

Raspberry Pi RTC Expansion Module User Manual

52PI-RPI-RTC

Maximum Power at Minimum Size

www.52pi.com

Raspberry Pi RTC Module User Manual

Contact Us

Raspberry Pi Geek Forum

kd@52pi.com

www.52pi.com

Version: 1.0

Data: 2015.6

Directory

1:Product Specification:	7
2:Produce Overview:	8
3:Produce Features:	8
4:Port:	8
5:Product Parameters:	8
6:Typical Application:	8
7:How to use:	9
8:More information:	11
8.1: schematic diagram:	11
8.2:test index:	12

Raspberry Pi Geek Forum Copyright Statement:

The document only describes the information about the product, however, it cannot guarantee the product function and performance. If the document content or the product feature and technical specification included in the document are changed, it will be without further notice.

Content in the document might be outdated. Our cooperation cannot promise to update this information.

Some information in the document might be disabled in your local area, that includes product and service. You can consult with the contact and agency in your local area.

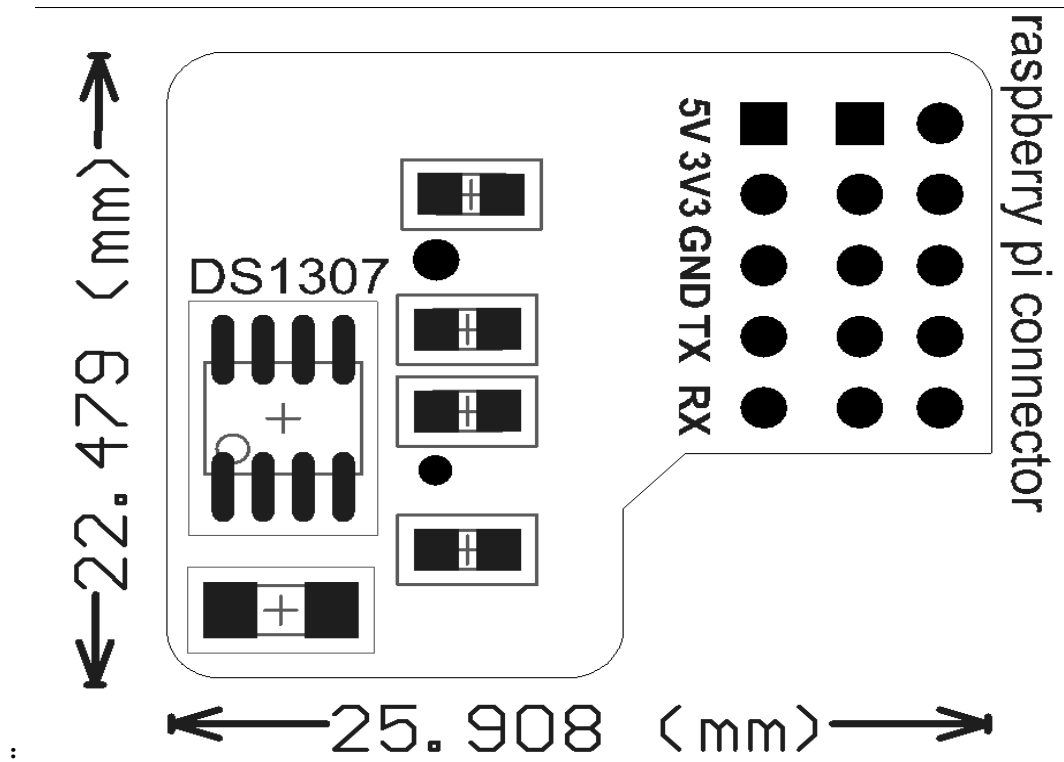
Copyright to the document belongs to Raspberry Pi Geek Forum. Users can only use this content after they or she get authorization from our company or other obligee. However, users cannot copy, paraphrase, or create similar devices or products.

