**[HOW TO](https://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi/overview) SET UP THE** [**DS3231**](https://www.raspberrypi-spy.co.uk/2015/05/adding-a-ds3231-real-time-clock-to-the-raspberry-pi/) **AS THE HARDWARE CLOCK OF THE RASPBERRY PI**

Wiring is simple:

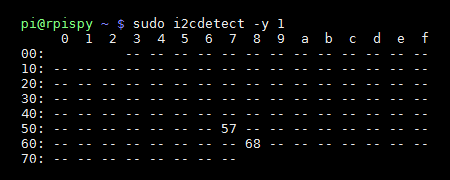
1. Connect **VCC** on the breakout board to the **3.3V** pin of the Pi (if using PCF8523 or DS3231)
2. Connect **GND** on the breakout board to the **GND** pin of the Pi
3. Connect **SDA** on the breakout board to the **SDA** pin of the Pi
4. Connect **SCL** on the breakout board to the **SCL** pin of the Pi

# Set up I2C on your Pi (Use the file in raspberry learned)

Verify your wiring by running

**sudo apt-get install python-smbus i2c-tools**

to install the helper software and then **sudo i2cdetect -y 1** at the command line, you should see ID #68 show up - that's the address of the DS1307, PCF8523 or DS3231!



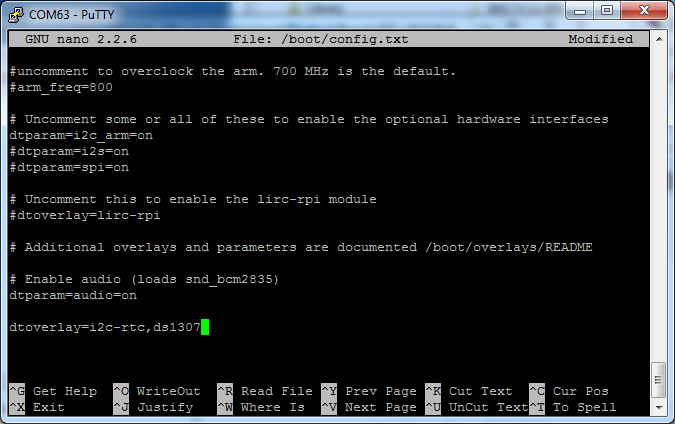
# Raspbian Jessie (Systemd)

You can add support for the RTC by adding a device tree overlay. Run

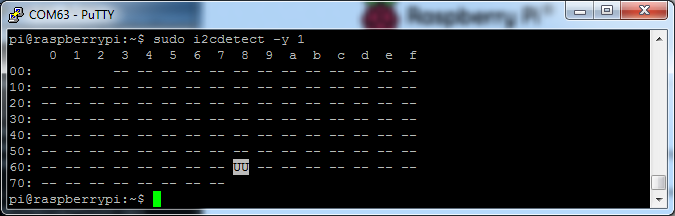
**sudo nano /boot/config.txt**

to edit the pi configuration to the end of the file:

dtoverlay=i2c-rtc,ds3231

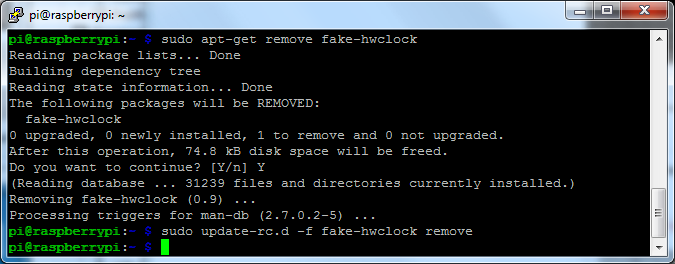
[](https://learn.adafruit.com/assets/35976)

Save it and run **sudo reboot** to start again. Log in and run **sudo i2cdetect -y 1** to see the UU show up where 0x68 should be

[](https://learn.adafruit.com/assets/35977)

