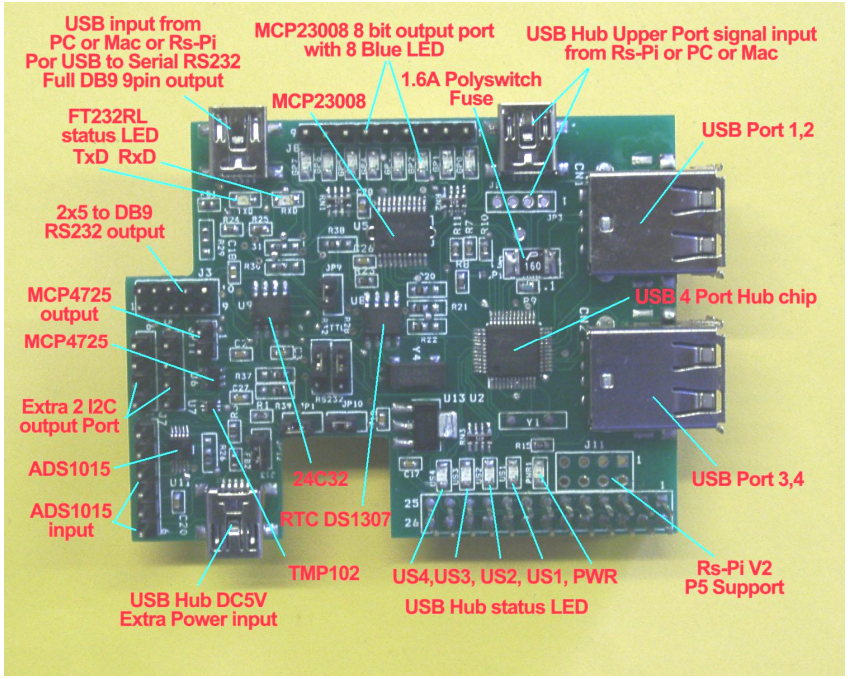
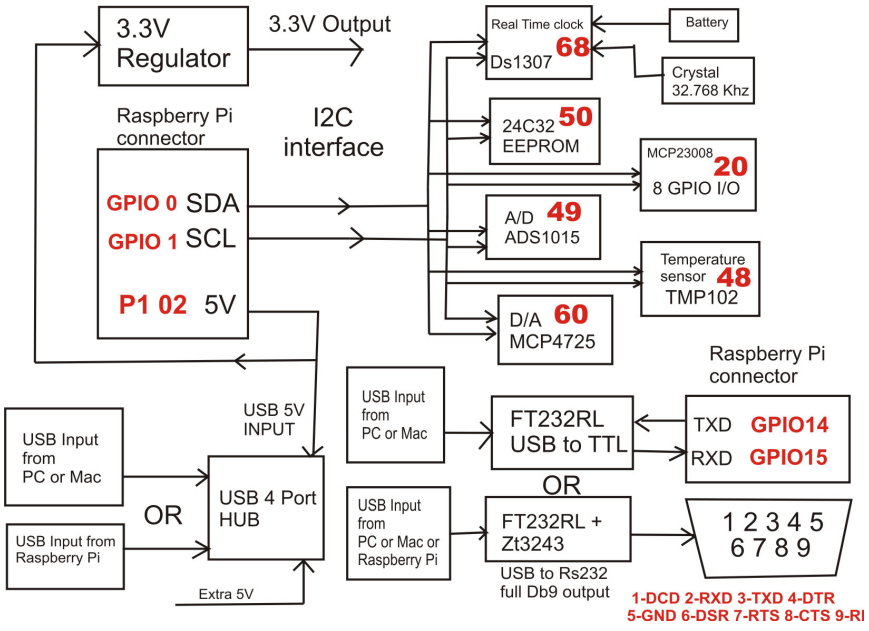
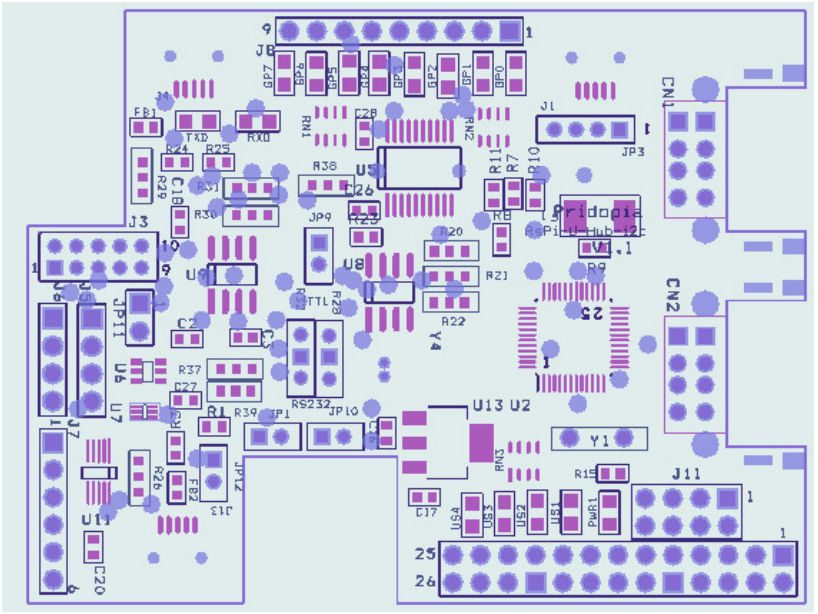
Rs-Pi USB- 4 Hub & I2C –RS232 User Manual







