Hamming with Linux

Why use Linux?

Out of Date Software

- Your Linux Distro’s package manager makes it extremely easy to stay up to date with the latest software

PipeWire

- This is the primary reason I would recommend Linux
- Easy to handle multiple soundcards and set priority of soundcards
- Easy to run multiple radios at the same time

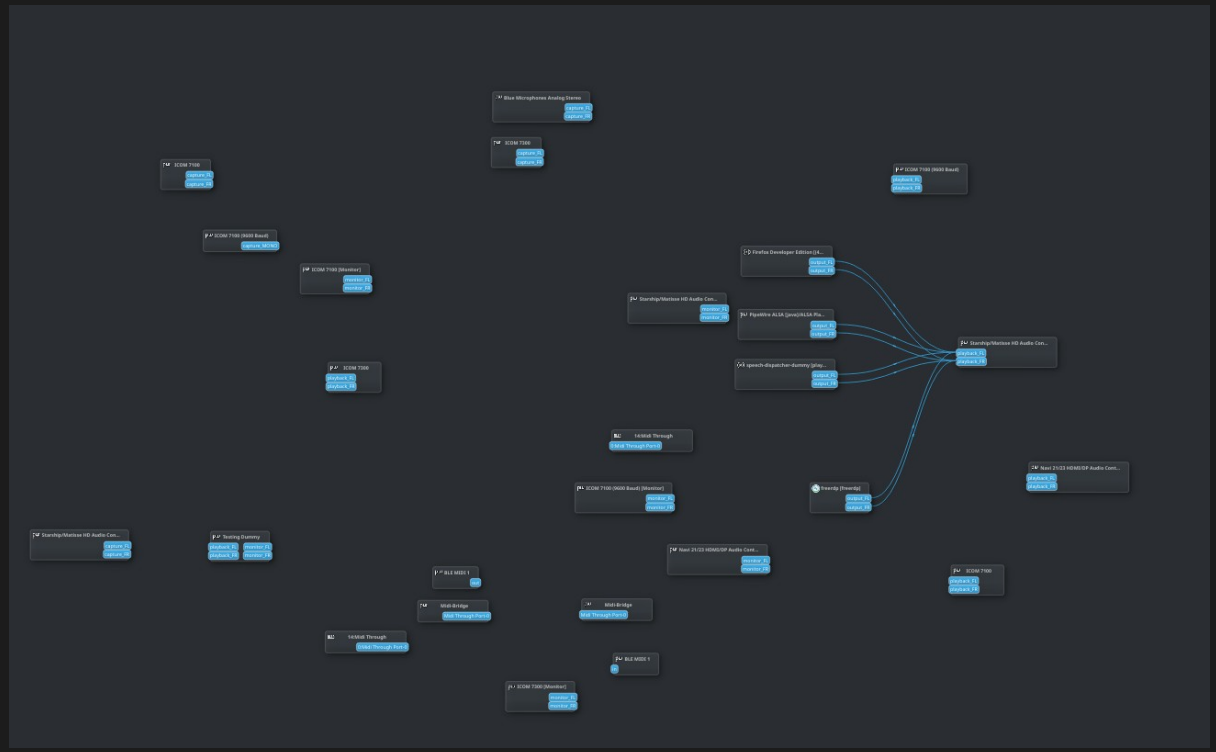
Embrace Open Source

- Ham Radio is about tinkering and experimenting, embrace an operating system that does the same
- If Linux is more embraced, we can expect more ham software to follow
  - Open Source Development will generally occur at a quicker rate
  - Software development have been trending towards open source

What do I do now?

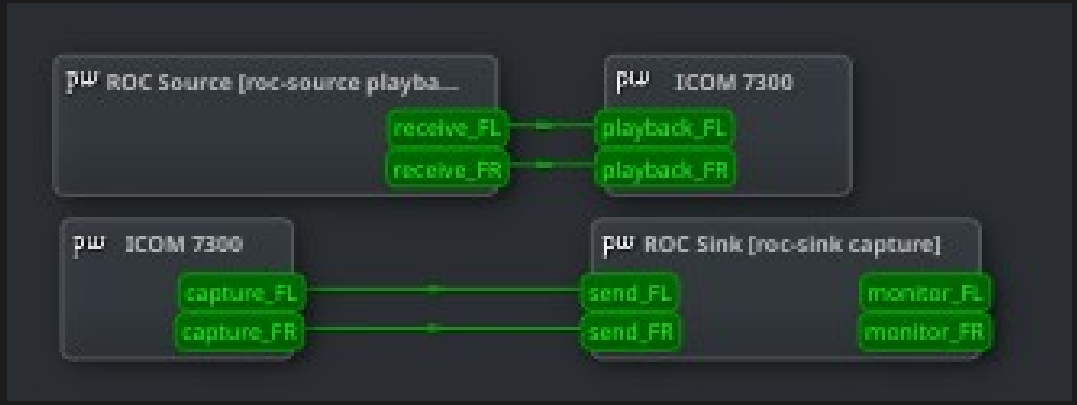
Linux Audio (Pipewire)

