How to set up the Raspberry Pi2 and Pi3 with Quisk. By Pete Juliano N6QW.

There is an amazing amount of inexpensive, yet powerful technology available today that causes such a magnetic attraction that you simply have to jump in with both feet. The Raspberry Pi2 with a quad core processor and many other on-board features had a huge neon display hanging over it saying "Buy Me" and so it was, and of course the next step was to couple this with some Softrock SDR transceivers that were lying in my junk box. Boom we were off to a new adventure. Little did I know that this was not simply plug and play and harkening back to when I had to install PowerSDR on my Windows XP Pro machine –reality sunk in again –why am I doing this? I am not the first to do this but I may be the first who smoked some parts, soldered their fingers together and has absolutely no clue about software especially Linux!

One of current crop of Linux based SDR programs is Quisk from N2ADR. I should also mention that while my initial efforts have involved the Pi2, in February 2016 the Pi3 has hit the market. I have one of those as well and these directions of "how to" apply to both models. The Pi3 in addition to having a quad core processor runs at 1.2 GHz (faster), has a built in WiFi (so no adapter required) has Bluetooth and is 64 bit capable. The earlier versions of the Raspberry Pi OS will not work with the Pi3. You must use the NOOBS OS 1.0.8 or later but the later versions are backward compatible with the Pi2.

Initially I struggled as a newbie with both the Pi2 and Quisk in my attempts to load Quisk on the Pi2. Quisk really is quite good and has many features. My problem is that I have never worked with Linux and many of the "everyone knows that" statements were unknown by me. By the time the Pi3 hit the market I had been through most of the early start-up problems.

Currently I now have rigs working on the Pi2 and Pi3 both using quisk. It is not without problems as initially I had difficulty in effecting the image rejection which I thought was the sound card, thus I bought a replacement model. Wouldn't you know –fools rush in and while getting the best seats are in the wrong theatre! Too late as the order was placed for a new usb sound card but it turns out it was one of my softrock transceivers. I now see greater than 50 dB of image suppression with the original card (StarTech 7.1), which actually exceeds the FCC criteria of 42 dB. So don't overlook the obvious.

Just like the Delta 44 by M-Audio was the sound card of choice for the SoftRocks (even for the Flex Radios) used with Windows and Power SDR, that card for the Pi's was identified as a Creative Labs X-Fi 5.1 Model 1090. Well I found the StarTech 7.1 to work better so you need not follow the crowd.

So given my late night head scratching and hair pulling, I thought it best to document the steps needed for the implementing the Pi with quisk. Unfortunately most of those who are using Quisk are either software engineers or seasoned veterans and simply "know what to do"! Thus this paper is aimed at folks who are like me and just want to have some fun but are starting at ground zero.

I simply don't understand terms like upload from the root using sudo nano /etc/udev/rules.d/local.rules the following lines of code h:skederwitch_on_the_framasaddle. You lost me at sudo which I think means you are a super user. This is a big hill to climb for new users –like me!

There are reflectors that support quisk and SDR where I have had some interesting experiences. When I posed various questions asking what I thought were serious questions about code or operations, one of the comments from a "quisk illuminati" was "that this is a useless thread." Frankly I don't believe there are ever dumb questions especially from a person that is trying to learn about new hardware and software. Too bad all of us didn't start coding at age 10. When I was 10 years old, coding was in its infancy using plug-board wiring and for the most part hard code didn't exist –neither did desktop

computers or cell phones. So please excuse an old timer who is just trying to learn something new.

Thus for those poor souls who like me who would like to embrace the new technology but lack in depth skills, I wanted to document how I made quisk work on my Raspberry Pi2 or Pi3.

The hardware part:

- I purchased a Pi2 Starter Kit from Element 14 which was a pretty good deal as you can fairly well get started right out of the box. However, having been there and done that I substituted a 32GB flash card for the stock 8 GB NOOBs pre-loaded card that came with the kit. Lots more room and there is now a later version of NOOBS 1.0.9 just hot off the press. You will need the latest version of NOOBS to work with the Pi3 but it is backward compatible with the Pi2. By the way NOOBS = New Out Of (the) Box Software. A Pi3 is also available as a starter kit.
- For a monitor I was using a 22 inch Samsung that has a DVI port. So the next purchased item was a HDMI to DVI cable + adapter. I got a shorter 3 foot cable and that works nicely. So now I can switch between my Windows machine and the Pi2. (Best Buy @ about $35 for the two). But I have now shifted to the Motorola Lapdock Atrix 4G. The Lapdock was intended as a docking station for one of the Motorola 4G phones. It has a two Micro style inputs for HDMI and USB. With this device you have the keyboard, mouse pad and 10 inch HDMI display all in one device. Brace yourself – with the Lapdock you will now have a Pi2 or Pi3 turned into a laptop computer. You will need to but some special cables to fit the male micro HDMI and USB connectors but these are readily available. A $100 bill will get you the Lapdock, power supply and cable adapters.
- Because only 4 USB ports are on the Pi2, I had a Targus USB port extender and so now have a total of 7 USB ports. This is a non-powered HUB but I am strongly advocating the purchase of a powered hub. Oh the Pi Foundation has a list of compatible accessories –check to make sure whatever you purchase is on that list. D-Link and Dynex have 7 port hubs listed as verified with the Pi2. When buying a powered hub look for ones that have individual Terminal Translators on each USB port. Many have a single terminal translator shared among the hubs. The shared approach will slow things down and may cause error messages.
- USB Sound Cards. My Pi2/Pi3's rigs are using two sound cards. The internal bcm2835 is used for the radio audio output. The external usb sound card is a StarTech 7.1 dolby card that has both stereo input and output ports. The external usb card is used for the main capture card. The StarTech does the heavy lifting and accommodates the microphone input. The StarTech was about $35. The StarTech which does the IQ tasks combined with the pulse audio software which is in the later download list does a superb job.
- Wireless USB is an Eidemax which is needed for the Pi2 but the Pi3 has an internal WiFi capability. I have a wired keyboard and wired mouse (generic). If I shifted to a Logitech wireless keyboard that would save a USB port. Moving to the Lapdock requires only one USB port for the keyboard and mouse. There are two extra USB ports on the rear deck of the Lapdock.
- 5" Touch Screen HDMI – a bear to install and way too small for quisk display. I went back to a larger HDMI and will save the 5" for some 'cutesy' future project – a waste of $60. For about $40 more you have the Lapdock with a 10 inch screen and an internal battery good for about 6 hours –perfect for portable QRP operations.

The software part:

- Using my windows machine and a program called SDFormatter I formatted the 32GB card. The NOOBS 1.0.6 was downloaded to the windows machine and installed on the formatted flash card. For the Pi3 you must use 1.0.8 or later –1.0.9 currently is on the Foundation download page.

- I then placed the 32 GB (actually 29.7GB now) in the Pi2 and applied power. It is pretty much automatic with regard to the installation –with the only note that at the bottom of the install screen you can select English (US) and US keyboard – otherwise you will have to do that now in **sudo raspi-config.**
- After the software is loaded (you will see a pop up screen that it was successfully loaded) the stock desktop (pretty sparse at the point) appears. The very first thing to do is to call up the terminal and type in **sudo raspi-config** and press enter. A menu will appear and you have many selections. There are 3 selections we'll address: 1) select internationalization and change the time zone for your location mine was US and Pacific new. 2) Select the keyboard option for your typical operation (US here) 3) Select Audio and force the Audio Output to the 3.5 MM jack (selection 2). You can also overclock the processor –but if you run the turbo version be sure to add a heat sink on the Broadcom processor. The Pi3 can't be overclocked (not easily like the Pi2) but also runs hot. Jameco Electronics sells a neat stick on processor heatsink for about $2. Close that out by selecting Finish and when you are asked about rebooting that will happen automatically upon closeout.
- The very next step is the install of the internet connection as this is needed to update the Pi2 and to download software. The Pi3 has a built in WiFi. At the upper right top of the pi browser is a network terminal symbol. Click on that and you are asked to enter your router password. Once that is done successfully (mine as suppled to me is a 10 digit alpha numeric and easy to screw up) the symbol will change from terminals to the flashing radio wave. Now you are cooking
- The first step is to insure you have the latest updates to the NOOBS and so calling up the terminal type in **sudo apt-get updates** and press enter. This action will update your Pi2/Pi3 to the latest configuration. If by chance you have an existing Pi2 with an older version of NOOBS then you would want to also **sudo apt-get upgrade.**
- Beyond the updates/upgrades from the Pi Organization you will also need other software so following the update or upgrade you will want to install a better and faster browser and a good choice is **iceweasel** (a variant of firefox—get it fire and ice and fox and weasel). Calling up the terminal type in **sudo apt-get install iceweasel** and press enter.
- At this stage the system operating in the background is python2.7. Python software was installed along with the basic NOOB 1.0.6 and later NOOBS. However if you somehow are using something else you will need python2.7 and **sudo apt-get install python2.7** will get you there. You can verify that you have python2.7 installed by using the terminal and typing in **sudo python** – this will show the version –in my case it came back **python 2.7.9.** So I was good to go. All future packages to be installed require the basic python2.7 shell. So get this step right!

