

A Pi SDR Standalone Rig

Pi SDR IQ Plus with Quisk, WSJT-X, FLDigi for with Softrock/Ensemble or RS-HFIQ rigs

The new firmware for Pi SDR that gives standalone operation, is including already installed and configured the Quisk, WSJT-X, Fldigi, Hamlib applications and Virtual audio streams.

It can be downloaded in order to be flashed to the microSD from this link

-> <https://mega.nz/#!NHBCRAQA!IHf6jtemtVdGqQr9txQ-Raeerpu1Iib0eFOEfF5RgSY>

The name of downloaded file is "pisdr_standalone_v1_1_lan.img.gz"

The cable connections are according to Hardware Setup Guides, yet without the PC.

Instead, there is need for a HDMI monitor and mouse/keyboard (I use a wireless dongle having both in combo a keyboard and a mouse from Logitech). The HDMI monitor is connected at the side of Pi SDR IQ Plus where is an HDMI out connector, also the keyboard and mouse can be connected to any of the front USB Host connectors.

The included applications, Quisk/WSJT-X/Fldigi are configured up to the point where they can operate together, eg their audio streams and CAT, though other configurations and calibrations should be performed by the operator, eg frequency/level/Rx image/Tx image/user callsign/locator setup for digimode apps etc.

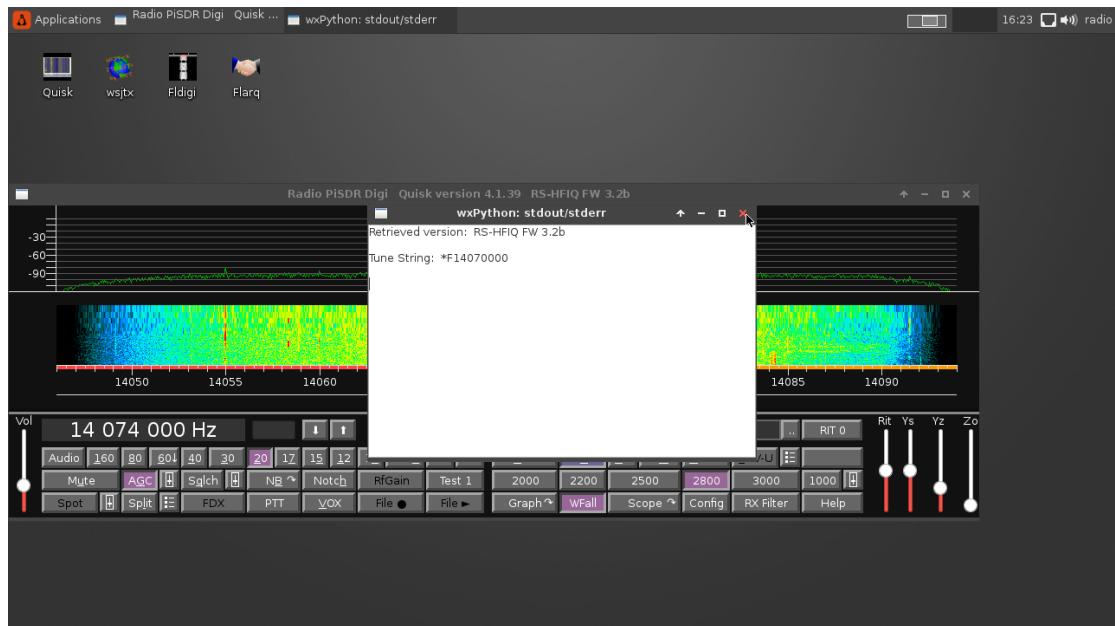
At first, upon startup, the firmware image is configured to have 'Blank' Quisk setup so user all that has to do is to select what radio is using by executing (double-click) the relevant desktop provided scripts, these scripts perform automatically the configuration required.

Each provided rig Quisk setup has a dual configuration installed, one for digimodes as default and one for phone operation, user can select ad-hoc each one that wishes to operate exclusively.

