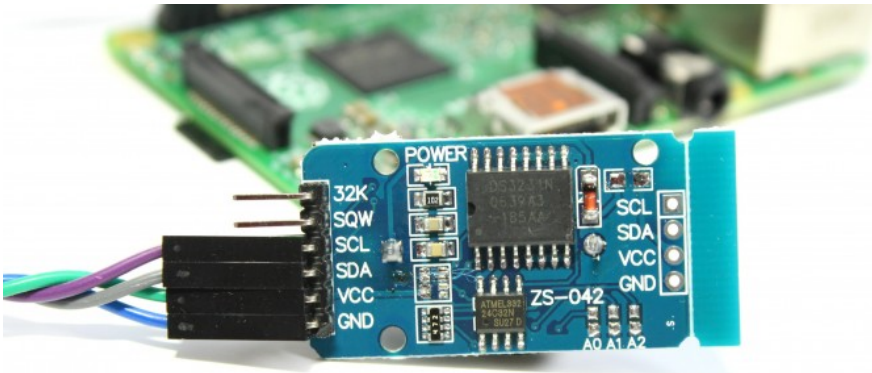
[HOME](#)[CATEGORIES](#) ▾[BERRYCLIP](#) ▾[BUY](#) ▾[TOOLS](#) ▾[TUTORIALS & HELP](#)[CONTACT US](#)[SITE MAP](#)YOU ARE AT: [Home](#) » [Hardware](#) » [Interfaces](#) » [I2C](#) » [Adding a DS3231 Real Time Clock To The Raspberry Pi](#)

Adding a DS3231 Real Time Clock To The Raspberry Pi

19

BY MATT ON MAY 25, 2015

I2C, TUTORIALS & HELP



While your Pi is connected to a network it will be able to set it's clock correctly using NTP. Without a network connection the system time and date will almost certainly be wrong. For some projects this is a problem especially if you are logging timestamps or performing other time sensitive operations.

This can be solved using a Real Time Clock (RTC) module. This will use a small coin cell battery to keep the time for the Pi even if it is turned off. When the Pi reboots it can set it's own internal clock using the time held in by the RTC.

The easiest way of implementing an RTC is to buy a pre-made module. Luckily RTC modules are relatively cheap and easy to obtain. Some modules will attach directly to the Pi's GPIO header but these may get in the way of other devices so I prefer more generic modules that keep the GPIO clear for other purposes.

DS1307 or DS3231?

Modules based on the DS1307 and DS3231 chips are popular devices and you'll see them for sale from various retailers. I purchased both types and quickly realised my DS1307 modules were useless. My advice would be to go for a DS3231 based module. They are more accurate and run happily from 3.3V. My two "Tiny RTC" DS1307 modules went straight in the bin.



Resistor Removal

Most generic I2C modules have pull-up resistors on the SDA and SCL pins but these aren't required as the Pi has it's own pull-ups. I didn't want these on-board resistors to interfere with the operation of the I2C bus so I removed them.

The module also has a basic charging circuit for use with a rechargeable LIR2032 coin cell. I wanted to use a normal

[RECENT](#) [POPULAR](#) [NEWS](#)

MAY 1, 2016

1

[Using the BBC Microbit with the Raspberry Pi](#)

APRIL 28, 2016

0

[Kano Computer Kit First Impressions](#)

APRIL 25, 2016

0

[8-megapixel Raspberry Pi Camera Module v2](#)

APRIL 11, 2016

1

[Write SD Card Images Using Etcher On Windows, Linux & Mac](#)

MARCH 24, 2016

0

[How to Change Your Character Skin In Minecraft Pi Edition](#)

