Power over Ethernet (PoE) Adaptor

For the Raspberry Pi model B+ , Pi2 and Pi3

User Manual

Circuit Surgery

www.circuitsurgery.com

Description

N.B.: In this manual the term "Raspberry Pi" will refer to the Raspberry Pi B+ and the Raspberry Pi 2

This Power over Ethernet adaptor fits directly onto the GPIO port of the Raspberry Pi and allows the Pi to have its power supplied through the network cable, so eliminating the need for additional power cables near the Pi. This adaptor can also supply external circuitry with 5 volts at up to 1.5 amps via the solder pads on the board. If required, a standard PCB mount terminal block with a pin pitch of 0.1" (not supplied) may be fitted here to allow for easy connection and disconnection of your circuit.

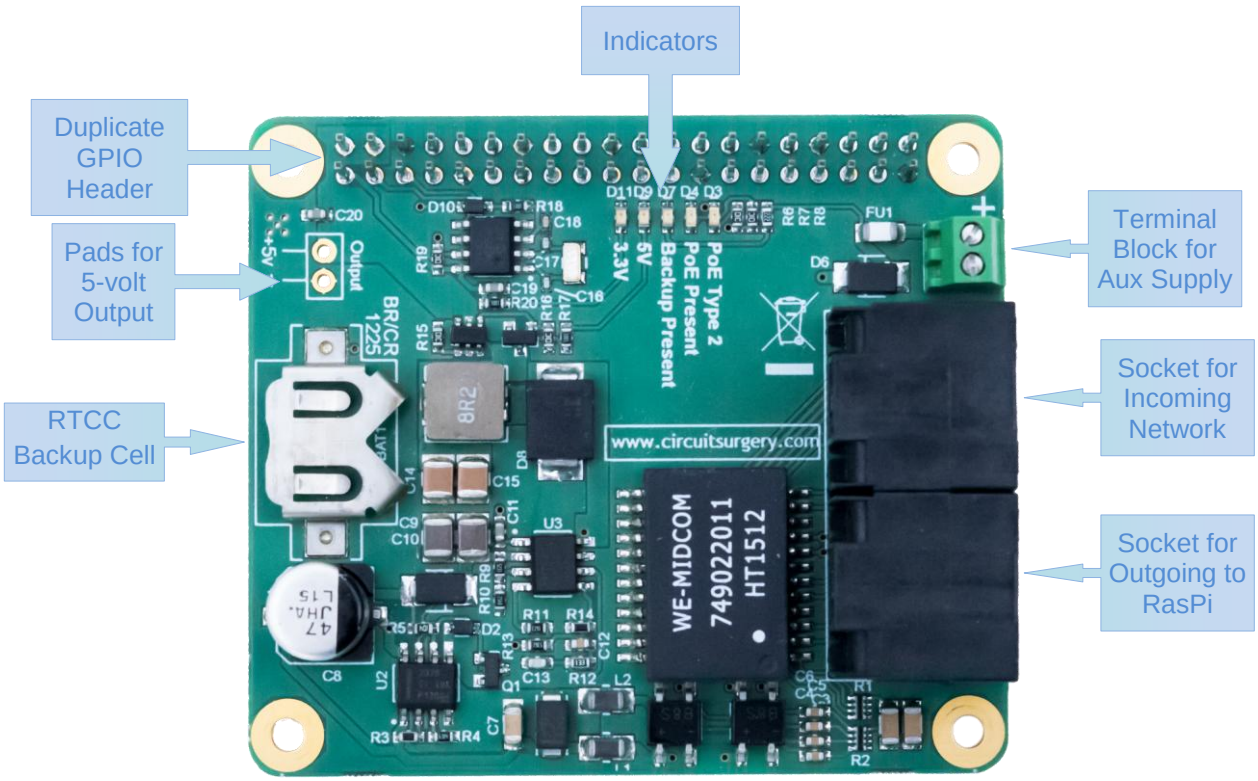
The adaptor also has provision for a secondary power source to be connected which can act as a back-up, should the PoE supply fail.

In addition, the adaptor incorporates a battery backed Real Time Clock, so that time keeping is not a problem when access to a time server is unavailable.

Main Features

- Compatible with IEEE 802.3af and IEEE 802.3at Power over Ethernet standards
- Can supply up to 2.5 amps* in addition to powering the Raspberry Pi
- Pads for 5v supply to external circuitry
- Real Time Clock & Calendar with alarm output
- RTCC is itself independently battery backed by means of a small coin cell
- Fits directly onto GPIO port
- GPIO port extended through the adaptor to allow full access to all pins
- Mechanical support for board by supplied spacers and screws
- Network connection to the Pi by short link cable supplied

*See the section headed "Power Supply"



Use

Fitting the module

This is quite straightforward.

1. Fit a spacer at each of the mounting holes on the Raspberry Pi.
2. Fit the adaptor onto the GPIO port and then use the other screws to fix the adaptor to the spacers. Depending on your application, it may be necessary to omit the two screws either side of the GPIO port.
3. Connect the cable link between the Pi's network port and the output of the PoE adaptor
4. Connect your PoE enabled network port to the input of the PoE adaptor

At this point, if there is PoE power on your network port, the Raspberry Pi will turn on.

Power Supply

The module is capable of supplying 2.5 amps at 5 volts in addition to the current needed to power the Pi itself. However, to make full, continuous use of this capacity additional cooling of the module will be required. With no additional cooling the additional continuous current capacity will be about 1.5 amps.

An output of 5 volts can be taken from the two solder pads as shown on the diagram. Alternatively, a standard PC mount terminal block may be fitted here to facilitate easy connection of external circuitry.

Backup Power

If a power-fail backup supply is required this should be DC, and between 12 and 48 volts.

The backup supply is connected to the terminal block on the top right corner of the adaptor with the positive lead connected to the terminal marked "+". Reversing the polarity will not harm the adaptor as there is reverse polarity protection. With a voltage applied, the red "Backup Present" indicator will illuminate.

Indicators

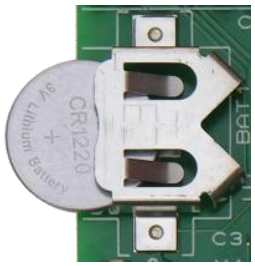
There are five LED indicators on this module. Looking at the module with the GPIO port at the top and reading left to right they are as follows:

- Green, "3.3v": indicates 3.3 volt supply from the Pi is present. This powers the RTCC.
- Green, "5v": indicates the 5 volt supply is present on the PoE module.
- Red, "Backup Present": indicates that a voltage is present on the screw terminals.
- Red, "PoE Present": indicates that a Power over Ethernet supply is OK.
- Red, "PoE Type 2": indicates that a high power Type 2 or IEEE 802.3at supply is present.

If the PoE supply fails and the backup supply takes over, the PoE LEDs will extinguish but the "Backup Present" and green voltage LED's will continue to glow, thus indicating that there is still power connected but that it's not coming from the network.

The Real Time Clock and Calendar

The RTCC is battery backed with a coin cell type BR1225 or equivalent, which is installed with the positive (+) side facing away from the board (see below for an illustration of the correct orientation) and pushed fully in. Whilst this is not essential for the operation of the RTCC, if it is not fitted and the adaptor loses all power the RTCC will need to be reset to the correct time when power is restored.



Software

The only software requirement is for the operation of the RTCC, and for completeness, the installation of the appropriate software modules will be described here.

The I²C Bus

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 Raspian 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