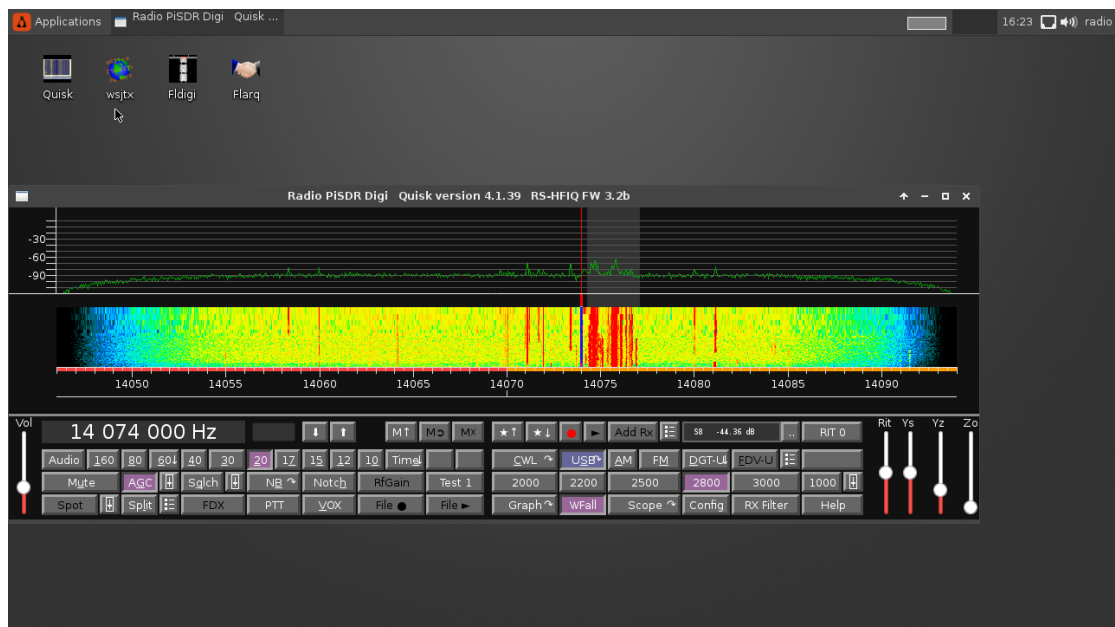


That is the first screen that should be shown after about 30sec from power on.

At that point, by double clicking on the Quisk desktop icon, the application will startup, as mentioned in digimode configuration so no sound will be heard.

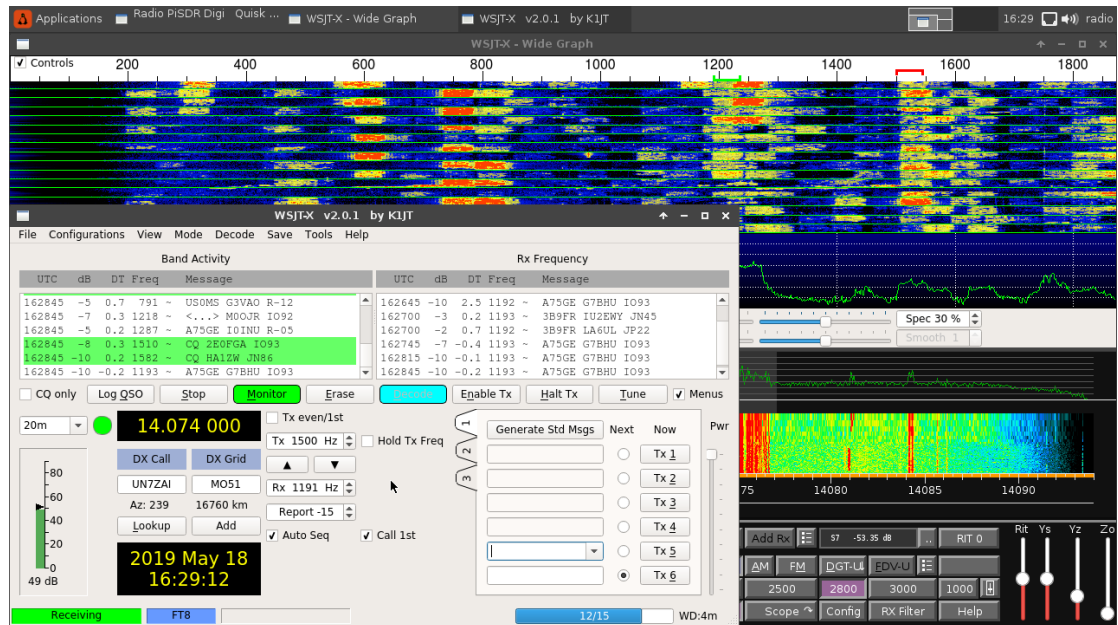


User should minimize or close the stdout/stderr python window upon popup. It is probably from the RS-HFIQ python code that is specific for rig comms.



At that point, user can double click the WSJTX desktop item and in a few seconds the application will start.

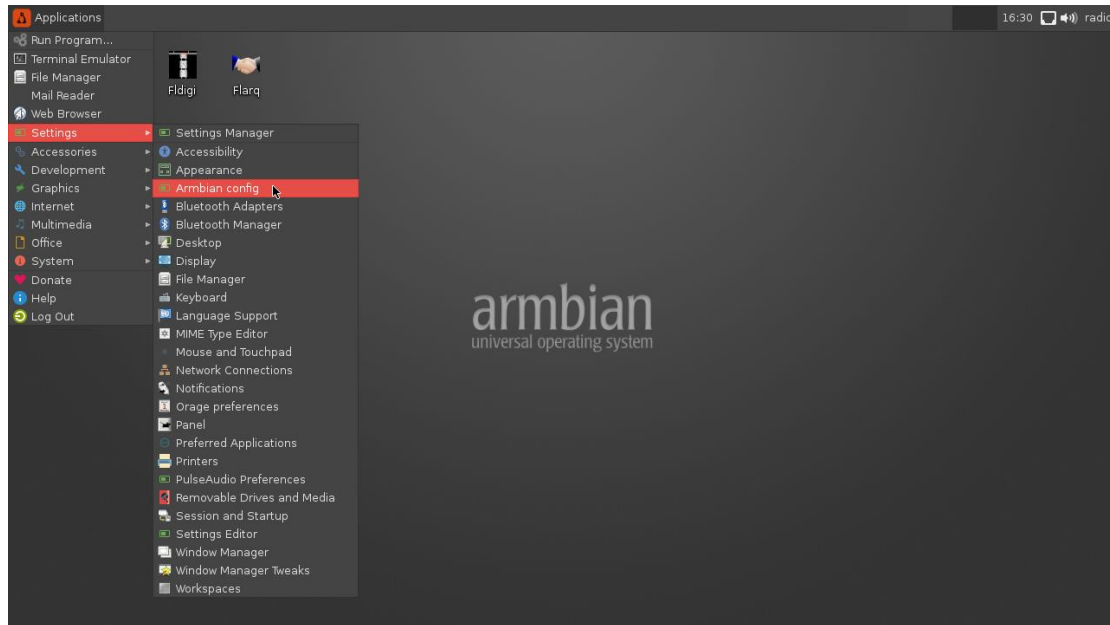
Since CAT and audio streams are already configured, after WSJT-X starts, user should select the band and decoded signals should show up after a while. Pay attention to the mode (LSB/USB) in Quisk and change it from there if required.