Qpwgraph



Linux Audio (Pipewire)

ROC Source and Sink



Linux Audio (Pipewire)

ROC Source & Sink

- Send and receive audio to and from network devices
  - Send ICOM 7300 audio to networked computer
  - ICOM 7300 can receive audio from networked computer
  - Hamlib handles PTT and Frequency Control
  - The ICOM 7300 is essentially operates as a poor man’s Flex Radio
  - About 600 KiB/s of bandwidth is necessary

Microwave Purpose?

With a microwave setup, Raspberry Pi, and Ethernet, and a computer at the other end, that gives you about 328 feet to work with, more if fiber is used, **with no loss**

Linux Audio (Pipewire)

Ideas?

- Use computer’s microphone as hand microphone and write a script that when the spacebar is held down, the radio is transmitting
- VHF to HF or HF to VHF Repeater?
- Multiple applications listening to the same signal
- Pipe Zoom Audio out to radio
- Other ideas?

Linux Audio (Pipewire) (Wireplumber)

Pipewire's Audio Manager

```
logan@Apollo ~> cat ~/.config/wireplumber/wireplumber.conf.d/53-icom7300.conf
monitor.alsa.rules = [
  {
    matches = [
      {
        alsa.long_card_name = "Burr-Brown from TI USB Audio CODEC at usb-0000:05:00.0-2.3.4, full speed"

      }
    ]
    actions = {
      update-props = {
        node.description = "ICOM 7300"
        node.name = "ICOM 7300"
        priority.session = 100
        priority.driver = 100
        node.pause-on-idle = false
        session.suspend-timeout-seconds = 0
        node.exclusive = true

      }
    }
  }
]
```

Hamlib (Rig Control)

Available on Windows, macOS

```
rigctl -m 3070 -r /dev/serial/by-id/usb-Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_IC-7100_22002393_A-if00-port0 -t 4536 -s 19200 -vvv
```

ICOM 7100 can be accessed on localhost at port 4536

Can be put into a service to be available on boot

Hamlib (Rig Control)

```
logan@Apollo ~/.config/systemd/user> cat hamradio-rigctld-icom-7100.service
[Unit]
Description=rigctld Ham Radio Rig Controller for ICOM 7100

[Service]
Type=simple
ExecStart=/usr/bin/rigctld -m 3070 -r /dev/serial/by-id/usb-Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_IC-7100_22002393_A-if00-port0 -t 4535 -s 19200
KillMode=process
TimeoutStopSec=0
```

Hamlib (Advanced Rig Control)

```
logan@Apollo ~/.config/systemd/user> telnet 127.0.0.1 4532
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
f
5357000
F 7074000
RPRT 0
F 1407400
RPRT 0
F 14074000
RPRT 0
```

```
logan@Apollo ~/.config/systemd/user> echo "f" | nc -w 1 127.0.0.1 4532
14074000
logan@Apollo ~/.config/systemd/user>
```

Hamlib (Rig Control)

Radio Operators are always having issues with getting the rig connected to the computer, especially in the field

Configure this correctly once and more than likely, the computer side will never be an issue

