



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Raspberry Pi RTC Module SKU: DFR0386

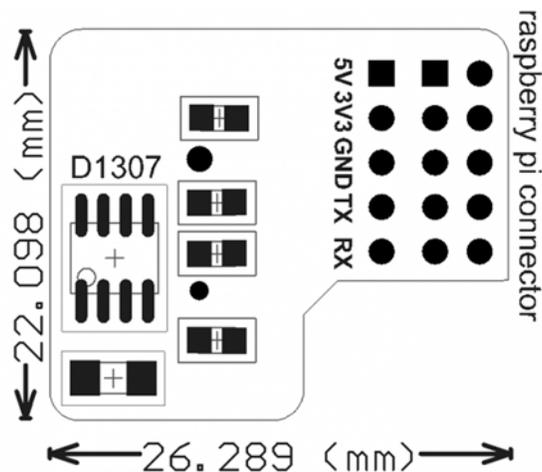
Introduction

The RTC module is specifically designed for Raspberry Pi. It communicates with Raspberry Pi through I2C bus. There is a Maxim DS1307 and CR1220 button cell on the board to keep the real time for a long time after the Raspberry Pi has its powerdown. Set a serial port, TTL convenient way online debugging.

Specification=

- RTC module: DS1307
- Battery model: CR1220 button cell
- Operating Voltage: 5V
- I2C address: 0x68
- Clock precision: $\pm 2\text{ppm}$ (0~40°C)
- Unit information: Second, Minute, Date, Week, Month and Year
- Two calendar clock
- Operating temperature: -10°C至+85°C
- Compatible with Raspberry Pi B/A+/B+/2B
- Interface: 2*5p 2.54mm