This pretty much installs the basic Pi2/Pi3 software. Our next series of installations will be to recognize the two sound cards, adding in software so you can watch youtube videos and some others beyond basic but not quite yet quisk software.

At this point it is a good idea to do a **sudo reboot** just to make sure all is still working OK. Another test is to call up iceweasel and do a search on qrz.com for your call sign. This will then prove that the internet is working and that you can navigate to where you want to be

Initially in the raspi-config we had you force the on-bard sound chip to the 3.5 mm phone jack, The default setting is to have the sound come out through the HDMI port. I am not using a HDMI monitor but a HDMI to DVI-D adapter so if I want to hear anything this must be invoked. My set-up actually is using two sound cards and so now we have to do several things: 1) is to tell the Pi2/Pi3 we are using two cards and the priority of those cards –ie which one is used first.

- Step 1 with the external usb card plugged into the Pi2/Pi3, at the command prompt type in **lsusb.** This command asks the Pi2/Pi3 to LIST (ls) all of the USB (usb) devices attached to the computer. You should see an entry for sound cards. Also look for the other devices such as keyboard, mouse, hubs, and the USB to I2C
- Step 2 identifies the cards to the computer as being used. The lsusb just says they are there but the next command says they are in use. At the command prompt type in **sudo nano /etc/modules** . This will bring up a data entry screen where you can see modules –typically at this point you may only see i2c. Navigate using the down arrow key to an entry below the i2c and type in the following (on two separate lines –after making the 1st entry hit the enter key). The 1st entry is **snd_bcm2835** and then hit enter. The second entry is **snd-usb-audio** and then hit enter. Note the 1st entry uses an underscore and the second dashes. Once you have done that we need to save that information and we do so by first simultaneously pressing ctrl and X at the same time. Next we simultaneously press shift and Y and finally Enter. We should now be back at the command prompt.
- Step 3 now establishes the order of the cards in terms of how the computer routes sound for default purposes. Here we go again enter at the prompt **sudo nano /etc/modprobe.d/alsa-base-conf** which now routes you to a pop up screen. You will make three entries with the index part being computer speak – "0" is the first configuration. So now make one entry per line. Using the arrow keys move down to an open area and the first entry is **options snd-usb-audio index=0** and then press enter. The second input is now **options snd_bcm2835 index=1** and after that entry press enter. To finish the job we must now do **Control X, Shift Y and Enter.** Once more reboot the computer!
- This now completes the basic set up for SDR hardware and all future installs are software packages needed to make quisk run. Many are libraries that have many embedded files and this process will take some time. During the course of installation at times you will be asked if you want to proceed with the install as a significant amount of SD card space may be chewed up. This is done mainly because some SD cards are only 2GB and so may be on the ragged edge. WE strongly recommended the use of the 32 GB SD card. For info purposes when I got done I used about 7+ GB –so even an 8 GB may be on the edge. A 16 GB card would be the smallest I would use.

The next series of software installation basically fall into three categories:

- Files (libraries) that are needed to execute/support the Quisk software
- The Quisk program (downloaded from the Internet) 4.04 is the latest.
- Other ancillary programs

The following are the Files/Libraries to Support quisk. These are done at the command prompt and are preceded with **sudo apt-get install** and you must have an internet connection. Try to install these packages in the order shown.