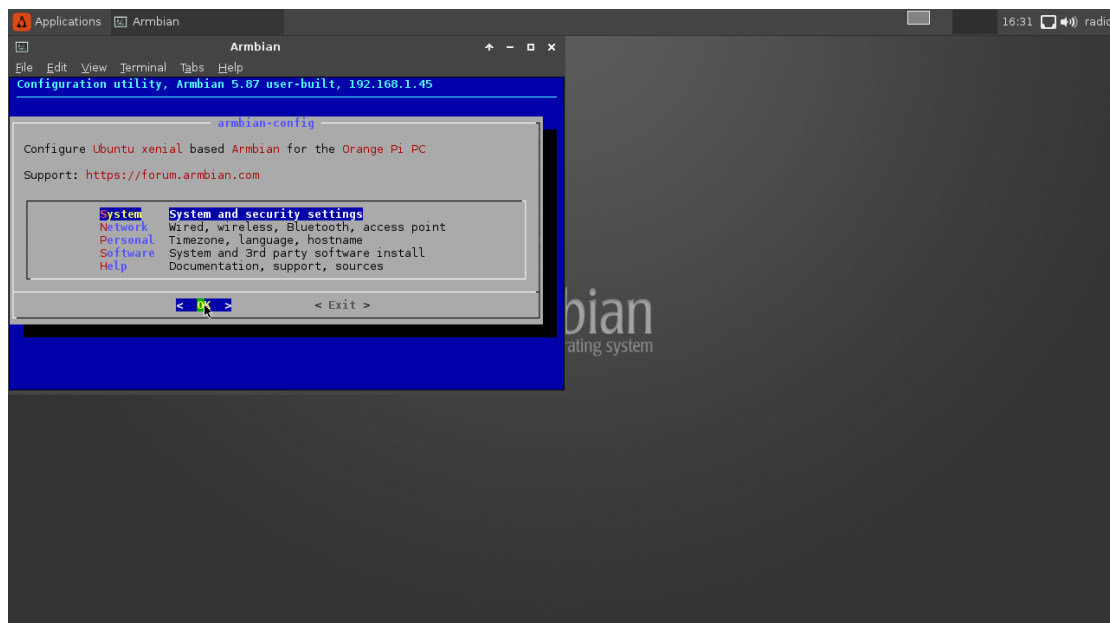
As it is firstly given, the firmware image has a relatively small screen size, a 720p resolution at 60Hz, “720p60”.

If using a larger monitor, user can change the screen resolution in the following way through the menu.

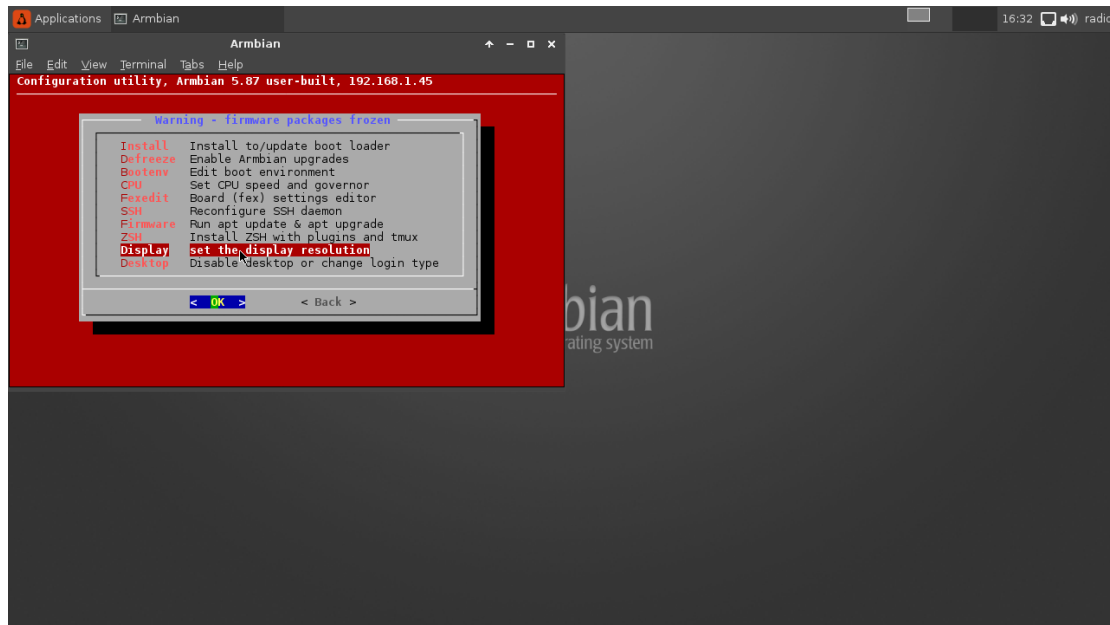
->Applications->Settings->Armbian config