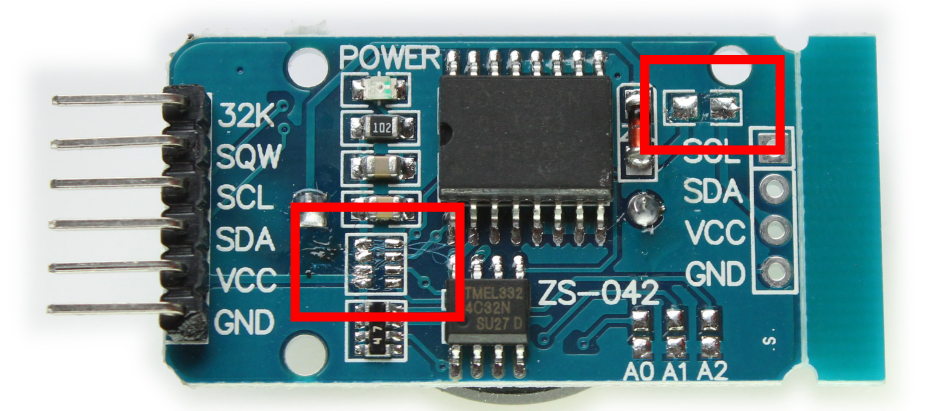
Search ...

Search

CATEGORIES

[3D Printing](#)[Add-ons](#)[BBC Micro:bit](#)[BerryClip](#)[Books](#)[Camera Module](#)

non-rechargeable cell so I removed another resistor to disable this charging circuit. The resistors were removed from the module using a soldering iron and some tweezers or small pliers. The location of the resistors is shown in the photo below :



The pull-up resistors are the block marked in red on the left. The charging circuit is the block marked in red on the right.

I2C Setup

As with all I2C devices you must configure the I2C interface. This is quite easy to do and explained in my [Enabling The I2C Interface On The Raspberry Pi](#) tutorial.

DS3231 Module Setup

In order to ensure you've got the latest updates you should run the following commands :

```
sudo apt-get update
sudo apt-get -y upgrade
```

Now we need to modify a system file using :

```
sudo nano /etc/modules
```

If it isn't already there add "rtc-ds1307" to the bottom so it looks something like :

```
snd-bcm2835
i2c-bcm2835
i2c-dev
rtc-ds1307
```

You can save and quit using CTRL-X, Y and ENTER.

Shutdown the Pi using "sudo halt" and remove the power when it has completed the process.

Hardware Setup

With the Raspberry Pi turned off you can now connect the module. It's a fairly simple configuration but it needs to be correct so double check it. Refer to my [GPIO Diagram](#) to ensure you get the correct pins on the Pi's header.

- General
- Cases
- Events
- General
- Hardware
- I2C
- Infographics
- Minecraft
- Model A+
- Model B+
- News
- Pi 2 Model B
- Pi Models
- Pi Zero
- Power
- Programming
- Python
- Raspbian
- Robotics
- Sensors
- Software
- SPI
- Tutorials & Help

TAGS

3D Printing

44780

audio

avconv

battery

berryclip

Birthday

bluetooth

cambridge

camera

CamJam

cpu

DigiMakers

display

GPIO

i2c

Kickstarter

lcd

LED

Li-on

Linux

Minecraft

Model A

Model B

mp4

pcb

photography

photos

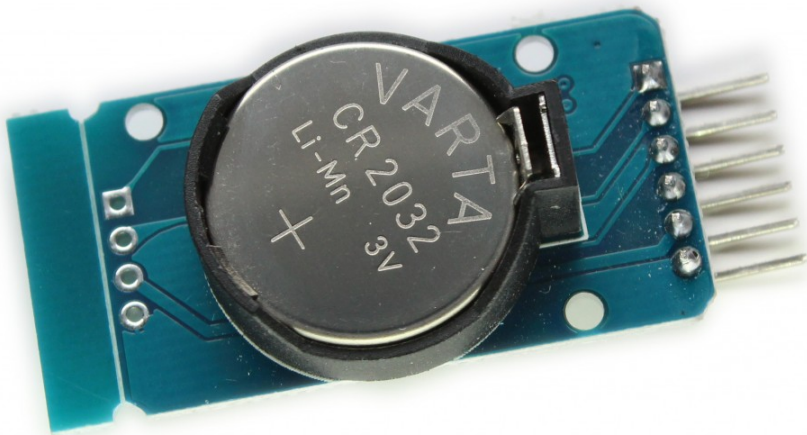
Pi-Lite

portable

power

DS1307	Pi GPIO
GND	P1-06
Vcc	P1-01 (3.3V)
SDA	P1-03 (I2C SDA)
SCL	P1-05 (I2C SCL)

Don't forget the 3V battery!



