



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

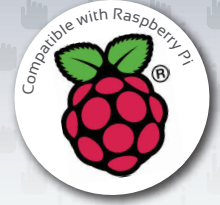
Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



# User Guide

## Pi RF-Transmitter Board



### Introduction

The RF transmitter add-on board can be used with the R-Pi to control simple Energenie radio controlled sockets using a small software program. We have written a simple program in Python to allow us to switch the sockets on and off with a single keyboard press.

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.



### Installing the 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.

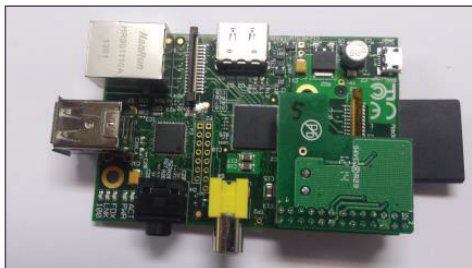


Fig. 1: R-Pi with piggy back RF-transmitter board

### How it is controlled by Software

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.

**GPIO Pin Header Rev 2.0**  
view from top

3V3	1	2	5V	
GPIO2	3	4	5v	
GPIO3	5	6	Ground	
GPIO4	7	8	GPIO14	
Ground	9	10	GPIO15	
Encoder signal D0	GPIO17	11	12	GPIO18
Encoder signal D3	GPIO27	13	14	Ground
Encoder signal D1	GPIO22	15	16	GPIO23
	17	18	GPIO24	Encoder signal D2
3V3	19	20	Ground	MODESEL mode select signal (OOK/FSK)
GPIO10	21	22	GPIO25	CE modulator enable (Output ON/QFF)
GPIO9	23	24	GPIO8	
GPIO11	25	26	GPIO7	
Ground				

Your software will use the outputs on the GPIO connector to drive the add-on board. We will write some code in Python to switch the two Energenie sockets on and off.

There are two parts to the transmitter board. The encoder and the modulator:

- 1) The encoder will accept 4 input signal levels programmed onto 4 of the GPIO lines (D0-D3) as shown above. It will then serialise them on a single line to the modulator part.
- 2) The modulator transmits the serialised signal. It needs to be programmed in Amplitude-Shift Keying (ASK) mode for the sockets using a GPIO signal. It also needs to be enabled by a separate GPIO signal.



Here are the pairs of codes using D0-D3 that can be sent to control sockets; we will use socket 1 and socket 2 in the sample program at the end:

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

Firstly, we need to install a Python module, RPi.GPIO, to enable software control of the GPIO pins for the Raspberry Pi. Open an LXterminal from the desktop and type the following lines:

```
sudo apt-get install python-rpi.gpio
```

Now, open a text editor such as leafpad and type the python code listed at the end called ENER002- 2PI.py into a new file under home/pi (or, more easily, download it from the [energenie4u.co.uk](http://energenie4u.co.uk) website and copy using a USB flash drive). This program will allow us to send coded commands to the sockets to program them and then switch them on and off when the return key on the keyboard is hit.

## Controlling the sockets

Launch your program by typing the following command at the prompt in the LXterminal window:

```
sudo python terminal ENER002-2PI.py
```

Then insert one of the radio controlled sockets into a mains wall socket which is switched on.

The socket must then be programmed to learn a control code from the transmitter. To do this the socket must be in learning mode indicated by the lamp on the front of the socket housing flashing slowly. If it is not doing this, press and hold the green button on the front of the housing (while the lamp is off), for 5 seconds or more and then release it when the lamp starts to flash at 1 second intervals. Then send a signal to it from your program by hitting the return key. Acceptance will be indicated by a brief quick flashing of the lamp on the housing which will then extinguish. Program first one socket then the other in this way, otherwise they will react to the same signal.

You can then toggle the sockets on and off by hitting the return key. You can also switch them manually on and off by briefly pressing the button on the front housing.

You can always reset the socket programming by holding down the green button for 5 seconds or more as mentioned before.

To increase the range of the transmitter you may wish to add an extra antenna to the circuit board. You can do this by soldering a piece of ordinary copper wire 13.5cm long into the hole marked ANT1 on the circuit board.

**IMPORTANT:** *The sockets will need to be inserted into separate mains wall sockets with a physical separation of at least 2 metres to ensure they don't interfere with each other. Do not put into a single extension lead.*

