

Installing Quisk for Ubuntu Linux Operating a Software-Defined Radio

Quisk is a software-defined radio application program with versions for Windows and Linux. The following example shows how to install and configure Quisk on an Ubuntu Linux system. Installation of Quisk on Windows is described in a separate article (<https://dog-asparagus-b2eb.squarespace.com/installing-quisk-for-windows/>).

What Is In This Article?

- Quisk Software
- Other Required Software

- Compiling Quisk
- Configuring Soundcards
- Customizing the Control Panel
 - The USB Connection
 - Tuning the Local Oscillator
- Example of a Configuration File

Quisk Software

Quisk is freeware by James Ahlstrom, N2ADR. A link to download the source distribution is on the Web site (<http://james.ahlstrom.name/quisk/>) along with installation/configuration instructions (<http://james.ahlstrom.name/quisk/docs.html>) and a help file (<http://james.ahlstrom.name/quisk/help.html>) for beginners. The information includes a list of required software packages, basic instructions for compiling Quisk, and a guide for configuring the program.

The documentation is written with novice Linux users in mind.

For this example, I installed Quisk 3.6.6 on an Ubuntu 12.10 system and tested it with a SoftRock RX Ensemble 2 radio.

Other Required Software

The required software is in the Ubuntu Software Center. You will need Python, which is not included in the Ubuntu 12.10 desktop image. The current versions of the required packages are python2.7, python2.7-dev, python-wxgtk2.8, fftw3-dev, libasound2-dev, and portaudio19-dev.

The Pyserial package mentioned in the general instructions is not required for this example.

Compiling Quisk

After downloading Quisk follow N2ADR's instructions. (Substitute the current version number):

```
mkdir ~/SDR/
```

```
mv ~/Downloads/quisk-3.6.6.tar.gz ~/SDR/quisk-3.6.6.tar.gz
```

```
cd ~/SDR/
```

```
gunzip quisk-3.6.6.tar.gz
```

```
tar xf quisk-3.6.6.tar
```

```
mv quisk-3.6.6 quisk
```

```
cd ~/SDR/quisk
```

```
make
```

```
make install
```

Configuring Quisk

Basic configuration files for some radios are provided in the Quisk directory, ~/SDR/quisk. The files in the SoftRock folder are

(1) `conf_fixed.py` for models with a fixed-frequency local oscillator,

(2) `conf_rx_ensemble2.py`, and

(3) `conf_rx_tx_ensemble.py`.

To set up a personal configuration file for an RX Ensemble2:

```
cp ~/SDR/quisk/softrock/conf_rx_ensemble2.py  
~/quisk_conf.py
```

The period before quisk_conf.py is required. The file demonstrates how to configure sound cards:

```
gedit ~/.quisk_conf.py
```

More information about sound cards is in the installation instructions (<http://james.ahlstrom.name/quisk/docs.html>) and the default settings file (`~/SDR/quisk/quisk_conf_defaults.py`).

A stereo sound card with a line-in jack is required for the I and Q audio signals from the radio. You can use an external stereo USB sound card with a laptop. Desktop PCs usually have a stereo line-in jack for an internal sound card you can use.

To find the Linux names of your sound cards, run Quisk:

```
cd ~/SDR/quisk/
```

```
python quisk.py
```

The terminal will fill with error messages because Quisk is not configured yet. When the console opens, click on the Config button in the bottom row, and select the Sound tab.

A listing of my configuration file is at the end of this article.

Customizing The Control Panel

You can change the colors and font sizes in the control panel for better visibility. The default settings are in the `quisk_conf_defaults.py` file along with an example of an alternate, dark color scheme by KB8RWQ.

The article (<https://dog-asparagus-b2eb.squarespace.com/installing-quisk-for-windows/>) on installing Quisk for Windows shows the Quisk 3.6.11 **user interface** (which is the same for Linux) and describes how to customize the colors and font sizes.

