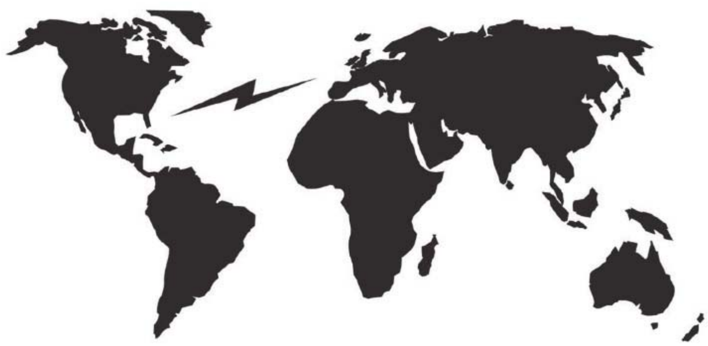
Charlotte, North Carolina

September 16 - 18, 2022

41st ARRL and TAPR Digital Communications Conference

Sponsored by
Amateur Radio Digital Communications

41st ARRL and TAPR Digital Communications Conference



ARRL
225 Main Street
Newington, CT 06111-1494 USA
Tel: 860-594-0200 www.arrl.org

Tucson Amateur Packet Radio
1 Glen Avenue
Wolcott, CT 06716 USA
Tel: 972 413-8277 www.tapr.org

Sponsored by:
Amateur Radio Digital Communications
5663 Balboa Ave Suite 432
San Diego, CA 92111-2705 USA
Tel: 858-477-9900 www.ampr.org

Printed in USA.

ISBN: 978-1-62595-138-0

First Edition

Greetings!

Welcome to the 41st annual ARRL and TAPR Digital Communications Conference (DCC). This year we are back to a live event for the first time in three years. After so much time in the virtual world, it will be good to see everyone in person again!

Since the virtual sessions of the past two years were so successful, we decided to do a combined live-stream and in-person conference. Look for the link to the live YouTube stream on-line at tapr.org.

We would like to thank Amateur Radio Digital Communications foundation (ARDC) for their support of this year's DCC. They have been immensely supportive in furthering TAPR's goals and ensuring that we have an in-person DCC this year. Just take a listen to the Sunday seminar for an idea of how much ARDC has contributed to the amateur radio community so far. And they are just getting started.

We would also like to thank Roland Kraatz, W9HPX, for his efforts acting as host for this year's event, and for being so patient while the 2020 DCC in Charlotte was postponed twice, to become this year's conference.

Breakfast on Saturday morning is provided courtesy of the Mecklenburg Amateur Radio Society (MARS) and the Charlotte Digital Radio Group. We thank them for their contribution!

The following pages contain the proceedings for this year's DCC. These papers represent not just those doing hard work, which they do, but those willing to take the time to present their results for all of us to see and build upon. This is the true spirit of advancing the radio art as furthered by ARRL, TAPR, ARDC and the DCC. We are grateful for their contributions, as well as those of the live speakers. It is written and live-presented papers combined that make the DCC such a fun and exciting event.

Whether you attend in person or watch the live stream, we hope you enjoy this year's DCC. Please let us know what you think, especially if you have suggestions to make it better next year.

73,

Scotty Cowling, WA2DFI
President, TAPR

Contents

Central Control Database for the HamSCI Personal Space Weather Station

***William D. Engelke*[1]**, *Cole Robbins*[1], *Nicholas Muscolino*[1], *Anderson Liddle*[1], *Travis Atkison*[1], *and Nathaniel A. Frissell*[2]

[1]The University of Alabama, Tuscaloosa, Alabama, USA
[2]The University of Scranton, Scranton, Pennsylvania, USA

Contributed Abstract for:
2022 TAPR DCC

The Ham Radio Science Citizen Investigation (HamSCI) Personal Space Weather Station (PSWS) is a modular ground-based platform for studying the geospace environment using ground magnetometers, medium frequency (MF) and high frequency (HF) radio receivers, Global Navigation Satellite System (GNSS) receivers, and very low frequency (VLF) radio instruments. The PSWS is a citizen science-based project, designed and developed as a partnership between the international amateur (ham) radio community and professional academic researchers. Data from each PSWS node is sent via internet to a public Central Control System, where the data can be easily downloaded by researchers. In this presentation, we discuss the architecture and show the features of the HamSCI PSWS Central Control System.

