

sBitx – HF SDR Transceiver

sBitx V3

from HF Signals

user review by
KI4UJY

HF Signals was founded by Ashhar Farhan, VU2ESE

Company

HFSIGNALS Corporation
8,The Green,Suite 8679
Dover DE 19901
Contact: sales@hfsignals.com,
hfsignals@gmail.com,
Phone: 630-923-1728

sBitx: (some Configuration Assembly Required)



The sBitx-v3 measures 10"x 6" x 2" and weights about 4 pounds (2Kgs) Comes preloaded with all software and configurations – just set CALL SIGN, Grid Square and your good to go.. (IF only this were true..)
SOME ASSEMBLY REQUIRED , USE ONLY AS DIRECTED , SIDE EFFECTS INCLUDE , LOST TIME..

Basics:

- sBitx v3 – Board Only \$180 | sBitx v3 – Full Radio with RPI4/2GB \$399*
 - Open Source
 - 25 watt max
 - Raspberry Pi4 w/ integrated touchscreen
 - CW, FT8, Digital , Upper, Lower Side Band and Am – baked in
 - Integrated Ham Radio Software Suite
 - Web-based Remote Operation*
 - Add a second screen with the Pi HDMI port*
- *modification required for second screen – web based operation is great!
RealVNC remote operation is not bad either..

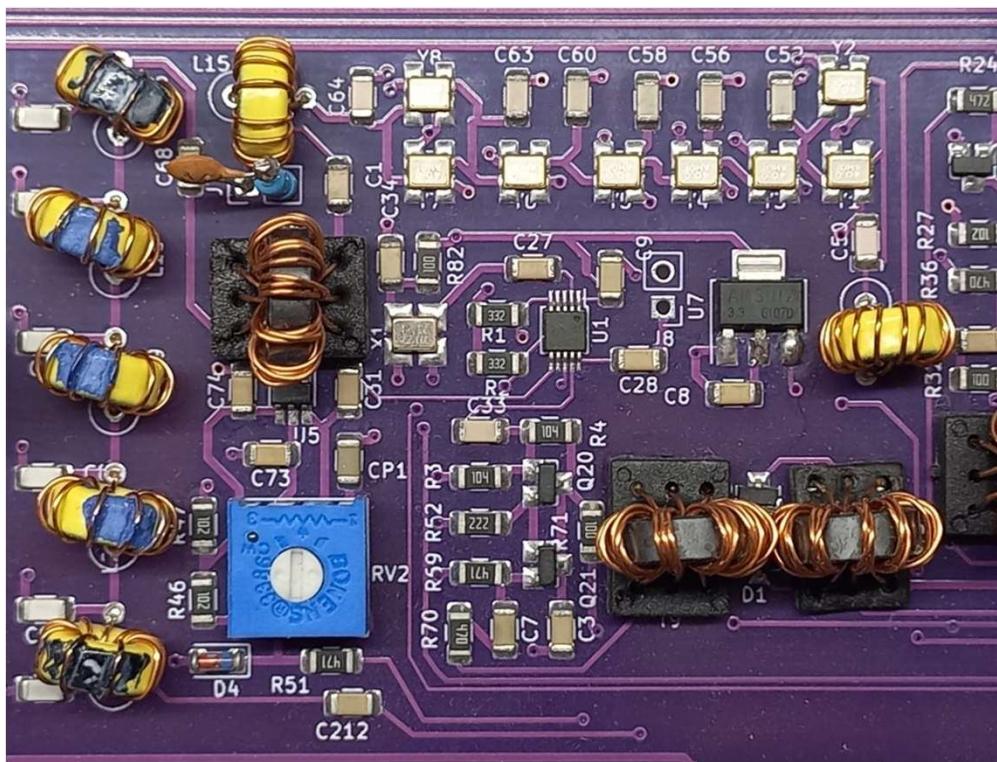
Frequencies

- Receive; 500 KHz to 30 MHz,
- Transmit: 3500-4000, 7000-7300, 10000-10150, 14000-14350, 18000-18200, 21000-21450, 24800-25000, 28000-29700 (KHz)
- Power 25 watts on 80M – 20M, 10 watts higher bands
- Two VFO's
- No internal tuner
- Based on 'Poor Man's SDR' WB2CBA?

Details

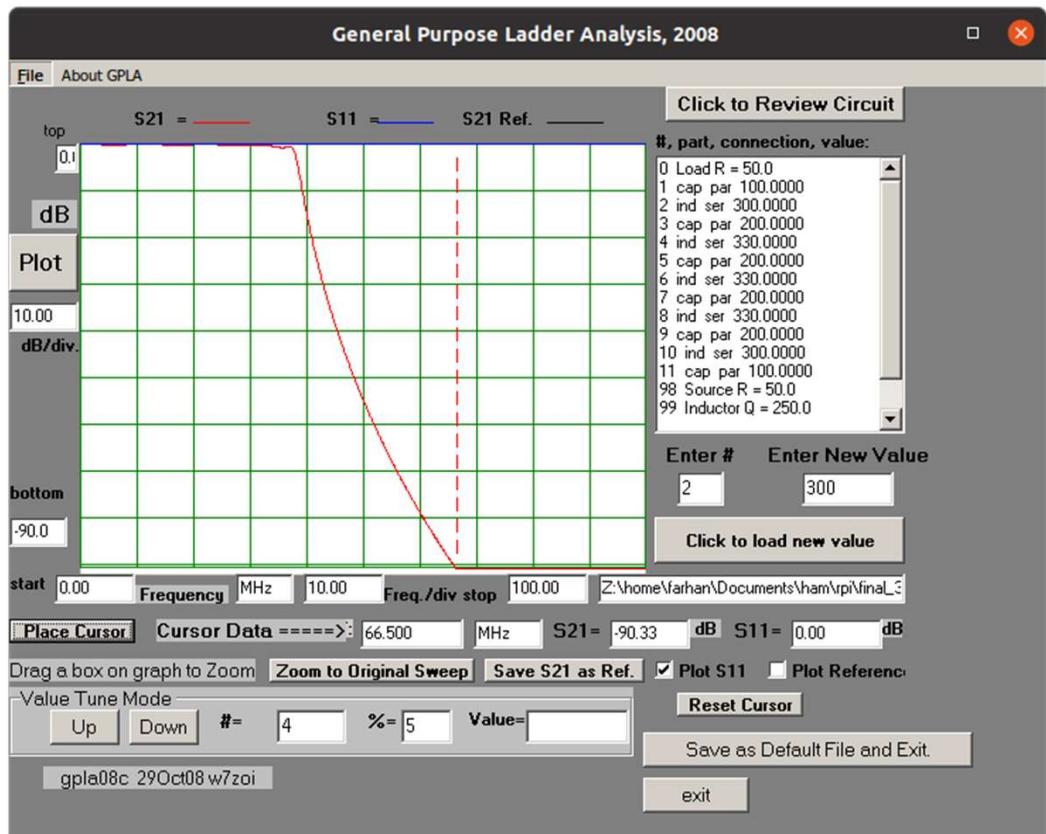
- Macro's , QSO's for each mode
 - Memories for each band
 - Built in Logger (FT8 Only?)
 - Receive 600 mA, transmit up to 6A
 - Built in mic and onscreen keyboard (gag)
 - External microphone and CW keyer input.
 - DIY integration with FlDigi, PAT (Winlink), POTA software (*)
 - The sBitx-v3 measures 10"x 6" x 2" and weights about 4 pounds (2Kgs)
- *diy vague directions (limited success!).. But integration in progress

