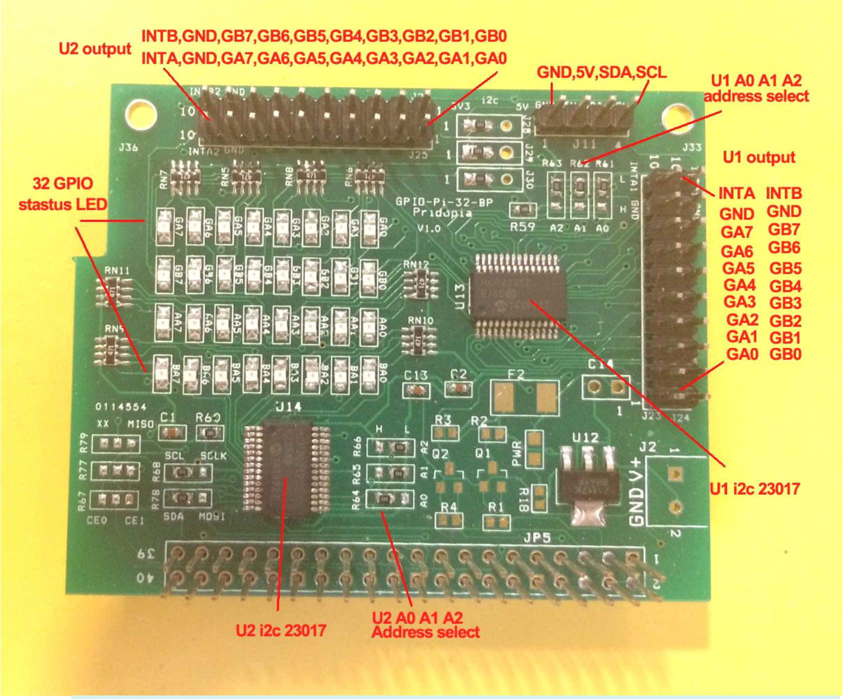
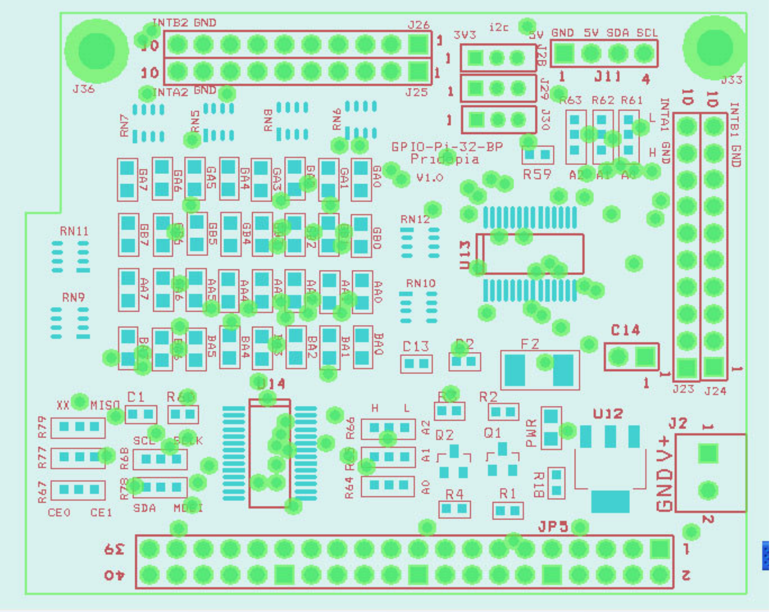
Rs-Pi – i2c 23017-2-BP User Manual