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

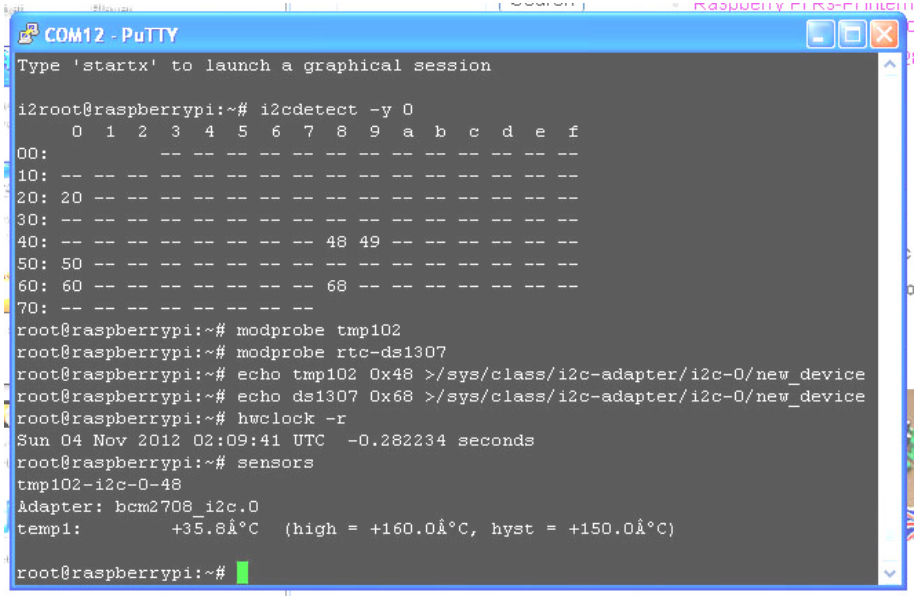
2. U8 RTC **DS1307** with CR1220 Battery
3. U7 **TMP102** Temperature Sensor
4. U9 24C32 32Kbit EEPROM **JP10** Disable Jumper
5. U6 **MCP4725** 12bit Digital-to-Analog Converter , JP11 (AOUT, GND) **JP1** Disable Jumper
6. U11 **ADS1015** 12bit Analog-to-Digital Converter ,

