

On-board Power Converter & RTCC Combo

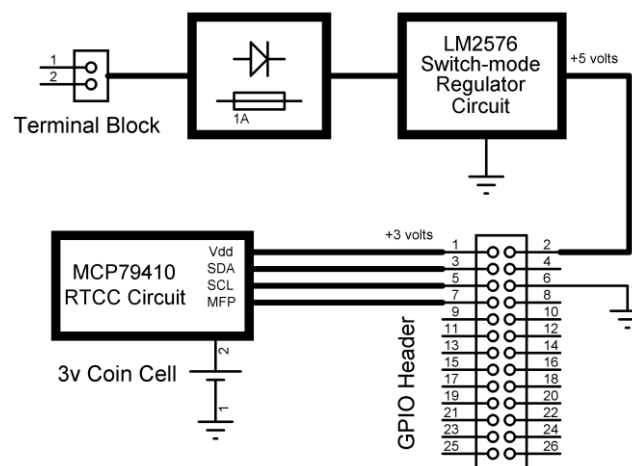
For the Raspberry Pi

Description

If you wish to use the Raspberry Pi for real world work in “the field” (possibly literally!) you will invariably be out of easy reach of mains power sockets and internet connectivity. This means that alternative means of powering and timekeeping will need to be found.

This board allows power to the Pi to be sourced from solar cells, batteries or low voltage A.C., as well as providing the timekeeping by means of the Real Time Clock and Calendar module.

It is designed for mounting directly on the Raspberry Pi’s GPIO header with that port duplicated on the module so that full access to the Pi’s I/O is still available.



Simplified Block Diagram

The Power Section

The switch mode power conversion is based on the LM2576S-5.0 switching regulator and is capable of delivering a stable 5 volts at more than enough current to power the Pi with other peripherals when connected to a suitable power source, such as a battery, with a voltage of between 8 and 30 volts. The module is also capable of regulating an alternating voltage such as that from the secondary winding of a low voltage transformer. Again the input voltage to the module must be in the range of 8 to 30 volts.

The RTCC Section

The Real Time Clock is based on the Microchip’s MCP79410 RTCC chip which not only keeps the time, but also allows two alarm times to be set and a hardware output for the Pi to detect and respond to the appropriate alarm. GPIO4 (header pin 7) is connected to the Multi-Function Pin of the RTCC chip and can be used to detect alarms.

Battery back-up is provided by a coin cell (not included) type BR1225 (or CR1225 and equivalents).

Use

The Power Section

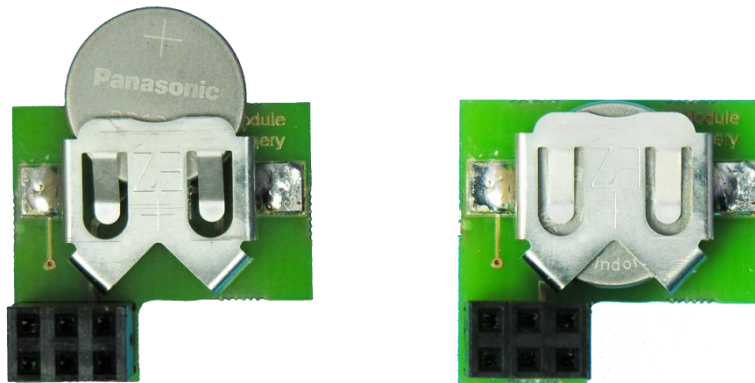
Connection of the power source to the module is by means of the screw terminal block. The orientation of the connecting wires is not important; due to the on-board rectifier, they may be connected either way round. An LED is provided as a visual indication that the module is supplying power to the Pi.

Specification

Input Voltage	8 - 30 Volts A.C. or D.C.
Output Voltage	5 Volts D.C.
Output Current	Max 2 Amps

The RTCC Section

Before fitting the module, the back-up battery should be installed as shown below. The battery required is a coin cell type BR1225. This should be fitted with the face displaying the “+” sign facing away from the printed circuit board.



NOTE:- Although the illustration above is of a different product, it does show the correct orientation of the battery

Once the module has been fitted the only thing left to do is to install the appropriate software to run the RTCC. A description of the process is as follows:

Before you can access the Real Time Clock chip it will be necessary to install the software tools for accessing the I2C bus if you've not already done so.