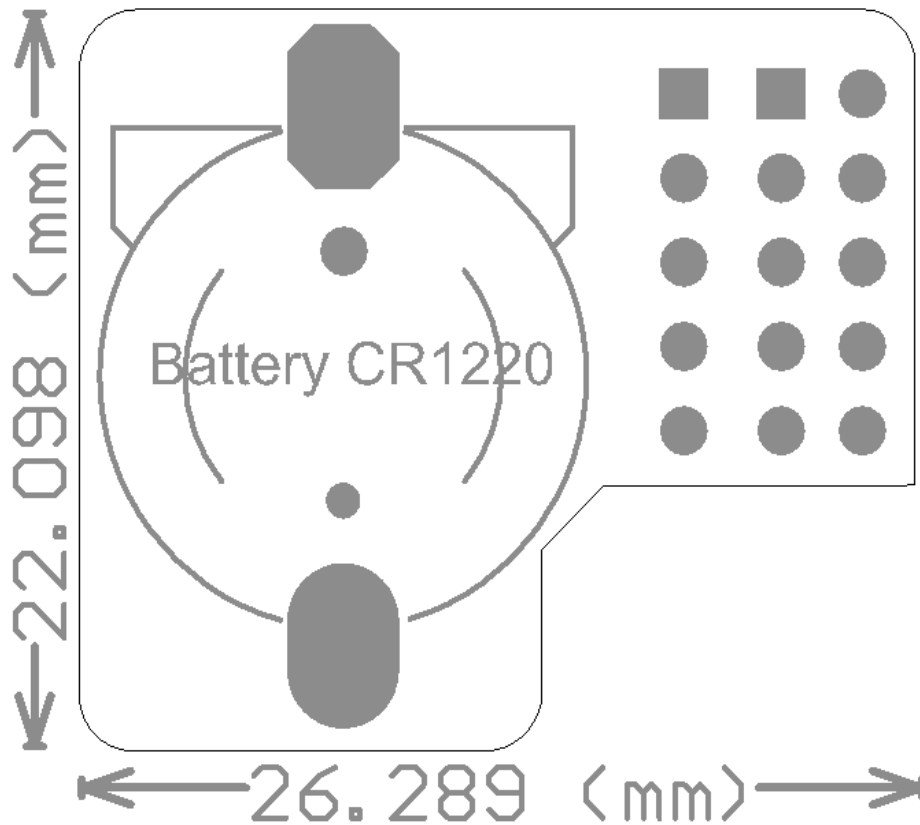
The document's final right to interpret and document use is the final interpretation, which belongs to Raspberry Pi Geek Forum.

More information:

Get more product and support, please contact Raspberry Pi Geek Forum (www.52pi.cn)

1: Product Specification:





3: Produce overview:

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 power down. Set a serial port, TTL convenient way online debugging

3: Produce Features:

- Use Maxim DS1307 chip
- Extends CR1220 button cell backup
- Can be operated by a shell
- Set a serial port connector
- Programmable square-wave output signal
- Consumes Less than 500nA in Battery-Backup Mode
- Automatic Power-Fail Detect and Switch Circuitry

- Programmable Square-Wave Output Signal

4: Port:

- Raspberry Pi B module 2X13 connection port
- Raspberry Pi A+/B+/2 module 2X13 connection port
- 2*5pin 2.54mm connector

5: Product Parameters:

- Accuracy $\pm 2\text{ppm}$ from 0°C to $+40^{\circ}\text{C}$
- Work voltage 5V
- Battery Backup Input for Continuous Time keeping
- Real-Time Clock Counts Seconds, Minutes, Hours, Day, Date, Month, and Year with Leap Year Compensation Valid Up to 2100
- Work temperature 0°C to $+70^{\circ}\text{C}$.

