Installation

Quisk is free open source software. It is hosted on the PyPi repository https://pypi.org. It can be installed using the standard Python setup tools. See specific directions below. Python is available as a series 2 release or a series 3 release, and there are 32-bit and 64-bit versions of each. Quisk uses either 32-bit or 64-bit Python 2.7. A Quisk compatible with Python 3 is under development.

Windows Initial Installation

First install Python 2.7 on your computer. If it is already installed, check the version. If it is much older than 2.7.15, please install a newer version. Visit http://www.python.org and download the Windows installer for Python 2.7. To get there from python.org, click on "Downloads", then "Windows", then "Latest Python 2 Release". Under "Files" click on "Windows x86 MSI installer" for the 32-bit version, or "Windows x86-64 MSI installer" for the 64-bit version. Select the 32-bit version unless you know you are using software that requires 64-bit. This is true even if you have 64-bit Windows. Quisk can use either 32-bit or 64-bit. Click on the downloaded MSI file to install Python. When installing, there is an option to "Add python.exe to Path". It is best to select this option so that you can start python by just typing "python" instead of the whole path from the install directory C:\python27\python.exe.

Next open a Command Prompt. This is located under "Windows System" on Windows 10; or use the search bar. Enter "python --version" and then "pip --version" to make sure that these programs are installed, and what version you have. If "python" is not found, it might not be on your Path. Try "C:\python27\python.exe --version". For pip, it is "C:\Python27\Scripts\pip.exe --version".

First upgrade some Python modules to the newest version, then install Quisk. Enter these commands. The upgrade of pip gave error messages on my machine, but it worked anyway.

```
pip install --upgrade pip

pip install --upgrade setuptools

pip install --upgrade wxPython

pip install --upgrade pyserial

pip install --upgrade quisk
```

You should then be able to start Quisk with the command "quisk" or "C:\python27\Scripts\quisk". To create a Quisk shortcut on your desktop, right-click an empty space and select "New" and "Shortcut". Use "C:\Python27\Scripts\quisk.exe" as the command and "Quisk" as the name. Create another shortcut for quisk_vna if you plan to use it. To install a newer version of Quisk, just use "pip install --upgrade quisk". You could check for newer versions of the other modules twice a year if you want.

To get started you must tell Quisk what kind of radio hardware you have. Press the Config button and select Radios. Then set your sound devices for that radio; the device for the radio speakers, the microphone and so forth. All configuration is (mostly) from the Config button. Ignore old directions and don't bother with a config file.

Windows Quisk Upgrade

To upgrade to a newer version of Quisk, use pip. Remember that if Python is not on your Path, you will need to type out the whole path C:\Python27\Scripts\pip.exe install --upgrade quisk.

```
pip install --upgrade quisk
```

Windows Uninstall Quisk

```
pip uninstall quisk
```

Linux Initial Installation

Python 2.7 is already installed on your computer. Either the 32-bit or 64-bit version will work. Enter "python" in a terminal and see what version you have. Install these packages. Some of these can be installed from Python using "pip", but it is better to use your package manager because pip and the package manager do not coordinate. If newer versions of these packages become available, Linux will notify you.

```
sudo apt-get install python-wxgtk3.0
sudo apt-get install libfftw3-dev
sudo apt-get install libasound2-dev
sudo apt-get install portaudio19-dev
sudo apt-get install libpulse-dev
sudo apt-get install libpython2.7-dev
sudo apt-get install python-usb
sudo apt-get install python-setuptools
sudo apt-get install python-pip
```

Then install quisk:

```
sudo -H pip install --upgrade quisk
```

This will install the Python files and all other files except for the C source. This is the easiest method if you do not want to work in C. To run Quisk, enter "quisk" in a terminal. You can create a panel launcher on your desktop with the command to run Quisk, /usr/local/bin/quisk.

To get started you must tell Quisk what kind of radio hardware you have. Press the Config button and select Radios. Then set your sound devices for that radio; the device for the radio speakers, the microphone and so forth. All configuration is (mostly) from the Config button. Ignore old directions and don't bother with a config file.

To edit the Quisk Python files you need to know where they are. To find out, import quisk and print it:

```
jim@IntelNUC:~$ python
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import quisk
>>> quisk
 module 'quisk' from '/usr/local/lib/python2.7/dist-packages/quisk/__init__.pyc'
>>>
```

Linux Quisk Upgrade

To upgrade to a newer version of Quisk, use pip:

```
sudo -H pip install --upgrade quisk
```

Linux Uninstall Quisk

```
sudo pip uninstall quisk
```

Linux Quisk Source Files

The previous method using pip is simple, but will not install the C source files. If you want those proceed as follows. Go to https://pypi.org/project/quisk and under "Navigation" select "Download files". Download the quisk-x-x-xx.tar.gz file. This is the source file. The ".whl" files are for Windows. Change directory to where the file was downloaded, probably "Downloads", and uncompress and untar it.

```
cd ~/Downloads
tar zxf quisk-x.x.xx.tar.gz
```