- libfftw3-dev
- libasound2-dev
- libportaudio2
- portaudio19-dev
- libncurses5-dev
- libwxbase2.8-0
- libwxgtk2.8-0
- python-wxversion
- python-wxgtk2.8

- libusb-dev
- libpulse-dev
- python2.7-dev
- pavucontrol

It should be noted that many of these libraries involve sound which of course is the heart of the sound card based SDR. The fft3W is the Fast Fourier Transform needed for the display. The pulse audio is a critical element for internally looping sound streams. For those familiar with the SDR via Windows/PowerSDR you had to install special software called VAC (virtual audio cable) that had to be purchased form a guy in Russia. Pulse Audio does this in Linux and so no additional software for the looping. For those into digital stuff like Fldigi or WSPR –this was critical to make those programs play. Now it is built in with a simple free download.

The next part involves the downloading of the quisk software and that may be found at the following location http://james.ahlstrom.name/quisk/index.html At this location you will find links to the latest quisk which up until about two days ago was 4.0.3 – Much of the older documentation has one install files and creating a quisk_conf.py file. That process was required for earlier versions but the 4.0.3 has much of that earlier drudgery work already included. I was confused by this and until some kind soul put me straight, did I realize there was an easier way. You can download 4.0.3 using iceweasel and select the option to save rather than extract to some arcane location buried in Python. The file will automatically be download into the home/pi/Download folder.

Next in the home/pi directory create a new folder called "quisk". Make sure it takes and that you can see it. The folder is created by opening the home/pi directory and up on the task bar is File click on that and a drop down menu has a choice Create > Folder – click on that selection and enter the name **quisk** and then click OK. You should now see that folder in the home/pi directory.

Next go to the Download folder and double click the folder and you should see a compressed version of quisk. Right click on the file and select "extract to another location" find the quick folder and say OK. Boom you will now have a single file folder which will do you absolutely no good. Thus double click to open all the sub files. (Note: one of the task bar icons is View click on that to engage show hidden files. Next go up to Edit and on the drop down menu click on "select all". Follow that with an Edit ~Copy files. Next place all of those files back into the quisk folder. So now the quisk folder will have booth the source 4.0.3 Folder plus all of the files in the folder.

Future navigations will be /home/pi/quisk –that is why we did all of this.

The way quisk has been set up is that it can be used with many radio including the Softrock version, the KX3, the Heremes, the Peaberry and even homebrew IQ type front ends. There are folders within the quisk folder to cover these various radios. Upon the initial launching of quisk there are generic inputs which must be changed to match the radios. Many of the earlier installed libraries involve the sound cards. You will need at least two cards and will even work with three cards. One of the cards is used for the radio's audio output and that can be the internal on board sound device. The second external card can handle the IQ processing as well as the microphone input. There is a button on the front panel called config which leads you various tabs where changes will be made. More on this in a later section.

We are not done installing software. Before we actually set up quisk we need to be able to control the USB function as this will be how we can make the changes in the panel. The information provided in the quisk documentation simply did not work for me and thanks to a kind soul a slightly different input cured the problem. At the command prompt type in the following line:

sudo nano /etc/udev/rules.d/99-softrock.rules a pop up screen will appear and you must type in the information exactly as below

SUBSYSTEM==”usb”,ATTR{idVendor}==”16c0”,ATTR{idProduct}==”05dc”,MODE=”0660”,GROUP=”dialout”

Once you have that typed in a double checked then hit Ctrl x and the Shift y and then Enter. This will take you back to the prompt screen. Now double check that you have loaded all of the software and we are ready to finalize the installation.

Most of the final work will be done in the quisk directory so start at the command prompt with

cd /home/pi/quisk this will no put you in the quisk directory and type in **make –j4** which is for the computer to match the hardware with the software. With this command you will see a flurry of activity on the screen to indicate things are being set up for installation. (The make –j4 is so that all four parts of the quad core processor are used in the setup.)

Once that has been complete we will now setup the quisk software. While in the quisk directory type in **sudo python setup.py install**. Once again there will be a flurry of activity on the screen and you will know if it was successfully installed. It is now a good idea to reboot the system to make sure everything is in its proper place.

Once the reboot has completed then it is time to launch quisk – but only after you have installed the soundcards, hooked the radio and connected all devices. Reboot the Pi2 with all else powered on and the check to see if you have everything that should be connected. You can do this at the command prompt by typing in lsusb, which is asking the Pi2 to list all of the usb devices connected. Look for the external sound cards and the usb interface to the softrock. This is an important step in that all devices must be present or quisk will not properly launch.

cd /home/pi/quisk
python quisk.py