Disable the "fake hwclock" which interferes with the 'real' hwclock

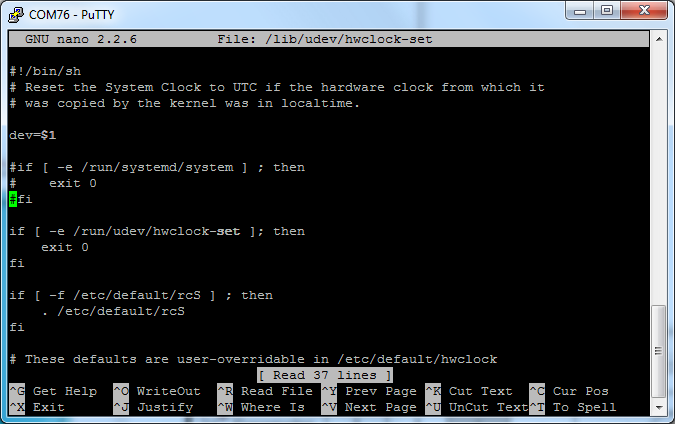
* sudo apt-get -y remove fake-hwclock
* sudo update-rc.d -f fake-hwclock remove
* sudo systemctl disable fake-hwclock

[](https://learn.adafruit.com/assets/36611)

Now with the fake-hw clock off, you can start the original 'hardware clock' script.

Run **sudo nano /lib/udev/hwclock-set**and comment out these three lines**:**

**#if [ -e /run/systemd/system ] ; then  
# exit 0  
#fi**

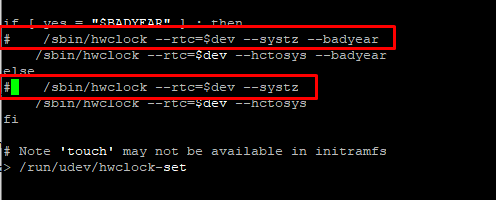
[](https://learn.adafruit.com/assets/36616)

Also comment out the two lines

/sbin/hwclock --rtc=$dev --systz --badyear

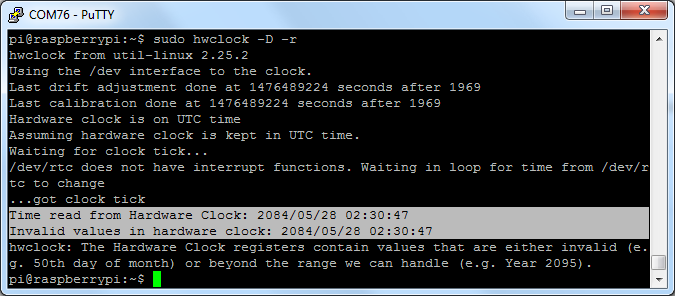
and

/sbin/hwclock --rtc=$dev --systz

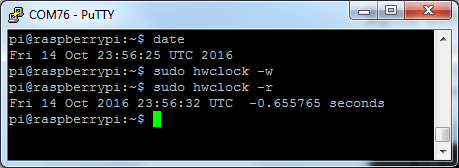
[](https://learn.adafruit.com/assets/75786)

## **Sync time from Pi to RTC**

When you first plug in the RTC module, it's going to have the wrong time because it has to be set once. You can always read the time directly from the RTC with **sudo hwclock -D -r**

[](https://learn.adafruit.com/assets/36612)

You can see, the date at first is invalid! You can set the correct time easily. First run date to verify the time is correct. Plug in Ethernet or WiFi to let the Pi sync the right time from the Internet. Once that's done, run **sudo hwclock -w** to **w**rite the time, and another **sudo hwclock -r** to **r**ead the time

[](https://learn.adafruit.com/assets/36613)

Once the time is set, make sure the coin cell battery is inserted so that the time is saved. You only have to set the time once

That's it! Next time you boot the time will automatically be synced from the RTC module

Now you can use the “date” command in console or

Use the the module datetime in python to call time out. (datetime.time,…)

If it cannot connect to the internet, it will use time from the RTC, if it can connect to the internet, it will use the time from the internet.

The RTC module can be removed safely when needed!