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 P5 GPIO connector (got 4 more GPIO pin)



i2c bus device detect status and active TMP102 & RTC DS1307

48 -> tmp102 50 -> 24c32 60 -> MCP4725 68 -> RTC DS1307

49 -> ads1015 20 -> 23008

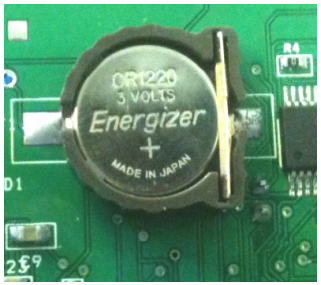
11. U10 **FT232RL** & U12 **ZR3243** J4 USB input from PC or Mac or RS-Pi for USB to RS232 COM port, just connect a cable (5pin mini USB-B to A USB) then you have DB9 full 9pin RS232 function.

U12 **ZR3243** RS232 Transceiver Driver , You can have DB9 full 9pin RS232 output for PC or Mac or RS-Pi USB cable input .

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

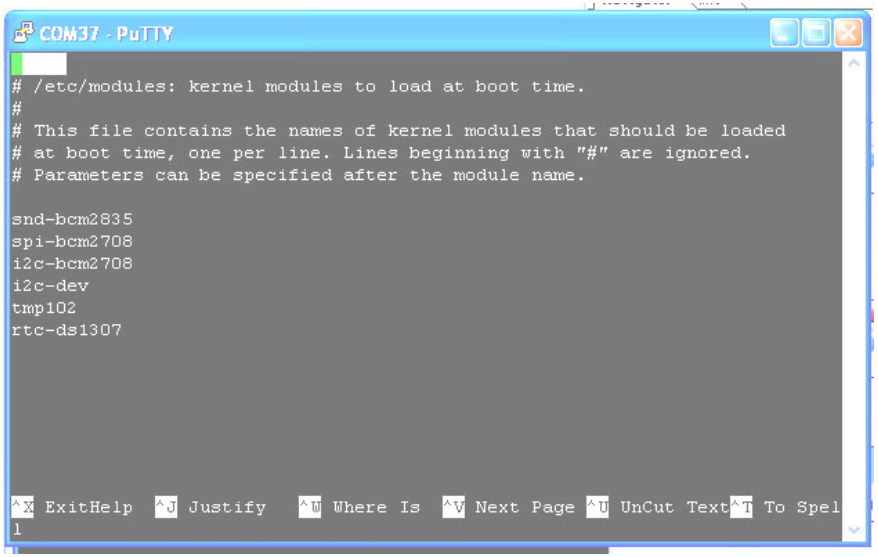
modprobe tmp102

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)

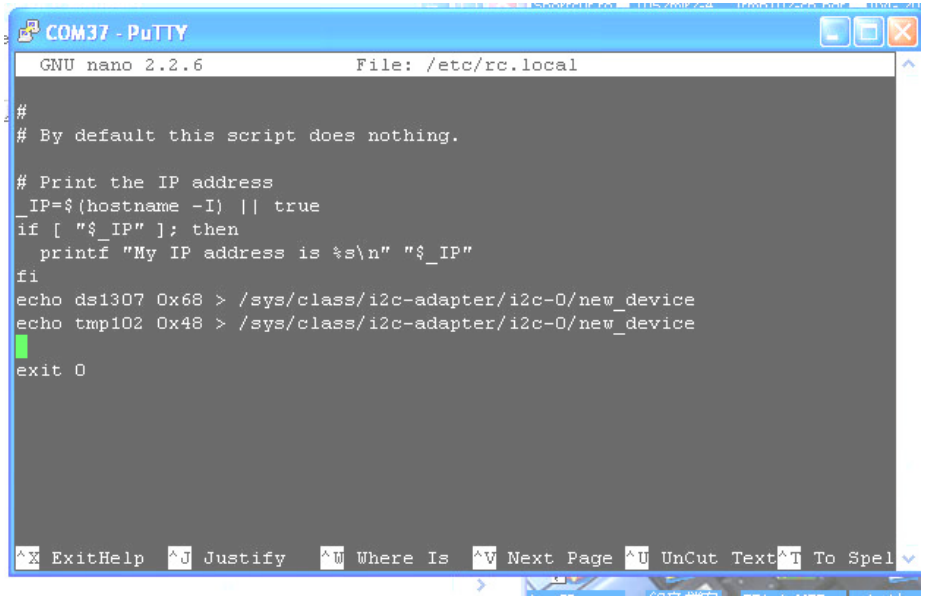
sensors show the temp

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



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**



* 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

- **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°C.

