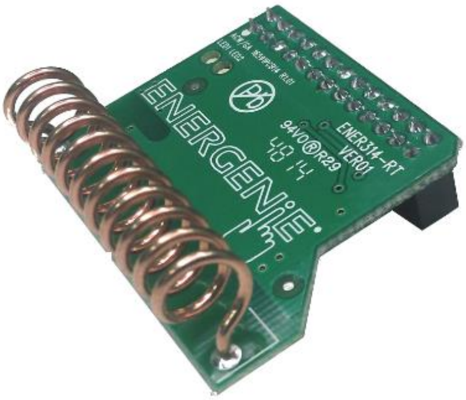
ENER002-2**PI-**RT Guide



Introduction

The ENER314-RT add-on board can be used to Control devices and monitor devices requiring 433 MHz ISIM band control directly from a Raspberry Pi equipped with the radio transceiver PCB which incorporates a radio receiver and a transmitter. All major RF communication parameters are programmable and most of them can be dynamically set. The ENER314-RT is fitted with a Multi-band transmitter module capable of FSK, GFSK, MSK, GMSK and OOK modulation. The ENER314-RT offers the unique advantage of programmable narrow-band and wide-band communication modes The ENER314-RT is optimized for low power consumption while offering high RF output power and channelized operation.

The unit will include the ENER314-RT board and 2 ENER002 radio controlled sockets. Users must install the latest BCM library in order to control the ENER314-RT. Users will be able to download the bcm libraries for BCM 2835-1.37 for older raspberry Pi's (Pre- RPi 2) required and software from our website: www.energenie4u.co.uk
If you own a Raspberry Pi 2 you must install the latest BCM Library. We found that bcm2835-1.5 works. You can download bcm2835-1.5 from here by following the instructions: https://gist.github.com/annem/3183536
The add-on board connects to the row of pins called the GPIO which can be controlled as either input or output lines under your software control.

s

Installing the Board

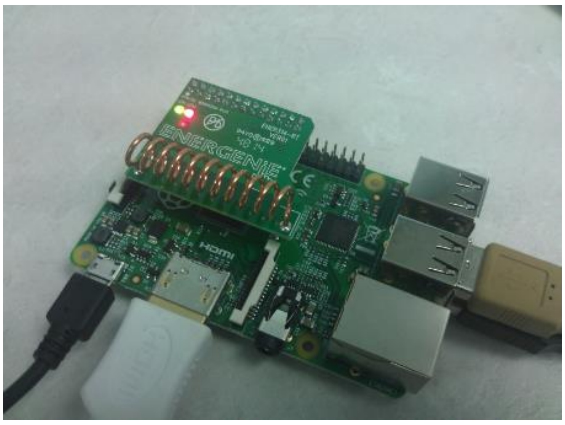
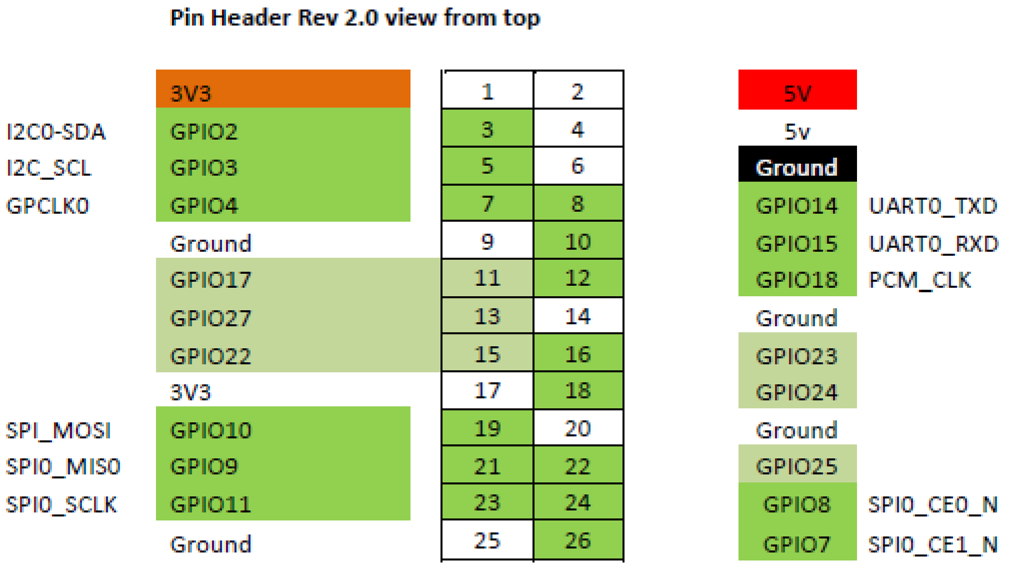


Figure 1 (Raspberry Pi B+ with RF Transceiver Board)

 Install the board on to the row of pins as show in the picture and connect your Raspberry-Pi as normal to a monitor, mouse, keyboard and USB power supply.

Note: The RF transmitter add-on board must be connected securely first before powering on the R-Pi. Connecting after the R-Pi is on may result in the device freezing.