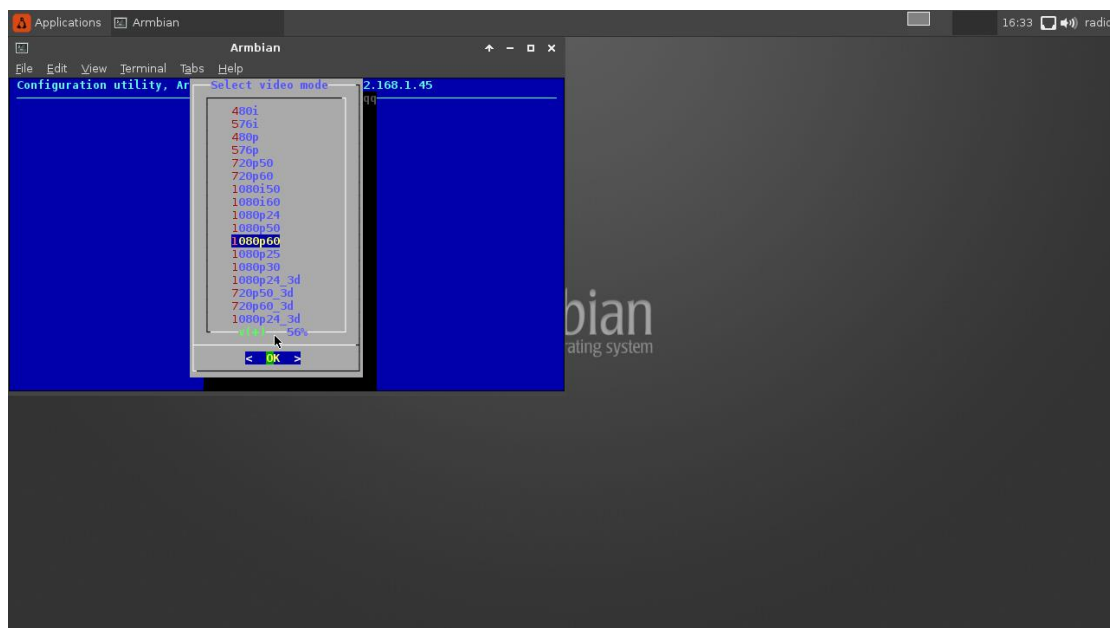
Navigation in this utility is with arrow keys, space and enter, first select the “System” settings with arrow keys and press enter



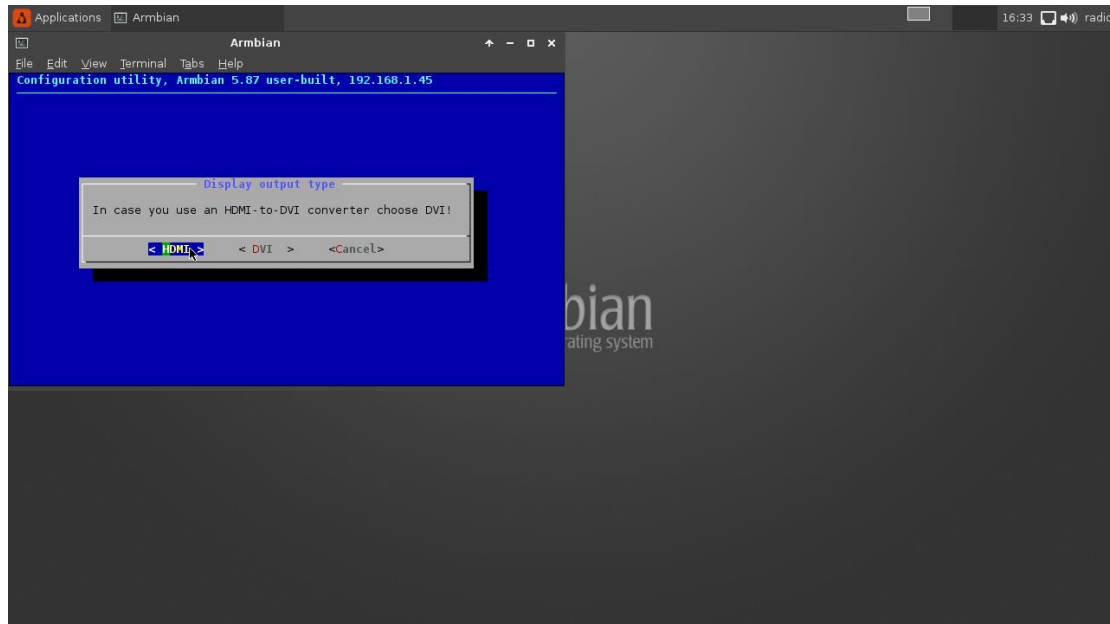
Then in the following screen select the “Display” and press enter