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 °C, 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°C

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

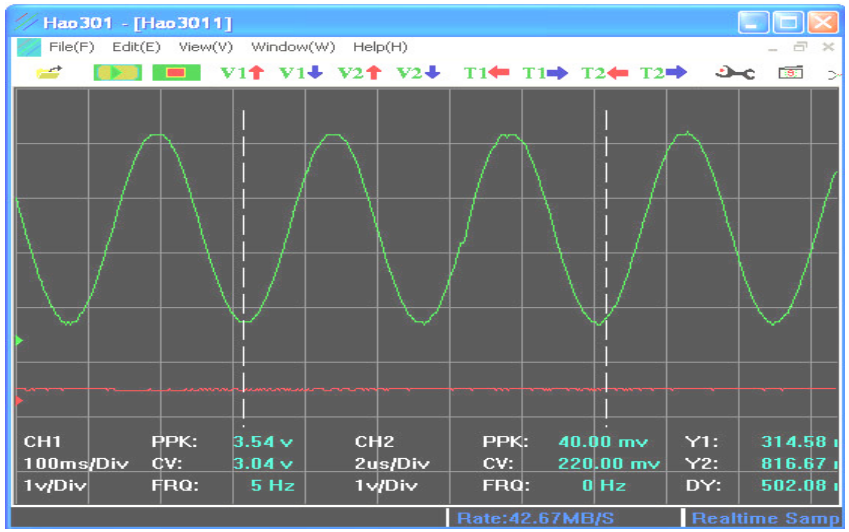
http://www.agilart.com/blog/tmp102-raspberry-pi

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

* 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

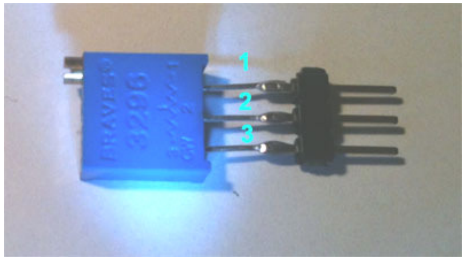
JP11 for analog output JP11 pin 1,2 (AOUT, GND)

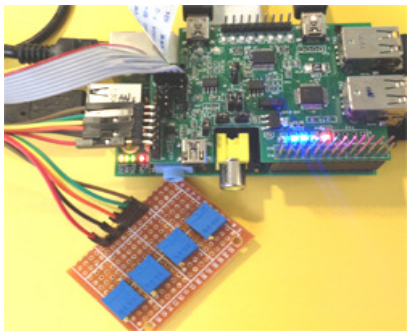


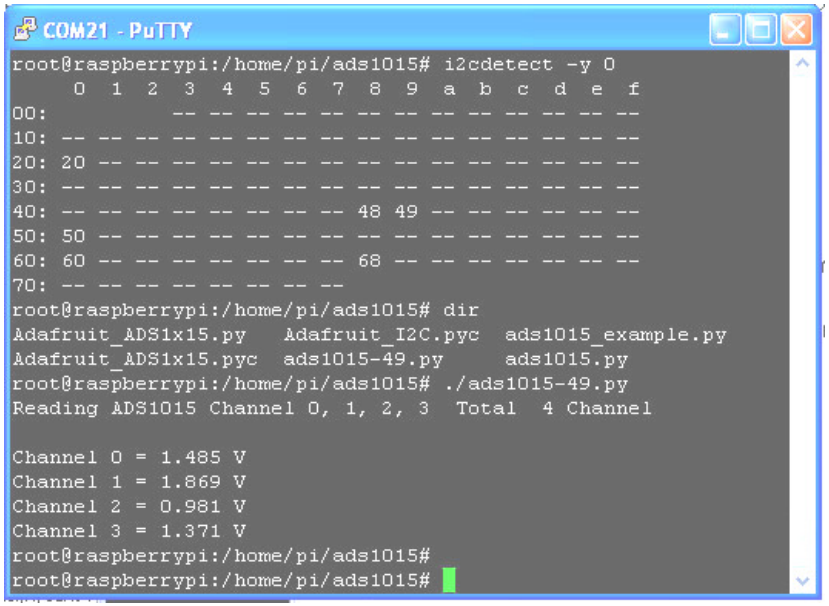
* **ADS1015 12bit Analog-to-Digital Converter**
demo 12 bit 4 channel input ads1015-49.py at address "49"