Dimension



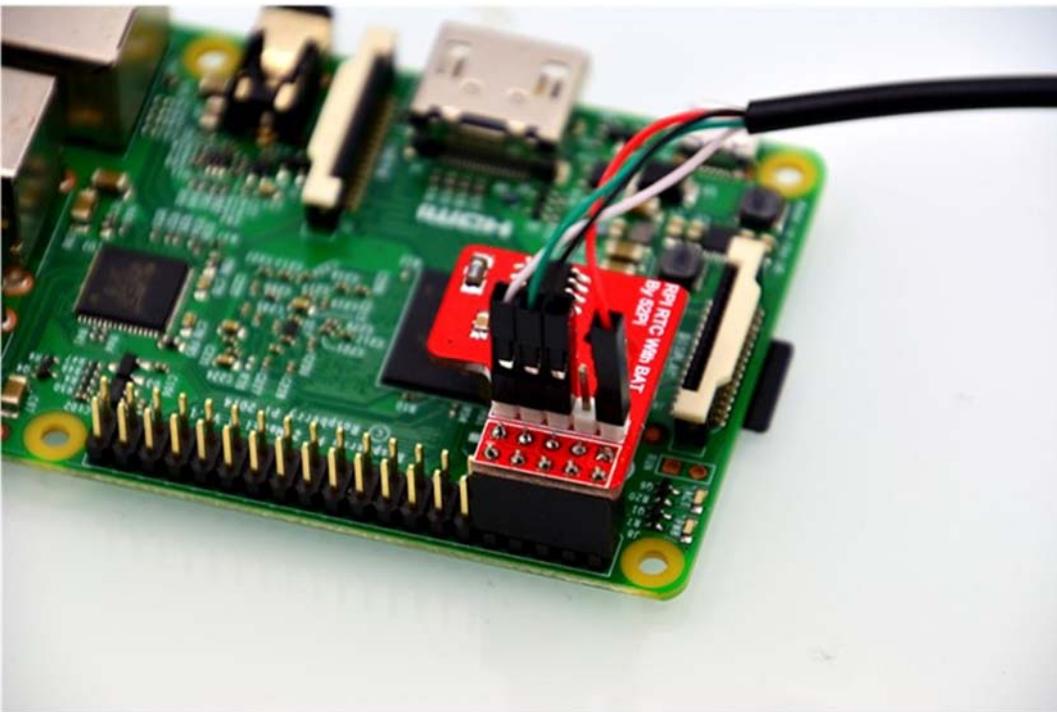
HOW TO USE

Connection

- Connect the module to your Pi



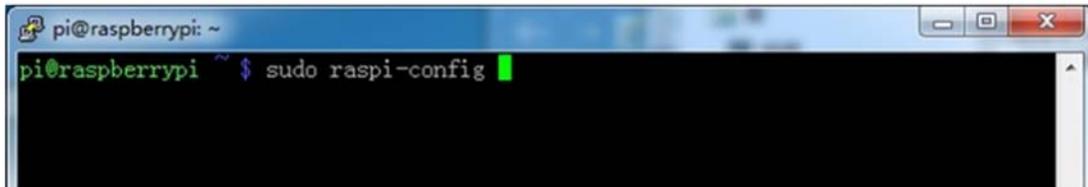
- The module leads to the TX&RX pins, you could set the information via this port.



NOTE: DO NOT power it again if the Raspberry Pi has been powered, or it will damage the module and Raspberry.

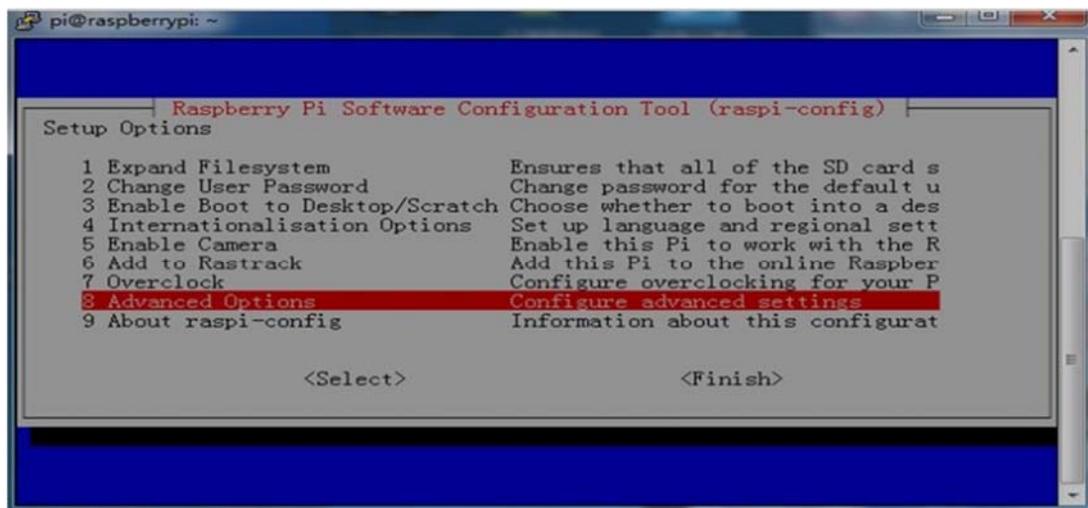
Test

- 1. Input "sudo raspi-config" to Open Raspberry Pi I2C interface



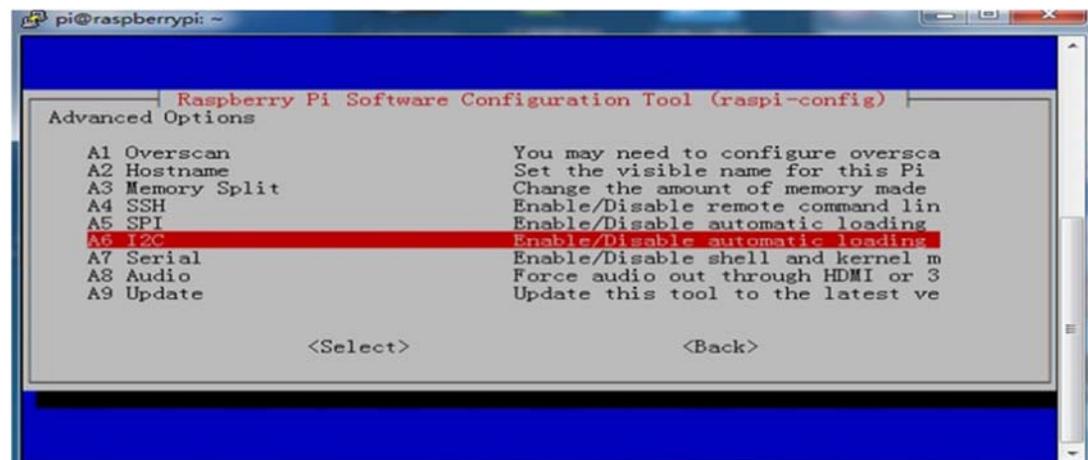
```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo raspi-config
```

