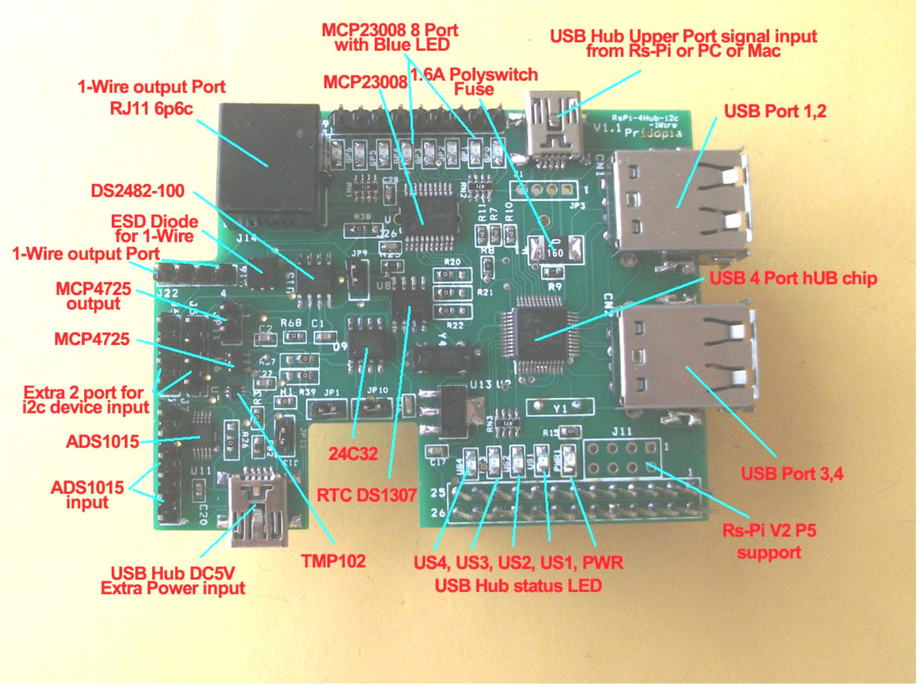
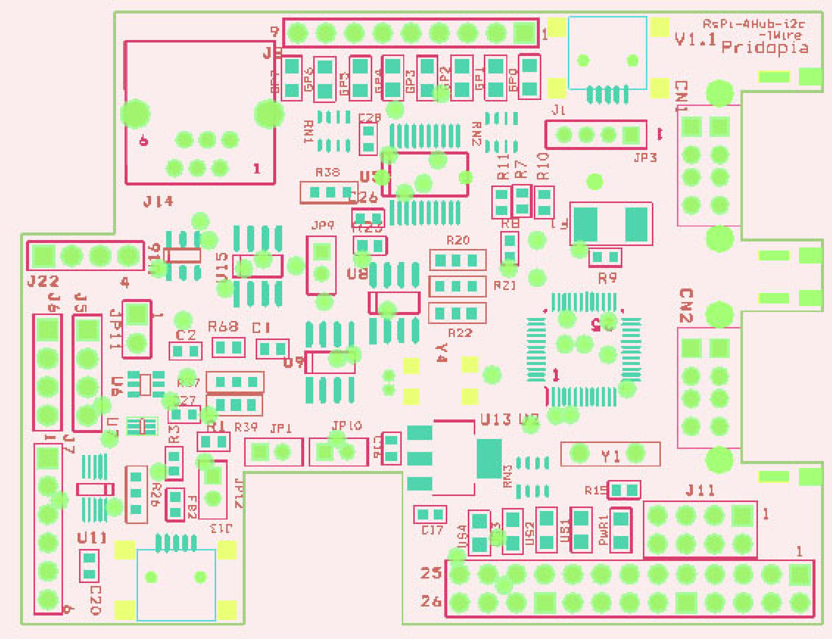
Rs-Pi USB- 4 Hub & AD/DA 1-Wire User Manual


MCP23008 8 Port with Blue LED
USB Hub Upper Port signal input from Rs-Pi or PC or Mac
1-Wire output Port RJ11 6p6c
MCP23008
1.6A Polyswitch Fuse
USB Port 1,2
DS2482-100
ESD Diode for 1-Wire
1-Wire output Port
MCP4725 output
USB 4 Port hUB chip
MCP4725
Extra 2 port for i2c device input
ADS1015
24C32
USB Port 3,4
ADS1015 input
RTC DS1307
Rs-Pi V2 P5 support
USB Hub DC5V Extra Power input
TMP102
US4, US3, US2, US1, PWR USB Hub status LED