6: Typical Application:

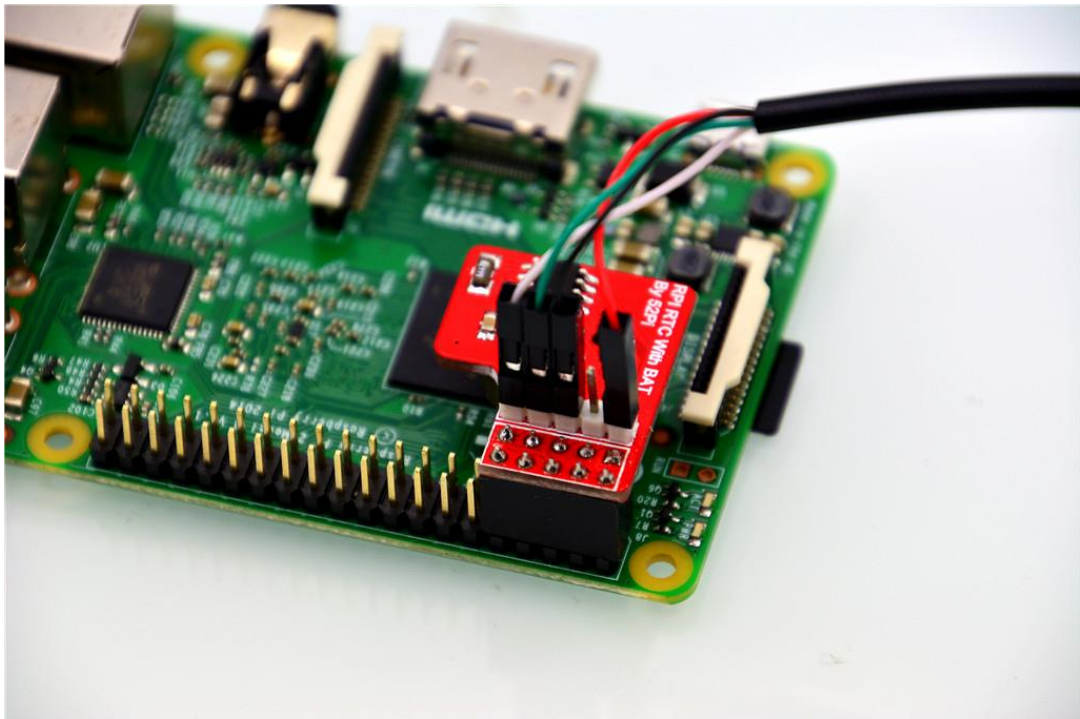
- Smart home
- Utility Power Meters
- Telematics

7: How to use:

Just insert the module to Raspberry Pi, like pictures below:



The instructions about serial port, like pictures below:



Serial Communication line Connect to module's 5V ,GND, TX,RX pin
(**caution**)You must not make raspberry pi powered by USB and serial line at the same time

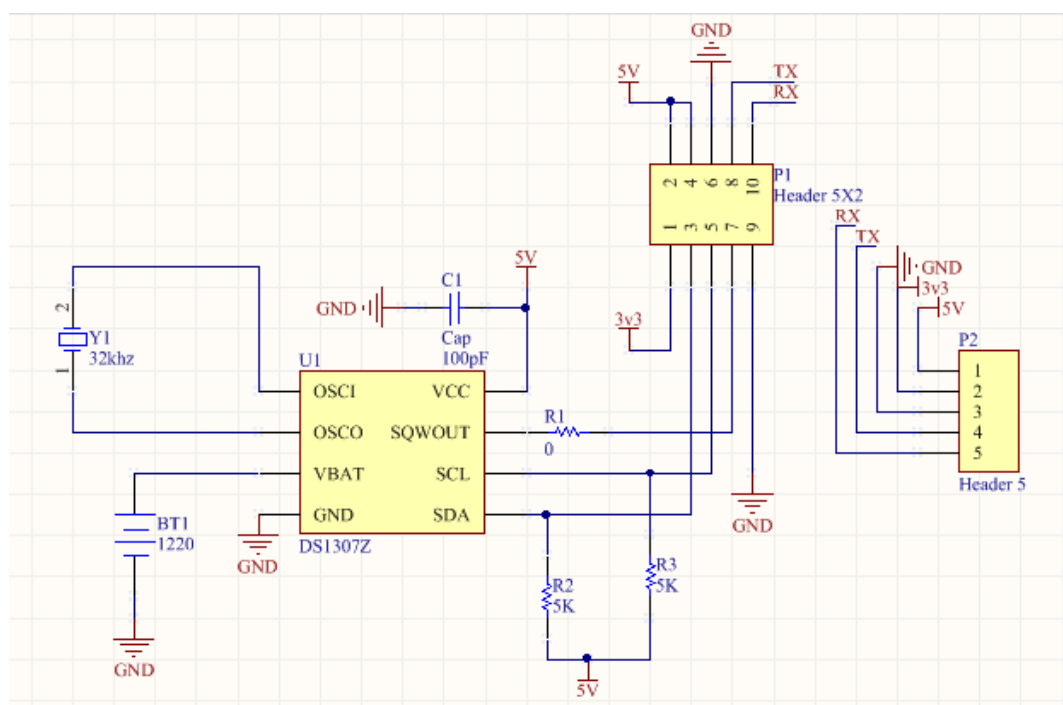
8: More information:

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal(BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2c, bidirectional bus. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. We select the 8-Pin SO package chip used in this Module and minimizes the required space

The RTC maintains seconds, minutes, hours, day, date, month and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format

The CR1220 backup battery will keep the RTC timekeeping no less than two years

8.1: Schematic diagram:

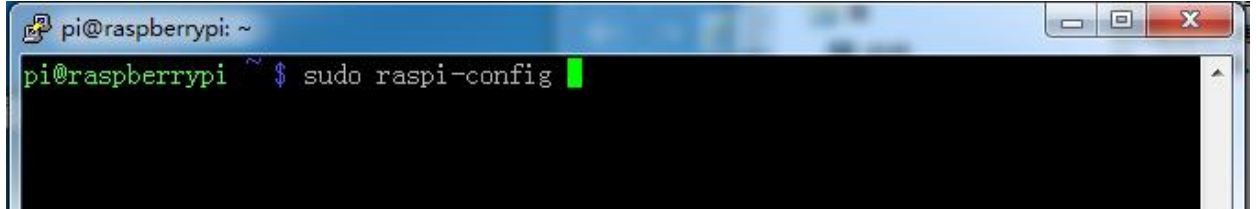


More information please check:

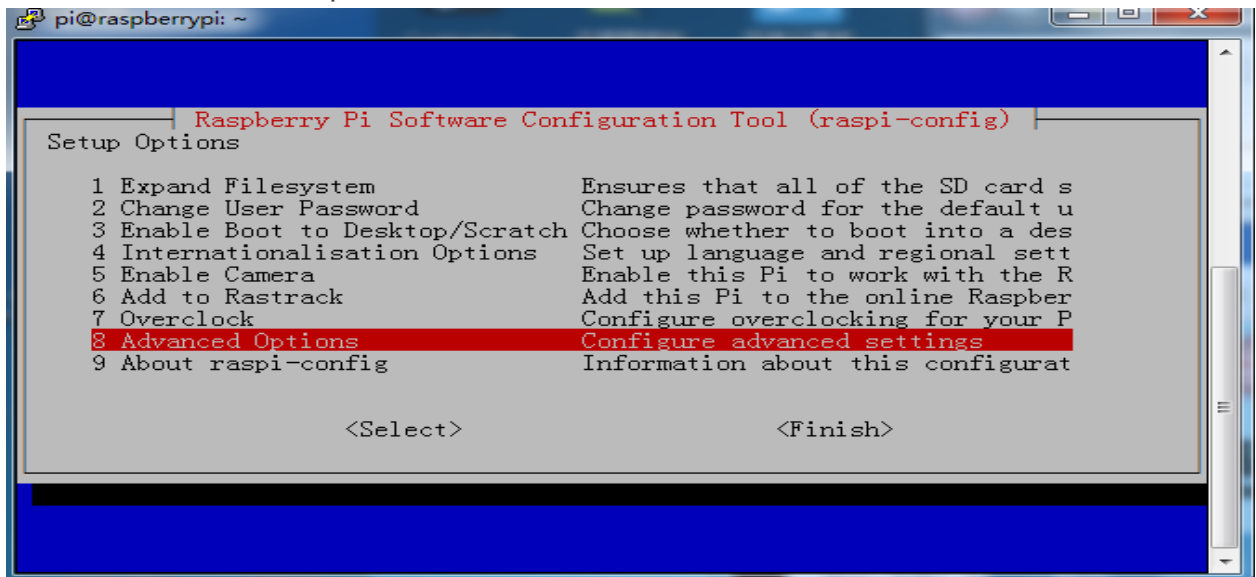
http://www.nxp.com/products/interface_and_connectivity/i2c/i2c_dacs_and_adcs/PCF8591T.html