- 2. Select "Advanced Options"



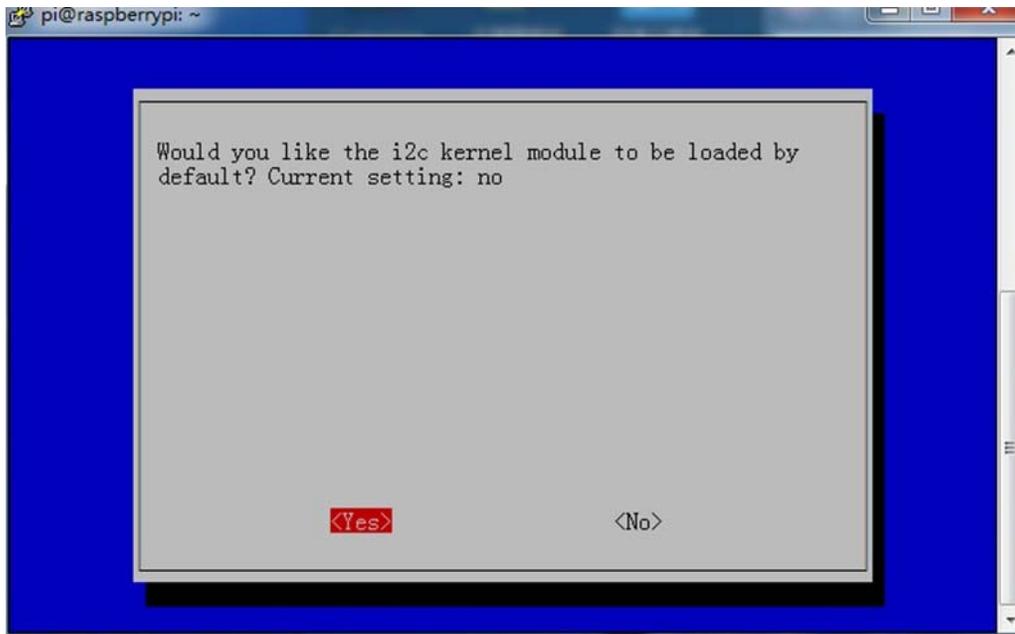
```
Raspberry Pi Software Configuration Tool (raspi-config)  
Setup Options  
1 Expand Filesystem           Ensures that all of the SD card s  
2 Change User Password       Change password for the default u  
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des  
4 Internationalisation Options Set up language and regional sett  
5 Enable Camera              Enable this Pi to work with the R  
6 Add to Rastrack            Add this Pi to the online Raspber  
7 Overclock                  Configure overclocking for your P  
8 Advanced Options           Configure advanced settings  
9 About raspi-config         Information about this configurat  
  
<Select>                       <Finish>
```

- 3. Select "I2C"