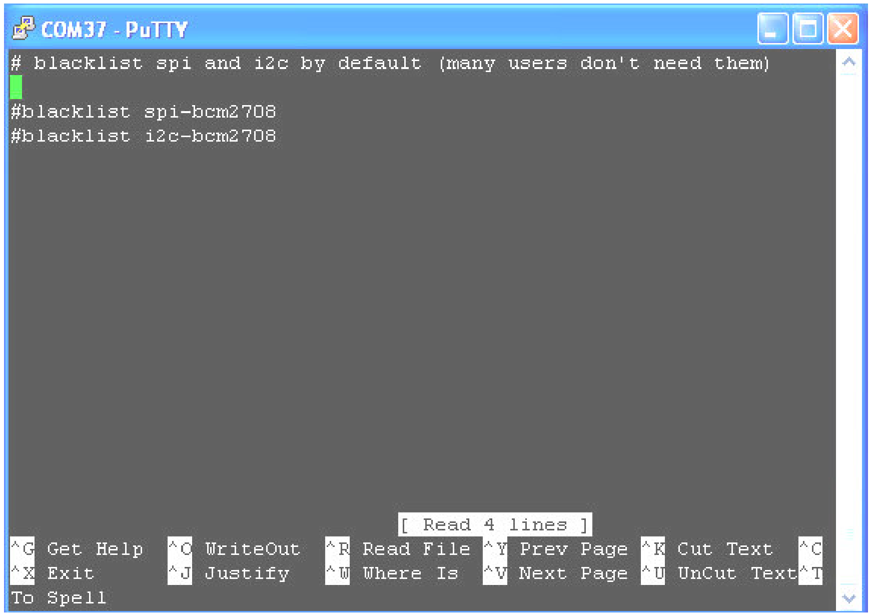


1. J23 GA0 ~ GA7,GND,INTA   U13 Port A
2. J24 GB0 ~ GB7,GND,INTB   U13 Port B
3. J25 GA0 ~ GA7,GND,INTA   U14 Port A
4. J26 GB0 ~ GB7,GND,INTB   U14 Port B
5. R61,R62,R63 (for U13 Address select A0,A1,A2)
6. R64,R65,R66 (for U14 Address select A0,A1,A2)
7. U13 (21) 23017 -1 Port A,B U14 (22)   23017-2 Port A,B

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
```

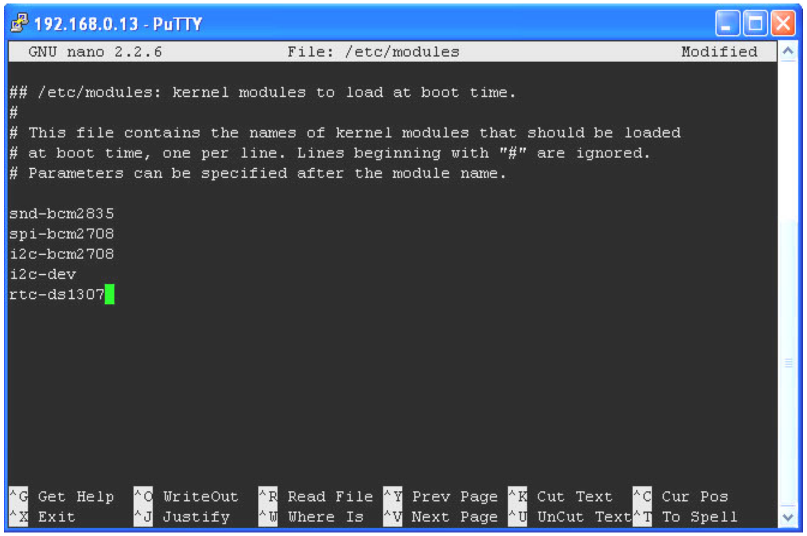


```
# blacklist spi and i2c by default (many users don't need them)