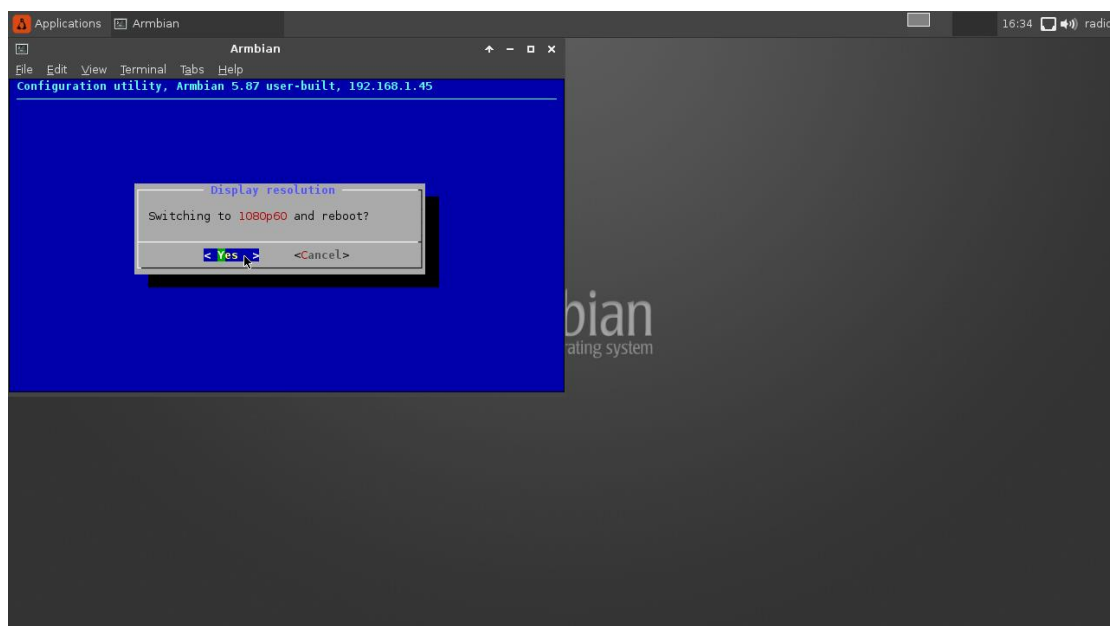
In this list select the resolution type of the attached HDMI monitor (as shown a new one 1080p60 is selected) and press enter



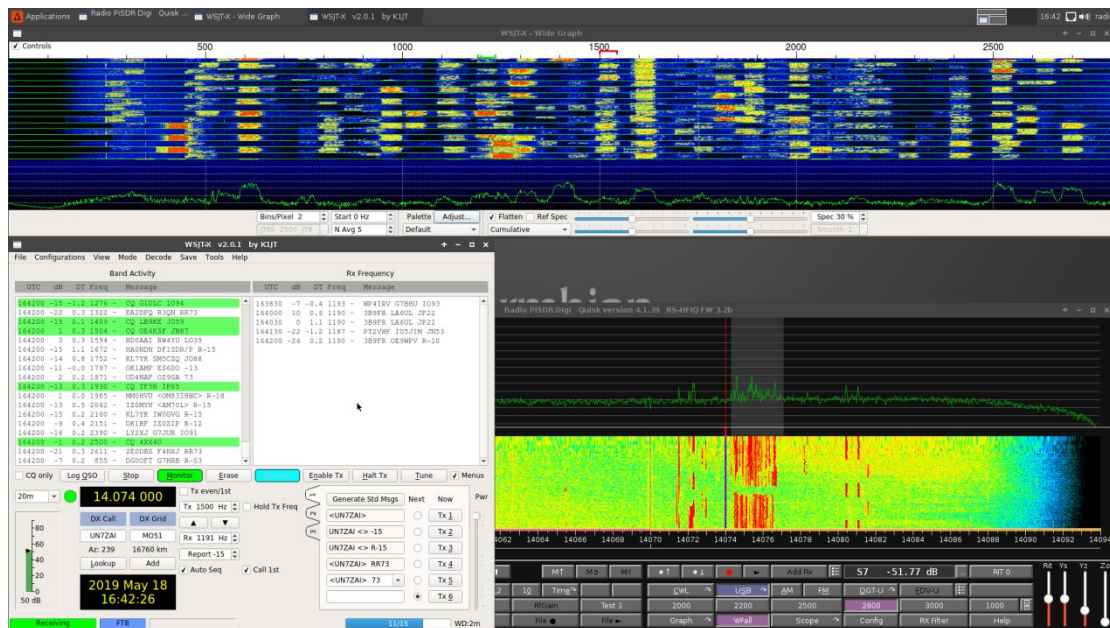
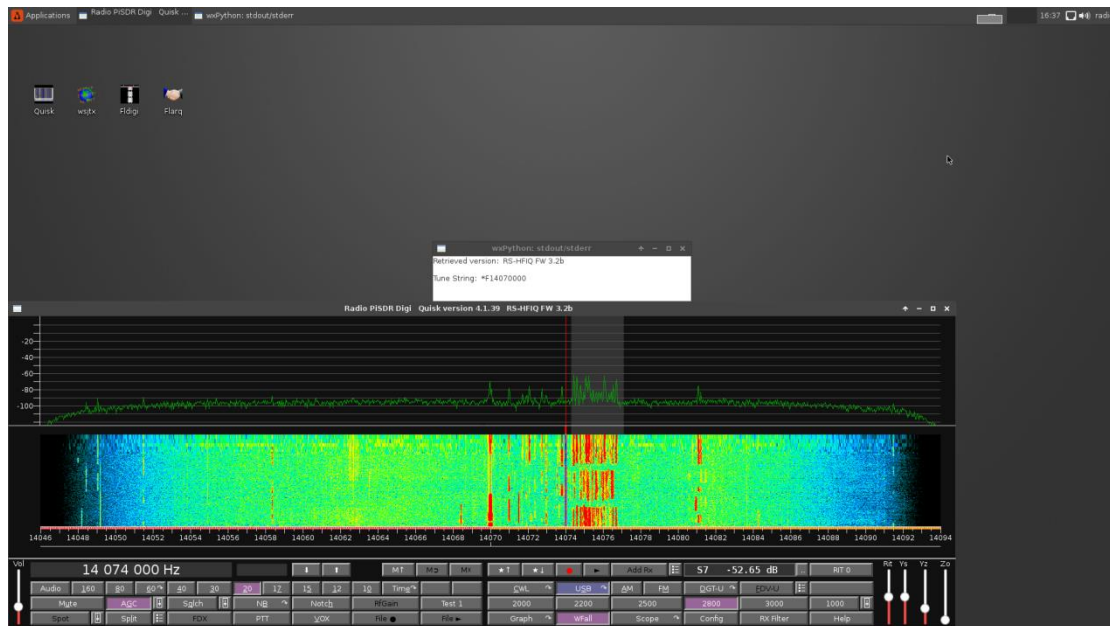
Enter again if you have a plain HDMI cable.