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

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

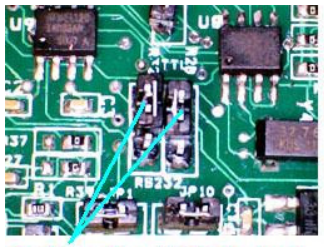




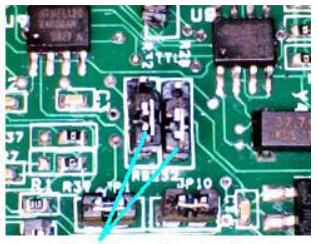


ADS1015 use 4 trimpot 10K OHM for 4 channel Analog input

*** FT232RL the driver for USB to TTL can download from our web site, use this function , you need plug –in USB cable install driver first, then plug-in Rs-Pi hub board to Rs-pi board. * the USB to RS232 you can choose USB from PC or Mac or Rs-Pi, then your PC or Mac or Rs-Pi can have RS232 port full 9pin DB9**



Setting for USB to TTL console Port



Setting for USB to RS232

Change R28 ,R32 Jumper you can change USB-TTL or USB-RS232 function

*** USB-TTL console function, access you Rs-Pi by a normal USB A to USB Mini B cable**
*** USB-RS232 make your PC or Mac or Rs-Pi have full function 9 pin RS232 Port**

For PC or Mac if you need driver you can download from our web site in number 19 FT232RL chipset

http://www.pridopia.co.uk/support.html

also the all the information for FT232RL
http://www.ftdichip.com/Products/ICs/FT232R.htm

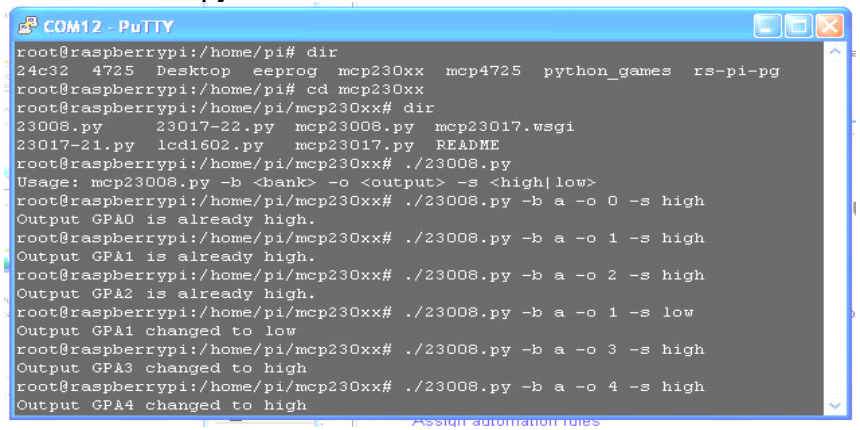
Driver for PC or Mac from FTDI web site
http://www.ftdichip.com/FTDrivers.htm