Interface Test

Power up the Pi and run the following command :

```
sudo i2cdetect -y 1
```

Note : If you are using a Rev 1 you will need to use “sudo i2cdetect -y 0”.

You should see something similar to this :

```
pi@rpispy ~ $ 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: -- -- -- -- -- -- 57 -- -- -- -- -- -- --
60: -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

In this example “68” is the hex address of the RTC module on the I2C interface.

I2C Device Setup

To ensure the DS1307 device is setup and the time synchronised when the Pi boots we need to edit another system file :

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

Add the following two lines before the exit 0 line :

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

python

Raspberry Pi Bootcamp

raspbian

ravpower

revision

sd card

security

sensor

soldering

SPI

temperature

ultrasonic

usb

video

RASPBERRY PI RELATED

Averageman Vs Pi

MattsBits – Pi Resources

Official RaspBerry Pi Site

Raspberry Pi Pod

RasPi.tv

Raspihub.com

RaspTut

Rastrack.co.uk

RPi School

TECH RESOURCES

How2Code

Technology Spy

ARCHIVES

Archives

Select Month

```
hwclock -s
```

so it looks something like :

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# 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
hwclock -s
exit 0
```

Note : If you are using a Rev 1 you will need to use `"/i2c-0/"` instead of `"/i2c-1/"`.

You can save and quit using CTRL-X, Y and ENTER.

Now reboot the Pi using :

```
sudo reboot
```

Now when you repeat the `i2cdetect` command (see above) the 68 will turn into UU :

```
pi@rpispy ~ $ 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:  --  --  --  --  --  --  --  57  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Time Zones and Daylight Saving Time

By default the Pi tends to show time as GMT or UTC. For me sat in Bristol this results in the time being 1 hour behind my local time. To tell your Pi what region you are in you can use :

```
sudo raspi-config
```

and then select "Internationalisation Options" followed by "Change Timezone". You can then select you location from the following screens.

I set my location to "Europe/London" and this results in "date" showing the time in BST (British Summer Time) rather than UTC. The time is then correctly adjusted by +1 hour.

Reading The Date And Time

You can read the Pi's system time using :

```
date
```

If you need to set the system time for any reason you can use the following command :

```
sudo date -s "29 AUG 1997 13:00:00"
```

Once correct you can write the system date and time to the RTC module using :

```
sudo hwclock -w
```

You should be able to read the date and time back from the RTC using :

```
sudo hwclock -r
```

By separating the commands with a semi-colon you can read back the system time and the RTC time at the same time. Hopefully they should match and look something like this :

The “hwclock -s” we added to “rc.local” sets the system time from the RTC module.

The Final Test

The final test is to determine if the RTC module is keeping time and that the Pi will use that time when it boots. The best way to do that is to :

- Power down the Pi
- Remove the power cable
- Remove the network connection
- Attach the Pi to a monitor and keyboard
- Leave it overnight
- Power it up and use “date” to see what time the Pi thinks it is

Hopefully your Pi is now displaying the correct date and time and will maintain it when the Pi is powered down.

SHARE.



[◀ PREVIOUS ARTICLE](#)

Using An I2C Enabled LCD Screen With The Raspberry Pi

[NEXT ARTICLE ▶](#)

Raspberry Pi Used In New Point Break Movie

RELATED POSTS

APRIL 11, 2016

1

Write SD Card Images Using Etcher On Windows, Linux & Mac

MARCH 24, 2016

0

How to Change Your Character Skin In Minecraft Pi Edition

MARCH 10, 2016

5

7-segment Display Modules and the Raspberry Pi