How to set up a Tiny Python Panadapter for Raspberry Pi!

By Josh Murray - M0JM0 - josh@m0jmo.co.uk - Twitter @JoshMurray
http://tigerstyleheavyindustries.wordpress.com
27th April 2014 - v1.8 (file name kept the same so links don't break)

I can't take all the credit for this process, as the majority of information has been pulled together from pieces scattered all over the Internet! I'm very new to Linux, so this was quite a steep learning curve for me. Unfortunately, I don't think I'll be particularly helpful if something in this document doesn't work for you; Google is your friend :)

This document describes the hardware, software and processes required to build a tiny computer with built in screen for use as a panadapter for radios with IQ output, such as the Elecraft KX3.

It uses the fantastic IQ Panadapter/Waterfall Software from Martin Ewing, AA6E - Thank you Martin, for creating this great tool.

Thanks too to Andrea IU4APC for proof reading and testing my documented process! Check out Andrea's amazing KX3Companion(.com) tool for Android devices!

OK, let's get started!

Go and buy the following items if you don't already have them:

- Elecraft KX3 or other radio with IQ output
- Cable to go between the radio's IQ output and the Sound Card input
- 512MB Raspberry Pi Model B Computer
- Pimoroni Pibow PiTFT Case
- Adafruit PiTFT 2.8" TouchScreen for Raspberry Pi
- Pack of 3mm Tactile Buttons (you'll need 1 but they generally come in packs of 5)
- 8GB Class 10 SD Card (minimum)
- Powered 4-Port USB Hub (the Pimoroni PiHub is very good)
- Micro USB Cable (the Pimoroni USB Noodle is nice)
- Griffin iMic (a USB Soundcard that is tested and works fine)
- "The PiHut" or Edimax USB WiFi Adapter for Raspberry Pi
- USB Keyboard and Mouse (I used wireless, that both used the same USB dongle)
- HDMI cable
- Monitor or TV with HDMI Input
- Another computer
- A USB Memory Stick (8GB)
- Internet Access
- Coffee
- Food
- Beer
- Oh, and some free Time! It shouldn't take **you** 20 hours though…

The easy bit…

1 - Download “Raspberry Pi NOOB” and unzip it on your computer (Google for the file!).
2 - Download and install “SD Formatter” on your computer (Google for the file!).
3 - Format your SD card using SD Formatter (full format, use the option).
4 - Copy the contents of the NOOB folder to the SD Card
5 - Insert the SD Card into the Raspberry Pi (RPi) and plug the powered hub in to the USB socket (with Keyboard, Mouse, WiFi and USB Soundcard attached). Plug the HDMI lead into the Raspberry Pi and into your monitor or TV.
6 - Insert the Micro USB Power cable into the Raspberry Pi (connected to a 5v 1A power supply) to Boot the RPi
7 - Select to install Raspbian OS and let it do its thing.
8 - Once installed, click OK.
9 - In raspi-config, change the following:
    (7) Overclock to 950MHz
    (8) Advanced Options
        (A2) Hostname - Enter a host name of your choice (e.g. **`panadapter`**)
10 - Login with Username **`pi`** and Password **`raspberry`**
11 - Type **`startx`** to start the GUI
12 - Double Click on the “WiFi Config” icon on the desktop, click Scan and select your wireless network by double clicking on it. Enter your WiFi password and click Add. All being well, after some handshaking, you’ll get an IP Address.
13 - Check you have Internet access… Click the icon on the far bottom left of the task bar (like the Windows Start Icon!) and select Internet>Midori
14 - In the address bar, type **`http://www.google.co.uk`** and hit enter. If you can get to Google, then that’s great and you have a working Internet connection and a working miniature computer! That was easy wasn’t it? The rest gets a bit more tricky!
15 - Start LXTerminal and type **`sudo apt-get update`** to update your RPi with the latest packages from the Internet.
16 - Install several packages.
Type (this is all one line!)

```
sudo apt-get install python-pygame python-libhamlib2 python-dev portaudio19-dev python-numpy
```

(Some of these may already have been installed). If asked whether it's OK to remove libjack-jackd2-0, Hit **`Y`** then **`enter`**. Go make a coffee, this takes a little while!
17 - We now need to install the PyAudio software. There may be a version in your repository (python-pyaudio), but we need version 0.2.7 or later, and the repository may not be up to that level. So, we need to download and install PyAudio "by hand":

You could download this directly on your RPi, but I could never get it to work, so on another computer go to:
`http://people.csail.mit.edu/hubert/pyaudio/`

Scroll down to "PyAudio Source" and click on "PyAudio Tarball.". Download the compressed archive pyaudio-0.2.7.tar.gz to a convenient folder, such as your home directory.