And the final confirmation since after that a system reboot will happen.



Upon the next start and by clicking the Quisk icon, the new screen size is applied



Also if phone operation is required, it is needed a third-party USB dongle with headphones/mic that should be connected to one of Pi SDR IQ Plus USB ports, since the on-board mic jack cannot be used in this configuration. If a user do not use Tx in phone modes (Rx only) and not having a USB audio dongle, it can place in Quisk configuration “Radio Audio Output” to “hw.0”, thus the onboard speaker output will be used.

The screenshot displays the Quisk software interface, which is used for managing radio equipment. The top bar shows the application name 'Quisk' and the version 'Quisk version 6.1.39'. The main window is titled 'Radio PiSDR Digi' and contains several tabs: 'Status', 'Config', 'Sound', 'Favorites', 'Tx Audio', 'Help with Radios', 'Radios', 'PiSDR', and '*PiSDR Digi*'. The 'Status' tab is currently selected, showing the following information:

- Sample interrupts: 123310
- Microphone level dB: -84
- FFT number of points: 3120
- FFT number of errors: 193
- Configuration file not found /home/radio/quisk_conf.py
- RS-HFQI FW 3.2b

Below the status information, there is a table showing the device configuration:

Device	Name	Sample rate	Latency	Errors
Capture radio samples	hw:1	48000	1088	1
Capture microphone samples	pulse:vac2_monitor	48000	42937	0
Play radio sound	pulse:vac1	48000	111756	211
Play microphone sound	hw:1	48000	2624	4

The bottom of the interface shows a control panel with various knobs and buttons for adjusting the radio's settings, including frequency (14 074 000 Hz), gain, and other parameters.

Applications Radio PiSDR Digi Quisk

Quisk w7x Fdigi Flarq

Radio PiSDR Digi - Quisk version 6.1.39 RS-48Q FW 3.2b

Status Config Sound Favorites Tx Audio Help with Radios Radios PiSDR *PiSDR Digi*

Quisk is running with settings from the radio PiSDR Digi. The hardware type is SoftRock Fixed.

When Quisk starts, use the radio **PiSDR Digi**

Add a new radio with the general type **SoftRock USB** and name the new radio Add

Rename the radio named **PiSDR** to the new name Rename

Delete the radio named **PiSDR** Delete

Restart Quisk with new settings Restart Quisk

14 074 000 Hz

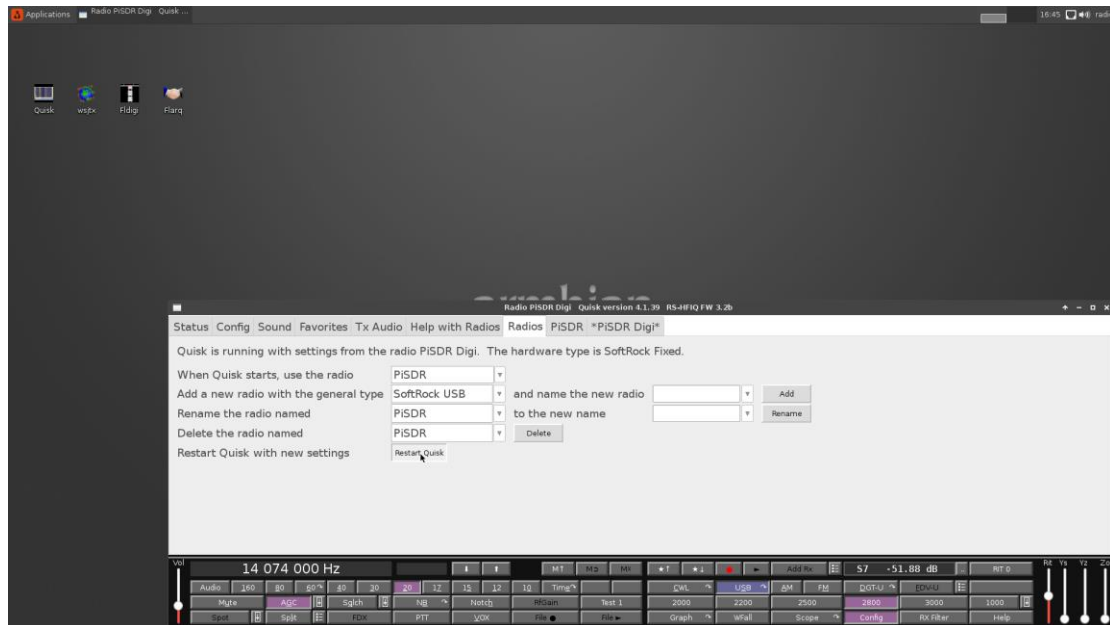
Audible 150 80 60 40 30 20 12 15 12 10 RT1 RT2 RT3

