Set RTC Time | Adding a Real Time Clock to Raspberry Pi | Adafruit Learning System

Web Clip

Now that we have the module wired up and verified that you can see the module with i2cdetect, we can set up the module.

- [ ] Don't forget to set up I2C in the previous step!

Raspbian Jessie (Systemd)

Thanks to kd8twg for the hints!

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

sudo nano /boot/config.txt

to edit the pi configuration and add whichever matches your RTC chip:

```
dtoverlay=i2c-rtc,ds1307
```

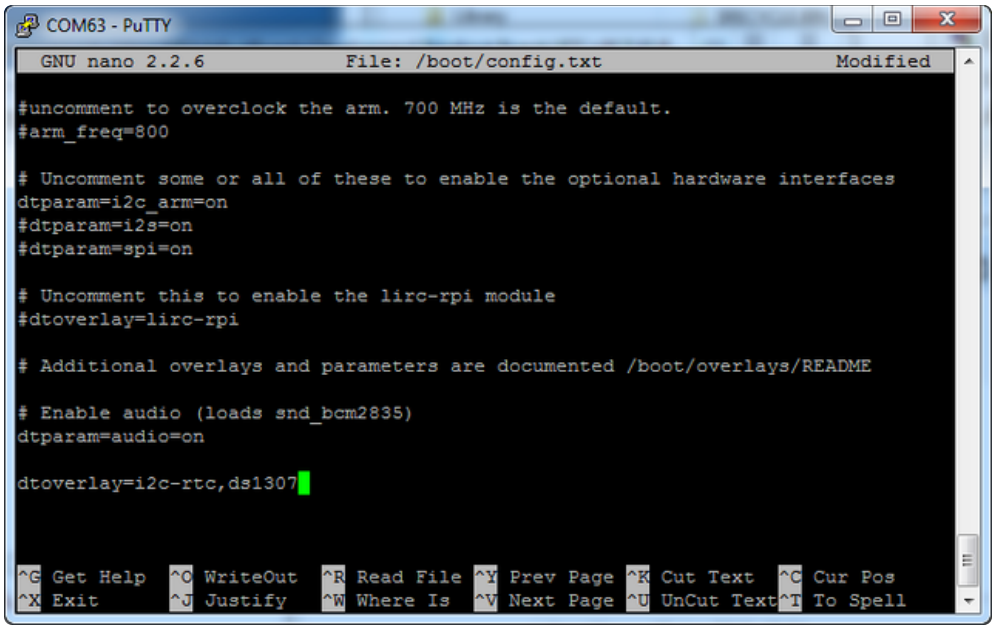