Copy this file to a USB stick and put the stick into your RPi.
Go to “Start” > Accessories > File Manager and look for your memory stick.
Copy the gz file across to your RPi home drive (/home/pi).

Open LXTerminal and Change directory (cd) to that folder by typing **cd /home/pi/** and then type **tar xzf pyaudio-0.2.7.tar.gz** to decompress and extract the folder PyAudio-0.2.7. Change (cd) to this folder by typing **cd PyAudio-0.2.8** and hit enter. Then type **sudo python setup.py install**

The process should complete in a minute or so.

Verify the installation by running Python and trying to import PyAudio:

Type **python** followed by the Enter key. Python will start and give its prompt: **">>>"**

Then type **import pyaudio** followed by the Enter key.

If there is no error message, you have successfully loaded PyAudio.

Hit the key combination **CTRL D** to exit Python.

18 - Install Network Time Protocol so your RPi has the correct time (once it connects to the Internet, since the Raspberry Pi has no real time clock).
Type **sudo apt-get install ntp**

This may already be installed.

19 - In the same way as you copied a file across from another computer to your RPi in Step 17 using a USB stick and the File Manager, copy across the compressed IQ application (get this from Martin Ewing, AA6E at **http://aa6e.net/wiki/File:Iq.tar.gz**) to your home drive on the Pi. (/home/pi/).
Open LXTerminal and type **cd /home/pi** (enter), then **tar xzf Iq.tar.gz** to decompress and extract the folder iq.

20 - Still in LXTerminal, change directory to the iq folder: **cd /home/pi/iq**
If you type **ls** (enter), you should see a bunch of .py (python) files listed.

21 - Connect the cable between the IQ output of your KX3 and the input on your USB Soundcard. Turn on your radio.

21 - Launch the iq application by typing the following line (this is all one line!):
nice -60 python iq.py --size=310 --n_buffers=16 --take=6 --index=1 --rate=33075 --WATERFALL --waterfall_acc=1 --FULLSCREEN

You should now see a window appear with a panadapter display at the top and a waterfall at the bottom. All being well, this should run reliably for hours on end without issue. If it crashes, then you'll have to start fiddling with the command to start the application, size, buffers, take and rate all impact performance. You're on your own there though I'm afraid… It took me hours of fiddling to get this working reliably on my RPi, and Andrea IU4APC helped refine it, so I hope it'll work properly on yours too. Hit **Q** in the IQ app to quit.

22 - Remember to shut down your RPi cleanly **EVERY TIME** you want to turn it off or reboot it, otherwise the file system on the SD card can become corrupted. Either use the Shutdown icon on the desktop, or if you're in LXTerminal or the command line, type in **sudo reboot** to

reboot, or **sudo halt** to turn off (it doesn't actually turn off, but the screen will go blank, then you can turn off the power).

Install the Screen and Case

23 - Right, now it's working, we can install the shiny Adafruit 2.8" Touch Screen and Pimoroni Pibow piTFT case.

I'm not going to re-type all the instructions for physically building the Adafruit screen here, as the instructions are pretty complex and photos really help, so just head over to their site and follow all the build instructions, including the supplemental instructions on installing the tactile 3mm switches:

https://learn.adafruit.com/adafruit-pitft-28-inch-resistive-touchscreen-display-raspberry-pi?view=all

Here are the commands needed to set up the software to support the screen, run it all in LXTerminal:

Type:

```
cd ~
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-bin-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-dev-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-doc-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi0-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/raspberrypi-bootloader-adafruit-20140227-1.deb
```

Next install the new kernel by typing **sudo dpkg -i -B *.deb**
This will take a few minutes so go make a sandwich or another coffee (maybe decaf this time!)

Type **sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~**

Type **sudo reboot** to reboot the RPi so it can all sink in!

Log back in (**pi / raspberry**) and without starting the GUI, run the following commands to install the screen driver. Note that the screen will go from white to black indicating the commands succeeded.

```
sudo modprobe spi-bcm2708
sudo modprobe fbtft_device name=adafruitts rotate=90
export FRAMEBUFFER=/dev/fb1
startx
```

The GUI should now display on the small screen :)

With the mouse, click the tiny power switch icon at the far bottom right of the screen and then click Logout to go back to the terminal.