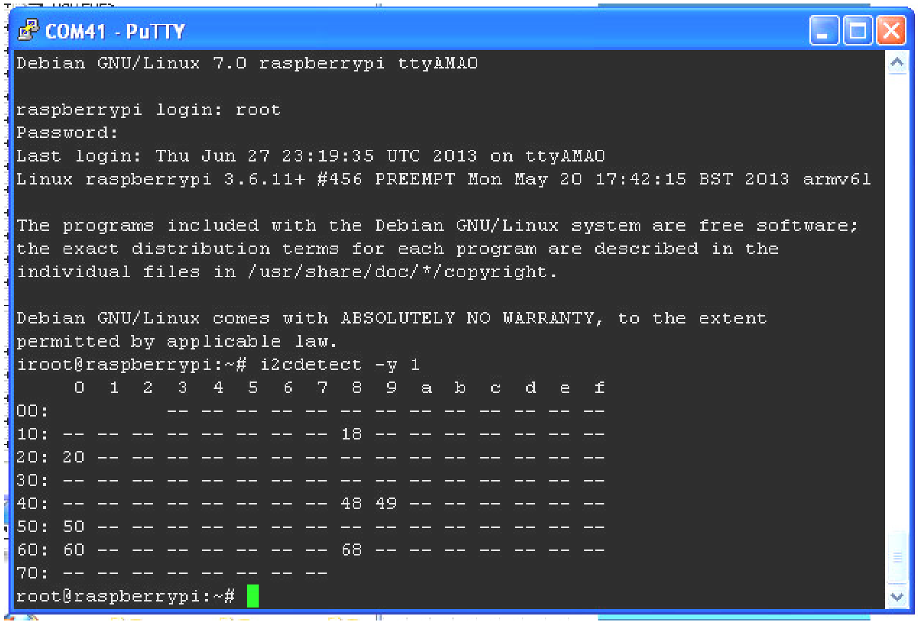
1. U5 **MCP23008 8 GPIO output**

J2 (GP0 ~ GP7) OUTPUT & GP0,GP1,GP2,GP3,GP,GP5,GP6,GP7 (8 LED OUTPUT) R20,R21,R22 (A0,A1,A2) address select **JP9** Disable Jumper

2. U8 RTC **DS1307** with CR1220 Battery
3. U7 **TMP102** Temperature Sensor
4. U9 24C32 32Kbit EEPROM **JP10** Disable Jumper
5. U6 **MCP4725** 12bit Digital-to-Analog Converter , JP11 (AOUT, GND) **JP1** Disable Jumper
6. U11 **ADS1015** 12bit Analog-to-Digital Converter ,

   J7 INPUT (AIN0, AIN1,AIN2,AIN3,GND,3V3) **JP12** Disable Jumper
7. J5 (5V, GND,SCL,SDL) J6(3v3,GND,SCL,SDL) I2C output
8. U2 - USB 4 Ports HUB chipset

   J1 (JP3) USB HUB upper port signal input from Rs-Pi
9. J13 Mini USB 5V input for USB HUB, you don't need plug 5V , the HUB already use 5V from Raspberry Pi, if your use USB device need more power, then you can plug-in 5V in this port.
10. J11 for RS-Pi V2 GPIO connector (got 4 more GPIO pin)

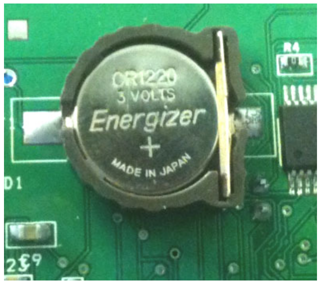


```
Debian GNU/Linux 7.0 raspberrypi ttyAMA0

raspberrypi login: root
Password:
Last login: Thu Jun 27 23:19:35 UTC 2013 on ttyAMA0
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.
iroot@raspberrypi:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- 18 -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 49 -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@raspberrypi:~#
```

i2c bus device detect status

18-> DS2482-100 48 -> tmp102 50 -> 24c32 60 -> MCP4725
68 -> RTC DS1307 49 -> ads1015 20 -> 23008

First Install battery for RTC , " + " mark on top
RTC DS1307 - 68   in i2cdetect –y 0   or i2cdetect –y 1   for Rs-Pi V2 you will see



68 in the screen   68 -> RTC DS1307

48 -> tmp102

This requires a Raspberry Pi running a kernel with the RTC module and DS1307 module included. This is not true of the "Wheezy" distros

or Occidentalis v0.1. This is for use with Occidentalis v0.2 or greater

then, load up the RTC module by running

sudo modprobe rtc-ds1307

Then, as root (type in **sudo bash**) run

echo ds1307 0x68 > **/sys/class/i2c-adapter/i2c-0/new_device** (if you have a rev 1 Pi)

echo ds1307 0x68 > **/sys/class/i2c-adapter/i2c-1/new_device** (if you have a rev 2 Pi)

hwclock –r   read time

hwclock –w   write time in RTC

hwclock –s   write time in System

hwclock --set --date="2013-08-21 08:00:12" --utc

write in custom Time in RTC

*TMP102 information

modprobe tmp102

echo tmp102 0x48 > **/sys/class/i2c-adapter/i2c-0/new_device**
(if you have a rev 1 Pi)

echo tmp102 0x48 > **/sys/class/i2c-adapter/i2c-1/new_device**
(if you have a rev 2 Pi)

sensors   show the temp

- **if your kernel without tmp102 module**

The '-y' option disables interactive mode for the command and the '0' is the I2C bus to scan. You can also run this command on the Pi's second I2C bus by specifying '1' instead.