PERSONAL SPACE WEATHER STATION (PSWS) WEB SITE UPDATE

DCC

BILL ENGELKE, AB4EJ

CHARLOTTE, NC, SEP. 2022

V9

1

WHAT WE WILL COVER TODAY

- National Science Foundation (NSF)-sponsored project to develop a network of personal space weather stations
- Web site development status; testing prototype hardware

2

PERSONAL SPACE WEATHER STATION PROGRESS REPORT

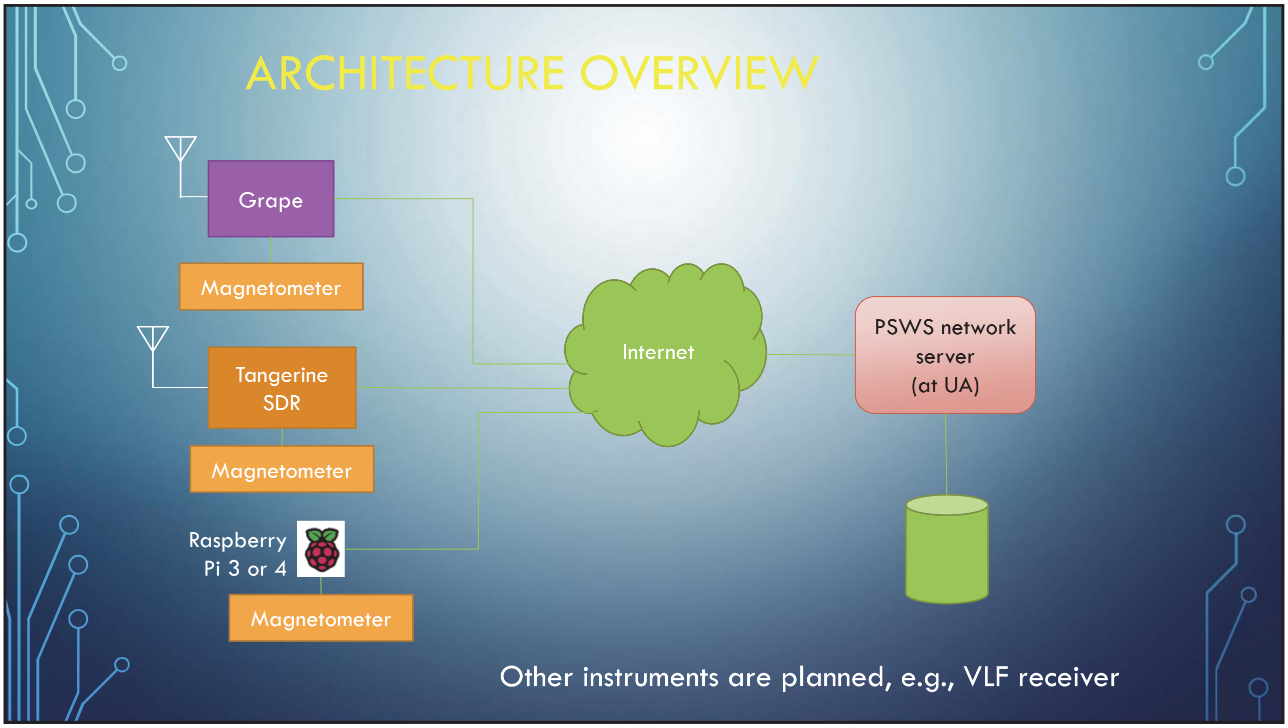
- Included design of 2 new software defined radios (one inexpensive, one high performance), plus a web site to hold collected data. Both radios use GPSDO.
- To be used for studying ionosphere via Doppler shift in WWV and other phenomena

3

STATUS

- The inexpensive SDR (CWRU) – is designed and is in final preparation
- The high performance SDR (TAPR) was delayed by about 2 years due to COVID supply chain, but we now have the parts!
- The magnetometer is in production, announcement at DCC Charlotte
- The software for all the systems is ready for testing; all open source
- The web site is in beta testing now

4


ARCHITECTURE OVERVIEW
Grape
Magnetometer
Tangerine SDR
Magnetometer
Raspberry Pi 3 or 4
Magnetometer
Internet
PSWS network server (at UA)
Other instruments are planned, e.g., VLF receiver

5


Personal Space Weather Station
Central Control System
Home
Stations
Observations
Users
Log In
Register your station
About
Welcome User : ab4ej-3 Log out
United States
Mexico
Cuba
mapbox

6

Personal Space Weather Station

Central Control System

Stations

Home | Stations | Observations | Users | Log In | Register your station | About

View your stations
Add new station

ID	User	Nickname	Grid	Elevation	Station status
N000003	ab4ej-3	N000003_A	EM63fi	112.0	Online
N000004	ab4ej-3	Grape N000019	EM63fj	100.0	Online
S000028	w2naf	W2NAF	FN21ei	480.0	Offline
S000030	AD0RR	Grape32	EM37ie	300.0	Online

View your stations
Add new station

7

HOW TO USE IT

- Sign up for account on web site
  - Add a station and one or more instruments
  - Grape, Magnetometer, TangerineSDR(future)
- Configure & start an uploader

8

Personal Space Weather Station

Central Control System

Observations

Home | Stations | Observations | Users | Log In | Register your station | About

Center Frequency:

10.000 MHz
5.000 MHz
2.500 MHz
15.000 MHz

Station Nickname: --------- Start Date: End Date: Submit Query

To download observation data, click on File/Observation link. Please be patient, it may take a while to zip a large observation.