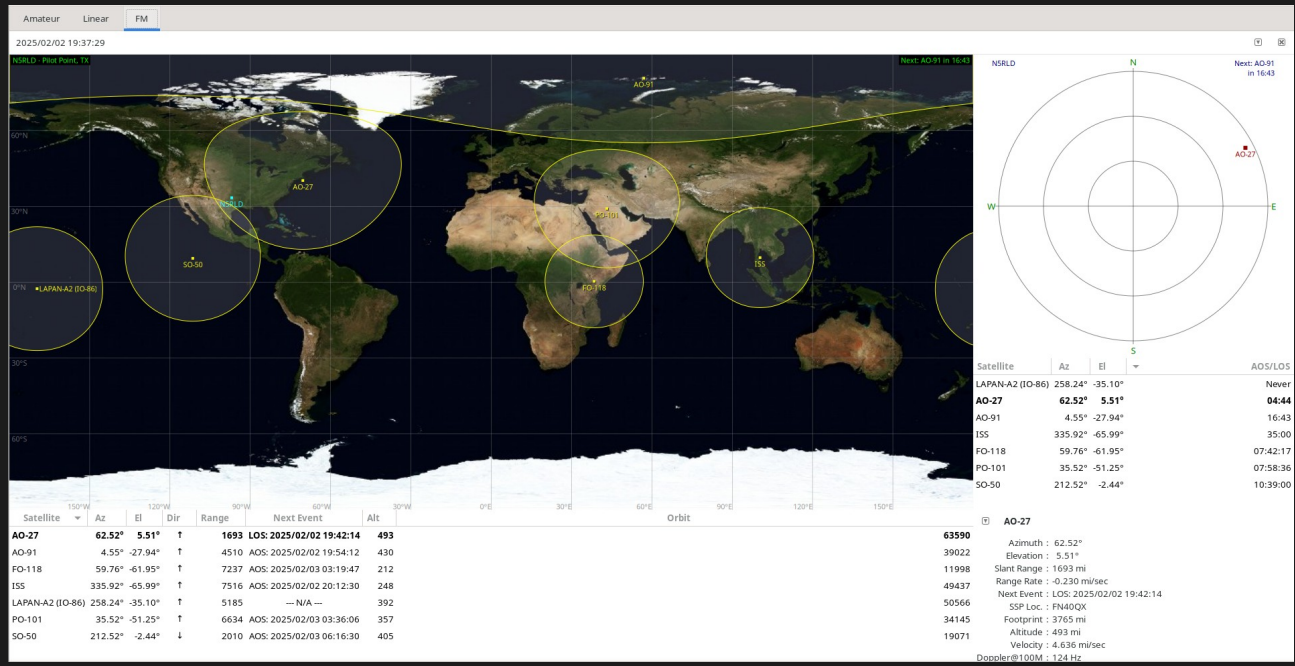
Majority of other radio software supports hamlib

- Doesn’t apply to N1MM

Satellite (Gpredict)

Can control rotor with rotctld

Can account for doppler



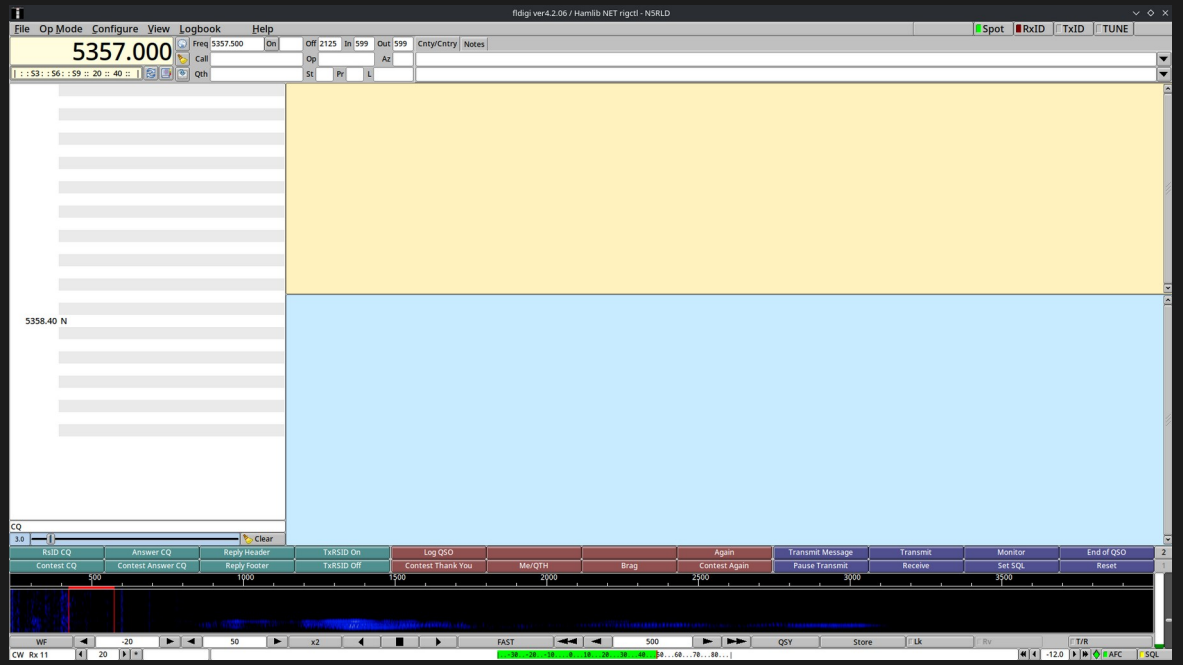
Digital (Fldigi)