8.2: Test bench:

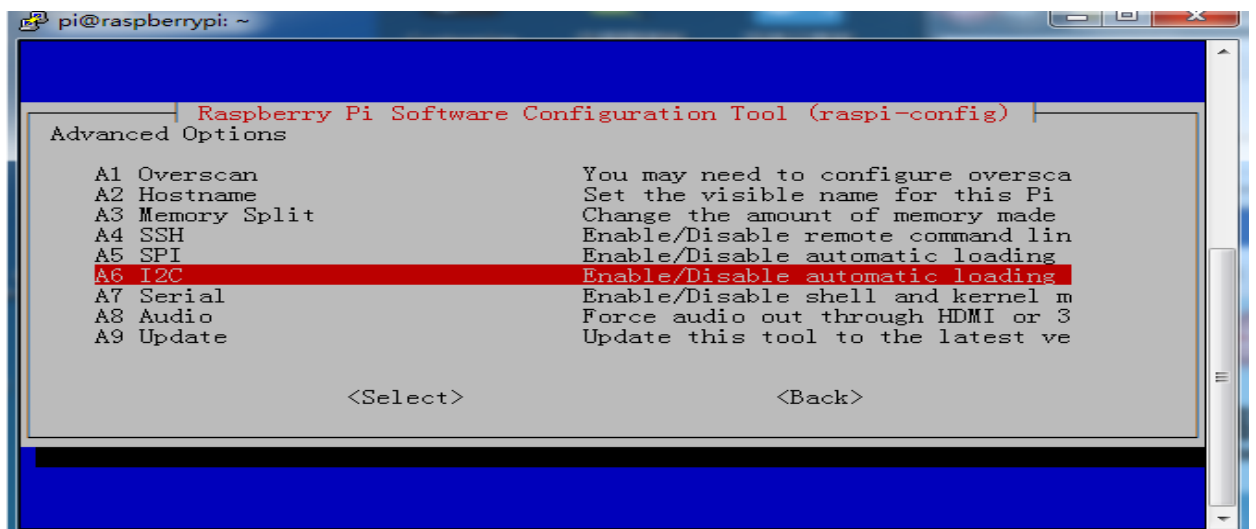
1. Type in [sudo raspi-config] set up raspberry pi and prepare to enable i2c interface