or

```
dtoverlay=i2c-rtc,pcf8523
```

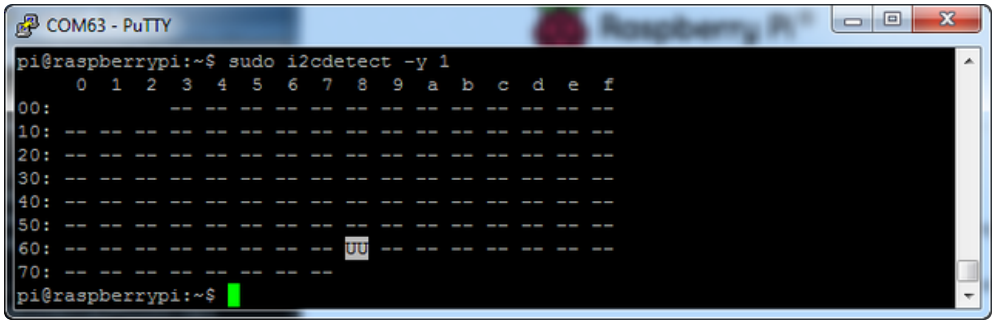
or

```
dtoverlay=i2c-rtc,ds3231
```

to the end of the file

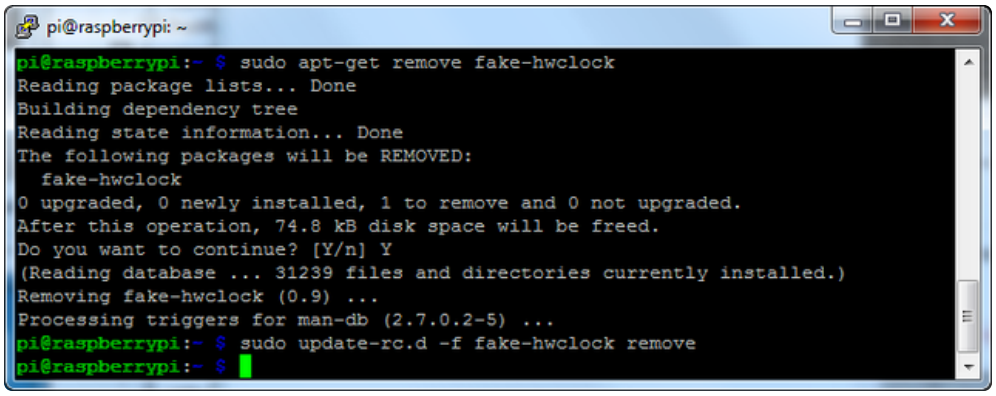


Save it and run `sudo reboot` to start again. Log in and run `sudo i2cdetect -y 1` to see the UU show up where 0x68 should be



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

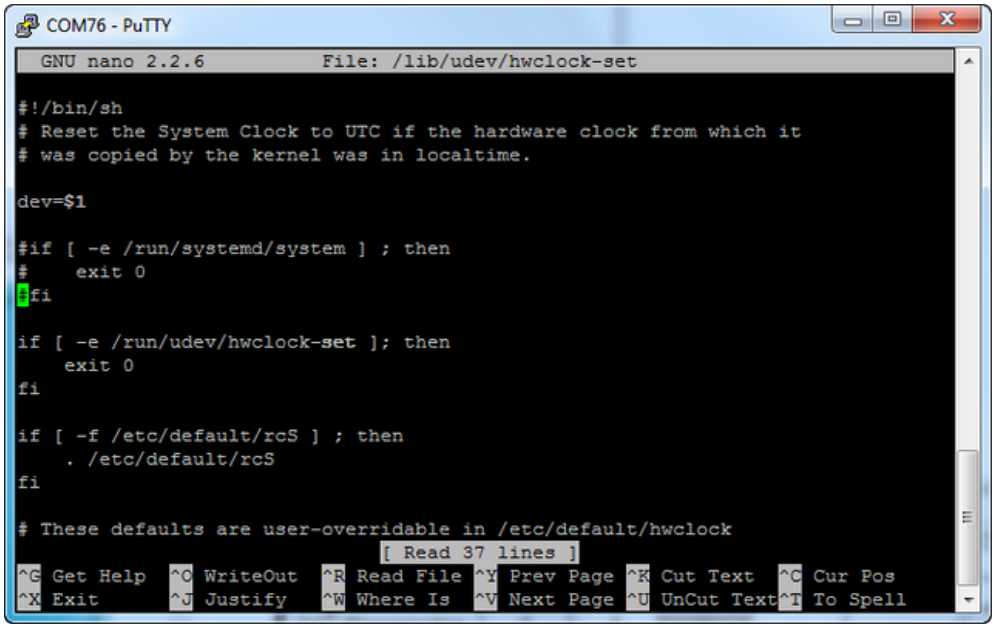
- `sudo apt-get -y remove fake-hwclock`
- `sudo update-rc.d -f fake-hwclock remove`
- `sudo systemctl disable fake-hwclock`