We can see that it has found our TMP102 device at address 0x48.

To read the temperature from our temperature sensor, we use the i2cget command to read a single byte (Byte 1 - full degrees) from the temperature register (0x00) of the device.

```
pi@raspberrypi ~ $ i2cget -y 0 0x48 0x00 b
0x16
```

Converting this hexadecimal value to decimal, we get our temperature of 22℃.

If you want a more precision on the temperature, you can read both the full and fractional bytes from register 0 as follows:

```
pi@raspberrypi ~ $ i2cget -y 0 0x48 0x00 w
0xa015
```

This gives us byte 2 (0xa0) and byte 1 (0x15), but as a 16bit hexadecimal number and in the wrong order. To convert to ℃, swap around the bytes, shift right by 4, convert to decimal and multiply by 0.0625.

E.g.

dec(0x15a0>>4) * 0.0625 = 21.625℃

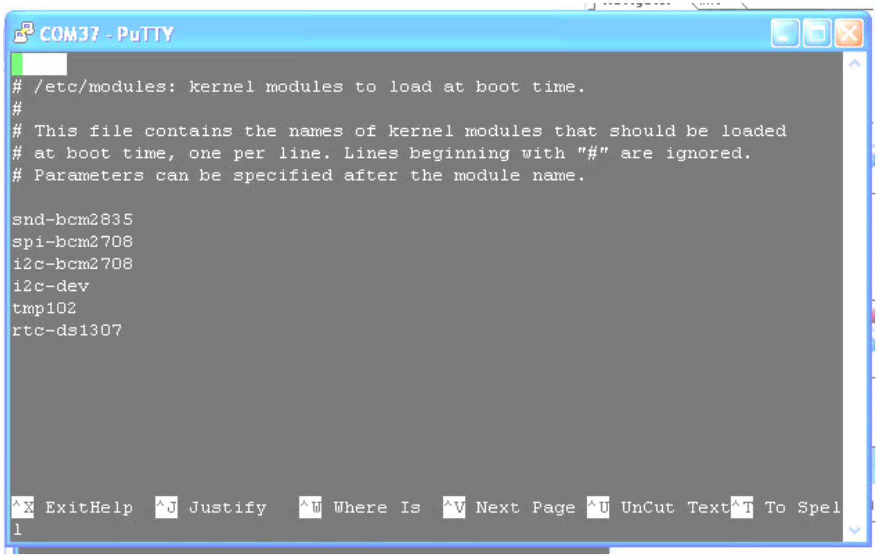
tmp102 information

http://www.element14.com/community/groups/raspberry-pi/blog/2012/07/26/is-it-done-yet-temperature-sensing-with-the-raspberry-pi#comment-16249

http://www.agilart.com/blog/tmp102-raspberry-pi

http://donalmorrissey.blogspot.co.uk/2012/09/raspberry-pi-i2c-tutorial.html

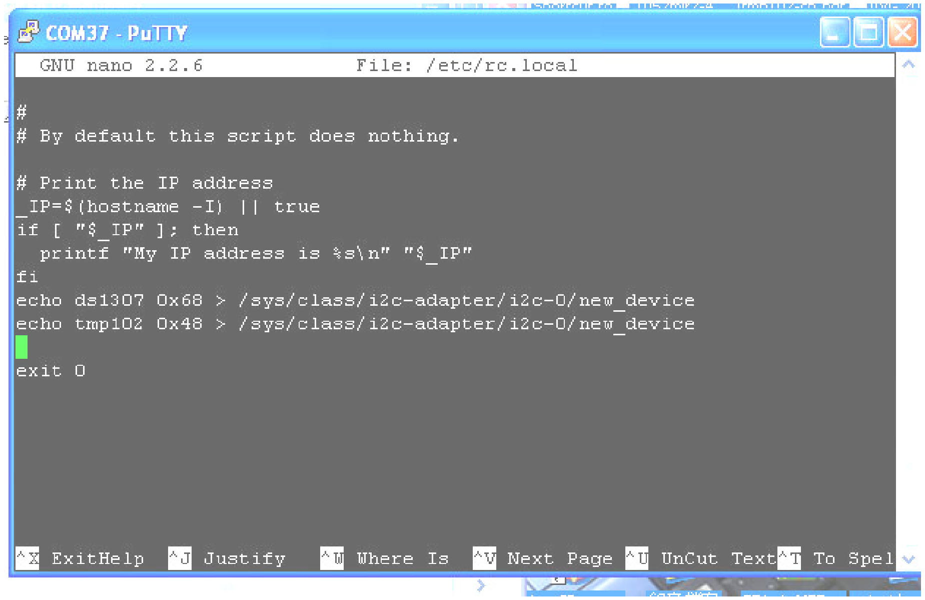
You'll want to add the RTC kernel module & temp tmp102 to the /etc/modules list, so its loaded when the machine boots. Run **sudo nano /etc/modules** and add **rtc-ds1307 & tmp102** at the end of the file



```
# /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
tmp102
rtc-ds1307










^X ExitHelp ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text^T To Spel
l
```