Off the Shelf Parts

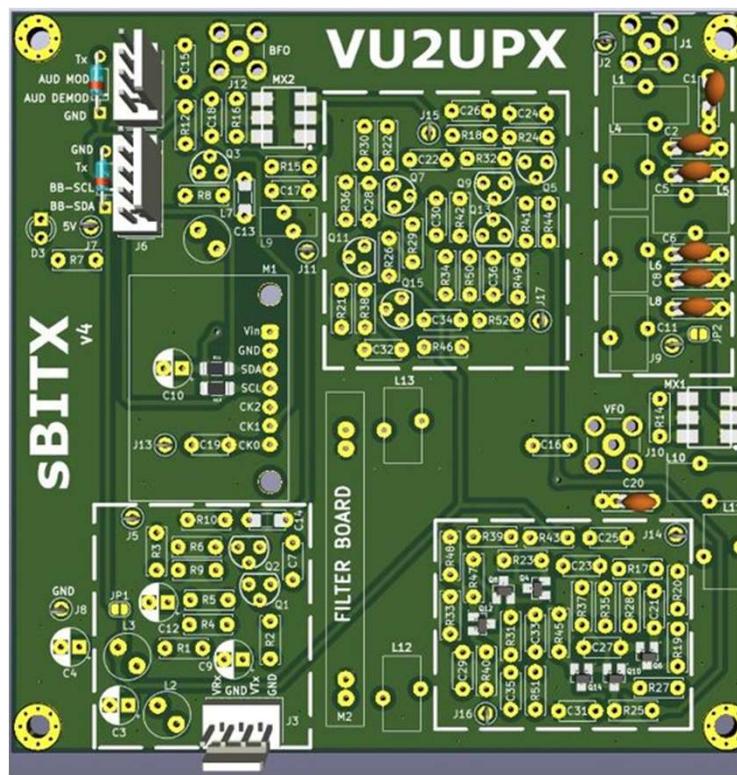


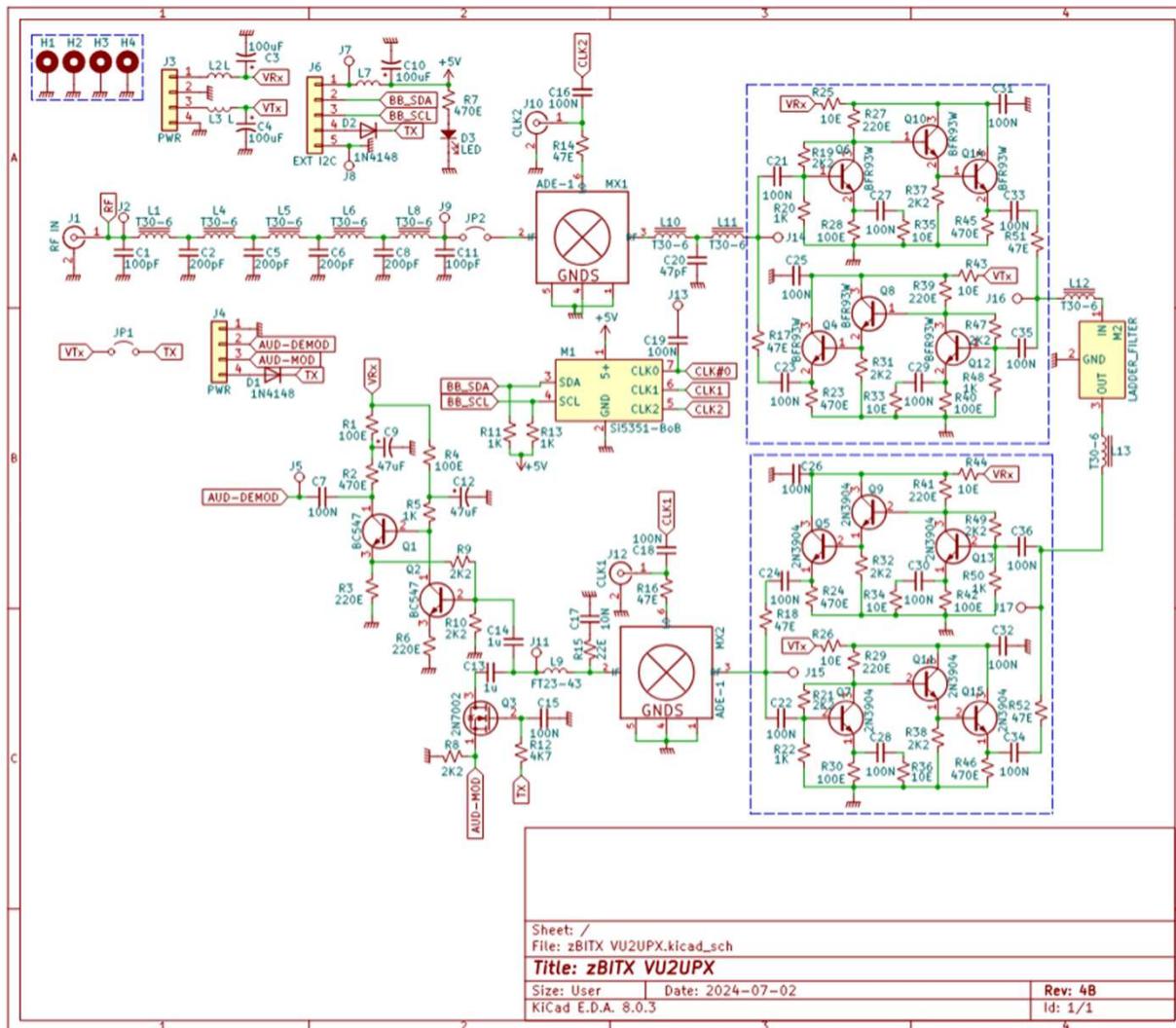
Radio Filter Portion

Low Pass Filter performance



SDR board sans parts





KI4UJY – Build – suitcase transportable – mini networking rack – go box (suitcase)



11" Home Networking Rack
Drawer/Shelf (table top or park bench operation)
Sbitx
Tunner (LDG) rated 100 watts
Rasberry Pi #2 open frame with sense HAT (why not)
Network Hotspot/Router
SDR receiver
Added a second screen – using a gamer portable HDMI screen powered by USB
No room for CW external keyer.. (built in keyer CW input
(Straight, Iambic A, Iambic B) 1/8" jack

Rear View



Anderson Power Pole power distribution
RF connections
120 Watt 12 volt regulator*
Cooling fins sBitx
USB to 12 volt adapter

*sBitx's low noise power supply rated for 11 to 13.8 (14.4 from my Lipo4fe is too high had to use a separate regulator to 13.8 for overvoltage protection)

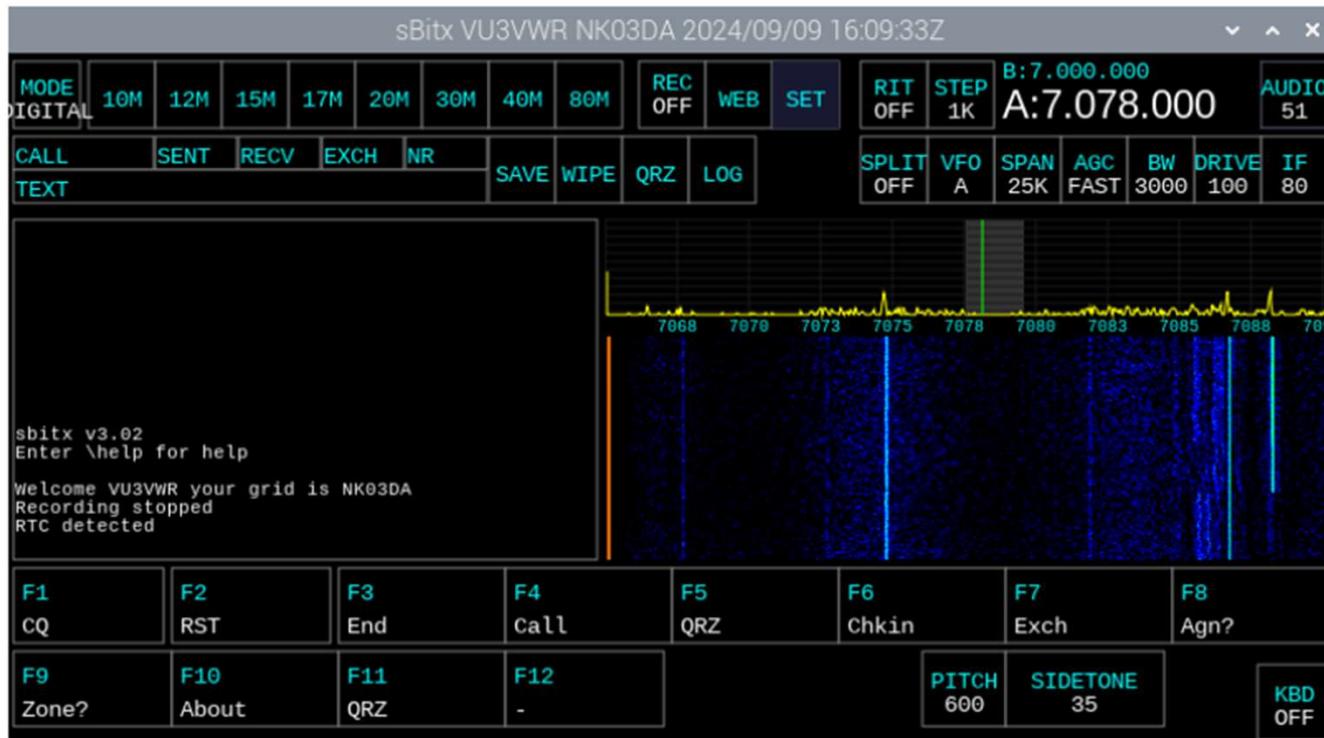
The Good..