USB Connection

1. The BrailleTTY background process in some versions of Ubuntu prevents other programs from using a USB serial port. To remove this utility:

sudo apt-get remove brltty

Reboot the computer.

2. The Quisk installation instructions explain how to add a rule in udev that gives you privileges to write to USB. Create the text file with administrative ownership.

sudo gedit

Enter the following text on a single line, typing no extra spaces:

SUBSYSTEM=="usb",ATTR{idVendor}=="16c0",

ATTR{idProduct}=="05dc",GROUP="dialout"

Click Save and enter (1) a file name like 99-softrock.rules and (2) the destination directory /etc/udev/rules.d

To join the dialout group and verify membership:

sudo usermod -a -G dialout [username without brackets]

groups [username without brackets]

Reboot the computer.

To run Quisk:

```
cd ~/SDR/quisk/
```

```
python quisk.py
```

Click the Config button in the bottom row of the console for a window confirming correct sound card and USB configuration.

Calibrating The SoftRock Local Oscillator

Before calibration of the local oscillator in my radio, the 14.0475 MHz signal from W1AW appears on the Quisk spectrum at 14.0596 MHz.

Current SoftRock radios use an Si570 chip for the variable frequency local oscillator. This chip uses an internal fixed-frequency crystal as a reference for synthesizing the local oscillator frequency. Before calibration, the Si570 is programmed to use a nominal crystal frequency of 114.285 MHz to calculate settings for the synthesizer. Recalibration is necessary because the actual crystal frequency varies from one Si570 device to another.

Calibrating an Si570 chip involves figuring out what the actual crystal frequency is and instructing the chip program to use it instead of the nominal frequency.

Here are three ways to calibrate the chip using a computer:

Method 1—Use Quisk and USBsoftrock. Ethan Blanton, KB8OJH, describes this method in his discussion (<http://kb8ojh.net/sdr/linuxsdr.html>) of Linux SDR. USBsoftrock, a free Linux utility by Andrew Nilsson, VK6JBL, can find the crystal frequency (Figure 1). USBsoftrock is in a Google code repository, and the link to it is in Ethan's article. Install libncurses5-dev and libusb-dev from the Ubuntu Software center before compiling it.

Figure 1. USBsoftrock calibration

```
pd@laptop:~$ usbsoftrock calibrate
Version      : 15.12
fXTALL = 114.187294
pd@laptop:~$
```

Then add these lines to the configuration file (gedit `~/.quisk_conf.py`), substituting the crystal frequency of your radio:

si570_direct_control = True

si570_xtal_frequency = 114187294

Method 2—Use Quisk and a known signal. Tune Quisk to a signal with a known frequency. Record the "real" frequency and the "tuned" frequency (where it appears on the spectrum), and calculate the crystal frequency with this formula:

$$\text{Xtal_Freq} = 114.285 \times (\text{Real_Freq} / \text{Tuned_Freq}) \text{ MHz}$$

For my radio:

$$\begin{aligned} \text{Xtal_Freq} &= 114.285 \times (14.0475 / 14.0596) \text{ MHz} \\ &= 114.187 \text{ MHz} \end{aligned}$$