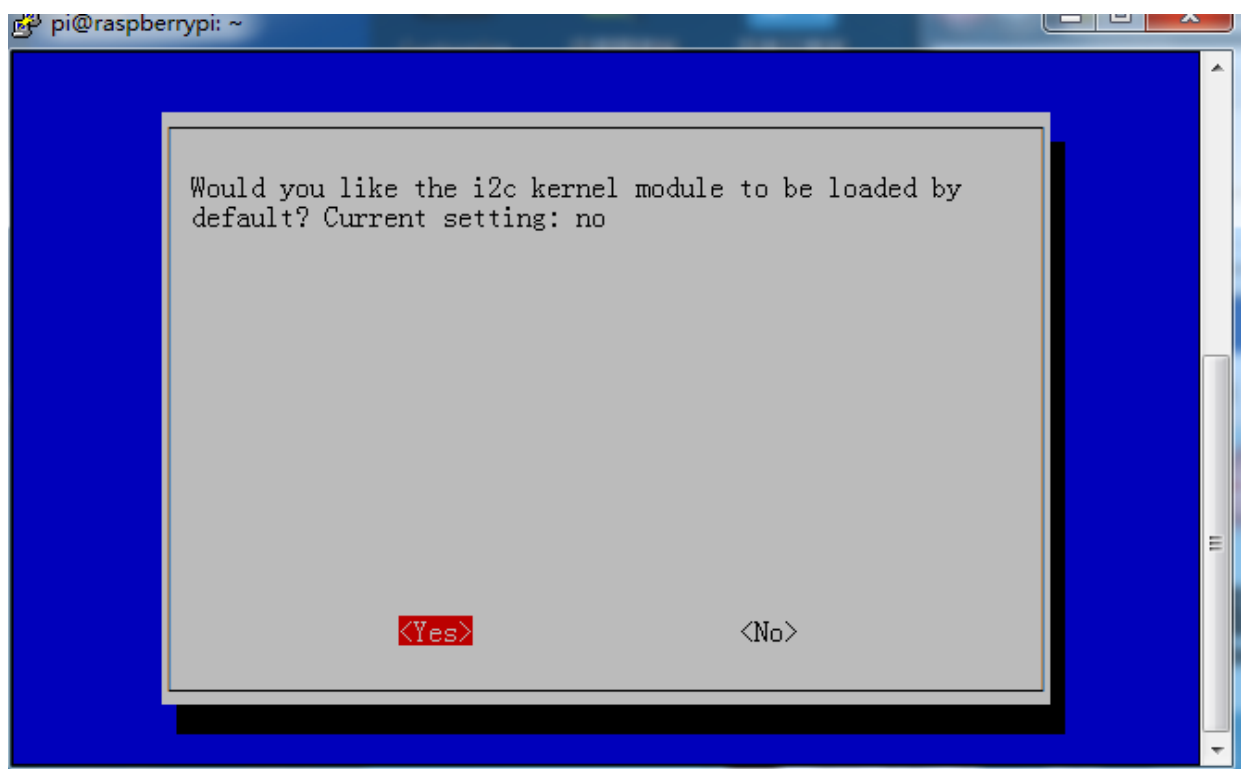
2. select Advanced Options



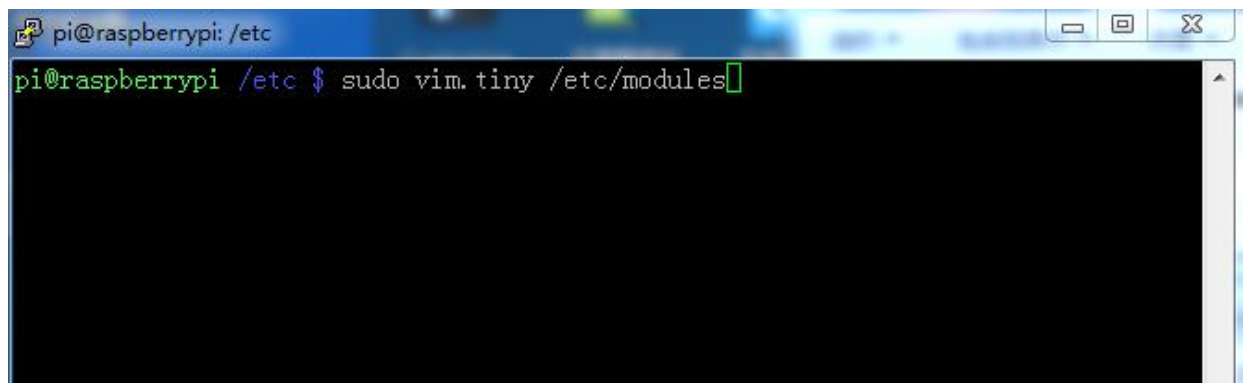
3. select I2c



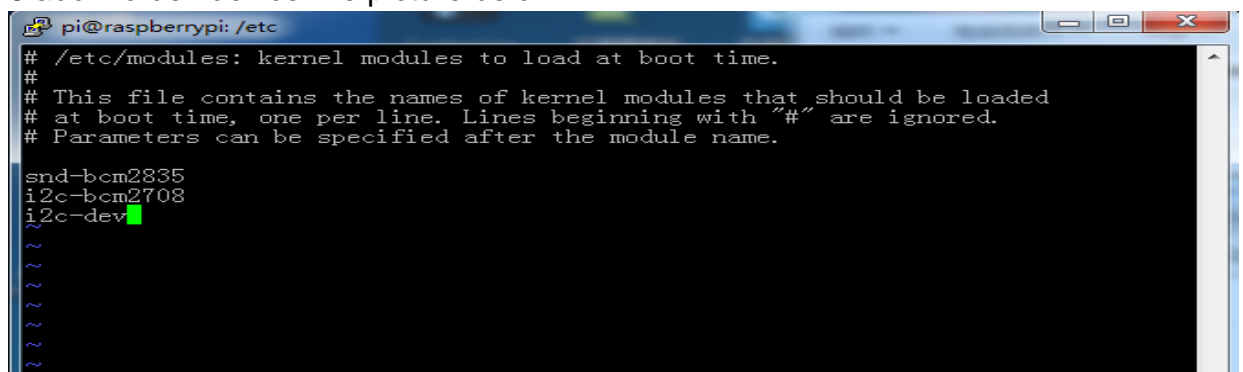
4. select yes enable I2C