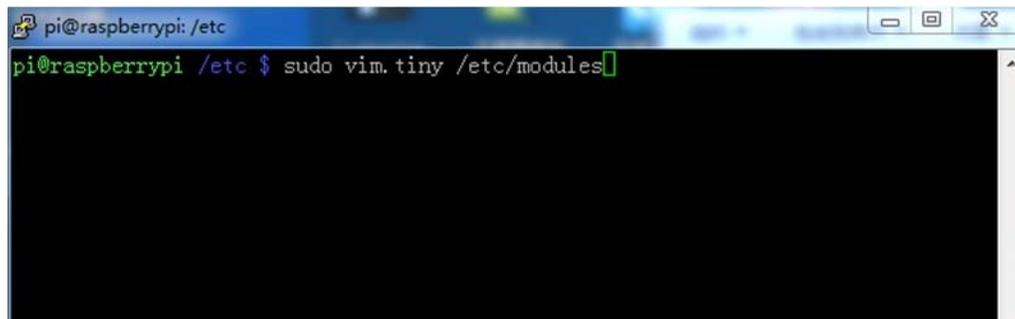


```
Raspberry Pi Software Configuration Tool (raspi-config)  
Advanced Options  
A1 Overscan                 You may need to configure oversca  
A2 Hostname                 Set the visible name for this Pi  
A3 Memory Split             Change the amount of memory made  
A4 SSH                      Enable/Disable remote command lin  
A5 SPI                      Enable/Disable automatic loading  
A6 I2C                      Enable/Disable automatic loading  
A7 Serial                  Enable/Disable shell and kernel m  
A8 Audio                    Force audio out through HDMI or 3  
A9 Update                   Update this tool to the latest ve  
  
<Select>                       <Back>
```

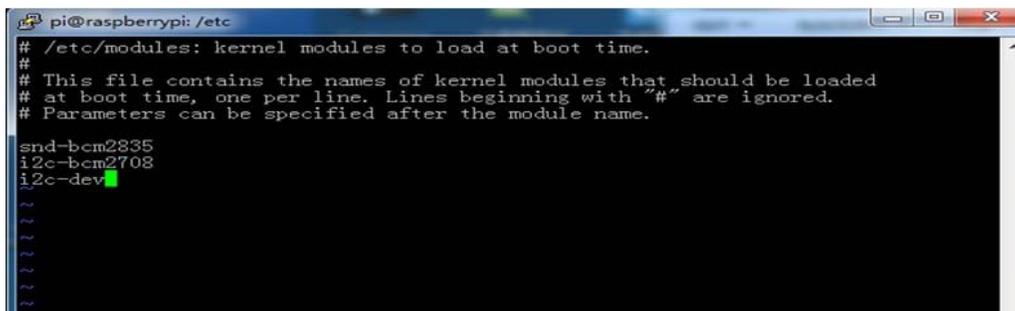
- 4. Select "YES"



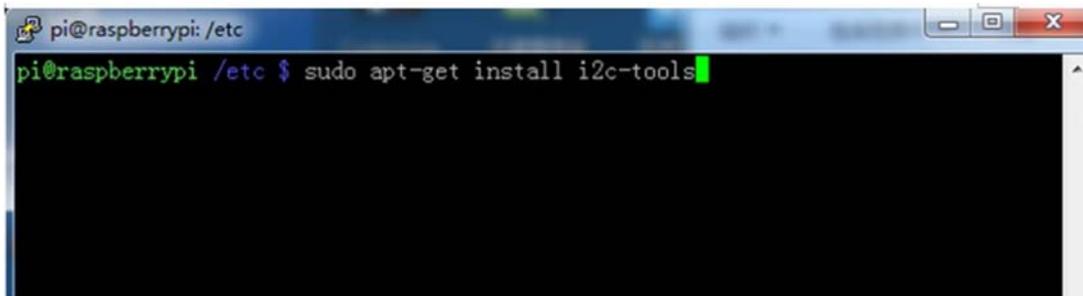
- 5. Input "sudo vim.tiny /etc/modules" to add the module