Support for Pipewire:

Limited

PULSE_SINK=

PULSE_SOURCE=



Digital (Dire Wolf)

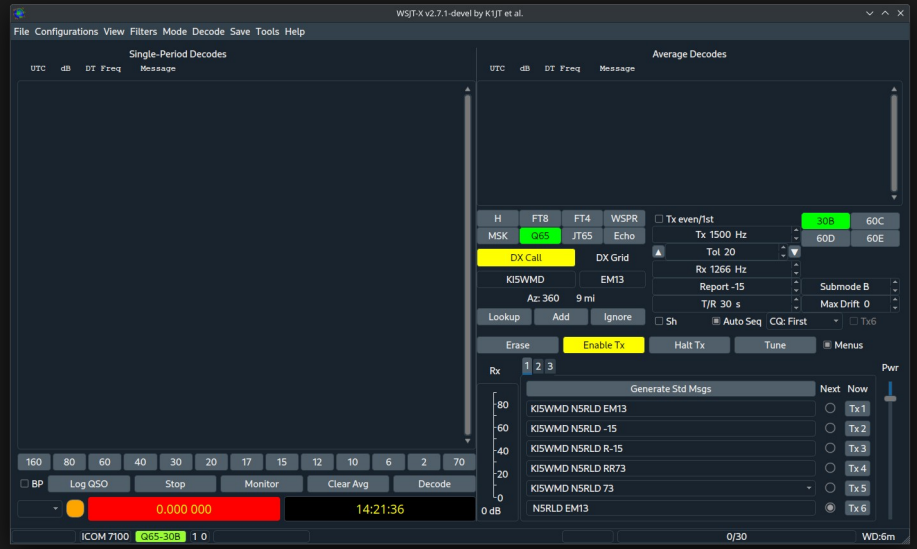
- Inadequate pipewire support
- To use with pipewire:

PULSE_SOURCE= PULSE_SINK= direwolf pulse pulse

```
Dire Wolf DEVELOPMENT version 1.8 A (Jan  9 2024)
Includes optional support for:  gpsd hamlib cm108-ptt dns-sd

Reading config file direwolf.conf
Trying tocalls.yaml
Trying data/tocalls.yaml
Trying ../data/tocalls.yaml
Trying /usr/local/share/direwolf/tocalls.yaml
Trying /usr/share/direwolf/tocalls.yaml
Audio device for both receive and transmit: pulse  (channel 0)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, A+, 48000 sample rate.

Note: PTT not configured for channel 0. (OK if using VOX.)
When using VOX, ensure that it adds very little delay (e.g. 10-20) milliseconds
between the time that transmit audio ends and PTT is deactivated.
For example, if using a SignaLink USB, turn the DLY control all the
way counter clockwise.

Using VOX built in to the radio is a VERY BAD idea.  This is intended
for voice operation, with gaps in the sound, and typically has a delay of about a
half second between the time the audio stops and the transmitter is turned off.
When using APRS your transmiter will be sending a quiet carrier for
about a half second after your packet ends.  This may interfere with the
the next station to transmit.  This is being inconsiderate.

If you are trying to use VOX with connected mode packet, expect
frustration and disappointment.  Connected mode involves rapid responses
which you will probably miss because your transmitter is still on when
the response is being transmitted.

Read the User Guide 'Transmit Timing' section for more details.

Ready to accept AGW client application 0 on port 8000 ...
Ready to accept KISS TCP client application 0 on port 8001 ...
DNS-SD: Avahi: Announcing KISS TCP on port 8001 as 'Dire Wolf on Apollo'
DNS-SD: Avahi: Service 'Dire Wolf on Apollo' successfully registered.
```

Digital (WSJT-X)