- Radio/ QSO / Logger integrated interface with SDR transceiver
 - CW
 - FT8
 - SSB
 - AM
 - Digital? (Intended to work with 3rd party applications)
 - External USB and Bluetooth keyboard via Pi
 - External Screen (wired and wifi internet)
 - External USB interfaces and storage.
 - Touch Screen integrated with pi
- Stay within your license and privilege !
- Documentation and user groups on-line abound

The best – Macros (CW)

- # Sample macro file for regular dxing
- F1 CQ, cq cq cq de {MYCALL}{MYCALL}{MYCALL} ar k
- F2 RST, * ur rst {SENTRST}{SENTRST} kn
- F3 End,! de {MYCALL} . tnx fer rpt . 73 es cu agn kn sk
- F4 Call, *
- F5 QRZ, qrz?
- F6 Chkin, ! de {MYCALL} ur rst {SENTRST} es tnx fer call . hw cpy? AR K
- F7 Exch,! de * ur rst {SENTRSTCUT}{EXCH} K
- F8 Agn?,agn?
- F9 Zone?,zn?
- F10 About, ! de * . my name is Michael Hamby. rig is sbitx es dipole . hw cpi? !
- de * k
- F11 QRZ,qrz?

Onboard: Typical Operating Screen



Features:

- Band
- Mode
- Presets
- Waterfall/bandscope
- Built in logger

Sbitx Learning Curve

- Menu's interact with Mouse/ touch screen and knobs.
- Mode specific
- Always operate with either an antenna connected or dummy load – accidental xmit is a real possibility
- Menu's are lagging , and especially if the system is loaded up with 3rd party applications running..
- Preference would have been to have a virtual radio front end and interface to wsjtx/Klog and Grid Tracker – but that would require custom software – but it is achievable ..
- DIY radio setup with some working software supplied and some not so working 3rd party items. More on that latter..

CW operating screen

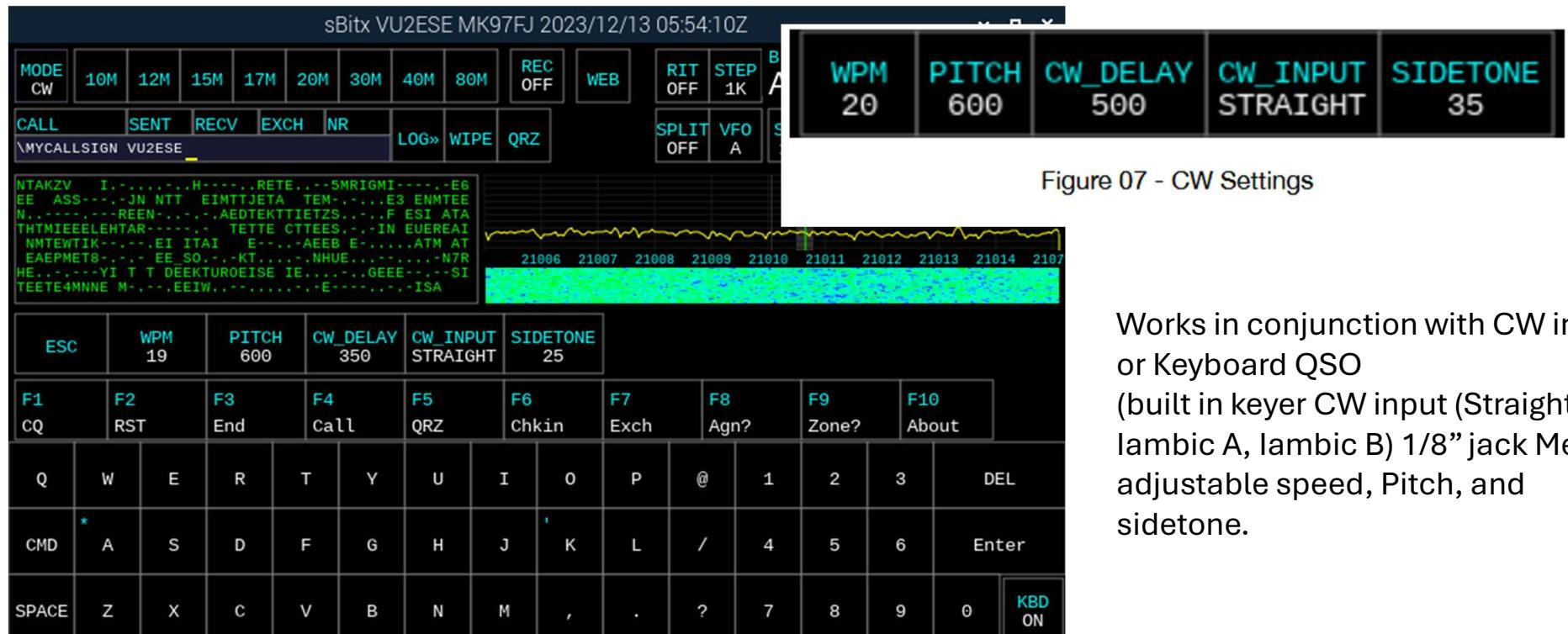


Figure 07 - CW Settings

Works in conjunction with CW input or Keyboard QSO
(built in keyer CW input (Straight, Iambic A, Iambic B) 1/8" jack Menu adjustable speed, Pitch, and sidetone.

CW Details

- The sidetone generation system provides audible feedback when sending Morse code with the K3NG CW Keyer. This audio output allows operators to hear what they're sending in real-time, even when operating in practice mode or when the transmitter output is disabled. For information about actually keying the transmitter, see [TX Keying and PTT](#).
- CW filters from 100 Hz to 3 KHz with minimum ringing
- Noiseless, electronic T/R and band switching with diodes.
- A very sensitive CW decoder
- Intersperse paddle/straight key sending with keyboard and macros
- Integrated logger with macros to generate standard messages.
- The microphone's PTT can also be used as a straight key if you don't have the key handy