Then you'll want to create the DS1307 device creation at boot, edit /etc/rc.local by running
sudo nano /etc/rc.local

and add **echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device** before **exit 0**



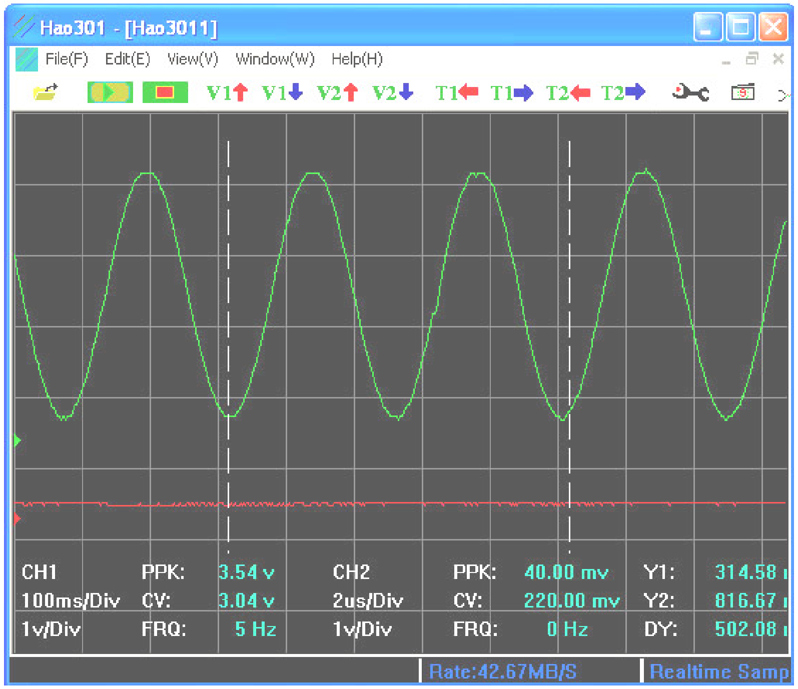
```
  GNU nano 2.2.6              File: /etc/rc.local

#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device
echo tmp102 0x48 > /sys/class/i2c-adapter/i2c-0/new_device

exit 0

^X ExitHelp ^J Justify  ^W Where Is ^V Next Page ^U UnCut Text^T To Spel
```

* Adafruit **Occidentalis v0.2** image support the TMP102 and RTC DS1307 if you need this driver, you can choose this.

The image can be download from
http://learn.adafruit.com/adafruit-raspberry-pi-educational-linux-distro/occidentalis-v0-dot-2

*   MCP4725 Digital to Analog Converter but our address are "60"   all the sample can   Download from our web site.

http://learn.adafruit.com/mcp4725-12-bit-dac-with-raspberry-pi
JP11 for analog output   JP11 pin 1,2 (AOUT, GND)