Mute **Agc** Split Sghch NB % Notch HScan Test 1 CWL USB AM FM DQTU EDNU

Spot Split FDx PTT YOK File Filew Graph WWall Scope RX Filter Help

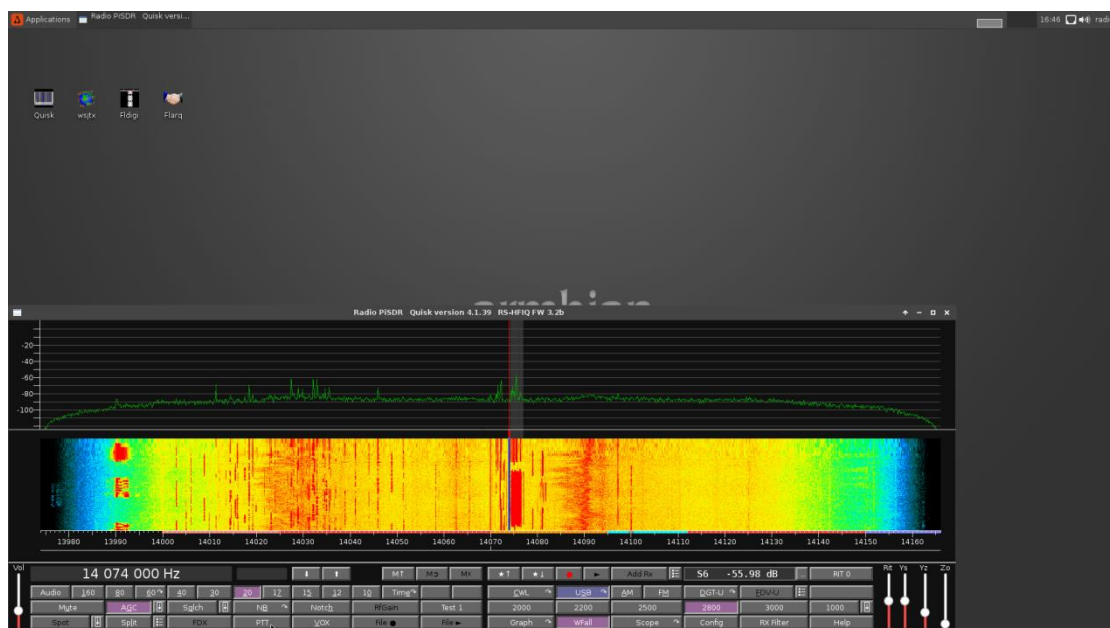
Vol -52.87 dB RT 0 RT 1 RT 2 RT 3

There user can select that the “PiSDR” configuration should be active next on instead of the PiSDR Digi and then click on the “Restart Quisk” button. Quisk will close and when it comes up again it will be in phone configuration.