Change directories to the Quisk directory:

```
cd quisk-x.x.xx
```

Now compile Quisk with the "make" command:

```
make
```

If "make" fails, you probably have missing packages or missing "-dev" packages. Try to figure out what is missing from the error messages. Now you can run quisk with the command:

```
python quisk.py
```

At this point you have a choice of where to install Quisk. You could just rename the quisk-x.x.xx directory to a more convenient name, like "quisk" in your home directory, and run Quisk from there with the command "python quisk.py". This is convenient if you want to alter the Quisk code. Or you could install Quisk into the Python system, and just enter "quisk" to run Quisk:

```
python setup.py build
sudo python setup.py install
```

Quisk Files

These are the Quisk files in the distribution:

quisk.py is the main program and is written in the Python language. Python is a powerful but easy to learn language, and I hope you have fun changing Quisk to make it do what you want. Python is also useful for general electronics calculations such as complex arithmetic. See www.python.org. Quisk.py uses the wxPython Python package to make the screen interface.
help.html is the help file for quisk. Press the "Help" button.

_quisk.so is the _quisk extension module for Linux, and _quisk.pyd is the extension module DLL used by Windows.
sdriq.so is the extension module needed for the SDR-IQ. It needs _quisk.so to be available when it starts.
makefile is the makefile, and you must run "make" to create a new _quisk.so unless you use a Python installer that creates _quisk.so itself.
setup.py is used by makefile and the Python installers.
quisk_conf_defaults.py is the basic configuration file imported into all other configuration files. Read it (but don't change it) to see what you can change in your own quisk_conf.py.
quisk_conf_*.py are various Quisk configuration files. Copy one of them to your own .quisk_conf.py and edit that file. I may publish new model files in the future, and you don't want your changes to be overwritten.
quisk_hardware_*.py are various quisk hardware control programs. If you have custom hardware, import one of these files into your quisk_conf.py. Or copy one of them to your own quisk_hardware.py, edit that file, and import it in .quisk_conf.py.
quisk.c, quisk.h are the files for the _quisk extension module used by quisk.py. The other C-language files are linked with these to make _quisk.so and _quisk.pyd.
sound.c is the general purpose sound code for all sources.
sound_portaudio.c is the sound card access code for PortAudio.
sound_pulseaudio.c is the sound card access code for PulseAudio.
sound_alsa.c is the sound card access code for the ALSA drivers.
sound_directx.c is the sound card access code for DirectX.
is_key_down.c is the hardware key checker for the PC. I use Ethernet to send the key status, but there is code for the parallel port and dummy code too.
sdriq.c, sdriq.h are the files that make sdriq.so and support the SDR-IQ.
microphone.c reads the microphone audio and sends it to your hardware using Ethernet. Change it for other sound access.

docs.html is Quisk documentation. Look for other *.html and *.txt too.
portaudio.py is a utility program. Run it to list your PortAudio devices. It is not used by the Quisk program.

Configuration
The Quisk "Config" button brings up a number of status and configuration screens. When starting with Quisk, you need to create a named "radio" and then set the configuration for that radio. You can have multiple radios to support several hardwares, or different settings for the same hardware. Use the "Help with Radios" button for documentation. A special radio named "ConfigFileRadio" is always available. It doesn't use the settings screens and takes all its settings from the configuration file. When Quisk starts, it reads an initial configuration from the Quisk file quisk_conf_defaults.py. Then it reads your configuration file if you have one. Then it reads settings from the configuration screens, which are stored in the file quisk_settings.json. You can set almost everything with the screens, but you can have a configuration file if you want. For Linux, the default configuration file name is ".quisk_conf.py" in your home directory; that is, "~/.quisk_conf.py". For Windows, the default configuration file name is quisk_conf.py in your My Documents folder.
Linux config file: ~/.quisk_conf.py
Windows config file: My Documents/quisk_conf.py
To help you get started, there are several configuration files included, such as quisk_conf_model.py for sound card, quisk_conf_sdriq.py for SDR-IQ, and quisk_conf_fixed.py for fixed VFO such as SoftRock. Do not change any of the quisk_conf_*.py files. Instead copy one of these files (but NOT

quisk_conf_defaults.py) to your own config file. Newer versions of Quisk will not overwrite your personal config file.

The file quisk_conf_defaults.py contains all Quisk's parameters, and you can read it to see what can be changed.

If you are controlling custom hardware, you will need to specify a hardware file in quisk_conf.py. The default is quisk_hardware_model.py. Look at the other quisk_hardware_*.py files. For example, quisk_hardware_fixed.py is for crystal controlled SoftRock. To use that hardware file, change your quisk_conf.py to include:

```
import quisk_hardware_fixed as quisk_hardware
```

There are comments in quisk_conf_model.py showing this change. If none of the hardware files do exactly what you want, copy one of them to your own quisk_hardware.py, edit that file, and include this line in quisk_conf.py:

```
import quisk_hardware
```

Newer versions of Quisk will not overwrite your quisk_hardware.py. Your hardware file enables you to customize Quisk without changing the Quisk program files.