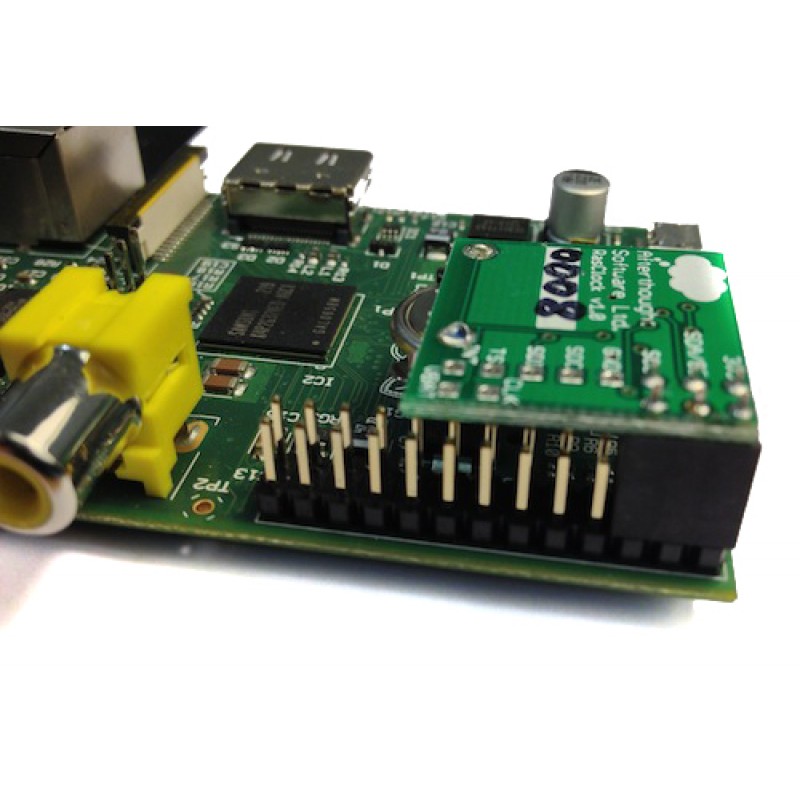
**The RasClock – Raspberry Pi Timekeeping with a Real Time Clock**

In order to achieve its miniature size and low price tag, several non-essential items usually found on a desktop computer had to be omitted from the Raspberry Pi. Laptops and computers keep time when the power is off by using a pre-installed battery powered 'Real Time Clock' (RTC). An RTC module is not included with the Raspberry Pi, which updates the date and time automatically over the internet via Ethernet or WiFi instead. Subsequently, your Pi will revert back to the standard date and time settings when the network connection is removed. For projects which have no internet connection, you may want to add a low cost battery powered RTC to help your Pi keep time!

The RasClock has been specifically designed for use with the Raspberry Pi and plugs directly in to the Raspberry Pi's GPIO Ports. This article will walk you through its installation!

**Step 1. Plug it in!**

Plug the coin battery into the RTC by matching the positive + on the battery with the positive + on the module. Then simply plug the RTC into the Raspberry Pi as depicted. It sits on the furthest 6 GPIO pins at the SD card end of the Pi. Make sure your Pi is switched off before installation, and the battery is firmly seated in the Rasclock to avoid any damage to the module.

**Step 2. Set-Up**

This RTC module is designed to be used in Raspbian. So, the first step is to make sure you have the latest Raspbian Operating System (OS) installed on your Raspberry Pi (<http://www.raspberrypi.org/downloads>). Currently, the module requires the installation of a driver that is not included in the standard Raspbian distribution; however a pre-compiled installation package is available which makes setup nice and easy.

Make sure your Pi has internet access, and grab the installation package off the internet.

***wget http://afterthoughtsoftware.com/files/linux-image-3.6.11-atsw-rtc\_1.0\_armhf.deb***

(The wget command allows you to grab a file off the internet by providing a URL)

***sudo dpkg -i linux-image-3.6.11-atsw-rtc\_1.0\_armhf.deb***

(the dpkg command enables the management of Debian packages. The -i installs the package, or upgrades it if it is already installed). This may take a couple of minutes to complete.

***sudo cp /boot/vmlinuz-3.6.11-atsw-rtc+ /boot/kernel.img***

(The cp command stands for copy. Here, we need to copy the RTC module's boot file to the Raspberry Pi boot directory).

The next step involves editing the text in the Raspberry Pi boot files. I usually use nano text editor for these minor changes - it’s basic, pre-installed and easy to master. System commands for nano are enabled by holding the CTRL key (denoted as ^ in nano) whilst pressing the relevant command e.g. CTRL+X to exit.

We need to configure Raspbian to load the RTC drivers at boot by adding the boot information to the /etc/modules configuration file.

***sudo nano /etc/modules***

(this will open the 'modules' file within nano text editor and allow you to make changes. To add text, simply use the arrows keys to browse to the next line in the boot file and add the following text, one per line. Then exit nano (CTRL+X) and don't forget to save those changes!)

***i2c-bcm2708***

***rtc-pcf2127a***

The final step in set-up is to register the RTC module when the Raspberry Pi boots and set the system clock from the RTC. When editing files always follow the instructions outlined at the top of the file denoted by #. For example, the file we are just about to edit requires any text to be put before the end of the file, denoted by 'exit 0'. Open the required file for editing:

***sudo nano /etc/rc.local***

For Rev 1. Raspberry Pi Boards Add the following text:

***echo pcf2127a 0x51 > /sys/class/i2c-adapter/i2c-0/new\_device***

***( sleep 2; hwclock -s ) &***

For Rev 2. Raspberry Pi Boards Add the following text:

***echo pcf2127a 0x51 > /sys/class/i2c-adapter/i2c-1/new\_device***

***( sleep 2; hwclock -s ) &***

Then reboot!

***sudo reboot***

**Step 3. Using the RTC**

After you reboot the Raspberry Pi, you should be able to access the module using the hwclock command. The first time you use the clock you will need to set the time. To copy the system time into the clock module:

***sudo hwclock –w***

To read the time from the clock module:

***sudo hwclock -r***

To copy the time from the clock module to the system:

***sudo hwclock –s***

That's it! You can now keep time using your Raspberry Pi with no Internet! Type hwclock into your resident search engine for a load more useful commands!

This article is sponsored by ModMyPi www.modmypi.com

All breakout boards and accessories used in this tutorial are available for Worldwide shipping from the ModMyPi Webshop.