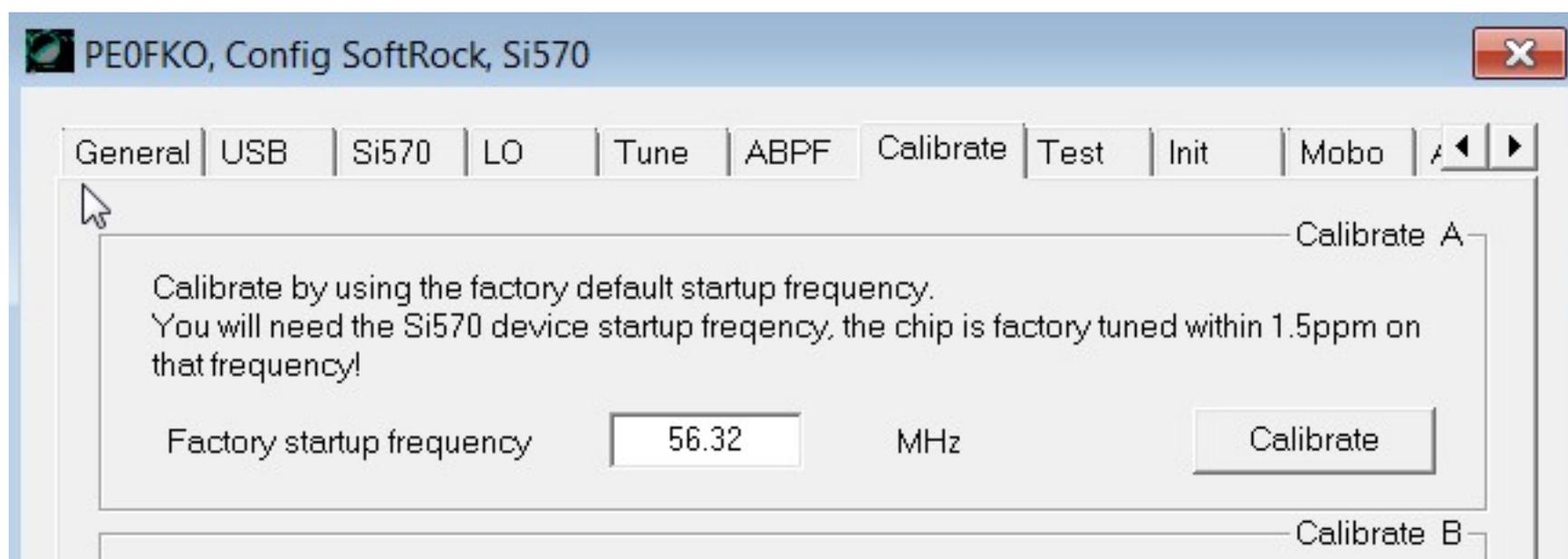
Enter this frequency in Hz in the configuration file as in Method 1.

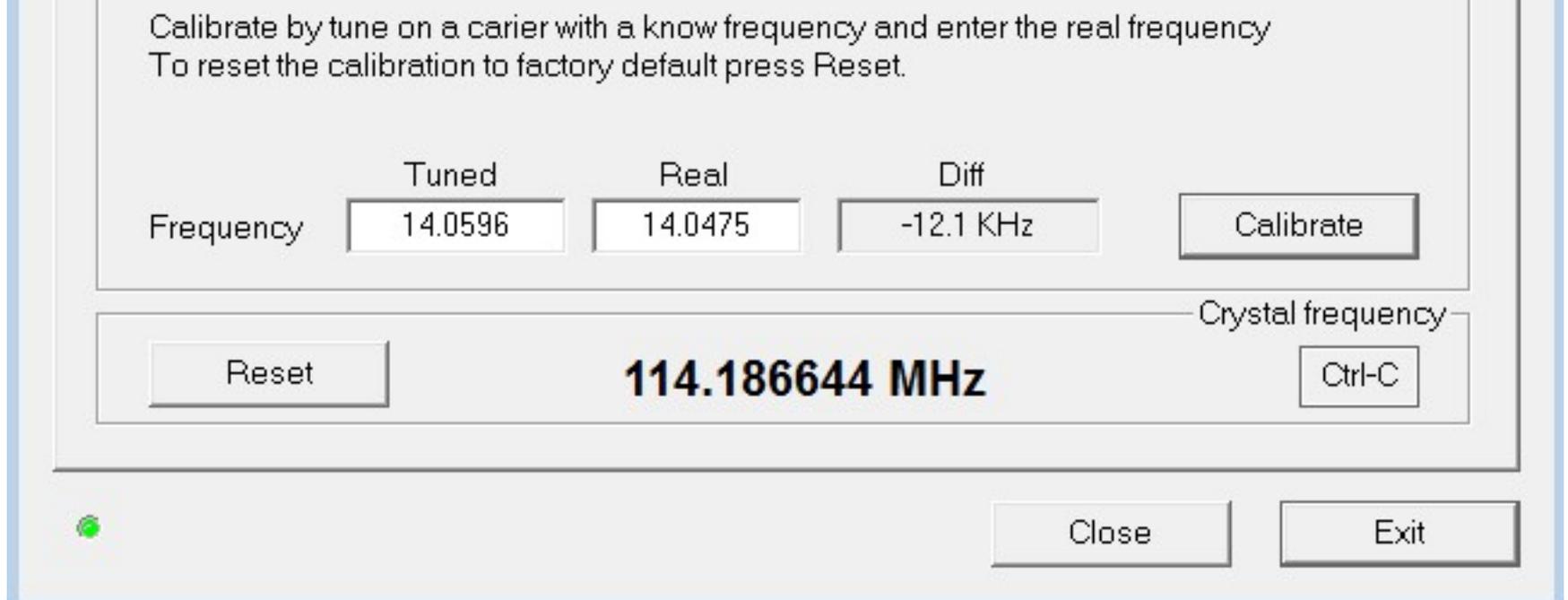
Method 3—Use the Windows CFGSR configuration utility. Tune Quisk to a signal with a known frequency, and as in Method 2 record the real and tuned frequencies.

