off

when ⚑ clicked
broadcast g17out g18out
broadcast g27out g22out

when space key pressed
broadcast g17off g18off
broadcast g27off g22off
```

(3) 5 GPIO Switch GPIO 7,8,9,10,11 demo

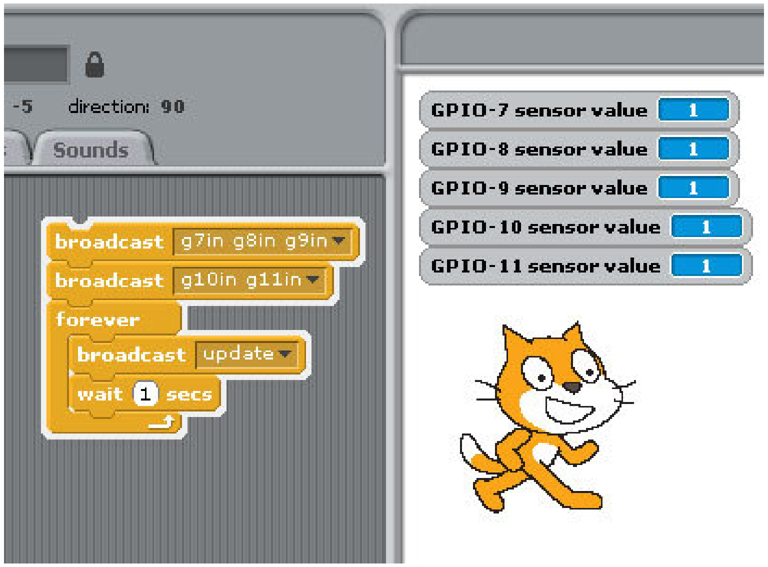


```
broadcast g7in g8in g9in
broadcast g10in g11in
forever
    broadcast update
    wait 1 secs
```

1.Setting GPIO 7,8,9,10,11 as input

2. broadcast "Update"

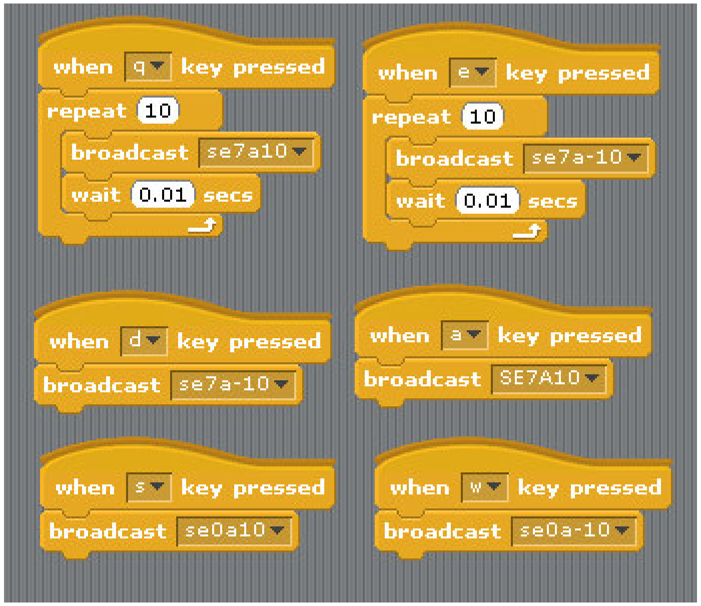
3. in Sensing --> Slider , you will see the GPIO-7 ,8, 9, 10, 11 in the list