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

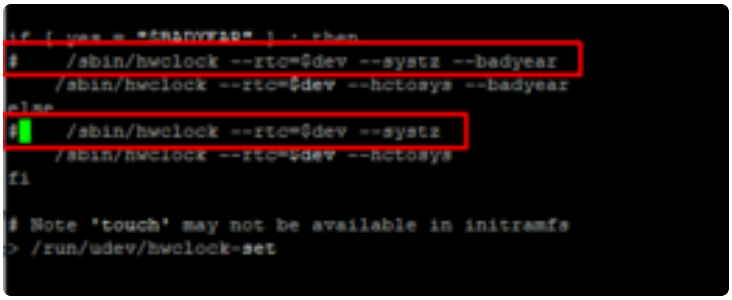


Also comment out the two lines

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

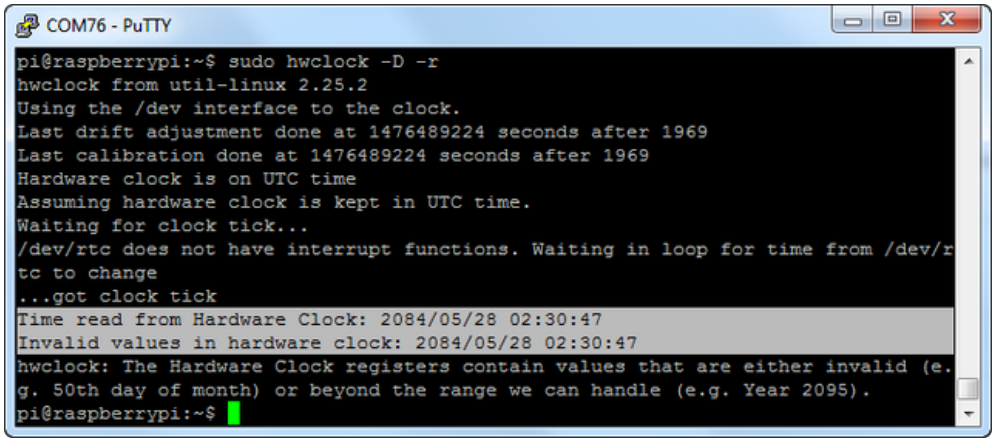
and

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

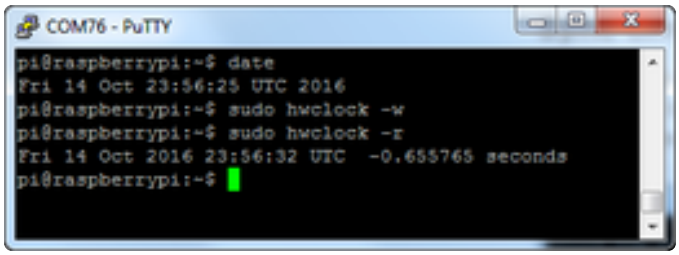


Sync time from Pi to RTC

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`



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



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

Raspbian Wheezy or other pre-systemd Linux

First, load up the RTC module by running

```
sudo modprobe i2c-bcm2708
sudo modprobe i2c-dev
sudo modprobe rtc-ds1307
```