#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
```

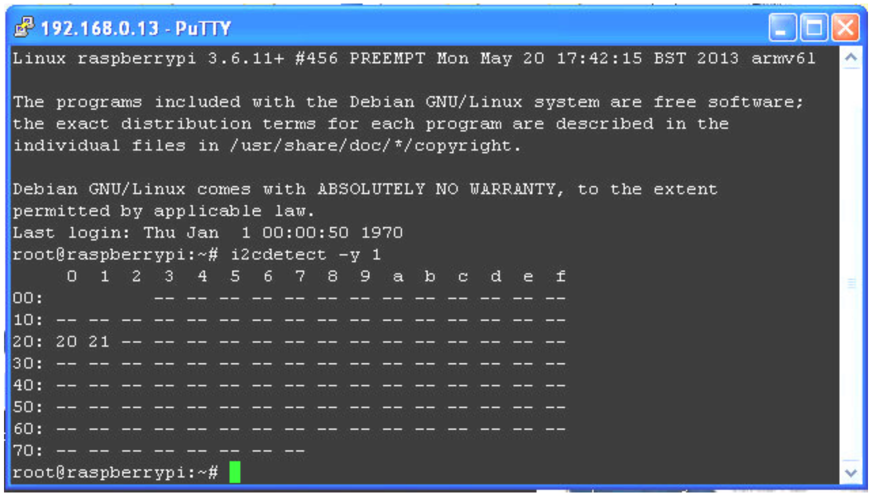


```
  GNU nano 2.2.6              File: /etc/modules                        Modified

## /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# 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-ds1307









^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^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



```
Linux raspberrypi 3.6.11+ #456 PREEMPT Mon May 20 17:42:15 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  1 00:00:50 1970
root@raspberrypi:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 21 -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@raspberrypi:~#
```

You should see something like 20 & 21 -> 23017 x2

Next install the python-smbus python module:

sudo apt-get install python-smbus

sudo apt-get install i2c-tools

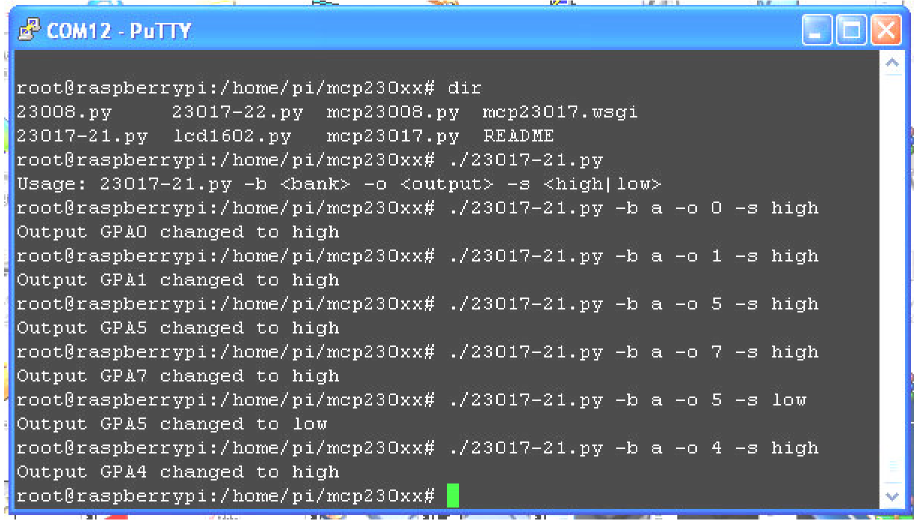
Now you are ready to use the i2c with python.