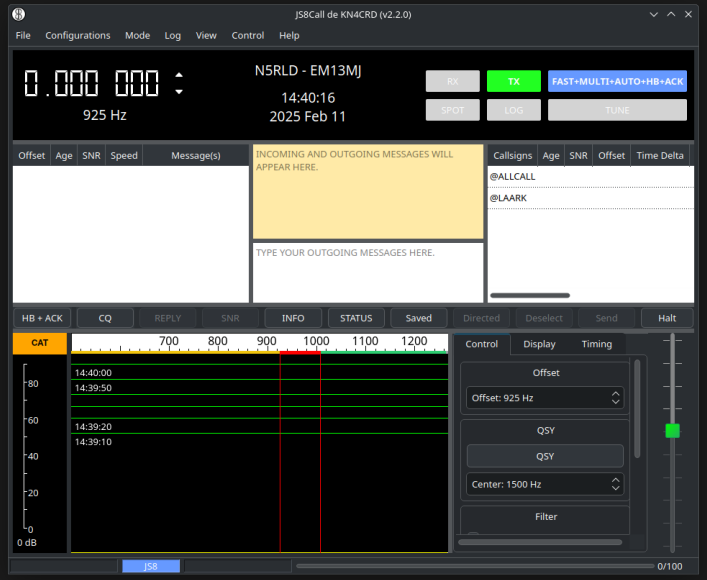
Native support for Pipewire



Digital (JS8Call)

Native support for Pipewire

Created as a result of WSJT-X being open source

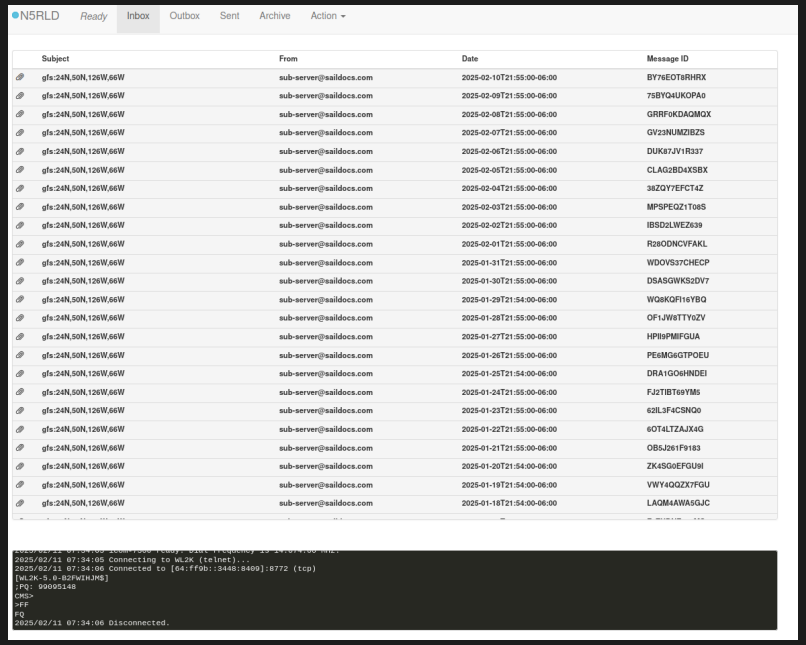


E-mail (Pat)

Not as good as Winlink

But, Winlink is not open source

Neither is VARA, for that matter



Advanced SDR (rtl_433)

What’s transmitting in the ISM band around you?

915 MHz, 433.92 MHz, 315 MHz

```
logan@Apollo ~ [3]> rtl_433 -d 0
rtl_433 version 24.10-53-g19a1dd60 branch master at 202502040951 inputs file rtl_tcp RTL-SDR SoapySDR with TLS
Found Rafael Micro R820T tuner
 SDR  Using device 0: Nooelec, NESDR SMArt v5, SN: 45822103, "Generic RTL2832U OEM"
Exact sample rate is: 250000.000414 Hz
[R82XX] PLL not locked!

time      : 2025-02-11 08:43:20
model     : LaCrosse-TX141Bv3                   Sensor ID : a6
Channel   : 1           Battery   : 1           Temperature: 21.70 C      Test?     : Yes

time      : 2025-02-11 08:43:20
model     : LaCrosse-TX141Bv3                   Sensor ID : a6
Channel   : 1           Battery   : 1           Temperature: 21.70 C      Test?     : Yes
bitbuffer_add_bit: Warning: row count limit (50 rows) reached
bitbuffer_add_bit: Warning: row count limit (50 rows) reached
bitbuffer_add_bit: Warning: row count limit (50 rows) reached
bitbuffer_add_bit: Warning: row count limit (50 rows) reached

time      : 2025-02-11 08:43:35
model     : Cotech-367959 ID        : 31
Battery   : 1           Temperature: 40.8 F       Humidity  : 94 %          Rain      : 81.0 mm       Wind direction: 263       Wind      : 1.2 m/s       Gust      : 1.7 m/s       Light Intensity: 8981 lux UV Index  : 0.9
Integrity : CRC

time      : 2025-02-11 08:43:49
model     : LaCrosse-TX141Bv3                   Sensor ID : a6
Channel   : 1           Battery   : 1           Temperature: 22.10 C      Test?     : No

time      : 2025-02-11 08:43:49
model     : LaCrosse-TX141Bv3                   Sensor ID : a6
Channel   : 1           Battery   : 1           Temperature: 22.10 C      Test?     : No

time      : 2025-02-11 08:43:52
model     : Cotech-367959 ID        : 31
Battery   : 1           Temperature: 40.8 F       Humidity  : 94 %          Rain      : 81.0 mm       Wind direction: 5         Wind      : 1.2 m/s       Gust      : 2.0 m/s       Light Intensity: 8810 lux UV Index  : 1.2
Integrity : CRC
```