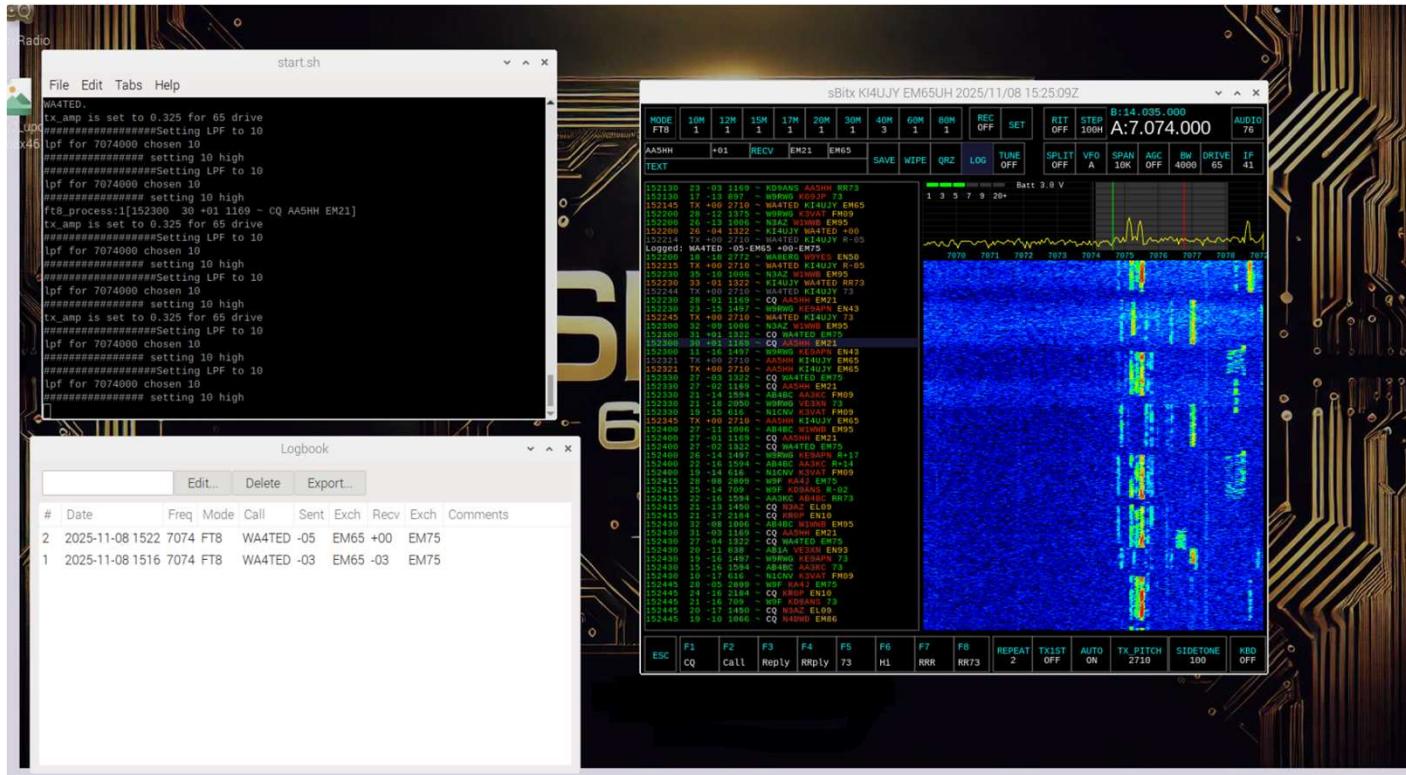
FT8 Integral Mode



Features:

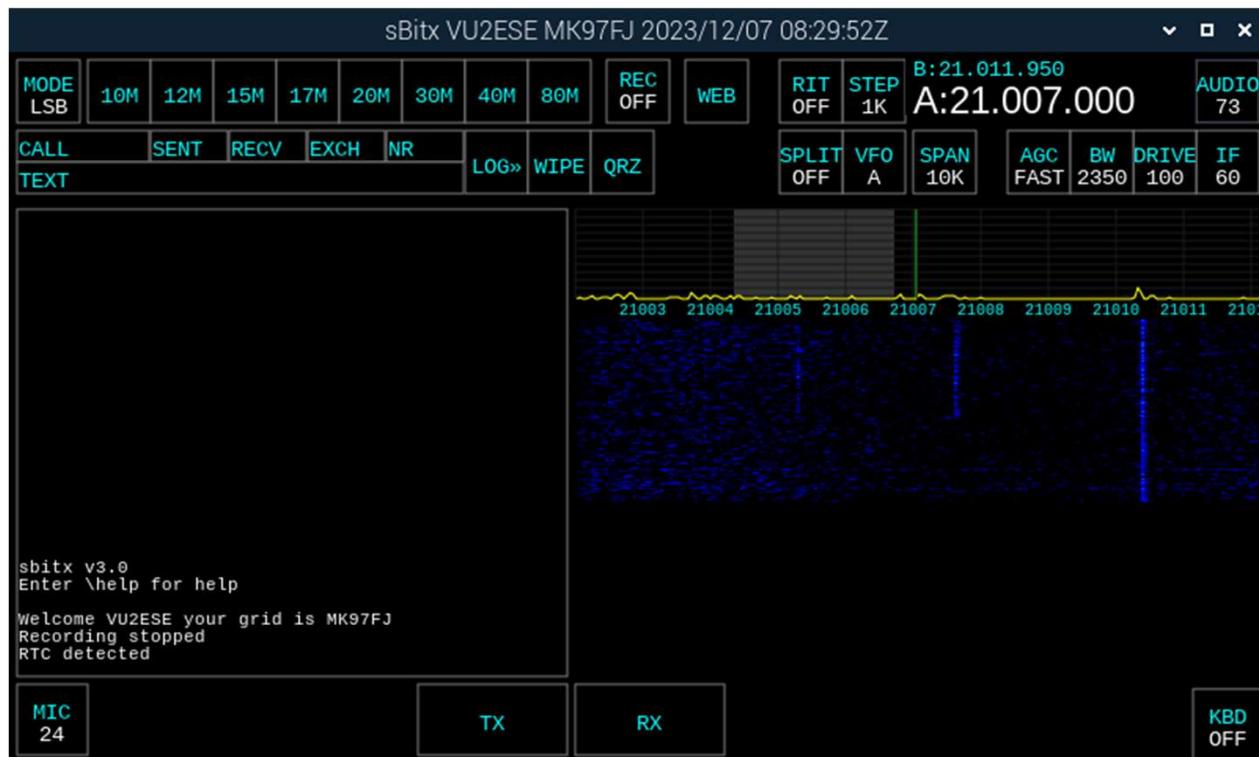
- Real time clock – sync with pi wifi (NTP)
- On screen decode
- Integral logging
- Auto Transmit on TX1st
- Auto QSO feature
- Menu's are configurable (contest, POTA)
- (on board screen is small – suggest using another computer in webpage mode!)