Next, download and install on a Windows computer the USB driver (<http://pe0fko.nl/SR-V9-Si570/#install>) and the Si570 USB Configuration Utility (<http://pe0fko.nl/CFGSR>). Connect the computer to the USB port on the radio, and start CFGSR. A small green dot in the lower left corner of the opening screen indicates a functioning USB connection. (The article (<https://dog-asparagus-b2eb.squarespace.com/installing-quisk-for-windows/>) on Quisk for Windows describes what to do if Windows does not install the USB driver.)

Select the Calibration tab, and click Reset if you have previously calibrated it using CFGSR. Enter the tuned and real frequencies in the Calibration B section. Clicking the Calibrate button displays the calculated crystal frequency (Figure 2) and stores it in the chip.

Figure 2. CFGSR calibration.





At this point the chip will remain calibrated when the power is off and if a different SDR program, such as HDSDR, is used.

A Configuration File Example

Here is an example of a configuration file for a SoftRock RX Ensemble2 receiver, with thanks to KB8OJH for posting his file (<http://kb8ojh.net/sdr/linuxsdr.html>):

```
from softrock import hardware_usb as quisk_hardware
```

```
usb_vendor_id = 0x16c0
```

```
usb_product_id = 0x05dc
```

```
softrock_model = "RxEnsemble2"
```

```
sample_rate = 48000
```

```
name_of_sound_capt = "hw:1"
```

```
name_of_sound_play = "hw:0"
```

```
# Swap the I and Q channels for correct sidebands
```

```
channel_i = 1
```

```
channel_q = 0
```

Acknowledgment

Thanks to Ethan, KB8OJH, for answering my questions.

Author Information

Peter DeNeef, AE7PD, is an Extra Class amateur radio operator in the U.S. This Web site has no ads or conflicts of interest.

rev. 1/21/216

Magnification Plug-Ins for Ubuntu (<https://dog-asparagus-b2eb.squarespace.com/compiz-magnification-in-ubuntu/>)

Linux Programs on HamRadioAndVision (<https://dog-asparagus-b2eb.squarespace.com/linux-page-hamradioandvision/>)

HamRadioAndVision Table of Contents (<https://dog-asparagus-b2eb.squarespace.com/home/>)

Powered by Squarespace (http://www.squarespace.com?channel=word_of_mouth&subchannel=customer&source=footer&campaign=4fd1028ee4b02be53c65dfb3)