Advanced SDR (GNU Radio)

Turn your HackRF into a SSB Transceiver

Used Pipewire to reconfigure the audio from the HackRF

Used Hamlib to configure PTT and frequency control

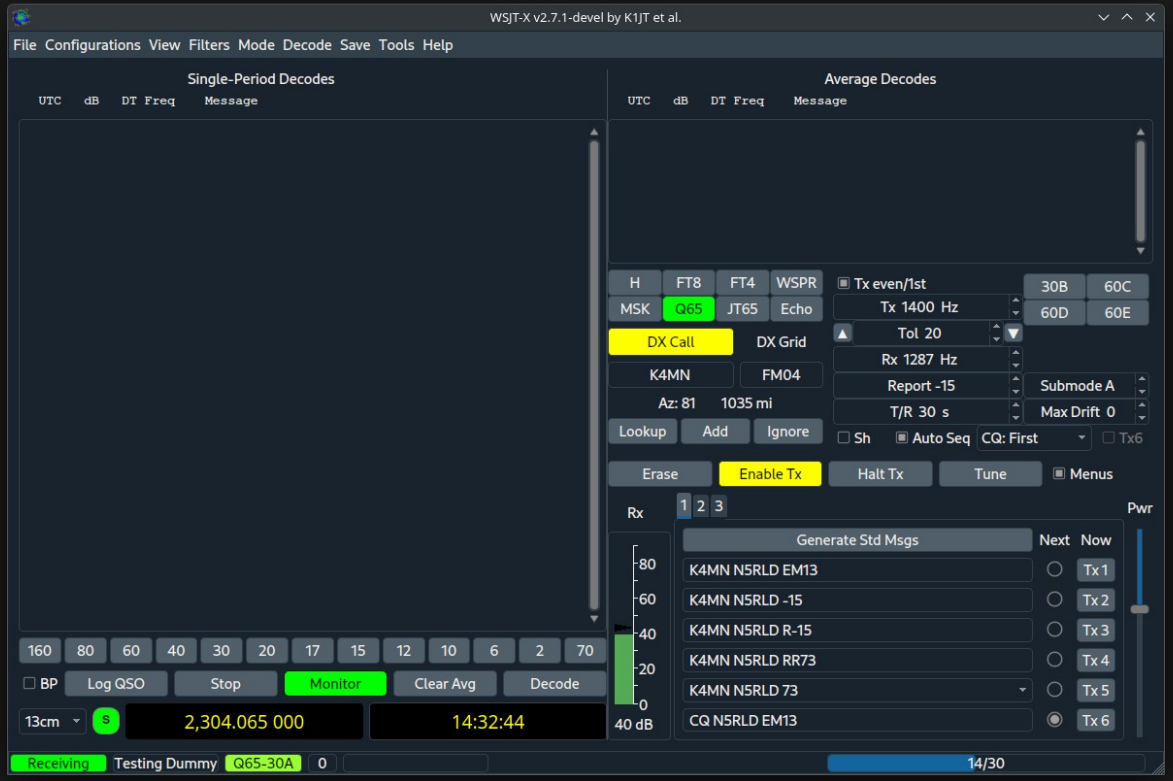
Capable of transmitting on HF and Microwave frequencies from 1 MHz to 6 GHz

- Need a filter and amp

Developed two flowgraphs to handle TX and RX

Wrote in Python with the help of Grok

Advanced SDR (GNU Radio)



Advanced SDR (GNU Radio)

Making Hamlib compatible:

- Spoof
  - \dump_state
  - \get_powerstat
  - CHKVFO 0
  - Split state
- Return Correctly
  - Transmit State
  - Current Frequency
- Can be modified to return real values