Some 23017 program information

http://nathan.chantrell.net/20120524/python-tools-for-the-mcp23017-io-expander/

http://nathan.chantrell.net/20120602/raspberry-pi-io-expander-board

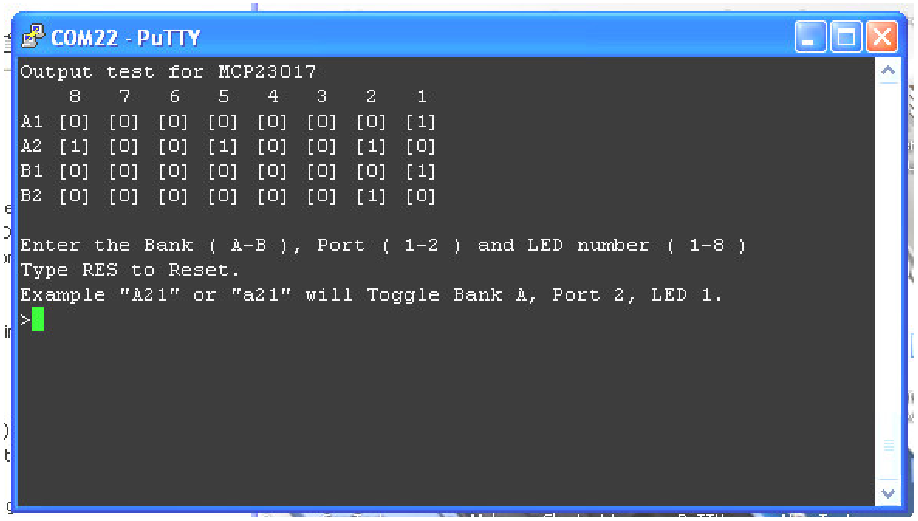
http://learn.adafruit.com/mcp230xx-gpio-expander-on-the-raspberry-pi/hooking-it-all-up



```
root@raspberrypi:/home/pi/mcp230xx# dir
23008.py     23017-22.py  mcp23008.py  mcp23017.wsgi
23017-21.py  lcd1602.py   mcp23017.py  README
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py
Usage: 23017-21.py -b <bank> -o <output> -s <high|low>
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 0 -s high
Output GPA0 changed to high
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 1 -s high
Output GPA1 changed to high
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 5 -s high
Output GPA5 changed to high
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 7 -s high
Output GPA7 changed to high
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 5 -s low
Output GPA5 changed to low
root@raspberrypi:/home/pi/mcp230xx# ./23017-21.py -b a -o 4 -s high
Output GPA4 changed to high
root@raspberrypi:/home/pi/mcp230xx#
```

use 23017-21.py set each bit to be High or Low

or use led-21.py and led-22.py led chaser program, can download from our web site

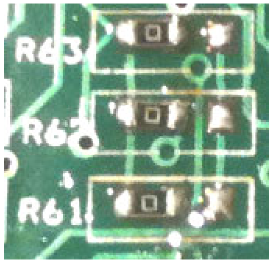


```
Output test for MCP23017
     8    7    6    5    4    3    2    1
A1 [0]  [0]  [0]  [0]  [0]  [0]  [0]  [1]
A2 [1]  [0]  [0]  [1]  [0]  [0]  [1]  [0]
B1 [0]  [0]  [0]  [0]  [0]  [0]  [0]  [1]
B2 [0]  [0]  [0]  [0]  [0]  [0]  [1]  [0]

Enter the Bank ( A-B ), Port ( 1-2 ) and LED number ( 1-8 )
Type RES to Reset.
Example "A21" or "a21" will Toggle Bank A, Port 2, LED 1.
>
```

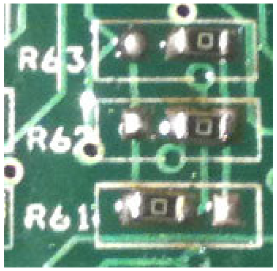
23017-2port-s-v103.py demo display all 32 GPIO status
and set each bit to be High or Low

