libradiohat - experimental version and tools

Added 1-March-2022

introducing *libradiohat*

This is a refactored set of libraries for accessing the RadioHat 1.0 hardware. It has many clean-ups and improvements and may now be built as a dynamic library.

There's also a primitive make file included that builds the library and test program but does not install it. For now, progams that use it must access things by absolute locations, or you must move the dynamic library to somewhere on the load library path.

This folder also includes a reweritten version of pit/rans called "transceiver" that uses the dynamic library for all radiohat access.

Two other experimental programs are included that expect for now to find the libradiohat.so in ~/dvl/libradiohat, but they can easily be modified to change this:

Other Tools

transceiver.grc / *transceiver.py* is an experimental gnuradio flow graph that attempts to put all the things needed to experiment with and test radiohat in one spot. It also includes some experimental PTT and VFO access from the flow graph. Unfortunately the ways needed to do this at the moment add a lot of overhead, and the program is a little unstable when run in the GRC environment. It seems to work OK by running the .py file, however.

Be aware, however, that the 1ms polling rate GRC uses for exporting variables as messages adds a lot of overhead and as a result this flow graph is occasional getting out of sync and losing alternate sideband rejection - usually if a lot of VFO tuning around is done at high speed. It's still very useful for testing.

Experimental *quisk* interface

QUISK is a folder containg a hardware configuration file for Quisk. There are two copies present under different names. It's generally operating OK with the QUISK version in the bullseye apt repository, but it's still very experimental. QUISK is very hard to configure and get working, but if you want to play with it, you may have some luck by moving the "radiohat" folder inside QUISK to the quisk configuration directory

and setting quisk up as a radio of type “Softrock fixed” with one of these config files chosen.

All these programs are intended for use with the default Alsa device and I suggest using a usb plug-in dongle. This simplifies understanding them and eliminates problems with t/r switching the code hardware audio paths. You can also use the built-in py headset sound or the HDMI sound, since they both include input devices for receiver and tone testing.

We’ll return to using the built-in headset jack and digital modes soon, but for now these simplifications are speeding development a great deal and ensuring a solid foundation for the work to come.