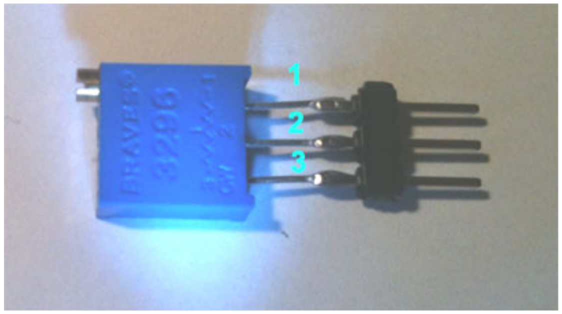


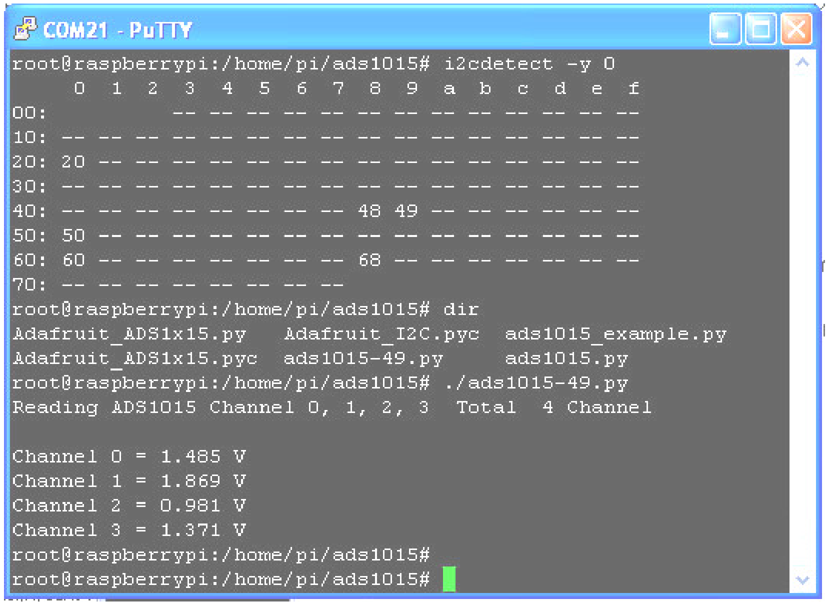
```
CH1          PPK:  3.54 v    CH2       PPK:  40.00 mv    Y1:  314.58
100ms/Div    CV:   3.04 v    2us/Div   CV:   220.00 mv   Y2:  816.67
1v/Div       FRQ:  5 Hz      1v/Div    FRQ:  0 Hz        DY:  502.08
                                       Rate:42.67MB/S    Realtime Samp
```

* **ADS1015 12bit Analog-to-Digital Converter**
demo 12 bit   4 channel input   ads1015-49.py   at address "49"

J7 INPUT (AIN0, AIN1,AIN2,AIN3,GND,3V3)

AIN0, AIN1, AIN2, AIN3 connect to   **trimpot 10K OHM   pin 2**
pin1 3v3   pin3 GND





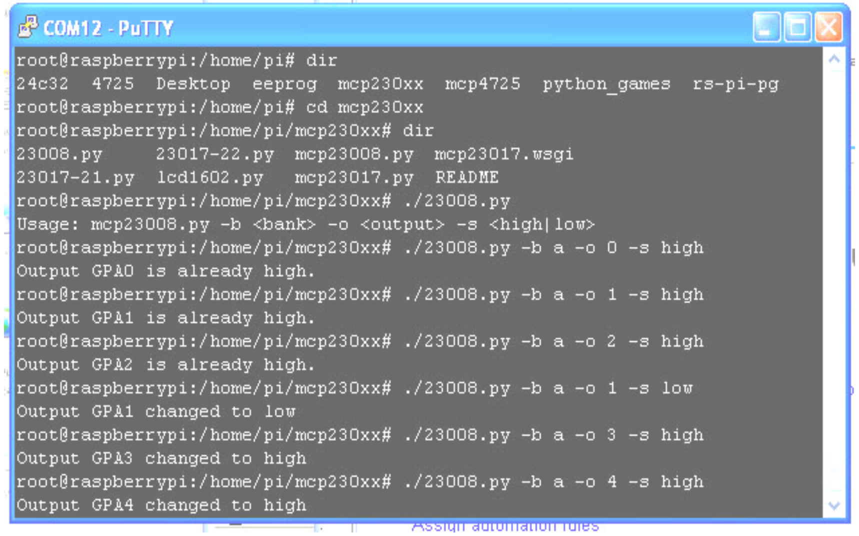
```
root@raspberrypi:/home/pi/ads1015# i2cdetect -y 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 49 -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@raspberrypi:/home/pi/ads1015# dir
Adafruit_ADS1x15.py   Adafruit_I2C.pyc  ads1015_example.py
Adafruit_ADS1x15.pyc  ads1015-49.py     ads1015.py
root@raspberrypi:/home/pi/ads1015# ./ads1015-49.py
Reading ADS1015 Channel 0, 1, 2, 3  Total  4 Channel

Channel 0 = 1.485 V
Channel 1 = 1.869 V
Channel 2 = 0.981 V
Channel 3 = 1.371 V
root@raspberrypi:/home/pi/ads1015#
root@raspberrypi:/home/pi/ads1015#
```

