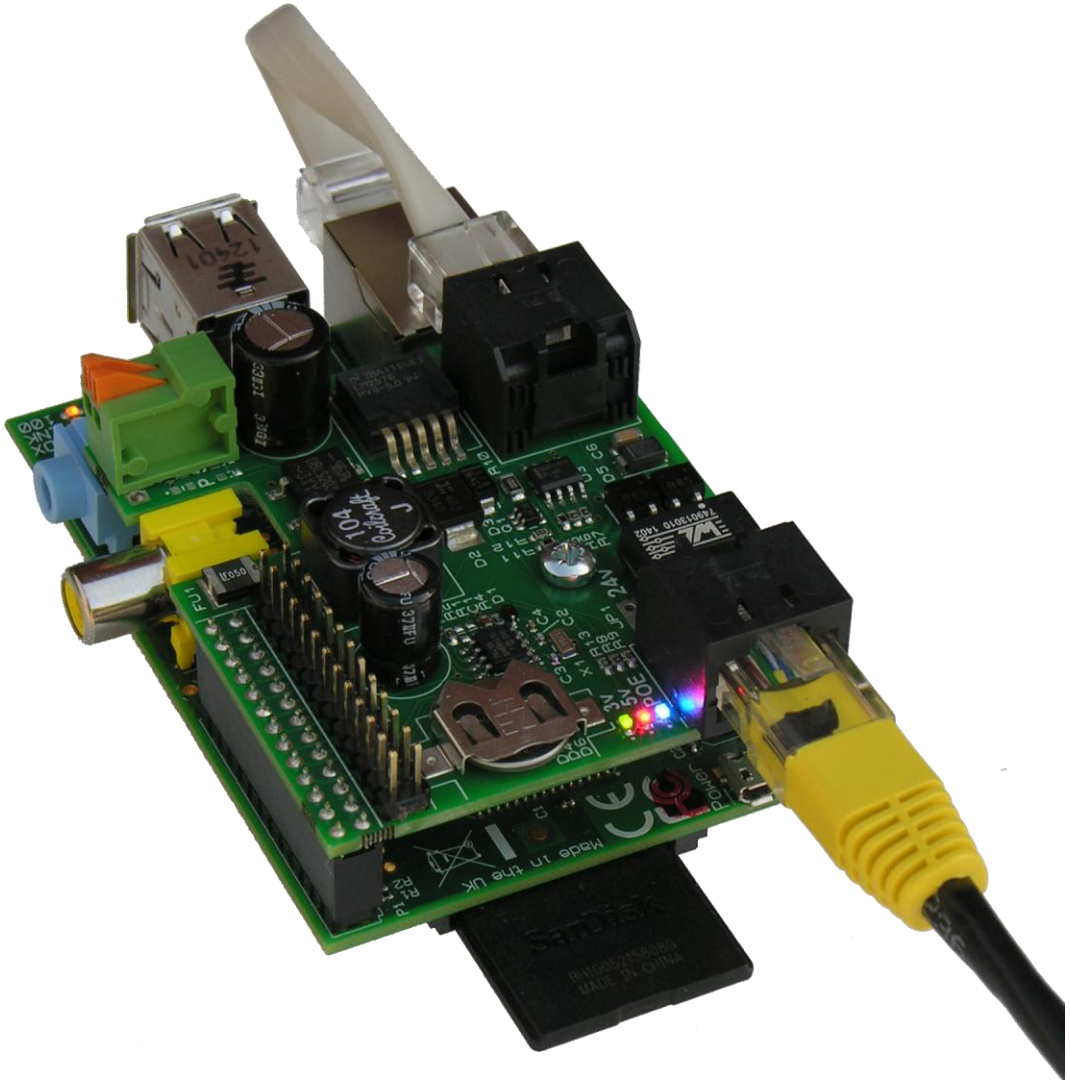
Power over Ethernet (PoE) Adaptor

For the Raspberry Pi

User Manual

Circuit Surgery

www.circuitsurgery.com



Description

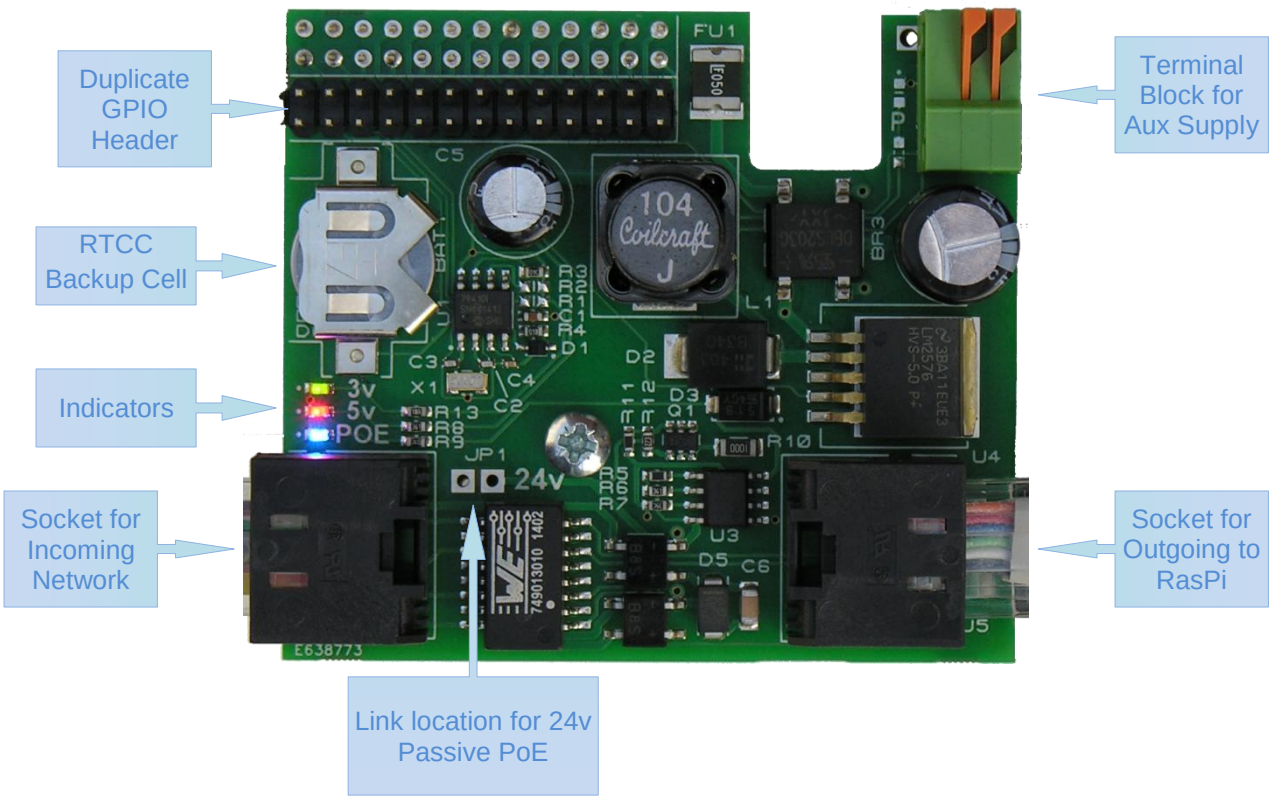
This Power over Ethernet adaptor fits directly onto the GPIO port of the Raspberry Pi and allows the Pi to have its power supplied through the network cable, so eliminating the need for additional power cables near the Pi.

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 Power over Ethernet standard
- Tool-free connection point for secondary (back-up) power supply
- Configurable for 24 volt passive PoE systems
- 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
- Duplicate port on adaptor allows full access to the Pi's GPIO port
- Mechanical support for board by supplied spacer and screws
- Network connection to the Pi by short link cable supplied



Use

Fitting the module

This is quite straightforward.

1. Put one of the screws through the Pi's mounting hole nearest the HDMI connector (this may be a tight fit and need to be screwed in) and screw on the spacer.
2. Fit the adaptor onto the GPIO port and then use the other screw to attach the adaptor to the spacer.
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.

Backup Supply

If a power-fail backup supply is required this may be either AC or DC, between 12 and 30 volts.

The backup supply is connected to the terminal block on the top right corner of the adaptor. Polarity of the connection is unimportant.