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

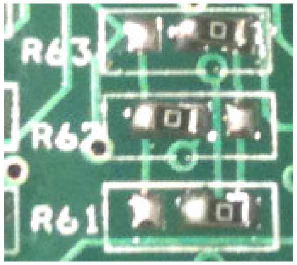
000 - 20



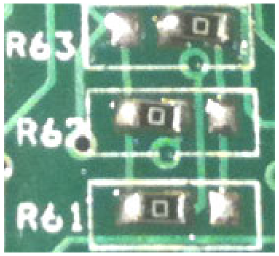
001 - 21



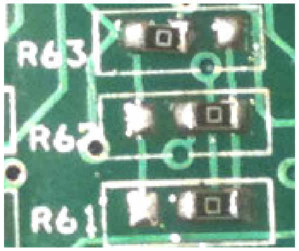
010 - 22



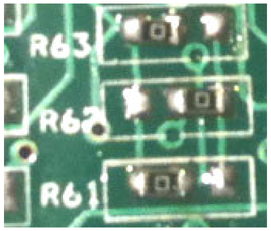
011 - 23



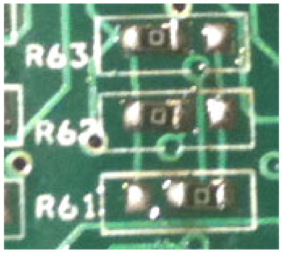
100 - 24



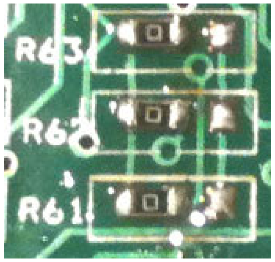
101 - 25



110 - 26



111 - 27



Download test program on our web site Python & C

Python
http://www.pridopia.co.uk/pi-23017-2-bp.html
C code
http://www.pridopia.co.uk/pi-i2c-23017x2-2803x2.html

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

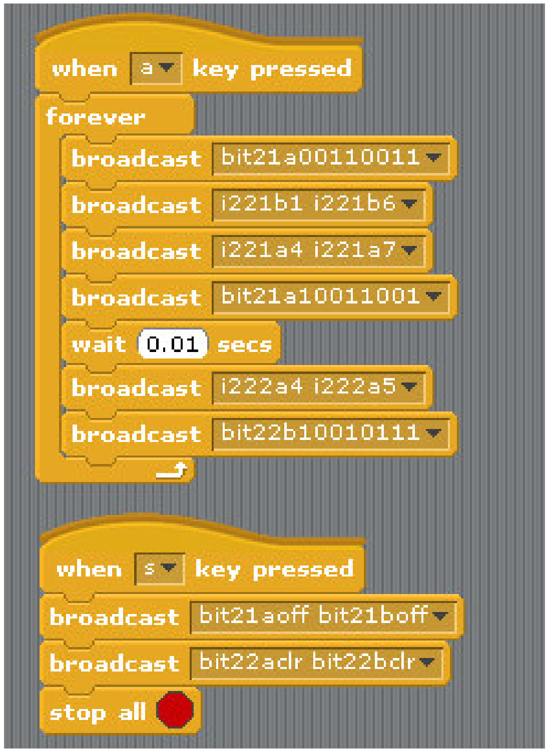
GPIO library - RPi.GPIO-0.5.6.tar.gz

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
# tar –xvf RPi.GPIO-0.5.6.tar
# cd   RPi.GPIO-0.5.6
# sudo python setup.py install
# sudo python xxx.py          ( xxx is your python program name)
```

New Pridopia scratch interface software you can download from our web site
http://www.pridopia.co.uk/rs-pi-set-scratch.html

```
Command "i2"+ "address(20-27)" + "a" +"bit(1 to 8)" for
Port A
 Command "i2"+ "address(20-27)" + "b" +"bit(1 to 8)" for
Port B
 Command "bit"+ "address(20-27)" + "a" +"bit(8 to 1)"
for Port A
 Command "bit"+ "address(20-27)" + "b" +"bit(8 to 1)"
for Port B

  i221a1 -->  i2c address 21 Port A bit 1 ON/OFF
  i222b4 -->  i2c address 22 Port B bit 4 ON/OFF

  bit22b01010101 -->  address 22 port B from bit 8 to 1
                      output  --> 01010101
  bit21a01010101 -->  address 21 port A from bit 8 to 1
                      output  --> 01010101
  bit21aoff  --> address 21 Port A all OFF/clear
  bit21boff  --> address 21 Port B all OFF/clear
  bit22aoff  --> address 22 Port A all OFF/clear
```



```
when a key pressed
forever
  broadcast bit21a00110011
  broadcast i221b1 i221b6
  broadcast i221a4 i221a7
  broadcast bit21a10011001
  wait 0.01 secs
  broadcast i222a4 i222a5
  broadcast bit22b10010111

when s key pressed
broadcast bit21aoff bit21boff
broadcast bit22aclr bit22bclr
stop all
```

U1 to U2 i2c 23017 address 21,22