Data rate	Center Frequency	Station	Instrument	Size	File/Observation	Start (UTC)	End (UTC)
1	—	N000003_A	Magnetometer2	293282	OBS2022-08-31T00:00.zip	2022-08-31T00:00:00+00:00	2022-08-31T16:36:00+00:00
10	5.000 MHz	Grape32	Grape32	718729929	OBS2022-08-04T13:00	2022-08-04T19:11:00+00:00	2022-08-31T16:29:00+00:00
10	5.000 MHz	Grape N000019	Grape19	725962491	OBS2022-08-04:13:00	2022-08-04T17:59:00+00:00	2022-08-31T16:19:00+00:00
1	—	N000003_A	Magnetometer2	428862	OBS2022-08-30T00:00.zip	2022-08-30T00:00:00+00:00	2022-08-30T23:50:00+00:00

1

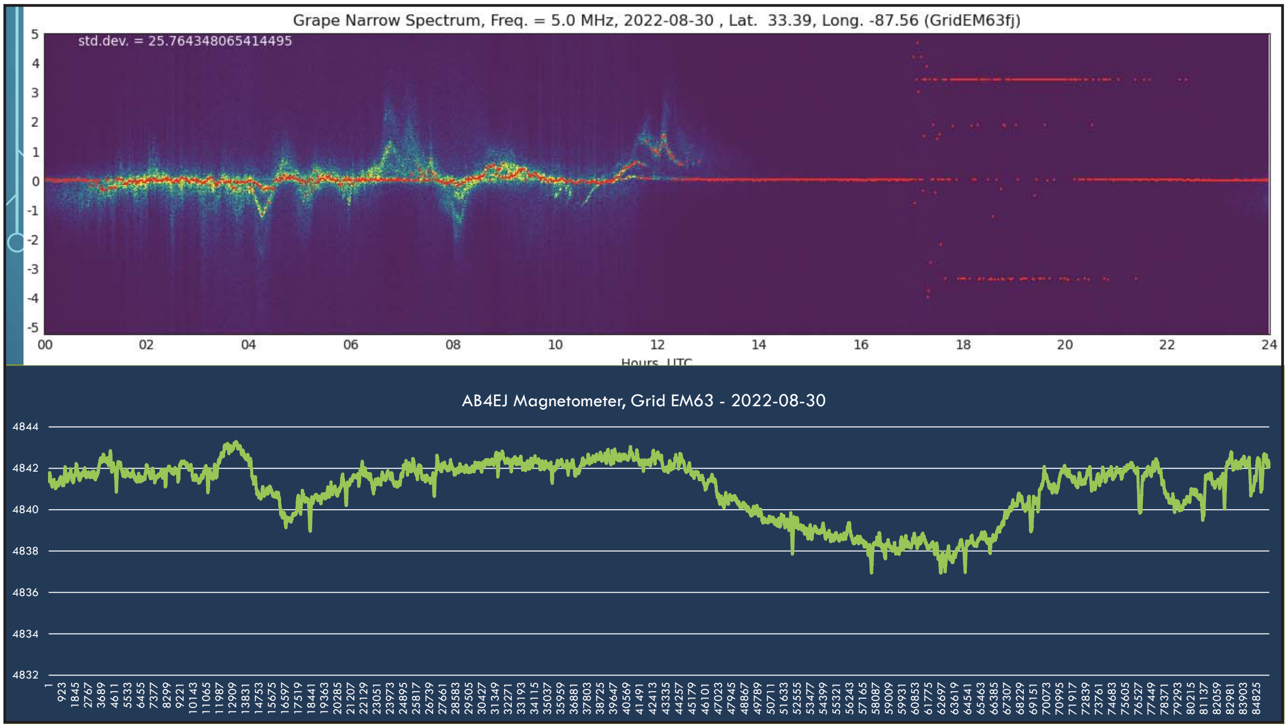
Page 1 of 15. next last »