- **23008 23008 8 bit GPIO**

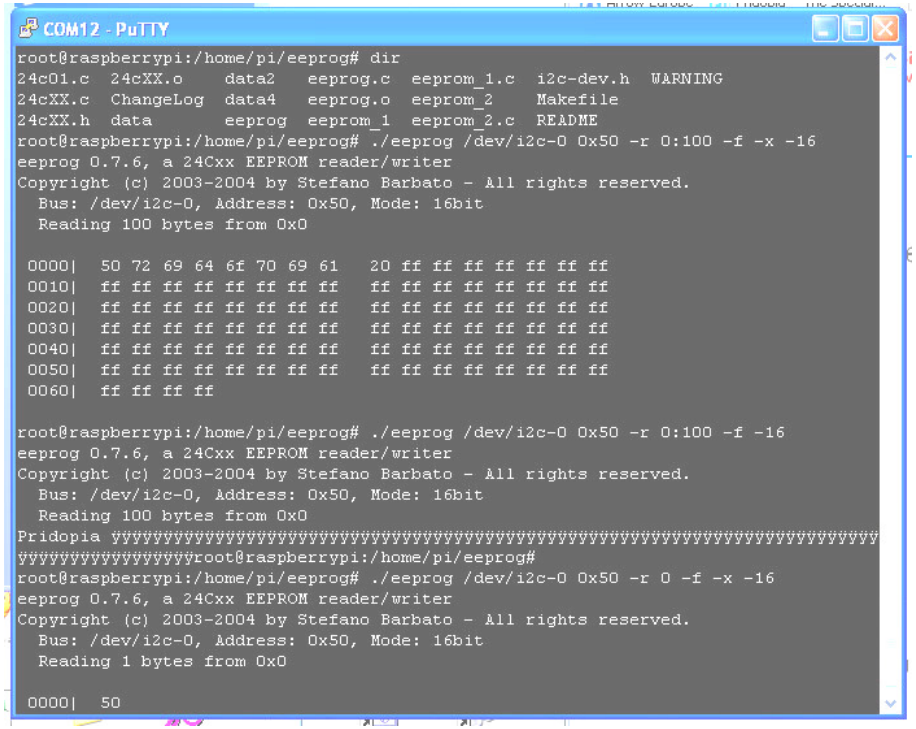
use 23008.py can set each bit output high or low

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

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



- **24c32 EEPROM**



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

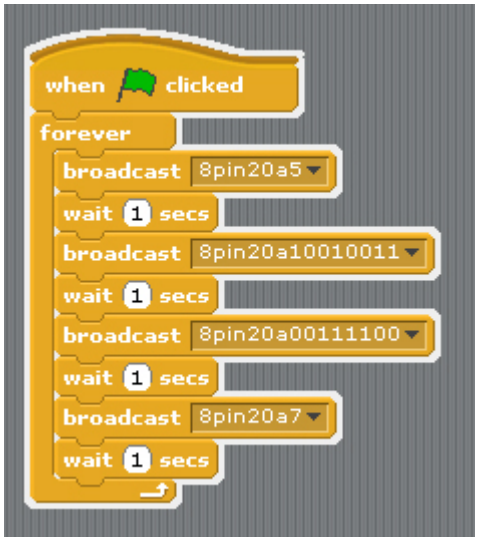
The Program can be download from

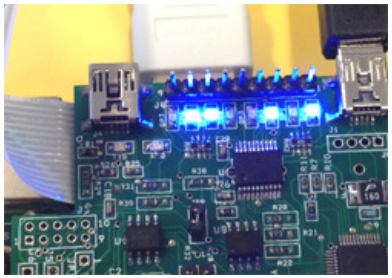
http://www.codesink.org/eeprog.html

New Pridopia **Pi_Scratch** interface software you can download from our web site, more easy to control by Raspberry pi

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

i2c address 20 23008 8 GPIO test





use command 8pin+"address"+a"bit (1 to 8)"

8pin+"address"+ a"10001000" --> 8bit mode

8pin20a5 --> output to address 20 bit 5

or use "8pin20a10010011" --> 8bit address 20 from 8 to 1 "10010011"

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

Package Content

1x Rs-Pi USB Hub & I2C AD/DA & FT232RL-R1 board
1x USB to MINI USB hub input cable (for USB Hub input & Power input)
1x 2.0mm pitch 2x5 to DB9 com port cable with 30cm length
1x CR1220 3V Battery
1x Manual