Connection is made by pushing the orange levers towards the back of the connector body and holding whilst inserting the wire. Releasing the lever allows it to return and grip the wire.

Indicators

There are three LED indicators next to the incoming network socket: a blue one, marked "POE" indicates the presence of a supply over PoE; a red one marked "5v" indicates the 5 volt output from the adaptor to the Pi; a green one marked "3v" indicates the 3.3 volts coming from the Pi, which is used to power the RTCC.

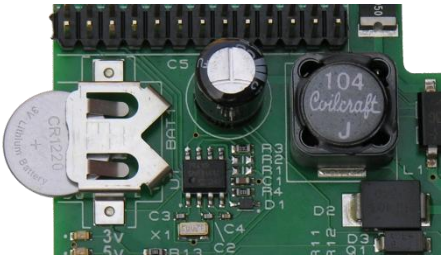
If the PoE supply fails and the backup supply takes over, the blue LED will extinguish but the red and green ones will continue to glow, thus indicating that there is still power connected but that it's not coming from the network.

24-Volt Passive PoE

In order to configure this adaptor to work with a 24v passive PoE system it is necessary to install a wire link onto the board between the two holes marked "24V". Once this is done, the adaptor will work as before but with the 24v system and not the 48v active IEEE 802.3af PoE system. Removing the link will restore 802.3af operation.

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. The cell should be fitted with the face displaying the "+" sign facing away from the printed circuit board.



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