We now need to make the modules auto load:
Type **sudo nano /etc/modules**

Add the two lines:

```
    spi-bcm2708
    fbtft_device
```

Then **Control-X Y [return]** to save

Type **sudo nano /etc/modprobe.d/adafruit.conf**
add the following line (it's all one line!)

```
    options fbtft_device name=adafruitts rotate=90
frequency=32000000
```

Then **Control-X Y [return]** to save

Now type **sudo reboot** to reboot the RPi.

We can set up the touchscreen for rotate=90 configuration by doing the following (for more delicate calibration or for other rotate=XX values, see the next section)
Create the directory and new calibration configuration file:

```
sudo mkdir /etc/X11/xorg.conf.d
sudo nano /etc/X11/xorg.conf.d/99-calibration.conf
```

and enter in the following lines, then save.

```
Section "InputClass"
        Identifier      "calibration"
        MatchProduct    "stmpe-ts"
        Option  "Calibration"   "3800 200 200 3800"
        Option  "SwapAxes"      "1"
EndSection
```

Save this file: **Control-X Y [return]**

You can now try to run X again with

```
FRAMEBUFFER=/dev/fb1 startx
```

The touch screen should now work too!

Quit X (the GUI) to go back to the terminal.

`To make X start up on the small screen every time, Type **sudo nano ~/.profile**

Add the following line near the top:

```
export FRAMEBUFFER=/dev/fb1
```

Save.

Type **sudo reboot** to reboot.

Log back in.

Type: **`sudo nano /etc/udev/rules.d/95-stmpe.rules`**

Insert the following line:

```
SUBSYSTEM=="input", ATTRS{name}=="stmpe-ts", ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```

Save.

Type: **`sudo rmmod stmpe_ts; sudo modprobe stmpe_ts`**

Then type: **`ls -l /dev/input/touchscreen`**

This should return some text with "event**n**" in yellow, where the **n** is a number.

Next type: **`sudo apt-get install evtest tslib libts-bin`**
Hit **`Y (enter)`** if prompted.

Once the installation is complete, type: **`sudo evtest /dev/input/touchscreen`**

If you touch the screen, you should see loads of text whizz by on the terminal.

`CTRL C` to end this test.

Now it's time to calibrate the screen!
DON'T USE ANYTHING METAL ON THE SCREEN!!
It is easily scratched… Go on, ask me how I know this!

Type:
```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_calibrate
```

Type:
```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_test
```

I couldn't find a way to close this application down after drawing a pretty star on the screen, so I had to ungracefully power off the RPi. This is bad practice :(

You can now calibrate the screen for the GUI.

Type:

```
wget http://adafruit-download.s3.amazonaws.com/xinput-calibrator_0.7.5-1_armhf.deb
```

```
sudo dpkg -i -B xinput-calibrator_0.7.5-1_armhf.deb
```

type: **`sudo rm /etc/X11/xorg.conf.d/99-calibration.conf`**

Type: **`startx`**

Open LXTerminal and type:

```
xinput_calibrator
```

Calibrate the screen and write down the numbers next to Option “Calibration” on a piece of paper!

Type: **sudo nano /etc/X11/xorg.conf.d/99-calibration.conf**

Enter the following lines:

```
    Section "InputClass"
                Identifier      "calibration"
                MatchProduct    "stmpe-ts"
                Option  "Calibration"   “3741 84 238 3841”
                Option  "SwapAxes"      "1"
    EndSection
```

Obviously the calibration numbers should be your numbers, off your piece of paper!
You might need to swap the numbers around on the line, or add a **#** at the start of the SwapAxes line to get the touch screen to work properly in X.

We now want to make the command line text look a bit better:
Quit X and in the command line, type: **sudo nano /boot/cmdline.txt**

At the end of the line, find the text that says **rootwait** and right after that, enter in: **fbcon=map:10 fbcon=font:VGA8x8** then save the file.

```
sudo reboot
```

Log in.