FT8 in action native SBITX Interface



Auto log and auto qso

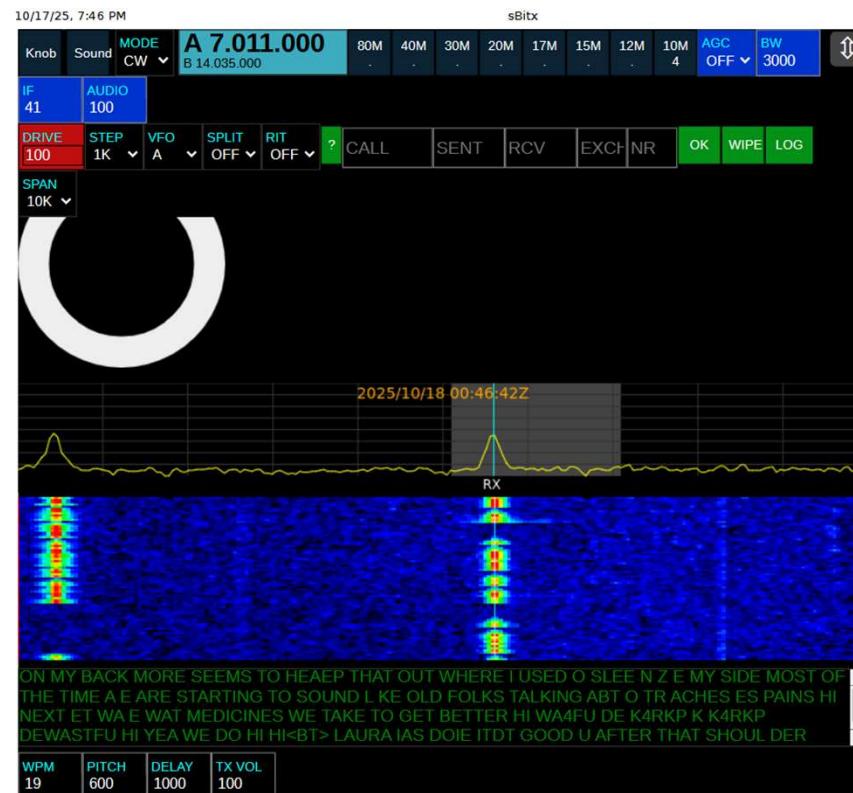
SSB Operation



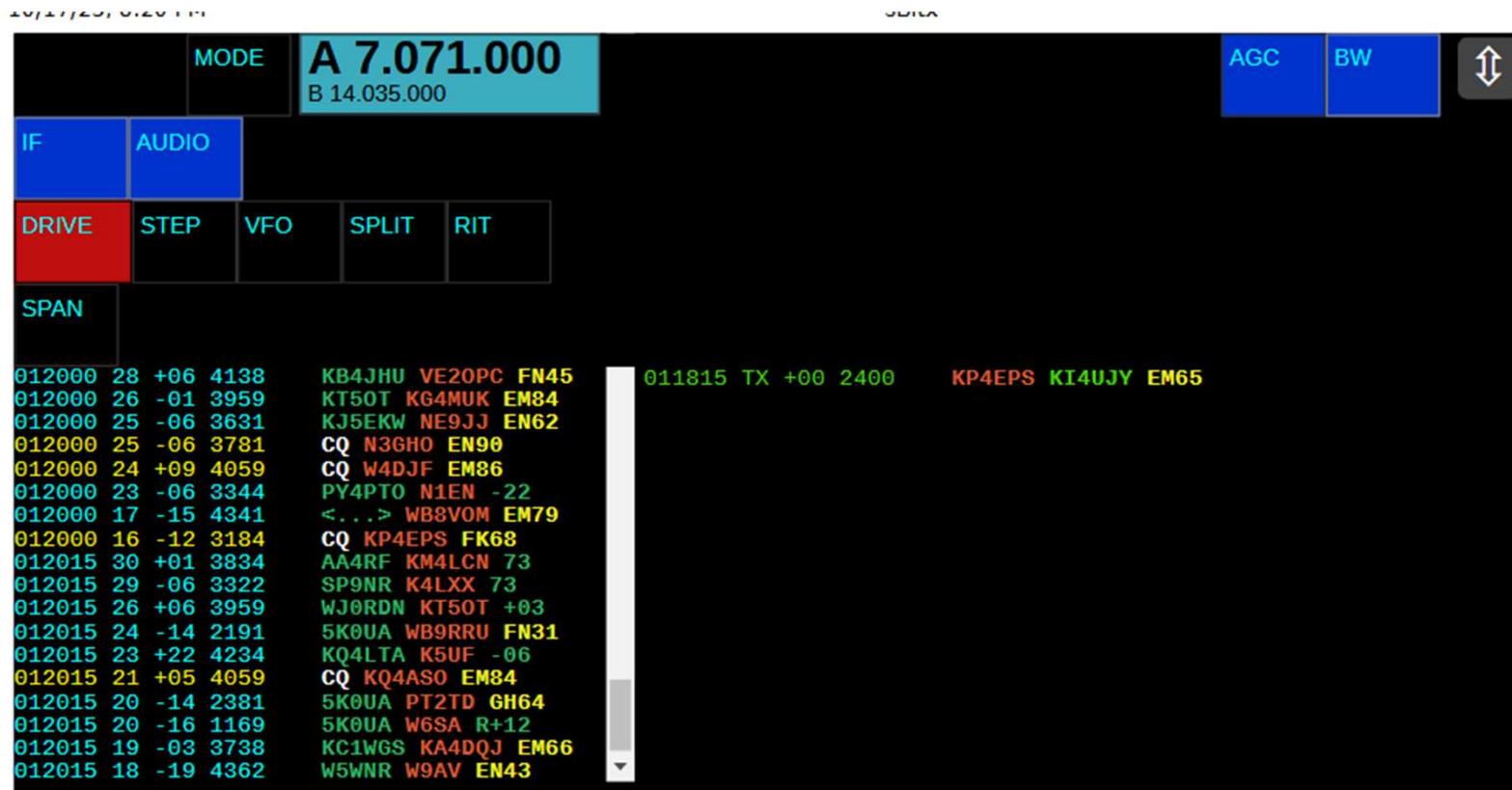
Features:

- Comes with external HT quality microphone
- Internal microphone
- Power limits per band..

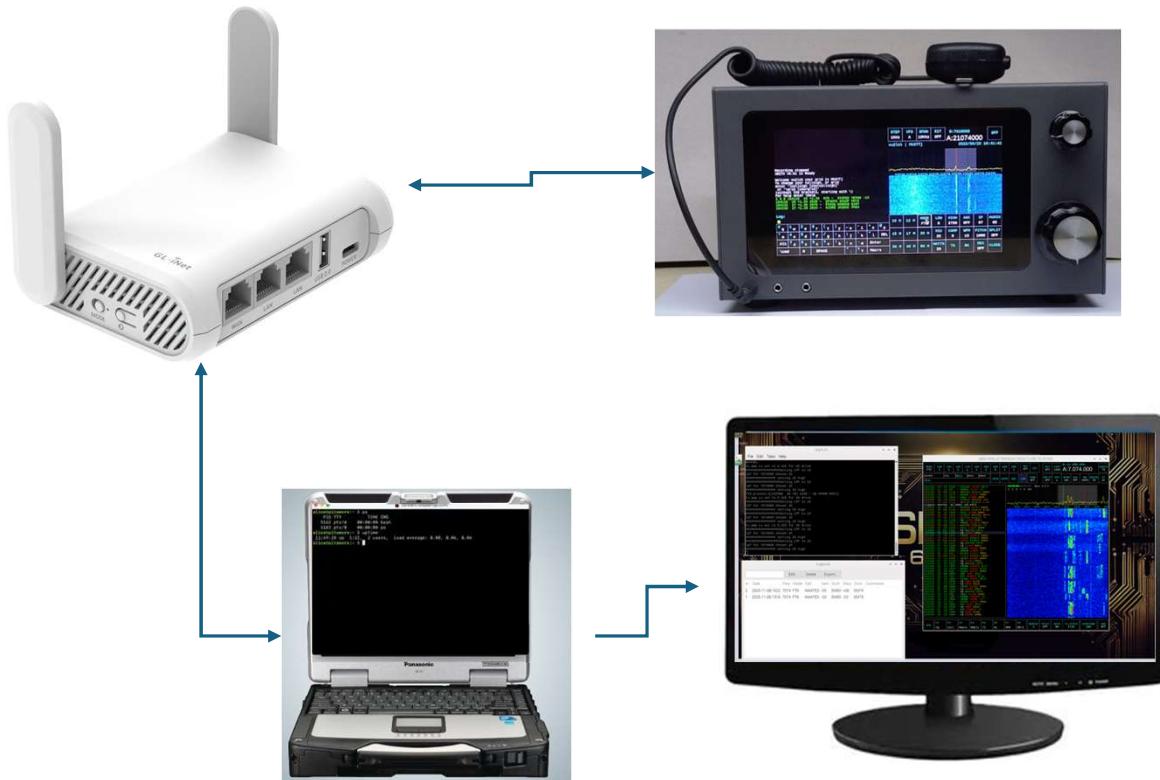
Tablet or Browser Operation (remote over wifi) (router with local network in my build)



