E-paper HAT real-time clock

The information here relates to the latest Raspbian Jessie.
If you are still using Raspbian Wheezy things will be a bit different.

Folks have been asking for information on how to use the real time clock on the E-paper HAT, so here's some information which should help. Because of the information stored in the HAT eeprom (which is read at boot time) all the GPIO used is preconfigured as input, output, i2c, spi and so on.
Also the appropriate driver modules are automatically loaded, including the one for the RTC. So the kernel is aware of a RTC (designated rtc0) at 0x68 on the i2c bus and takes charge of it.
If you type `i2cdetect -y 1` you should see UU in the 0x68 slot of the table to show that there is a device present and being used by the operating system.

```
pi@RasPi2B2:~$ 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: -- -- -- -- -- -- -- --
```

If you wanted to talk to the device over i2c from a user program, you would need to unload the driver module by typing `sudo modprobe -r rtc_ds1307` as follows:

```
pi@RasPi2B2:~$ sudo modprobe -r rtc_ds1307
pi@RasPi2B2:~$ 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: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

`i2cdetect -y 1` will now show 68 at the appropriate location, indicating that the device is now available to the user at i2c address 0x68.

Remember to reinstate the driver afterwards with `sudo modprobe rtc_ds1307` if you want the operating system to maintain timekeeping for you.

So why rtc_ds1307 when the chip is a DS3231? Well, they are, for most purposes compatible, whereas the driver for the DS3232 which is supposed to also work with the DS3231, doesn't seem to function properly at the present time.

Now, for reliable off-net operation there are a few things to do.
Since we now have a real hwclock, we want to remove the tasks that run for the fake-hwclock so that they don't cause confusion.

As root (type `sudo su`) do the following:

```
# Disable fake-hwclock
# remove package
apt-get remove  --purge fake-hwclock
```

Still as root, type `nano /lib/udev/hwclock-set` and comment out the following three lines:

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

so that they look like this

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

Save and exit. Next, type `timedatectl set-ntp 1` to activate the network time protocol (ntp) function. On boot up the system clock will be set from the RTC. If there is a network connection, the ntp daemon will adjust the system time and RTC every few minutes. If there is no network, the system time will not get adjusted and accuracy is largely dependent on the RTC keeping good time (to around 3 minutes a year) and resetting the system clock at every boot. If you wish to check the status of the hardware and system clocks, you can type

```
timedatectl status
```

at a command prompt. The output looks like this:

```
pi@RasPi2B2:~$ timedatectl status
      Local time: Sat 2015-12-12 17:28:11 GMT
  Universal time: Sat 2015-12-12 17:28:11 UTC
        RTC time: Sat 2015-12-12 17:28:11
       Time zone: Europe/London (GMT, +0000)
     NTP enabled: yes
NTP synchronized: yes
 RTC in local TZ: no
      DST active: no
 Last DST change: DST ended at
                  Sun 2015-10-25 01:59:59 BST
                  Sun 2015-10-25 01:00:00 GMT
 Next DST change: DST begins(the clock jumps one hour forward) at
                  Sun 2016-03-27 00:59:59 GMT
                  Sun 2016-03-27 02:00:00 BST
```

If you wish to adjust the system clock to match the RTC in the absence of a network connection and without rebooting, you can use the old style command:

```
sudo hwclock –hctosys
```

The timestamp on the empty file `/var/lib/systemd/clock` shows when the ntp daemon last updated the system clock and RTC.

```
pi@RasPi2B3 ~ $ ls -l /var/lib/systemd/clock
-rw-r--r-- 1 systemd-timesync systemd-timesync 0 Dec 13 09:11 /var/lib/systemd/clock
```

There will be more to follow in another document on talking to the RTC, but in the meantime you can find the datasheet for the clock chip here:
http://datasheets.maximintegrated.com/en/ds/DS3231M.pdf