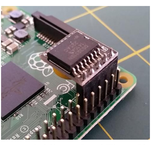

Adding a Real Time Clock to your Raspberry Pi

By The Pi Hut • Feb 18, 2011 • 2 comments

When you switch on your PC or Mac, do you have to change its time? Highly unlikely! Do you have to on your Raspberry Pi? Possibly.

Each time your Raspberry Pi boots it connects to what is called an NTP - Network Time Protocol - server and asks it the time. But what happens if your Pi is not connected to the internet? You're out of luck (or is that time?). Your Pi has no way of knowing what time it is.

But there is a solution in the form of a 'Real Time Clock' or RTC board. These inexpensive add-on boards plug on top of your Pi's GPIO pins and contains a clock chip and a tiny battery that tell the Raspberry Pi what time it is. The Pi Hut sells such a board.

Setting up I2C

The RTC board uses what is known as the I2C (pronounced 'eye squared see') protocol to communicate between itself and the Raspberry Pi. The I2C allows the Raspberry Pi to communicate with a lot of different devices at the same time that are also connected to the same I2C pins on the Raspberry Pi.

Simon Monk has written very good instructions about how to set up your Raspberry Pi to be able to use I2c on the Adafruit site. Follow the full instructions, including checking that everything has worked by going through the manual instructions too.

Plugging in the Board

It is always best to turn your Raspberry Pi off before plugging anything onto the GPIO pins. Always do this using the `sudo halt` or `sudo shutdown -h now` commands.

Now plug the board onto your Raspberry Pi GPIO pins on the inside corner with the RTC board pointing inwards, as shown below.

Setting up the RTC

You now need to configure your Raspberry Pi to recognise and use the board.

Turn on the Raspberry Pi and verify that the board is seen by running `sudo i2cdetect -y 1` at the command line. You should see ID #68 show up (the I2C address of the DS1307 Real Time Clock chip):

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

If you have an older Rev. 1 Raspberry Pi, you will have to run `sudo i2cdetect -y 0` as the I2C bus address changed from 0 to 1 from rev 2 onwards.

Setting the RTC Clock

The RTC module must be loaded by the kernel by running:

```
sudo modprobe rtc-ds1307
```

Now you need to be running as the super user; type in:

```
sudo bash
```

and then, if you have a Rev.2 Raspberry Pi or later:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1
/new_device
```

Or if you have the older Rev.1 Raspberry Pi, use:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0
/new_device
```

Type `exit` to stop being the 'root' user.

Check the time on the RTC device using:

```
sudo hwclock -r
```

If this is the first time the module has been used, it will report back Jan 1 2000, and you'll need to set the time to the current time.

The easiest way is to connect it up to Ethernet or Wifi. It will automatically set the time from the network. You can check the time using the `date` command:

```
root@raspberrypi:~# date
Wed Feb 23 12:09:28 UTC 2011
root@raspberrypi:~#
```

Run `sudo hwclock -w` to write the system time to the RTC board.

You can then verify it with `sudo hwclock -r`:

```
root@raspberrypi:~# date
Wed Feb 23 12:09:28 UTC 2011
root@raspberrypi:~# hwclock -r
Wed 23 Feb 2011 12:09:35 PM  -0.125776 seconds
root@raspberrypi:~#
```

So, now the RTC clock is set, is that it? Well, no. The next time you boot your Pi, it will not know that it needs to read the clock from the RTC, so if it is not connected to the internet it will again be showing the wrong time.

Making Your Raspberry Pi Use the Real Time Clock

You will need to add the RTC kernel module to the file `/etc/modules` so it is loaded when the Raspberry Pi boots. In a terminal window run `sudo nano /etc/modules` and add `rtc-ds1307` at the end of the file:

```
GNU nano 2.2.6          File: /etc/modules
# /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
i2c-bcm2708
i2c-dev
rtc-ds1307
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Next you will need to add the DS1307 device creation at boot by editing the `/etc/rc.local` file by running

```
sudo nano /etc/rc.local
```

and add the following lines to the file:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1
/new_device
sudo hwclock -s
```

```
date
```

Just before the `exit 0`. ***Note:*** *If you have a Rev 1 Raspberry Pi, replace `i2c-1` by `i2c-0` above.*

```
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-1/new_device
sudo hwclock -s
date
exit 0
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

To save the file, press Ctrl+X, Y then return.

To test this out, shutdown your Raspberry Pi, unplug any ethernet cable or wifi dongle and then turn the Raspberry Pi back on. Use the following command in a terminal window to check the date is still set:

```
date
```

Now, the next time you reboot your Raspberry Pi it will read the time from the Real Time Clock.

2 comments

The Pi Hut
February 10, 2023 at 11:05 am

@Hugh Burnham You're absolutely right, it does now use a CR927 battery but it seems we didn't update the page and specs when that change was made. Poor show from us! We'll get it all updated today. Thanks for the nudge.

Hugh Burnham
February 9, 2023 at 2:20 pm

We bought a couple of these to use in a project at work. I was trying to work out how long the supercapacitor would hold the date and time for. It is an important parameter and should be in the spec. If for example the super cap is 0.1F, then at the discharge rate specified in the LC datasheet then it would last only about 4 or 5 days. However I notice that the dimensions of the "supercapacitor" on the board are exactly the same as a CR927 lithium cell, and I suspect that is actually what is fitted. In which case it will last about 4.5 years. (35mAh, 0.84uA). Much better.
Perhaps the misleading reference to supercapacitor should be removed from the advertising?

Leave a comment

All comments are moderated before being published.

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Name | E-mail

Message

SUBMIT

Featured Products

Raspberry Pi 3 Model B+
£40 - Incl. VAT
NOTIFY ME

Mini RTC Module for Raspberry Pi
£5 - Incl. VAT
ADD TO CART

© The Pi Hut 2023