Then, as root (type in **sudo bash**) run

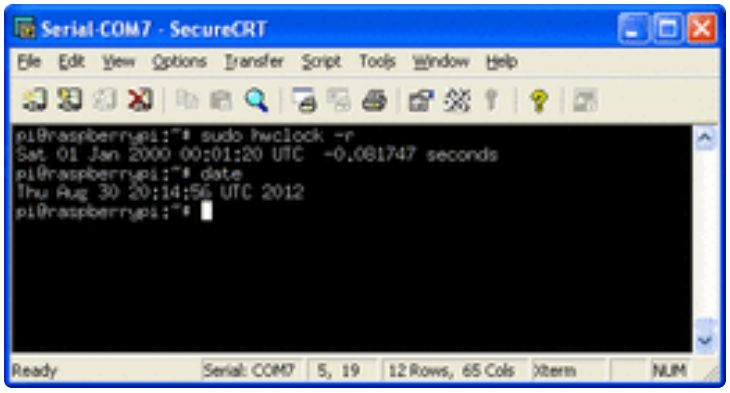
```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

If you happen to have an old Rev 1 Pi, type in

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

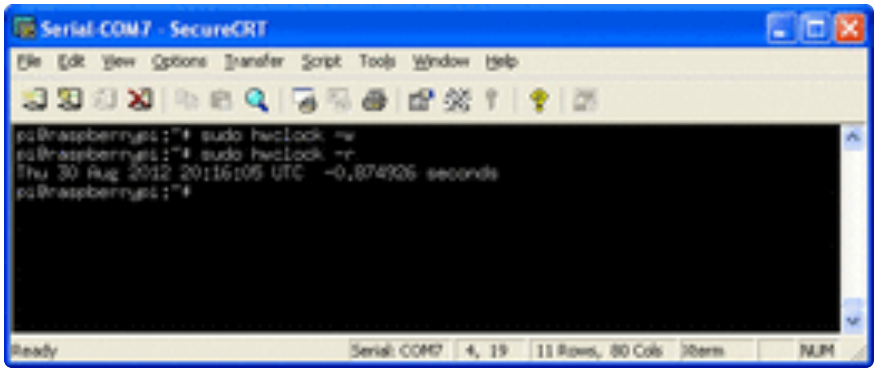
You can then type in **exit** to drop out of the root shell.

Then check the time with **sudo hwclock -r** which will read the time from the DS1307 module. 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

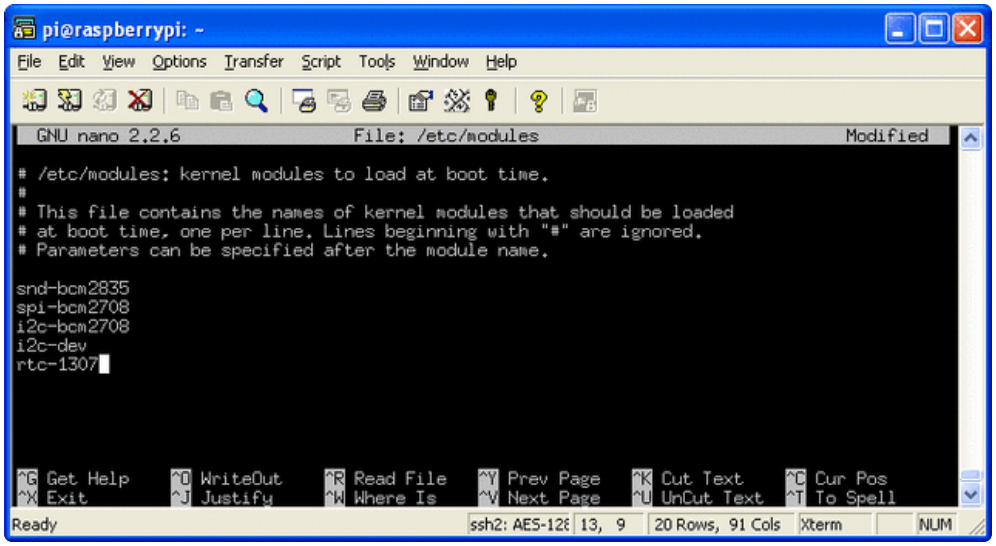


First you'll need to get the right time set on the Pi, the easiest way is to connect it up to Ethernet or Wifi - it will automatically set the time from the network. Once the time is correct (check with the **date** command), run **sudo hwclock -w** to write the system time to the RTC

You can then verify it with **sudo hwclock -r**



Next, you'll want to add the RTC kernel module to the /etc/modules list, so its loaded when the machine boots. Run **sudo nano /etc/modules** and add **rtc-ds1307** at the end of the file (the image below says rtc-1307 but its a typo)



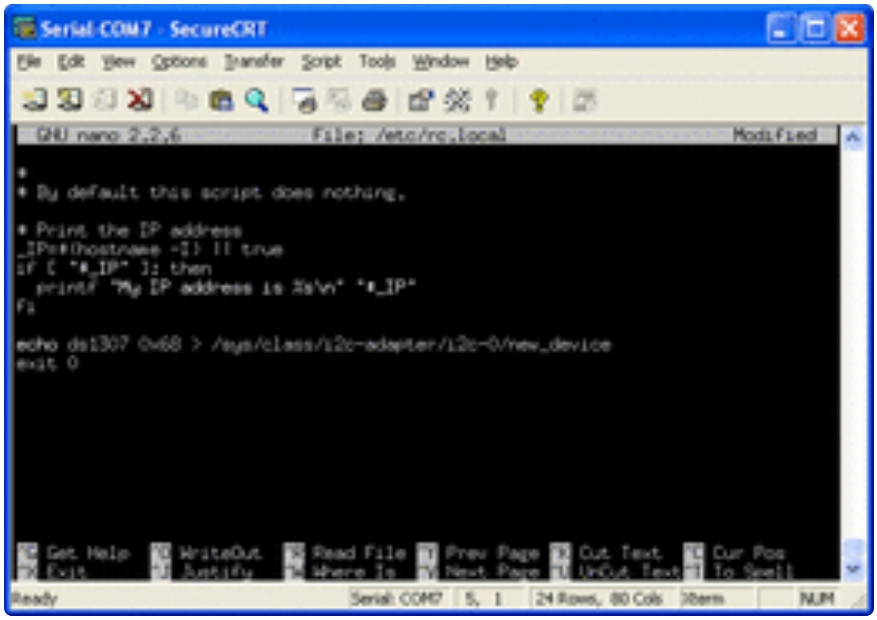
Older pre-Jessie raspbian is a little different. First up, you'll want to create the DS1307 device creation at boot, edit /etc/rc.local by running

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

and add:

> `echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device` *(for v1 raspberry pi)*
> `echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device` *(for v2 raspberry pi)*
> `sudo hwclock -s` *(both versions)*

before `exit 0` (we forgot the hwclock -s part in the screenshot below)



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