The pin header connects to the add-on board as follows to allow you to control the GPIO lines as outputs to drive the radio frequency transmitter.



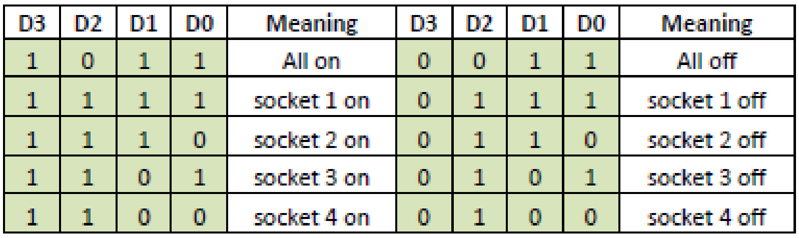
Pin Header Rev 2.0 view from top

	3V3	1	2	5V	
I2C0-SDA	GPIO2	3	4	5v	
I2C_SCL	GPIO3	5	6	Ground	
GPCLK0	GPIO4	7	8	GPIO14	UART0_TXD
	Ground	9	10	GPIO15	UART0_RXD
	GPIO17	11	12	GPIO18	PCM_CLK
	GPIO27	13	14	Ground	
	GPIO22	15	16	GPIO23	
	3V3	17	18	GPIO24	
SPI_MOSI	GPIO10	19	20	Ground	
SPI0_MISO	GPIO9	21	22	GPIO25	
SPI0_SCLK	GPIO11	23	24	GPIO8	SPI0_CE0_N
	Ground	25	26	GPIO7	SPI0_CE1_N

Figure 2 (GPIO pin header)

The board will communicate with the ENER002 radio controlled sockets using Each board transmits a frame of information using On-Off-Keying (OOK) which is a basic form of Amplitude Shift Keying (ASK). This frame includes source address (20 bits) and control data (4 bits).

Here are the pairs of codes using D0-D3 signals that can be sent to control sockets.



D3	D2	D1	D0	Meaning	D3	D2	D1	D0	Meaning
1	0	1	1	All on	0	0	1	1	All off
1	1	1	1	socket 1 on	0	1	1	1	socket 1 off
1	1	1	0	socket 2 on	0	1	1	0	socket 2 off
1	1	0	1	socket 3 on	0	1	0	1	socket 3 off
1	1	0	0	socket 4 on	0	1	0	0	socket 4 off

The receivers within the ENER002 have 4 address slots and the above codes are the combinations to control them.

The software package on our website contains the bcm libraries vital for functioning with the GPIO pins on the Raspberry Pi. bcm 2835 is a C library for Raspberry Pi (RPi). It provides access to GPIO and other IO functions on the Broadcom BCM 2835 chip, allowing access to the GPIO pins on the 26 pin IDE plug on the RPi board so you can control and interface with various external devices. It provides functions for reading digital inputs and setting digital outputs, using SPI and I2C, and for accessing the system timers. We have recently been made aware of the bcm library change for the Raspberry Pi 2. The bcm library (2835-1.37) does not support the new device tree introduced in the BCM2836 based kernels, so it gets the wrong peripheral base address for GPIO and SPI peripherals (it changed address on the new quad-core device). As far as we can see BCM2835-1.5 does work on the Raspberry Pi 2.

Setting up the sockets

Hold down the green power button on the adapter housing for 6 seconds until the red LED light on the socket begins to flash at 1 second intervals. The adapter has now entered learning mode. Wait until the program has run on the Raspberry Pi. The red LED lamp on the adapter will stop flashing and turn off. The adapter is has now paired with the selected button pair on the Raspberry Pi which can now be used to switch power to the socket part on or off.

The ALL on/off button can be used to switch all paired adapters on or off at one time.

Pressing the ON or OFF button on the adapter housing will also cause the socket power supply to toggle on and off.

Setting up Raspberry Pi

To set up the Raspberry Pi for the first time before running the software please download the software package found on our website. Copy all the files unto the Raspberry Pi. Install bcm2835 drivers if not already installed. To do this open the LXTerminal application found on the raspberry Pi desktop. Enter into bcm2835 directory. Once in the directory type the following command: `tar zxvf bcm2835-1.37.tar.gz`. This will unzip the folder.Then type `cd bcm2835-1.37`.Then Type the following commands.