- **23008 23008 8 bit GPIO**

use 23008.py can set each bit output high or low

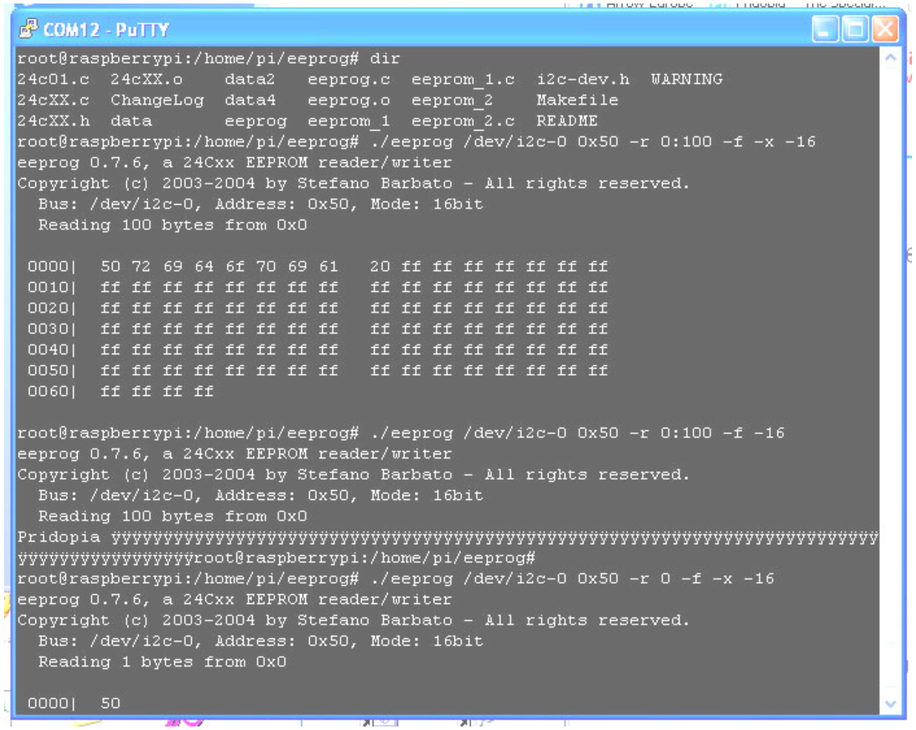
use 23008-20.py LED chaser program for Rs-Pi V1

use 23008-1-20.py for Rs-Pi V2



```
root@raspberrypi:/home/pi# dir
24c32  4725  Desktop  eeprog  mcp230xx  mcp4725  python_games  rs-pi-pg
root@raspberrypi:/home/pi# cd mcp230xx
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# ./23008.py
Usage: mcp23008.py -b <bank> -o <output> -s <high|low>
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 0 -s high
Output GPA0 is already high.
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 1 -s high
Output GPA1 is already high.
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 2 -s high
Output GPA2 is already high.
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 1 -s low
Output GPA1 changed to low
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 3 -s high
Output GPA3 changed to high
root@raspberrypi:/home/pi/mcp230xx# ./23008.py -b a -o 4 -s high
Output GPA4 changed to high
```

- **24c32 EEPROM**



```
root@raspberrypi:/home/pi/eeprog# dir
24c01.c  24cXX.o    data2   eeprog.c  eeprom_1.c  i2c-dev.h  WARNING
24cXX.c  ChangeLog  data4   eeprog.o  eeprom_2    Makefile
24cXX.h  data       eeprog  eeprom_1  eeprom_2.c  README
root@raspberrypi:/home/pi/eeprog# ./eeprog /dev/i2c-0 0x50 -r 0:100 -f -x -16
eeprog 0.7.6, a 24Cxx EEPROM reader/writer
Copyright (c) 2003-2004 by Stefano Barbato - All rights reserved.
  Bus: /dev/i2c-0, Address: 0x50, Mode: 16bit
  Reading 100 bytes from 0x0

 0000|  50 72 69 64 6f 70 69 61   20 ff ff ff ff ff ff ff
 0010|  ff ff ff ff ff ff ff ff   ff ff ff ff ff ff ff ff
 0020|  ff ff ff ff ff ff ff ff   ff ff ff ff ff ff ff ff
 0030|  ff ff ff ff ff ff ff ff   ff ff ff ff ff ff ff ff
 0040|  ff ff ff ff ff ff ff ff   ff ff ff ff ff ff ff ff
 0050|  ff ff ff ff ff ff ff ff   ff ff ff ff ff ff ff ff
 0060|  ff ff ff ff

root@raspberrypi:/home/pi/eeprog# ./eeprog /dev/i2c-0 0x50 -r 0:100 -f -16
eeprog 0.7.6, a 24Cxx EEPROM reader/writer
Copyright (c) 2003-2004 by Stefano Barbato - All rights reserved.
  Bus: /dev/i2c-0, Address: 0x50, Mode: 16bit
  Reading 100 bytes from 0x0
Pridopia ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿroot@raspberrypi:/home/pi/eeprog#
root@raspberrypi:/home/pi/eeprog# ./eeprog /dev/i2c-0 0x50 -r 0 -f -x -16
eeprog 0.7.6, a 24Cxx EEPROM reader/writer
Copyright (c) 2003-2004 by Stefano Barbato - All rights reserved.
  Bus: /dev/i2c-0, Address: 0x50, Mode: 16bit
  Reading 1 bytes from 0x0

 0000|  50
```