- 6. Add "i2c-dev" device

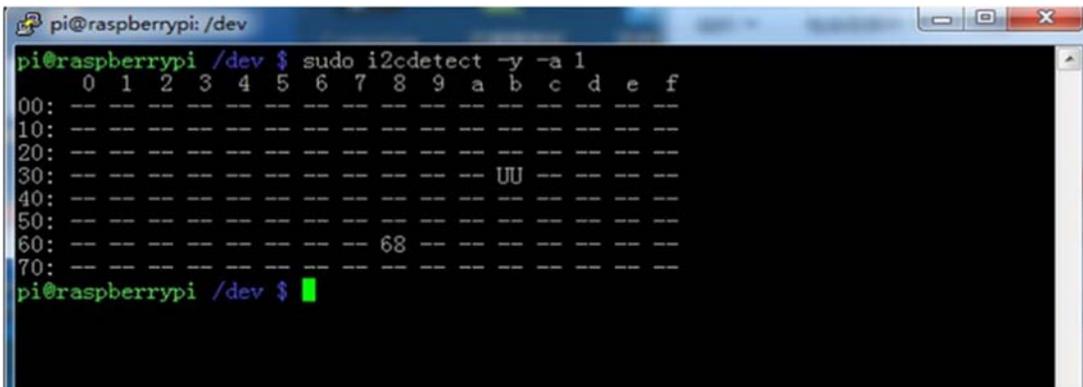


- 7. Install I2C tools, input "sudo apt-get install i2c-tools"



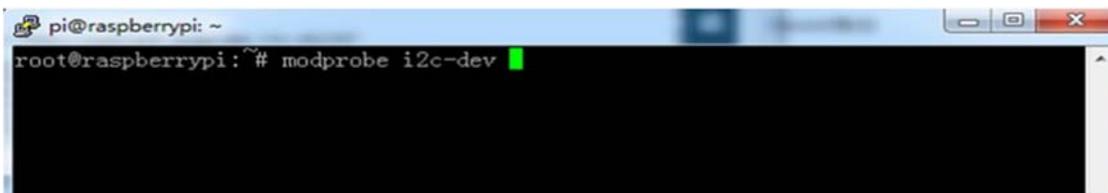
```
pi@raspberrypi: /etc
pi@raspberrypi /etc $ sudo apt-get install i2c-tools
```

- 8. Input "sudo reboot" to reboot Raspberry Pi; Input "sudo i2cdetect -y -a 1" after a reboot. If everything goes well, the module will be detected normally.



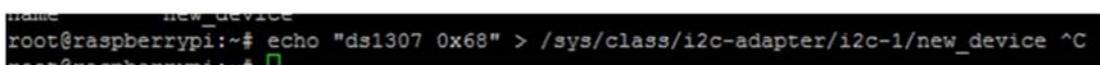
```
pi@raspberrypi: /dev
pi@raspberrypi /dev $ sudo i2cdetect -y -a 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi /dev $
```

- 9. Input "sudo su--" to get "root"; input "modprobe i2c-dev" to load I2C device.



```
pi@raspberrypi: ~
root@raspberrypi:~# modprobe i2c-dev
```

- 10. Input "echo "ds1307 0x68" >/sys/class/i2c-adapter/i2c-1/new_device" to load to Raspberry Pi system I2C device.

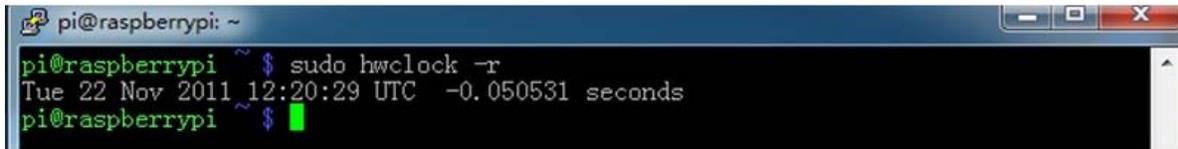


```
root@raspberrypi:~# echo "ds1307 0x68" > /sys/class/i2c-adapter/i2c-1/new_device ^C
root@raspberrypi:~#
```

- 11. Now you can use "hwclock" command to use this module, refer to "man hwclock" for more details.

"hwclock -r" Get RTC module time

"hwclock -w" Set system time



```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo hwclock -r  
Tue 22 Nov 2011 12:20:29 UTC -0.050531 seconds  
pi@raspberrypi ~$
```