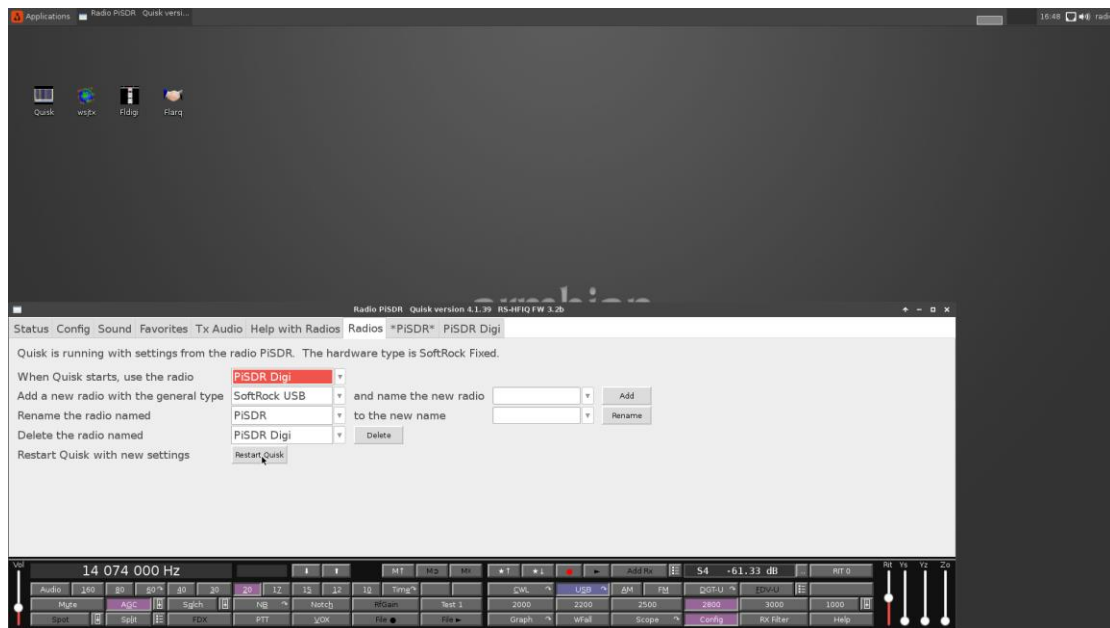
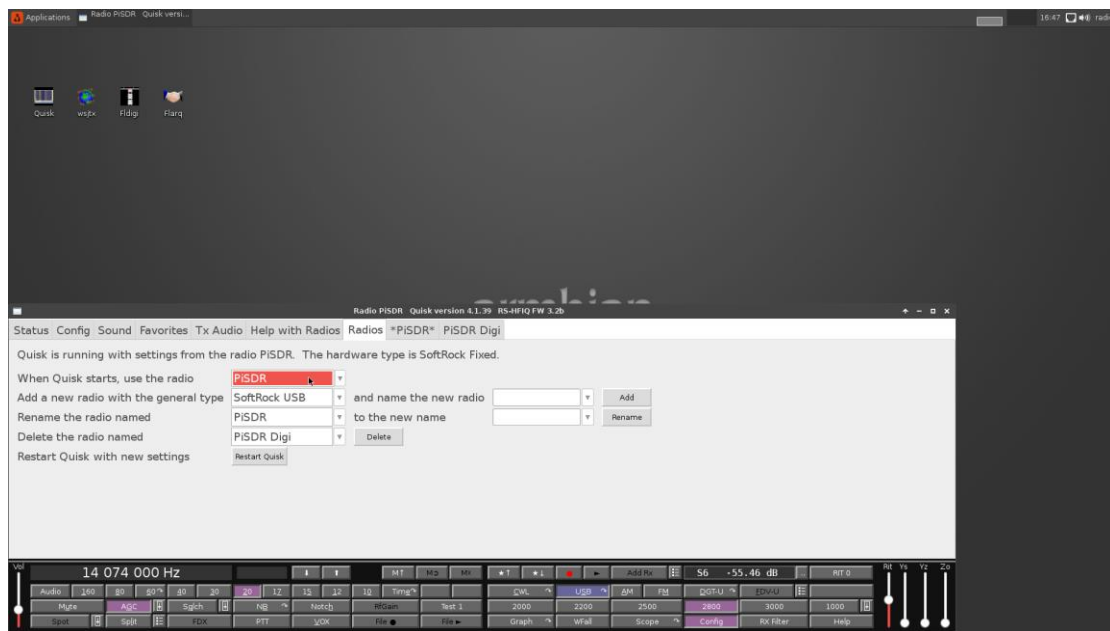


And here is Quisk in phone configuration.

A difference between the two configurations, appart from the audio streams, is the sample rate, in PiSDR Digi the samplerate/bandwidth is 48KHz whereas in PiSDR it is 192KHz. User can change that if it is needed via the relevant Quisk “Config” menu.



Also, user can change back to Pi SDR Digi via the same method



After each time “Restart Quisk” button is used to switch Quisk configuration, Quisk closes and restarts automatically. Even though that happens, after the restart, close manually and start manually Quisk as a precaution to properly store settings.

The userID, is 'radio' and its password is '123456789', it should be needed when changing HDMI screen size/resolution.

User can also connect via PuTTY over the network to the Pi SDR using the above credentials. This way it can change HDMI screens by executing “sudo armbian-config”

```
radio@pisdr: ~
login as: radio
radio@192.168.1.45's password:

  _ _ _ _ _
 | | | | |
 | | | | |
 | | | | |
 | | | | |

System load: 0.18 0.48 1.01 Up time: 1:25 hour
Memory usage: 15 % of 997MB IP: 192.168.1.45
CPU temp: 45°C
Usage of /: 63% of 3.4G

[ 0 security updates available, 2 updates total: apt upgrade ]
Last check: 2019-05-18 17:04

[ General system configuration (beta): armbian-config ]

Last login: Sat May 18 18:28:31 2019 from 192.168.1.11

radio@pisdr:~$ sudo armbian-config
[sudo] password for radio: 
```

[illegible]

There are plenty of configuration items that could be tweaked, though these I hope are the most essential for an easy startup.

The standalone firmware works with Softrock/EnsembleRxTx or RS-HFIQ rigs, for those two types there are relevant configurations that are enabled via desktop executable scripts. These scripts and configurations are located in /home/radio/quisk_configurations folder if a user wants to examine or modify them.

Overall, the supported rigs are those that there is Quisk support for them.

For anything regarding additional configuration a user can read and study the relevant documentation pages from the installed applications, so here are their links

Quisk

-> <http://james.ahlstrom.name/quisk/docs.html>

WSJT-X

-> <http://physics.princeton.edu/pulsar/k1jt/wsjitx-doc/wsjitx-main-2.0.0.html>

FLDigi

-> <http://www.w1hkj.com/FLdigiHelp/>