Make sure your Pi is connected to the internet and from the command line, type:

- ***sudo apt-get update***
- ***sudo apt-get install i2c-tools***

This will ensure the operating system is up to date, and the second line will install “i2c-tools” software library

For the Raspbian operating system, run Nano and modify the “Modules” file by entering:

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

Add these two lines to the end of the code:

- ***i2c-bcm2708***
- ***i2c-dev***

Now reboot the Pi so that the new library can “settle in”!

If there is a file called *“/etc/modprobe.d/raspi-blacklist.conf”* then this needs to be edited and the two lines:

- ***blacklist spi-bcm2708***
- ***blacklist i2c-bcm2708***

should be commented out by adding a “#” in front of them. So again at the command line prompt, enter:

➤ ***sudo nano /etc/modprobe.d/raspi-blacklist.conf***

...and then edit the two lines to read:

- ***#blacklist spi-bcm2708***
- ***#blacklist i2c-bcm2708***

Save the file, then type:

➤ ***sudo i2cdetect -y 1*** (***sudo i2cdetect -y 0*** if you have an early version Raspberry Pi)

This will show a table with any i2c addresses that are in use. The RTCC module uses address 0x6f .

In order to be able to access i2c-tools via Python, you will need to install the “python-smbus” package. Type:

➤ ***sudo apt-get install python-smbus***

That completes the basic setup for accessing the i2c bus and allowing direct access to the RTCC chip, which may now be tested using the Python script at <http://www.circuitsurgery.com/sw/rtcc-alm-test.zip>

This program firstly reads all the registers, prints them to the screen and also saves them to a file. The “ST” bit is then checked and if clear the program sets it to start the clock running. Likewise the “VBATEN” bit is set in order to enable the battery backup facility.

The user is then prompted to enter the current date and time and having collected the input, the registers are loaded with the information.

- ❖ **Note that for testing purposes I suggest entering a fictitious time at this stage, the reason will become clear later!**

The list of registers and their associated contents are displayed for a second time and then the current time (or, more accurately, the time entered by the user) is displayed continuously, updating every second.

Escape back to the system prompt by using “Ctrl + Z”

This is a quick (and dirty!) way to check the functionality of the RTCC chip, and also serves as a starting point for your own software.

“hwclock”

In order to be able to use the “**hwclock**” command, a driver will need to be installed. There is not a specific driver for the MPC7941x chip, but the one for a different chip (the DS1307) is suitable.

From the command prompt, type:

- ***sudo modprobe rtc-ds1307***
- ***sudo bash*** (enters Bash as Root)
- ***echo mcp7941x 0x6f > /sys/bus/i2c/devices/i2c-1/new_device*** (for Rev.2 Pi’s) OR...
- ***echo mcp7941x 0x6f > /sys/bus/i2c/devices/i2c-0/new_device*** (for Rev.1 Pi’s)
- ***exit*** (Return to standard user)

Now you will be able to issue the command:

➤ ***sudo hwclock -r***

which will read the contents of the clock, which should correspond with the time entered at the first testing stage, and:

➤ ***sudo hwclock -w***

which will write the system time to the RTCC. Now, repeat the “***hwclock -r***” command and this time you should see that the output has changed to reflect the system time. The following steps allow the various modules to be loaded, so that the RTCC is available at system boot. Type:

➤ ***sudo nano /etc/modules***

Add the following line:

▪ ***rtc-ds1307***

Save and close the file.

Now open /etc/rc.local for editing:

➤ ***sudo nano /etc/rc.local***

Immediately before the line – “***exit 0***” add the following lines:

▪ ***echo mcp7941x 0x6f > /sys/class/i2c-dev/i2c-1/device/new_device***
▪ ***hwclock -s***

These two lines will create the device on boot and then write the time from the RTCC to the system.

Your Pi is now set up with a hardware Real Time Clock and can tell the time regardless of whether it's connected to the internet or not!

Full details of the MCP79410 RTCC chip can be found at:

<http://www.circuitsurgery.com/docs/mcp79410-rtcc.pdf> , section 5 of which deals with the various registers and addresses.