FT8 com log – JHU heard from KY

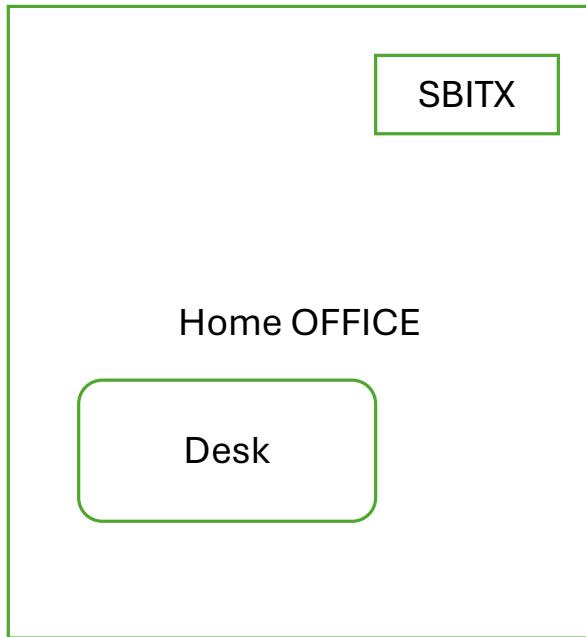


Sbitx Network Operation – Block diagram



- Sbitx V3.0
- GL.iNet GL-SFT1200 (Opal) Portable WiFi Travel Router, Mini VPN Wireless Router for Fiber Optic Modem, Mobile Internet WiFi Repeater, Dual Band Openwrt...
- PC or linux box running RealVNC viewer
- Or Web Page Radio Controls
- Cat6 or wifi

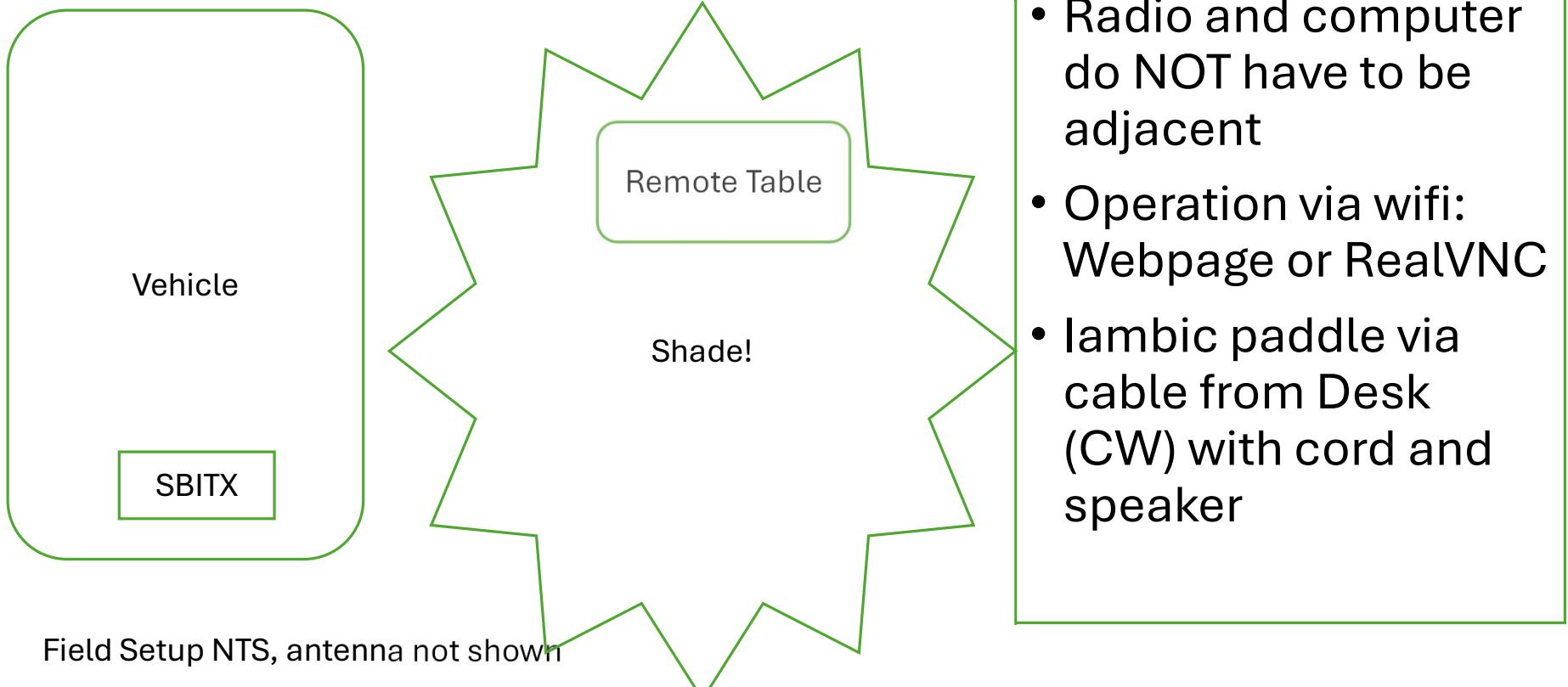
Network operations and local wifi ..



Home Setup Real advantage (one desk/one chair for home office/radio)

- Radio and computer do NOT have to be adjacent
- Operation via Webpage or RealVNC
- Iambic paddle via cable from Desk (CW) with cord and speaker
- (radio can even be in a different room..)

Parks on the Air/Camping..



The Ugly

- Integration with 3rd Party Ham radio applications is ... DIY
- Plug and Pray..
- Requires Linux and application layer understanding to troubleshoot.. (If only there was an engineer with some computer science / linux background handy)
- Haven't dove into the linux system logs yet to see where the faults occur...
- Ham Lib Rig Ctrl (appears to be where the fault lies)

3rd Party Applications where things get weird and wacky – some assembly required – frustration, heartbreak, and disappointment abound ..

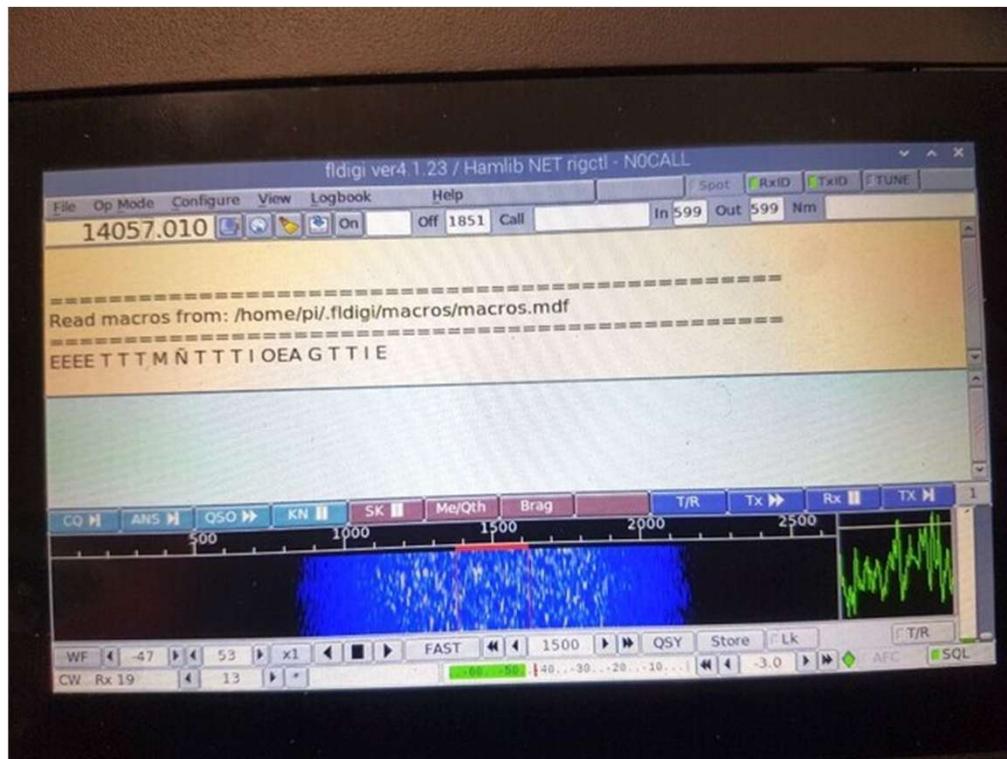
Using Third-party Software/Modems (per documentation)