5. type in [sudo vim.tiny /etc/modules] to open file modules

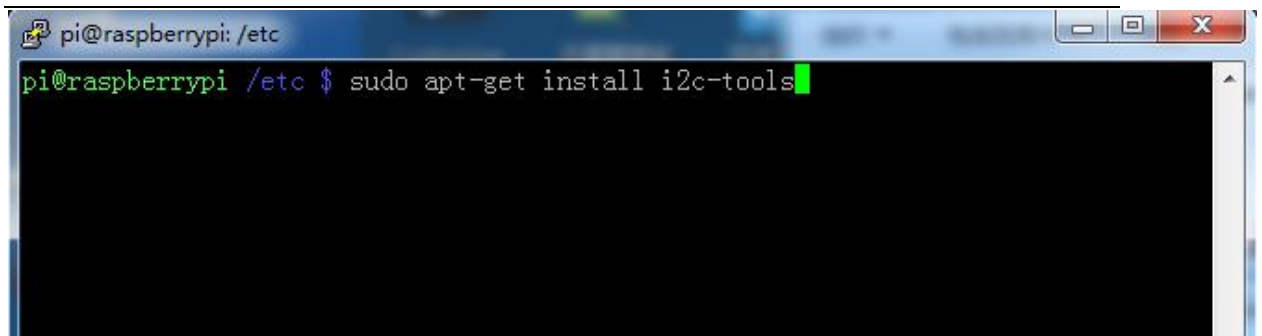


6.add i2c-dev device like picture below



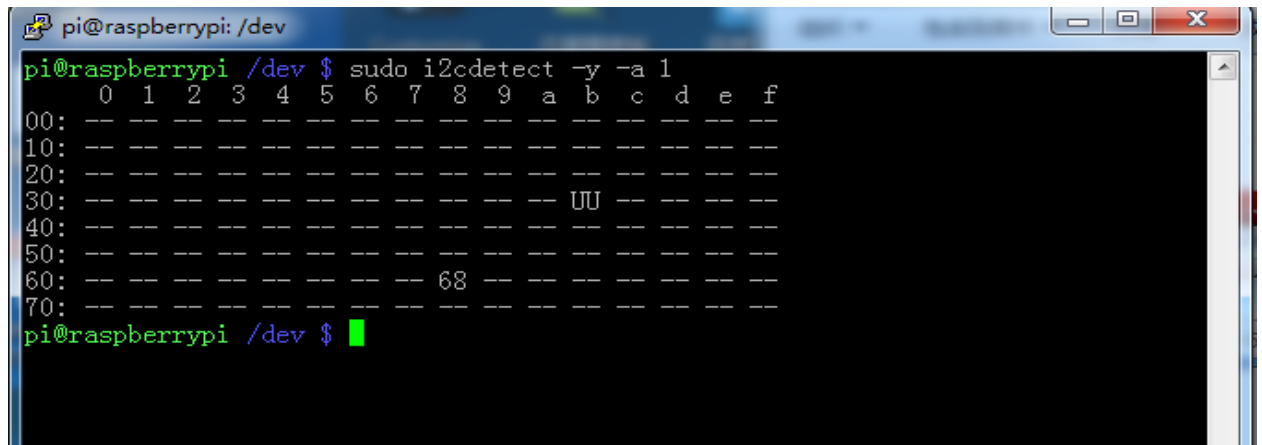
7. install i2c-tools ,type in [sudo apt-get install i2c-tools]