```
broadcast g7in g8in g9in
broadcast g10in g11in
forever
    broadcast update
    wait 1 secs
```

GPIO-7 sensor value 1
GPIO-8 sensor value 1
GPIO-9 sensor value 1
GPIO-10 sensor value 1
GPIO-11 sensor value 1

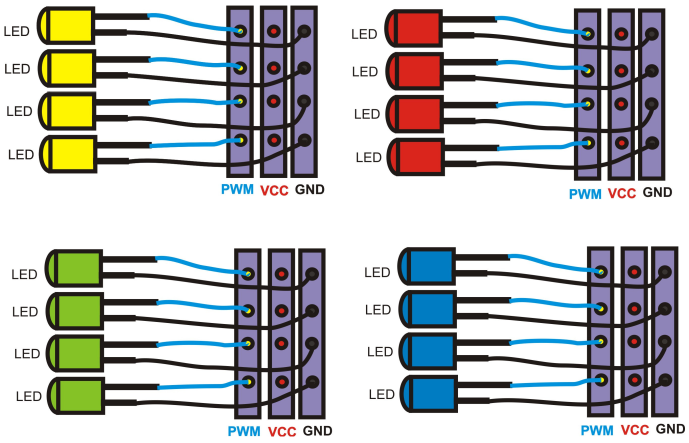
(4) PWM / Servo control demo



```
when q key pressed
repeat 10
    broadcast se7a10
    wait 0.01 secs

when e key pressed
repeat 10
    broadcast se7a-10
    wait 0.01 secs

when d key pressed
broadcast se7a-10

when a key pressed
broadcast SE7A10

when s key pressed
broadcast se0a10

when w key pressed
broadcast se0a-10
```

```
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
```

(5) 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 & **E**nd channel 0 ~ 15 16 channel

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

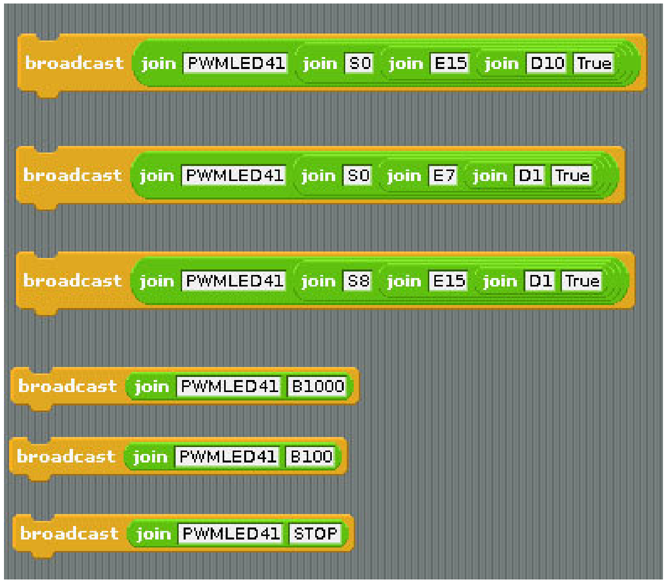
LED Brightness control Command

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

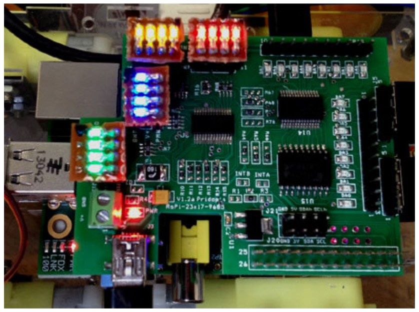
PWMLED41B1000

Stop command

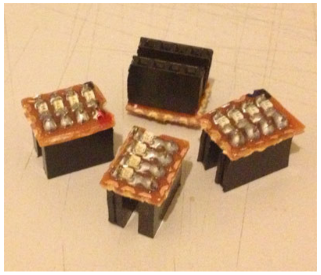
"PWMLED41STOP"



```
broadcast join PWMLED41 join S0 join E15 join D10 True

broadcast join PWMLED41 join S0 join E7 join D1 True

broadcast join PWMLED41 join S8 join E15 join D1 True

broadcast join PWMLED41 B1000

broadcast join PWMLED41 B100

broadcast join PWMLED41 STOP
```



16 channel PWM board



LED module (Blue, Green,Yellow,Red)