The MFP (Alarm Facility)

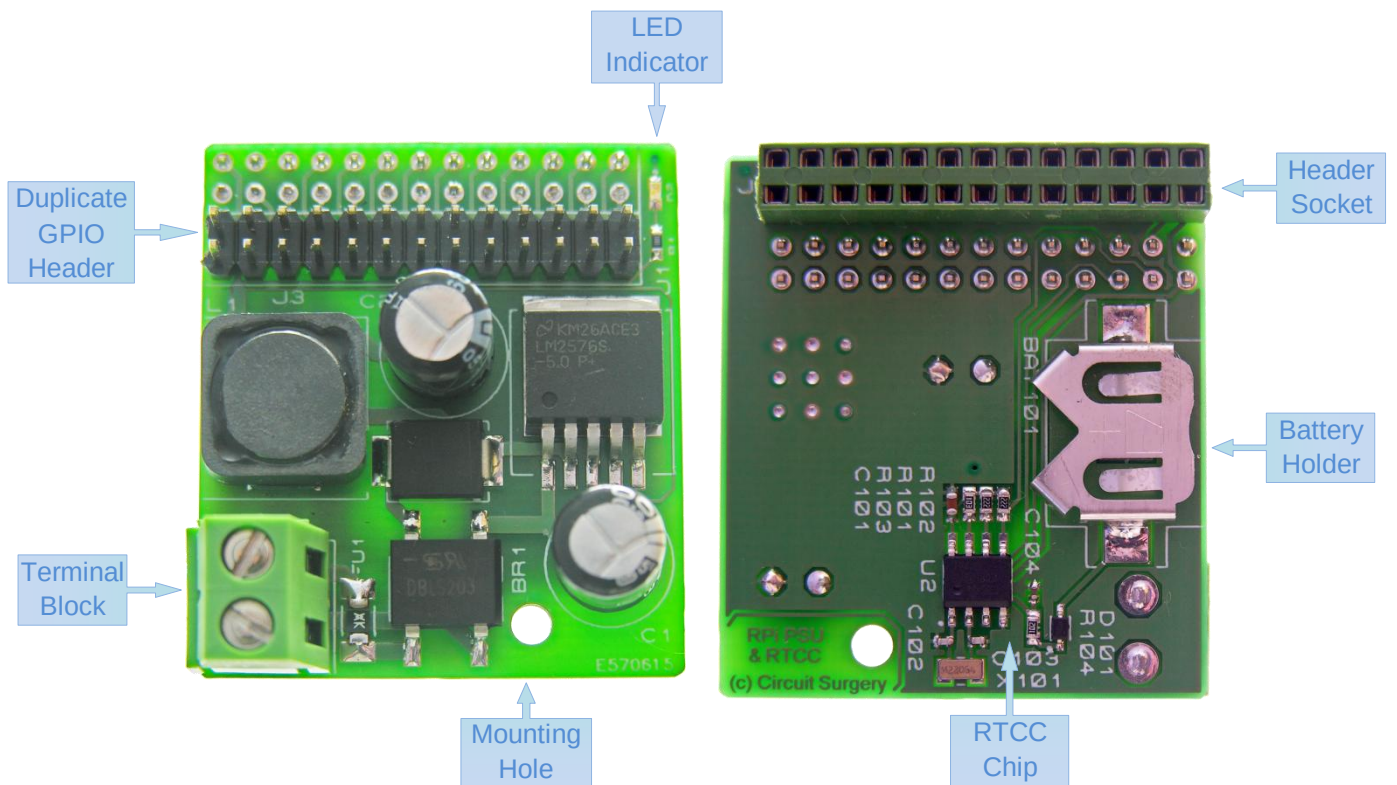
The MCP79410 RTCC chip used on this module has an “MFP” (Multi Function Pin) which is connected to the Raspberry Pi's GPIO4 pin (header pin 7). The MFP may be used to send an alarm signal or a clock signal to the RPi.

Two alarm times can be set in the RTCC chip, and when the time is reached the state of the MFP is changed to signal the alarm to the RPi.

Alternatively, the MFP can be set to send a clock signal to the MFP. This can be at one of four frequencies – 1Hz, 4.096kHz, 8.196kHz or 32.768kHz

For full details of how to set the MFP, please refer to the [data sheet](#)

Views of module



© Circuit Surgery 2012. No part of this publication may be reproduced in any form, whether physically or electronically, without prior written consent.

**The Raspberry Pi name and the raspberry logo are trademarks of the Raspberry Pi Foundation.
I have no affiliation with the Raspberry Pi Foundation.**