use eeprog 0.7.6 can read/write for 24c32 or 24cxx

The Program can be download from

http://www.codesink.org/eeprog.html

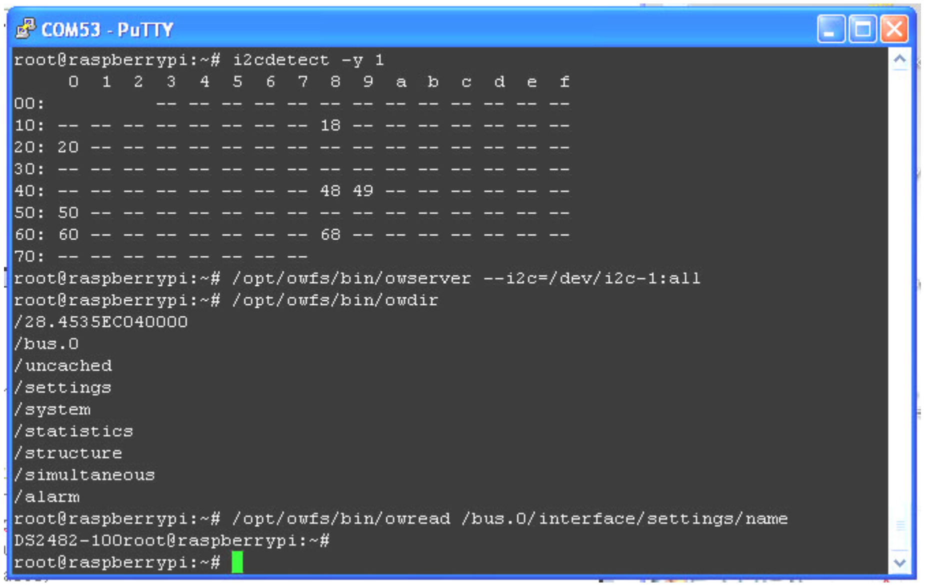
*. U15 DS2482-100 I2C to 1-Wire bridge device

J22 1-wire Port pin 1 - pin4

(5V,GND, OW (1-Wire Data, ESD Protected). RT (1-Wire Return/Ground ,ESC protected)

J14 –RJ11 pin 1 to 4 (5V,GND, OW (1-Wire Data, ESD Protected). RT (1-Wire Return/Ground ,ESC protected)

* U16 DS9503P ESD protection diode



```
root@raspberrypi:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- 18 -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 49 -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@raspberrypi:~# /opt/owfs/bin/owserver --i2c=/dev/i2c-1:all
root@raspberrypi:~# /opt/owfs/bin/owdir
/28.4535EC040000
/bus.0
/uncached
/settings
/system
/statistics
/structure
/simultaneous
/alarm
root@raspberrypi:~# /opt/owfs/bin/owread /bus.0/interface/settings/name
DS2482-100root@raspberrypi:~#
root@raspberrypi:~#
```

18 -> DS2482-100 I2C 1-Wire bridge chip

/28.4535EC040000 - Connect & Detect DALLAS 18B20P TEMP Sensor

```
root@berry:~# /opt/owfs/bin/owserver --i2c=/dev/i2c-0:ALL
root@berry:~# /opt/owfs/bin/owdir
```

It appears that OWServer has found 1 1-wire busses, exactly what we're expecting to happen.

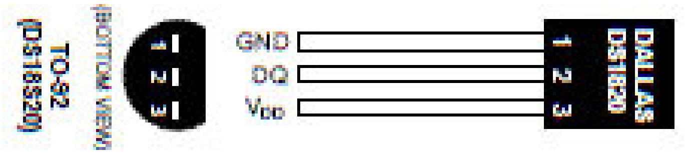
Lets see if we can get some more details.

Which chip on the breakout board is bus.0

```
root@berry:~# /opt/owfs/bin/owread /bus.0/interface/settings/name
DS2482-100
```

How about the i2c addresses of each bus entry. bus.0

```
root@berry:~# /opt/owfs/bin/owread /bus.0/interface/settings/address
/dev/i2c-0:18
```



Installation of the OWFS (One Wire File System)

First you need to install the following packages:

sudo apt-get update

sudo apt-get install automake autoconf autotools-dev gcc-4.7 libtool libusb-dev libfuse-dev swig python2.6-dev tcl8.4-dev php5-dev i2c-tools

If promoted answer Yes on any questions during the install.

Download the latest version of OWFS to your usr/src directory

cd /usr/src

sudo wget -O owfs-latest.tgz http://sourceforge.net/projects/owfs/files/latest/download

Unpack with the following command:

sudo tar xzvf owfs-latest.tgz

Next you must configure OWFS: (replace X.XXXX with the version number you downloaded)

cd owfs-X.XXXX

sudo ./configure

If everything is correct, you should get a result like this:

Current configuration:

Deployment location: /opt/owfs

Compile-time options:

Caching is enabled

USB is DISABLED

etc.

