

## easyRadio Advanced Connect2Pi Dongle Quick Start Instructions.



These instructions demonstrate the process to get up and running quickly using the connect2Pi radio module from LPRS on a Raspberry Pi running Rasbian.

It should be possible that other flavours of linux/raspberryPi should be similar.

### Step 1:

#### Plug the Connect2Pi module into either of the USB sockets on the Raspberry Pi

Note: Rasbian and most recent builds of RaspberryPi Distro's have the FTDI drivers built in.

If yours does not, the easiest fix is to 'sudo apt-get dist-upgrade' and get the latest distribution, otherwise follow this link:

[http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_220\\_FTDI\\_Drivers\\_Installation\\_Guide\\_for\\_Linux%20.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_220_FTDI_Drivers_Installation_Guide_for_Linux%20.pdf)

### Step 2: Open up a terminal. At the Prompt type dmesg:

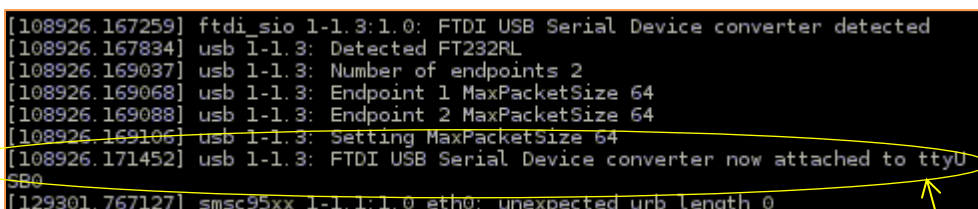


```

pi@raspberrypi: /
File Edit Tabs Help
pi@raspberrypi / $ dmesg

```

You should get a response containing something like:



```

[108926.167259] ftdi_sio 1-1.3:1.0: FTDI USB Serial Device converter detected
[108926.167834] usb 1-1.3: Detected FT232RL
[108926.169037] usb 1-1.3: Number of endpoints 2
[108926.169068] usb 1-1.3: Endpoint 1 MaxPacketSize 64
[108926.169088] usb 1-1.3: Endpoint 2 MaxPacketSize 64
[108926.169106] usb 1-1.3: Setting MaxPacketSize 64
[108926.171452] usb 1-1.3: FTDI USB Serial Device converter now attached to ttyUSB0
[129301.767127] smsc95xx 1-1.1:1.0 eth0: unexpected urb length 0

```

Note the line: FTDI USB Serial Device converter now attached to **ttyUSB0**

## Step 3:

### Install some Python tools that are not included in the Raspbian distro:

At the prompt type: `sudo apt-get install python-setuptools`

This will allow you to do step 4

## Step 4:

### Install pyserial module into python 2.x

This will allow you to run a python script which imports the serial module

Type: `sudo easy_install -U pyserial`

## Step 5:

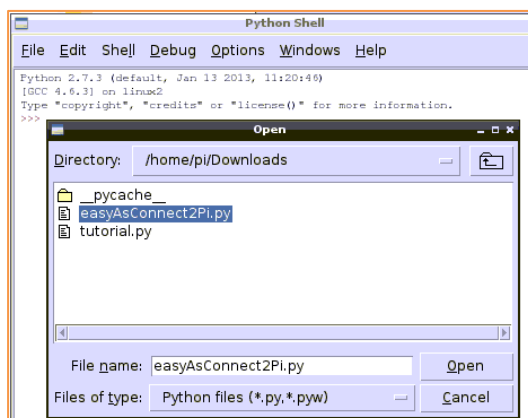
### Start programming (From now on this also applies if running Python in windows or on a mac)

First download `easyAsConnect2Pi.py` from [lprs.co.uk](http://lprs.co.uk) connect2pi download section

Launch 'IDLE' (not 'IDLE3') from Start>Programming or from the desktop icon



This will open a Python shell.



From the menu choose `file>open` and navigate to your file:

Open the file. A second window will open:

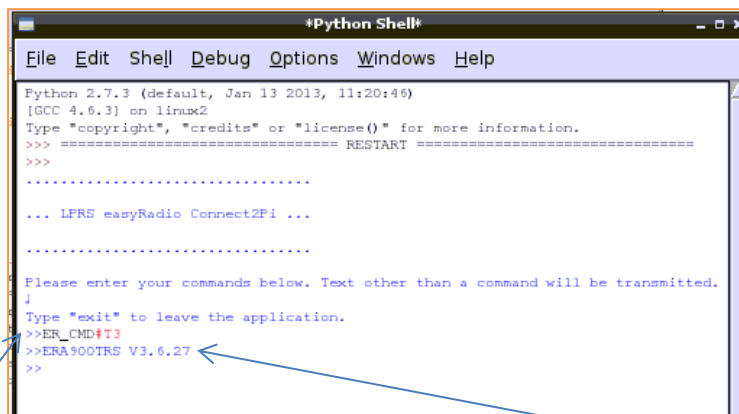
```
import serial

# configure the serial connection
ser = serial.Serial(
    port="/dev/ttyUSB0",
    baudrate=19200,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout = 0.5
)
```

Locate the code (near the top) and modify ttyUSB0 to be the same as you saw in in step 2

Hit the F5 Key to run (You will be prompted to save the file – click ok)

The Python Shell will now run your code:



```
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
...
... LPRS easyRadio Connect2Pi ...
...
Please enter your commands below. Text other than a command will be transmitted.
↓
Type "exit" to leave the application.
>>ER_CMD#T3
>>ERA900TRS V3.6.27
>>
```

Enter the command ER\_CMD#T3 and it will respond with the module's firmware e.g ERA900TRS V3.6.27

All valid commands will be processed and the responses displayed, otherwise they will be treated as data and transmitted automatically by the radio module.

Explore the file to discover how this is coded and modify for your own purposes.

Other command examples:-

- ER\_CMD#C?      Get Frequency channel
- ER\_CMD#U?      Get Uart value. Default is 19.2 kbps
- ER\_CMD#P?      Get Power level

Note: For complete table of ER commands see document **easyRadio Advanced Datasheet 3.6** or later, which is downloadable from [www.lprs.co.uk/easy-radio/downloads.html](http://www.lprs.co.uk/easy-radio/downloads.html)

## **Copyright**

The information contained in this data sheet is the property of Low Power Radio Solutions Ltd and copyright is vested in them with all rights reserved. Under copyright law this documentation may not be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form in whole or in part without the written consent of Low Power Radio Solutions Ltd.

The circuitry and design of the modules are also protected by copyright law.

## **Disclaimer**

*Low Power Radio Solutions Ltd* has an on going policy to improve the performance and reliability of their products; we therefore reserve the right to make changes without notice. The information contained in this data sheet is believed to be accurate however we do not assume any responsibility for errors or any liability arising from the application or use of any product or circuit described herein. This data sheet neither states nor implies warranty of any kind, including fitness for any particular application.

easyRadio modules are a component part of an end system product and should be treated as such. Testing to fitness is the sole responsibility of the manufacturer of the device into which easyRadio products are fitted, as is also the deployment into the field. **Any liability from defect or malfunction is limited to the replacement of product ONLY, and does not include labour or other incurred corrective expenses.**

## **Contact Information:**

Please Contact:-

### **Low Power Radio Solutions Ltd**

Two Rivers Industrial Estate  
Station Lane  
Witney  
Oxfordshire  
OX28 4BH

Tel: 01993 709418 Fax: 01993 708575

Website [www.lprs.o.uk](http://www.lprs.o.uk) email [info@lprs.co.uk](mailto:info@lprs.co.uk)