- Hundreds of open source software, ranging from FreeDV to FlDigi to WSJT-X, are available on Raspberry Pi. To run them on sBitx:
 - Set sBitx Mode to DIGITAL
 - Enable Hamlib in the third-party software's setting.
 - From the third party software, choose **Hamlib NET rigctl** as the radio.
 - Set the Network server to 127.0.0.1, port 4532 (standard for NET rigctrl)
 - Set the **PTT Method** to **CAT**
 - Set the **Mode** (if prompted) to **USB**
 - Choose **plughw: CARD=Loopback_DEV=1** as the Input/Record Audio. Alternatively, you can also choose **Loopback: PCM(hw:1,1)**
 - Choose **plughw: CARD=Loopback_DEV=0** as the Output/Playback Audio. Alternatively, could also choose **Loopback: PCM (hw:2,0)**

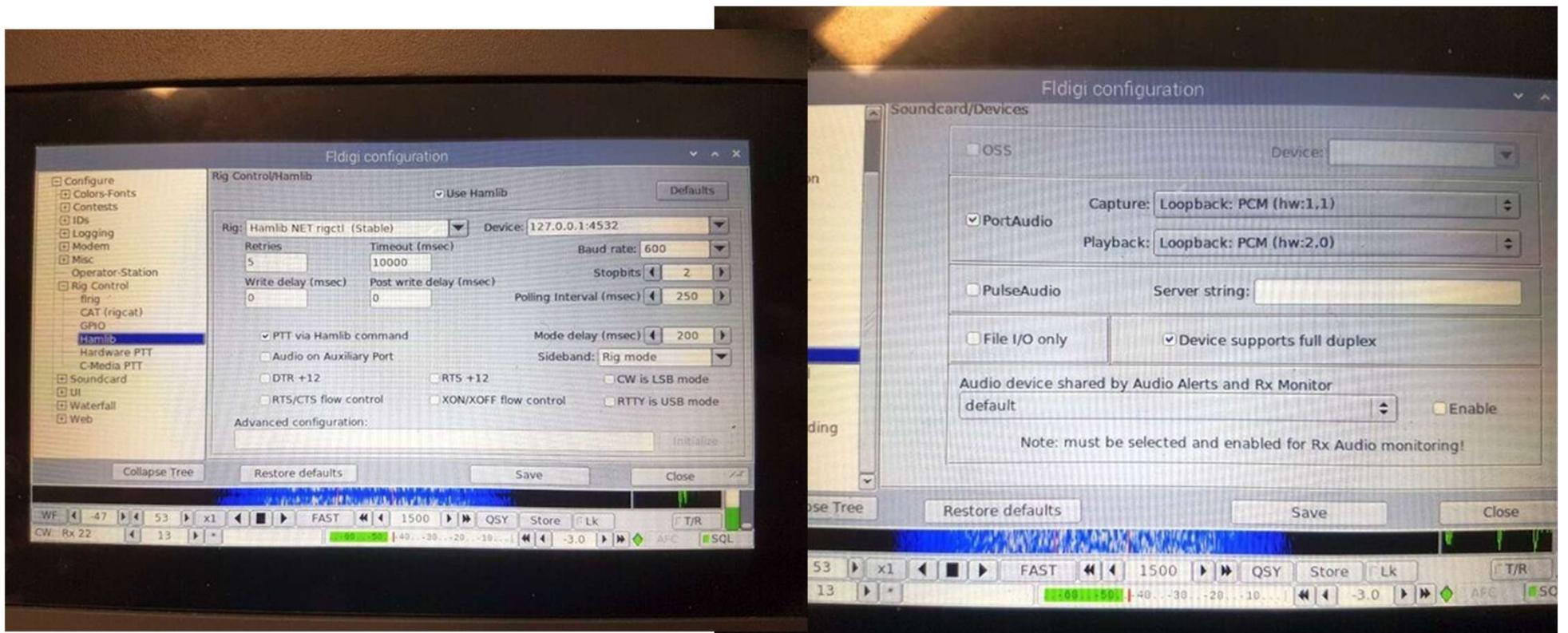
3rd Party Software Hopes and dreams

- In search of a working Hamlib NET rigctl solution (progress update)
 - WSJT-X – appears to work (does decode) – doesn't transmit!
 - Fldigi – appears to work (does decode) – doesn't transmit!
 - PAT (winlink in linux) – your own your own kid..
- Use of Hamlib ties the port up , changing mode requires a power cycle – especially true of WSJT (you must reboot to restart an instance)
- Very Laggy takes some time for 3rd party and Sbitx to ‘sync’ up
- Error messages from Ham Lib
- Searching the internet for wisdom .. Even consulted the oracle (chat GPT)
- https://groups.io/g/BITX20/topic/sbitx_v3/107702921

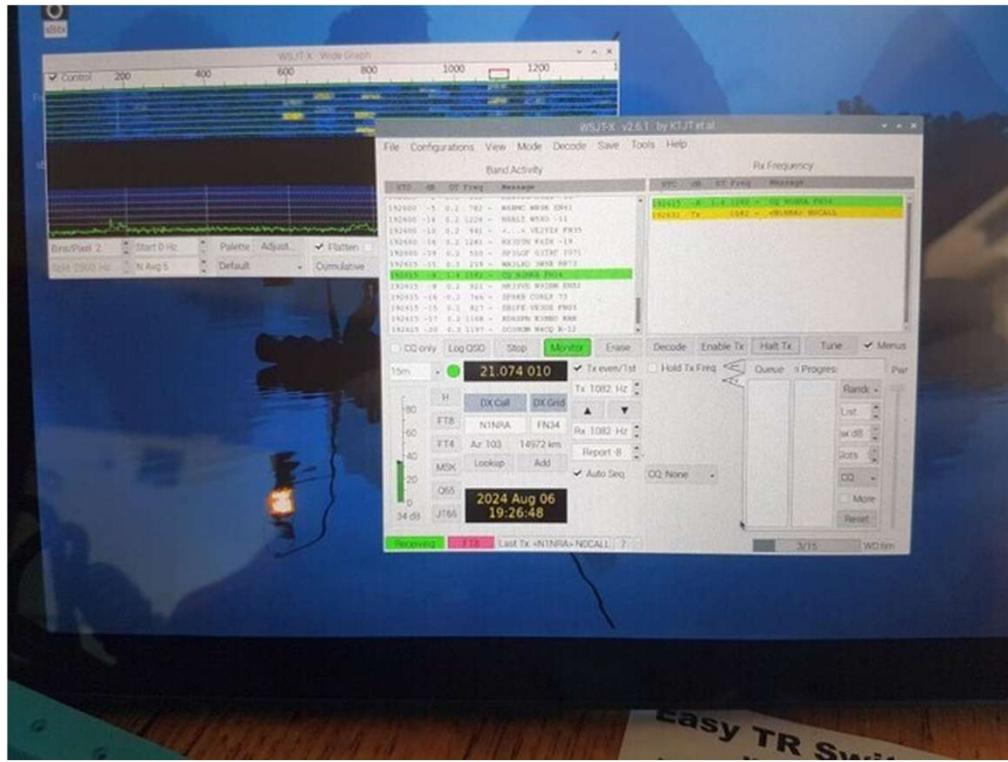
Fldigi – Oldy/Moldy .. But some lucky souls have got it working.. I haven't as of 10/27/2025