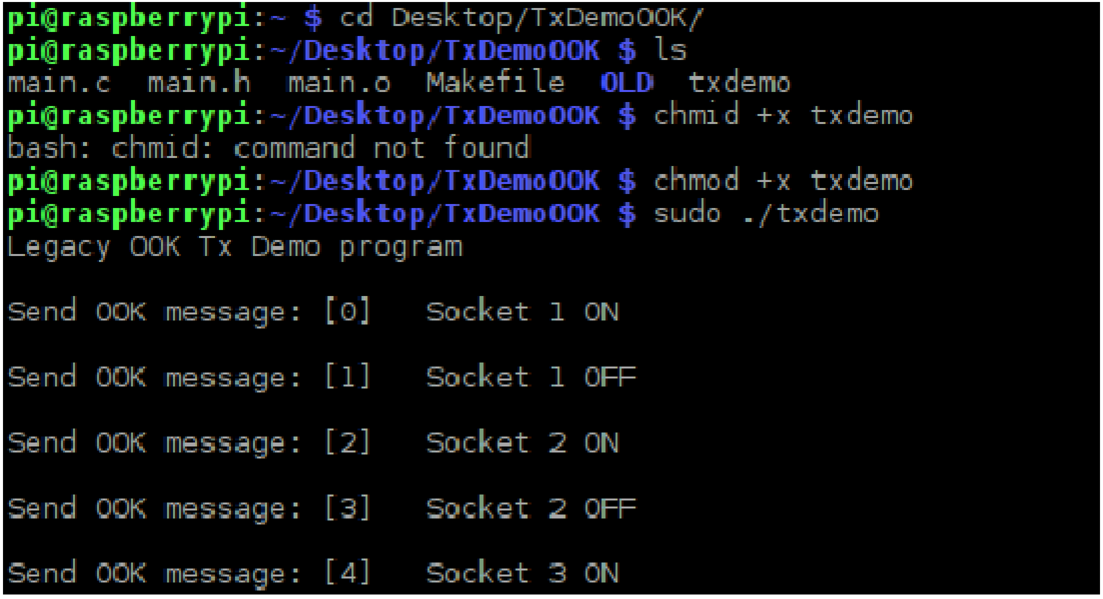
```
./configure
make
sudo make check (This command should pass the test)                    // test should
pass
sudo make install
```

This version of the bcm will work for Raspberry Pi A,A+,B and B+. You will need to install bcm2835-1.5 for Raspberry Pi 2. The bcm libraries have now been installed. Now exit the bcm directory using `cd ..`

Enter into the 'TxDemoOOK' directory. Then type the following commands:

```
make
chmod +x txdemo
sudo ./txdemo
```

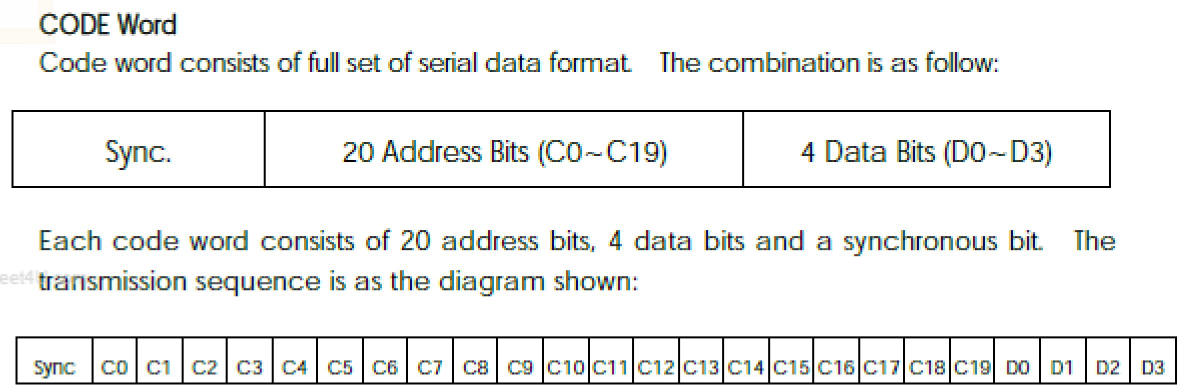
This will initiate the program. Press Ctrl+C to exit the program. The sockets
can now be paired with the Raspberry Pi. The first socket with Socket 1 pair and second socket with socket 2 Pair.



```
pi@raspberrypi:~ $ cd Desktop/TxDemoOOK/
pi@raspberrypi:~/Desktop/TxDemoOOK $ ls
main.c  main.h  main.o  Makefile  OLD  txdemo
pi@raspberrypi:~/Desktop/TxDemoOOK $ chmid +x txdemo
bash: chmid: command not found
pi@raspberrypi:~/Desktop/TxDemoOOK $ chmod +x txdemo
pi@raspberrypi:~/Desktop/TxDemoOOK $ sudo ./txdemo
Legacy OOK Tx Demo program

Send OOK message: [0]   Socket 1 ON

Send OOK message: [1]   Socket 1 OFF

Send OOK message: [2]   Socket 2 ON

Send OOK message: [3]   Socket 2 OFF

Send OOK message: [4]   Socket 3 ON
```

The program will send 10 different commands (ON & OFF channels 1-4 and All). There are a maximum of 4 channels per address. Users can alter the address of the Pi to increase the number of channels by changing the following line:

```
// Set up the 20 bit address field quasi random
        for (i = 5; i <= 14; ++i){
                buf[i] = 8 + (i&1) * 6 + 128 + (i&2) * 48;        // address 20b * 4 = 10 Bytes
        }
```



CODE Word

Code word consists of full set of serial data format. The combination is as follow:

Sync.	20 Address Bits (C0~C19)	4 Data Bits (D0~D3)

Each code word consists of 20 address bits, 4 data bits and a synchronous bit. The transmission sequence is as the diagram shown:

Sync	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	D0	D1	D2	D3