Raspberry Pi RTC Module User Manual



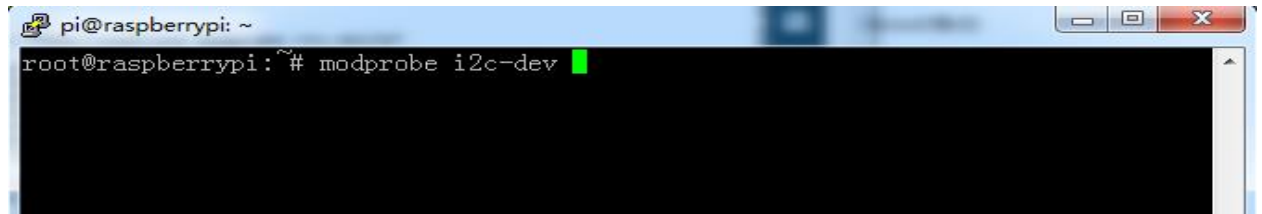
```
pi@raspberrypi: /etc
pi@raspberrypi /etc $ sudo apt-get install i2c-tools
```

8. type in [sudo reboot] wait the raspberry pi restart
Then type in [sudo i2cdetect -y 1], if RTC works ,it should be like picture below



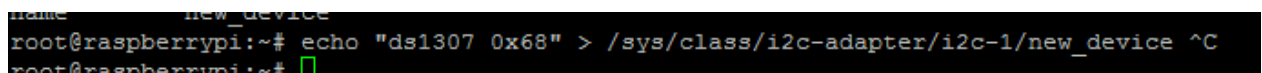
```
pi@raspberrypi /dev $ sudo i2cdetect -y -a 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: -- -- --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

Type in [sudo su -] change root user
Type in [modprobe i2c-dev] to load I2C device



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

Type in [echo "ds1307 0x68" > /sys/class/i2c-adapter/i2c-1/new_device] add RTC to system



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

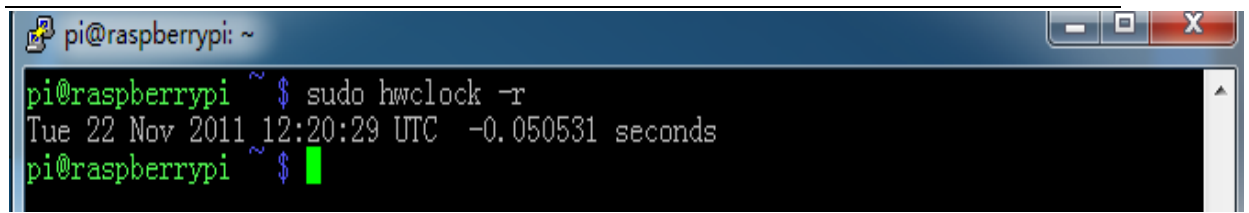
Then you can use “hwclock” command to use this RTC module
If you want to know more about hwclock command you can type in “man hwclock” to get details

A brief description like following:

Command [hwclock -r] to get RTC time

Command [hwclock -w]set the system time to RTC time

Like following



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