

How to use RTC module to work as a stable clock on Raspberry Pi

As there is no clock module on Raspberry Pi, files may crash due to the time error. Therefore, you need a RTC module to work as a stable clock. It takes little space and it is quite easy for you to learn how to use it. The product looks like the below picture:



DS1307 RTC Module

This RTC module is specifically designed for Raspberry Pi. You may connect it with Raspberry Pi via I2C bus. It includes DS1307 and CR1220 button battery to ensure that Raspberry Pi can display the right time. There are five PINS: 5V , 3.3V , GND, RXD and TXD on this RTC for easier installation.

Electronic Components:

1×Raspberry Pi A+/B/B+/2B/3 Board

1×New RTC for RPi Module

1×USB-to-TTL USB Cable

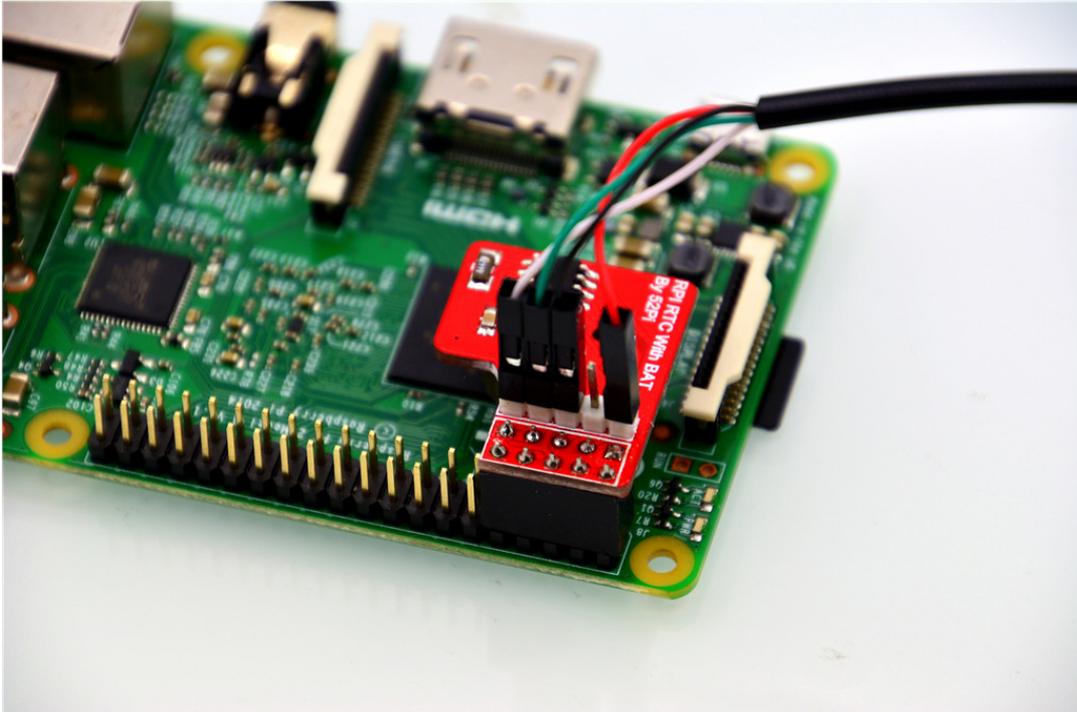
System Requirements: Raspbian System

Installation Manual:

All you need is to insert the RTC module. Please refer to the following illustration:



USB-to-TTL Wire illustration:



Note:

- RTC 5v to USB-to-TTL Red Cable (5v)
- RTC GND to USB-to-TTL Black Cable (GND)
- RTC TX to USB-to-TTL Green Cable (TX)
- RTC RX to USB-to-TTL White Cable (RX)

Operating Procedures:

1. Burn the Raspbian System first. Boot and log in.
2. Edit the configuration file file /boot/config.txt

Here is the picture for your reference:



```
pi@raspberrypi:~$ !grep
grep -v "#" /boot/config.txt |grep -v "^$"
hdmi_safe=1
disable_overscan=0
overscan_left=16
overscan_right=16
overscan_top=16
overscan_bottom=16
hdmi_force_hotplug=1
dtparam=i2c_arm=on
dtparam=spi=on
device_tree=bcm2710-rpi-3-b.dtb
dtoverlay=seed-tft28b,speed=32000000,rotate=180
dtoverlay=pi3-miniuart-bt-overlay
dtoverlay=i2c-rtc,ds1307
force_turbo=1
dtoverlay=ads7846,penirq=22
dtparam=audio=on
```

The setup is mainly based on /boot/overlay/README. This RTC module is supported by official Raspberry Pi. All you need is to edit the /boot/config.txt and it will take effect. Here are the excerpts from README for your reference:

Name: i2c-rtc
Info: Adds support for a number of I2C Real Time Clock devices
Load: dtoverlay=i2c-rtc,<param>=<val>
Params: ds1307 Select the DS1307 device

The parameters we need to debug are as follows:

device_tree=bcm2710-rpi-3-b.dtb
dtoverlay=i2c-rtc,ds1307
dtparam=i2c_arm=on

Please ensure that your /boot/config.txt includes the entire above configuration.

Restart the system and then pass the following test:

dmesg |grep rtc

If you are able to view the following information, it shows that the driver of RTC module has been successfully loaded.



```
pi@raspberrypi:~$ dmesg |grep rtc
[ 6.663981] rtc-ds1307 1-0068: rtc core: registered ds1307 as rtc0
[ 6.664028] rtc-ds1307 1-0068: 56 bytes nvram
```

Set the system time by date command in the order of mmDDHHMMYYYY.ss

```
pi@raspberrypi:~$ sudo date 051014302016.20
Tue May 10 14:30:20 UTC 2016
```

You may refer to the manual for specific date parameters.

Final step is to synchronize the system time to the hardware clock (RTC clock)

```
pi@raspberrypi:~$ sudo hwclock -w step 1
pi@raspberrypi:~$
pi@raspberrypi:~$
pi@raspberrypi:~$ date && sudo hwclock -r step 2
Tue May 10 14:37:54 UTC 2016
Tue May 10 14:37:54 2016 -0.010701 seconds
pi@raspberrypi:~$
```

Through these two commands, clock synchronization has been completed. You can turn off your computer without any concern of date loss caused by timestamp error.