Fldigi – settings – Use Hamlib/SoundCard



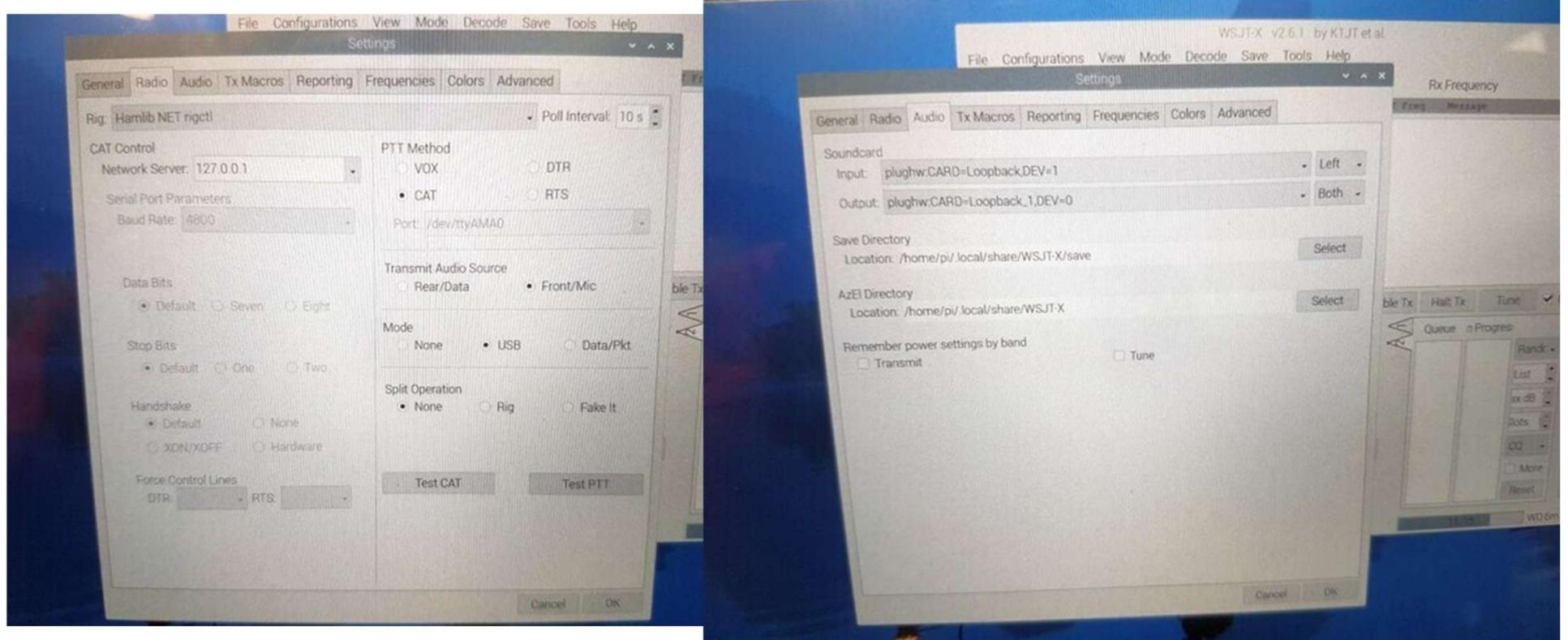
FT 8 WSJTx – would like to run wspr with this rig.. (to dream the impossible dream..)



"modes" called **FST4**, **FST4W**, **FT4**, **FT8**, **JT4**, **JT9**, **JT65**, **Q65**, **MSK144**, and **WSPR**

FT8 is very popular – contact based QSO - minimalistic

WSJT-X – settings



Wsjtx_notes.txt

- Before installing
- sudo apt --fix-broken install
- Download the 32 bit package from the wsjt-x page (the one with armhf in the name),
- sudo dpkg -i wsjt...
- you may have to run if it complains about the locales
- sudo pkg-reconfigure locales
- On WSJT-X
 - File > Settings > Radio, Choose "Hamlib NET", the NetServer is set to "127.0.0.1"
 - File > Settings > Radio, Set PTT Method to CAT, Mode to USB
 - File > Settings > Audio, Input Set to "plughw:CARD=Loopback,DEV=1", "Left"
 - File > Settings > Audio, Output Set to "plughw:CARD=Loopback1,DEV=0", "Both"
- add the line to /etc/rc.local
- sudo modprobe snd-aloop enable=1,1,1 index=1,2,3

AuxCom Winlink (Pat) the good the bad the really really .. ugly

- Pat uses ham lib rigctl to interface to the sbitsx sdr transceiver
- Out of the box the interface appears to be misconfigured
- Gui Pat control is ‘clunky’
- Since ham lib rigctl and ARDOP are both Tcp/IP based there is a potential for running Pat remote to the Sbitx from another installation of Pat (a second computer)
 - Pat configures localhost statement to be replaced by sbitsx IP address
- ARDOP appears to have been superseded ..
- Makes this a potential game changer for AuxCom
- IF ONLY IT WORKED!!! BUT HOPE SPRINGS ETERNAL...

Ham Library – the secrete sauce..

- **RIGCTL**
 - Rigctl [-hillLnouV] [-m id] [-r device] [-p device] [-d device] [-P type] [-D type] [-s baud] [-c id] [-t char] [-C parm=val] [-v[-Z]] [command|-]
 - Control radio transceivers and receivers. rigctl accepts commands from the command line as well as in interactive mode if none are provided on the command line.
 - Keep in mind that Hamlib is **BETA** level software. While a lot of backend libraries lack complete rig support, the basic functions are usually well supported.
 - Please report bugs and provide feedback at the e-mail address given in the BUGS section below. Patches and code enhancements sent to the same address are welcome.

Winlink (Pat) the good the bad the ugly

- **Starting the web server**
- Start Pat in **http** mode:
- \$ pat http
- 2016/03/15 01:34:40 Starting HTTP service (<http://localhost:8080>)...
- What you long to see:
- pat --listen winmor,ardop,ax25,telnet http
- 2016/03/15 01:40:49 ft897 ready. Dial frequency is 5.347.00 MHz.
- 2016/03/15 01:40:50 WINMOR TNC v1.5.7.0 initialized
- 2016/03/15 01:40:50 ARDOP TNC (ARDOP TNC_0.4.3.8-BPQ) initialized
- 2016/03/15 01:40:50 Listening for incoming traffic (winmor,ardop,telnet,ax25)...

Pat Under the hood: (rough notes) – work in progress

- Rig control appears to be broken in the out of the box configuration
- <https://themodernham.com/setup-pat-winlink-on-raspberry-pi-with-rigcontrol/>
- <https://github.com/la5nta/pat/wiki/ARDOP>
- <https://github.com/la5nta/pat/wiki/The-command-line-interface#configure>
- <https://github.com/la5nta/pat/wiki/Rig-control>
- Rig control notes:
- Get a list of radio identifiers:
- rigctl -l
- Find the Sbitx on the list (id ###)
- Start the daemon:
- rigctld -m ### -r /dev/ttyUSB0 -s 4800 **(this is dubious)**
- Now that rigctld is ready to accept connections on it's default port (4532), you'll need to configure Pat.
- To make Pat aware of your rig, open up the config (~/.wl2k/config.json or use pat help to find the config directory) and add a new rig under the hamlib_rigs-section:
 - "hamlib_rigs":{
 - "my_precious_rig": {"address": "localhost:4532", "network": "tcp"}
 - },

Sbitx Summary

- It's inexpensive*,
- Fairly Compact* , Decent Power
- Okay for POTO/ camping (suitcase packable)
- Native FT8 has a learning curve
- CW will probably be great!
- Goal is to use it with a remote Linux Tough Pad computer (with a sun readable screen)
- Using external monitor for setup and troubleshooting)
- Dummy load a requirement or antenna get all these interfaces working.
- 3rd Party Software does not require Signal Link USB or CAT interface (all virtual) both a blessing and an obscene curse..

References

- <https://www.hfsignals.com/index.php/sbitx-v3/>
- Online support group:
 - <https://groups.io/g/bitx20>
- Full source code, circuit and explanations on github.com/afarhan/sbitx
- <https://www.vu2ese.com/index.php/2022/07/20/the-sbitx-the-sdr-for-the-homebrewer/>
- <https://www.sbitx.net/>
- <https://hamlib.sourceforge.net/html/rigctl.1.html>
- <https://github.com/la5nta/pat/wiki/ARDOP>