```
sudo dpkg-reconfigure console-setup
```

Select Teminus 6x12 font by pressing the following:
UTF-8
Guess optimal character set
OK
Terminus
6x12 (frame buffer only)

```
sudo reboot
```

We now want to make the left-hand switch on the screen act as a power switch:

```
sudo nano /etc/modules
```

add:

```
rpi_power_switch
```

Save

```
sudo nano /etc/modprobe.d/adafruit.conf
```

and enter in the line
```
options rpi_power_switch gpio_pin=23 mode=0
```

Save

```
sudo modprobe rpi_power_switch
```

Pressing the button at the bottom left of the screen while in the command line should now initiate the shutdown process on the RPi.

24 - Putting the Pibow case together is pretty self explanatory. Just install the first few layers, then insert your Raspberry Pi, then another layer or two, then insert your fully assembled screen into the GPIO pins and then carry on with the last few layers of case. Screw down and you're good to go!

25 - How to automatically login to Raspberry Pi as pi user.
Step 1: Open a terminal session and edit the inittab file.
sudo nano /etc/inittab
Step 2: Disable the getty program.
Navigate to the following line in inittab
1:2345:respawn:/sbin/getty 115200 tty1

And add a **#** at the beginning of the line to comment it out
#1:2345:respawn:/sbin/getty 115200 tty1
Step 3: Add the login program to inittab.
Add the following line just below the commented line
1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1
This will run the login program with pi user and without any authentication
Step 4: Save and Exit.

Press **Ctrl+X** to exit nano editor followed by **Y** to save the file.

Reboot the pi **sudo reboot** and it will boot straight on to the shell prompt pi@raspberrypi without prompting you to enter username or password.

26 - To make the IQ application fill the Adafruit screen properly:
sudo nano /home/pi/iq/iq.py and change line 262 of the iq.py application to:

```
SCREEN_SIZE = (320, 240) if opt.waterfall \
```

Save

27 - Create a bash script to launch iq with all the right switches:

```
cd /home/pi/iq
sudo nano startiq.sh

#!/bin/bash
# script to start IQ
echo "Starting IQ…"
nice -60 /home/pi/iq/iq.py --size=310 --n_buffers=16 --take=6
--index=1 --rate=33075 --WATERFALL --waterfall_acc=1 --FULLSCREEN
```

The line starting with "nice" is all one line, all the way to "fullscreen".

Save!

Type: **sudo chmod 755 startiq.sh**

Type: **sudo cp startiq.sh /usr/bin/**

Type: **sudo mkdir ~/.config/autostart**

Type: **cd ~/.config/autostart**

Type: **sudo nano iq.desktop**

in it, add the following:

```
[Desktop Entry]
Name=IQ Autostart
Comment=Automatically starts IQ
Exec=startiq.sh
Icon=lxterminal
Terminal=false
Type=Application
Categories=Configuration
```

Save

```
cp iq.desktop /home/pi/Desktop
cd /home/pi/Desktop
sudo nano iq.desktop
```

Change **Name=IQ Autostart** to **Name=IQ**

Save

Delete unnecessary icons from the desktop, so that the IQ icon becomes visible.
I have Shutdown, LXTerminal, Midori, WiFi Config and IQ icons on the desktop.

28 - Set up Auto Start of X:

```
sudo apt-get install xserver-xorg-video-fbdev

cd /usr/share/X11/xorg.conf.d

sudo nano 99-fbdev.conf
```

Enter the following into nano:

```
Section "Device"
  Identifier "myfb"
  Driver "fbdev"
  Option "fbdev" "/dev/fb1"
EndSection
```

Save!

```
sudo raspi-config
```

Select Boot to Desktop and reboot.

Job done! Your RPi should now boot directly into the IQ application!

Get a beer, you've earned it!!

Oh, while you're enjoying your beer…. **MAKE AN IMAGE OF YOUR SD CARD!!!**

If you use a Mac, this is a good guide on how to image your card:

http://computers.tutsplus.com/articles/how-to-clone-raspberry-pi-sd-cards-using-the-command-line-in-os-x--mac-59911

Next Steps…

Infra Red Remote Control!
I have just purchased an "Adafruit irKey with Remote" (available from Pimoroni or direct from Adafruit). It needs the following items changing in the iq.py file to be able to control the application:

```
cd /home/pi/iq
```

```
sudo nano iq.py
```

Change

```
elif event.key == pg.K_r:
```

to

```
elif event.key == pg.K_BACKSPACE:
```

Change

```
if event.key == pg.K_q:
```

to

```
if event.key == pg.K_x:
```

Save

This allows you to use the **Up**, **Down**, **Left**, **Right**, **Enter**, **Return** and **Stop** buttons on the infra red remote control within the application (**Return** button maps to **R** and **Stop** button maps to **Q**). No keyboard or mouse required!

USB to Serial for Frequency Display
It is very simple to use the Elecraft USB to Serial cable between the Pi and the KX3, just add the **--HAMLIB** switch when starting the iq.py application. I get strange noise on the waterfall when connecting this cable though (multiple vertical lines shown), so I don't use this. I get this on my other computers when using PSK31, so I think it's an issue with my cable rather than this setup.