Next you need to compile OWFS which will take approx. 30 minutes with the following command:

sudo make

Next install OWFS which will take a few minutes

sudo make install

Once the installation has completed you need to create a mountpoint for the 1wire folder:

sudo mkdir /mnt/1wire

In order to use the 1wire devices without root privileges you have to change the FUSE settings, edit the fuse configuration file with:

sudo nano /etc/fuse.conf

Update this line: #user_allow_other and remove the # from the start, then save your changes

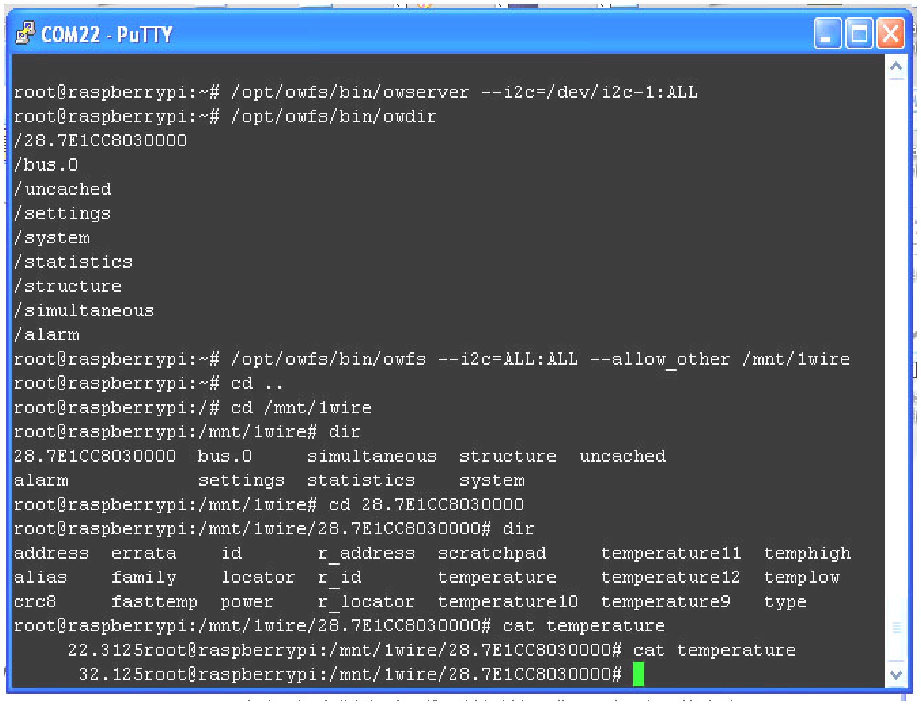
You can now start using OWFS to access your i2c devices and any connected sensors:

sudo /opt/owfs/bin/owfs --i2c=ALL:ALL --allow_other /mnt/1wire/

Using a terminal window navigate to the /mnt/1wire/ directory and use the ls command to list all connected devices.

If you have a temperature sensor connected you should have a folder starting with 28.xxxxxx

cd into this folder and then enter cat temperature to read the temperature of the sensor.



```
root@raspberrypi:~# /opt/owfs/bin/owserver --i2c=/dev/i2c-1:ALL
root@raspberrypi:~# /opt/owfs/bin/owdir
/28.7E1CC8030000
/bus.0
/uncached
/settings
/system
/statistics
/structure
/simultaneous
/alarm
root@raspberrypi:~# /opt/owfs/bin/owfs --i2c=ALL:ALL --allow_other /mnt/1wire
root@raspberrypi:~# cd ..
root@raspberrypi:/# cd /mnt/1wire
root@raspberrypi:/mnt/1wire# dir
28.7E1CC8030000  bus.0     simultaneous  structure  uncached
alarm            settings  statistics    system
root@raspberrypi:/mnt/1wire# cd 28.7E1CC8030000
root@raspberrypi:/mnt/1wire/28.7E1CC8030000# dir
address  errata    id       r_address  scratchpad     temperature11  temphigh
alias    family    locator  r_id       temperature    temperature12  templow
crc8     fasttemp  power    r_locator  temperature10  temperature9   type
root@raspberrypi:/mnt/1wire/28.7E1CC8030000# cat temperature
     22.3125root@raspberrypi:/mnt/1wire/28.7E1CC8030000# cat temperature
      32.125root@raspberrypi:/mnt/1wire/28.7E1CC8030000#
```

cat temperature -- 22.312 & 32.125

/opt/owfs/bin/owserver --i2c=/dev/i2c-1:all

/opt/owfs/bin/owdir

Will display all 1-wire information in your system

To keep Rs-Pi USB Hub board working properly, you need make sure the Vcc input for Rs-Pi above 4.75V,
JP3 pin 1 Vcc, pin4 GND

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 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

Package Content

1x Rs-Pi USB Hub & I2C AD/DA -1-wire board
1x USB to MINI USB hub input cable (for USB Hub input & Power input)
1x CR1220 3V Battery
1x Manual