Raspberry Pi Real Time Clock Module DS1302

Raspberry Pi Real Time Clock

To keep costs down, one of the bits of hardware omitted from the Raspberry Pi is a Real Time Clock (RTC).

When you boot up your Raspberry Pi it will have forgotten what the current date and time was and will default (on mine at least) to 30 November 1999.

The Raspian "wheezy" installation that we have been using has NTP installed. This stands for Network Time Protocol and means that if your Raspberry Pi is connected to the internet it should periodically update its internal clock to that of one of the many clock servers out on the internet.

If you are not connected to the internet, then you may want to connect your Raspberry Pi to a Real Time Clock chip which has a battery backup so that the date and time are kept accurate.

We have a great module (DS1302 Real Time Clock Module with Battery Backup) which is perfect for the Raspberry Pi as it runs at 3V so no logic level conversion is required and has a built in battery backup. The module uses the DS1302 Real Time Clock IC and you can read the DS1302 Datasheet here

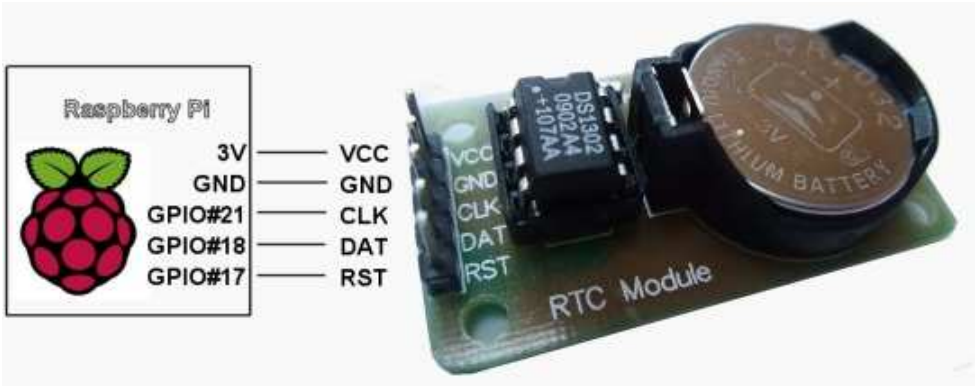
The DS1302 chip uses a slightly quirky serial interface which we won't go into here, but it is all detailed in the datasheet. Luckily for us, an article in the Sept 2012 issue of Everyday Practical Electronics (EPE) discusses using this chip with the Raspberry Pi and provides a nice program (written in C) which can be used to set the RTC date and time on the module and also to read the date and time and set the Raspberry Pi's date and time.

Note: If you find you cannot set the date and time, there is a write protect register value which needs to be cleared (register 0x8E). We have had a few of these modules where this write protect has been turned on. You can clear it by setting register 0x8E to 0 in code, or by simply removing the battery and re-inserting it (The default value for the register is off)

Wiring up the RTC Module

There is a wiring diagram on the EPE magazine page, but unfortunately it is wrong as they have CLK connected to GPIO#17 and RST (CE) connected to GPIO#21. These two connections should be swapped.

The correct wiring (using our module) is shown below



Source Code

The source code for the Real Time Clock program can be downloaded here Real Time Clock Source Code

Extract the rtc-pi.c. file from the zip archive and copy it onto your Raspberry Pi's SD card

You now need to compile the code. Compile it using

cc rtc-pi.c

This will create a file called a.out. Rename this to rtc-pi using

mv a.out rtc-pi

And make it executable using

chmod +x rtc-pi

Setting the Date and Time

To set the data and time on the module use

sudo ./rtc-pi CCYYMMDDhhmmss

And to read the date and time from the module and set the Raspberry Pi clock

sudo ./rtc-pi

You should set this command to run when the Raspberry Pi boots up. There are many ways to do this, so we will leave it up to you to decide which method is best for you.

Changes for Raspberry Pi V2 boards

For those who are going to use this with Raspberry Pi v2:

In version 2 Pi GPIO27 is in place of GPIO21 (same pin). Code in rtc-pi.c should be changed. Look for #defines SCLK_OUTPUT, SCLK_HIGH and SCLK_LOW. These are the correct values for Pi v2 using GPIO27:

#define SCLK_OUTPUT *(gpio+GPIO_SEL2) &= 0xFF1FFFFFL; *(gpio+GPIO_SEL2) |= 0x00200000L

#define SCLK_HIGH *(gpio+GPIO_SET) = 0x08000000L

#define SCLK_LOW *(gpio+GPIO_CLR) = 0x08000000L

I also found that a pullup resistor is needed between DAT and VCC (= 3.3V). A 10k..30k resistor seems to work fine.