9

USING OBSERVATIONS

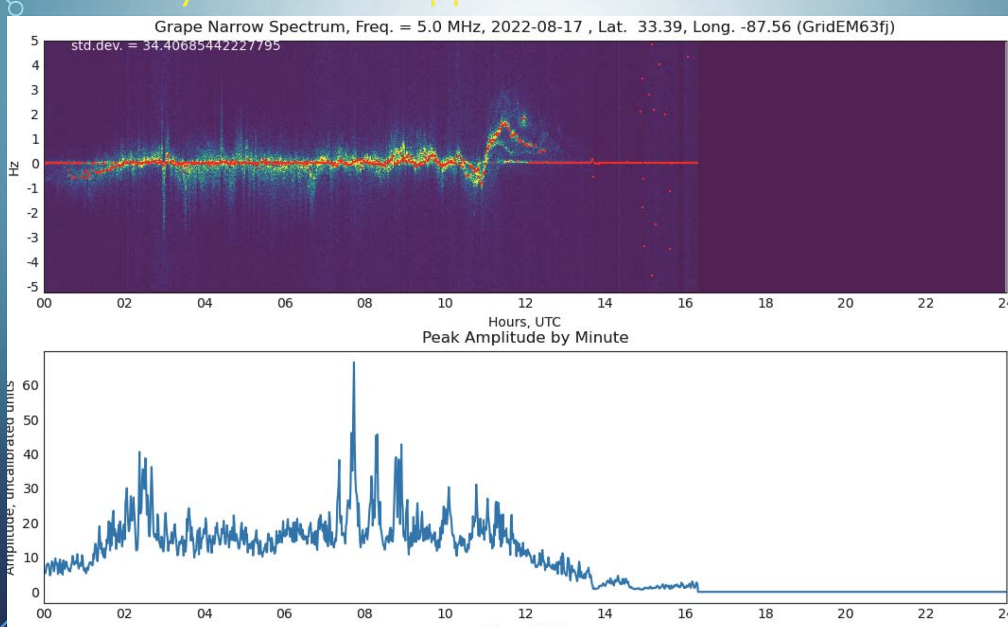
- Click on observation of interest
- Wait! (Large observation may take several minutes to be compressed)
- File shows up in your download directory
- Unzip file & run your data analysis
- Spectrum files can be examined by making a waterfall plot
- Magnetometer files can be analyzed using Excel (csv files)
- Some examples…

10

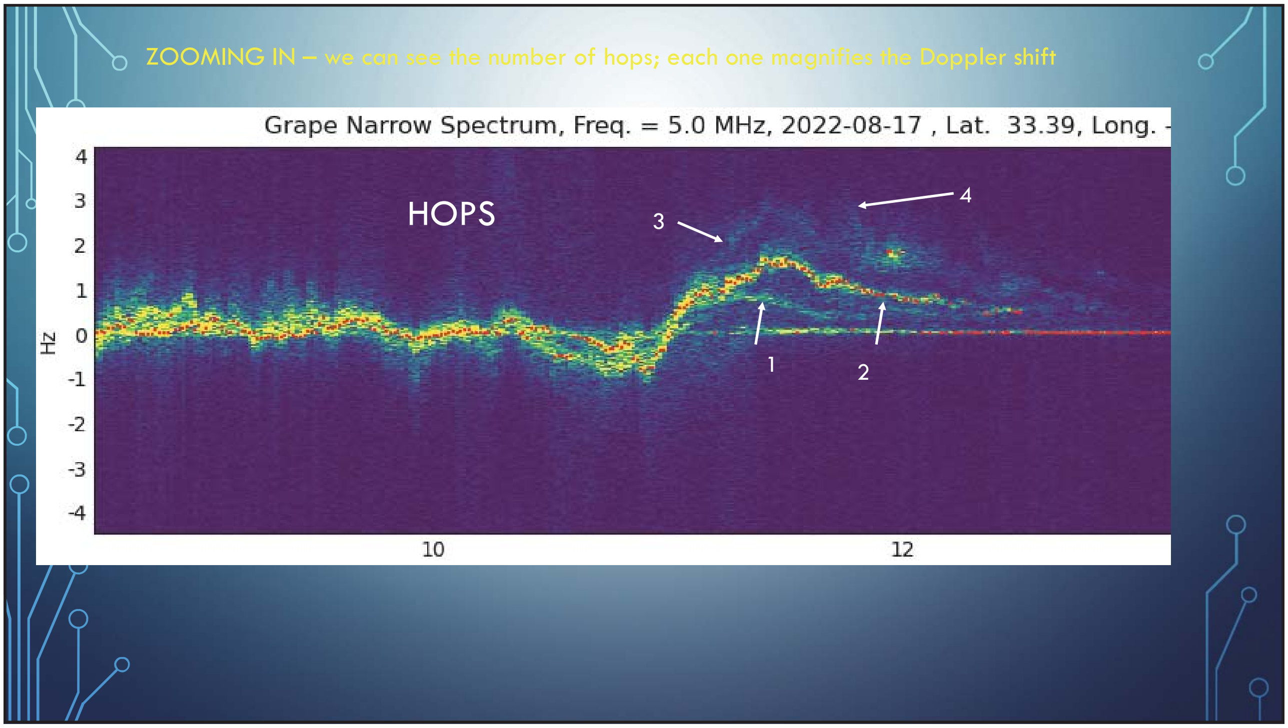


11

Data Analysis: WWV Doppler Shift at sunrise

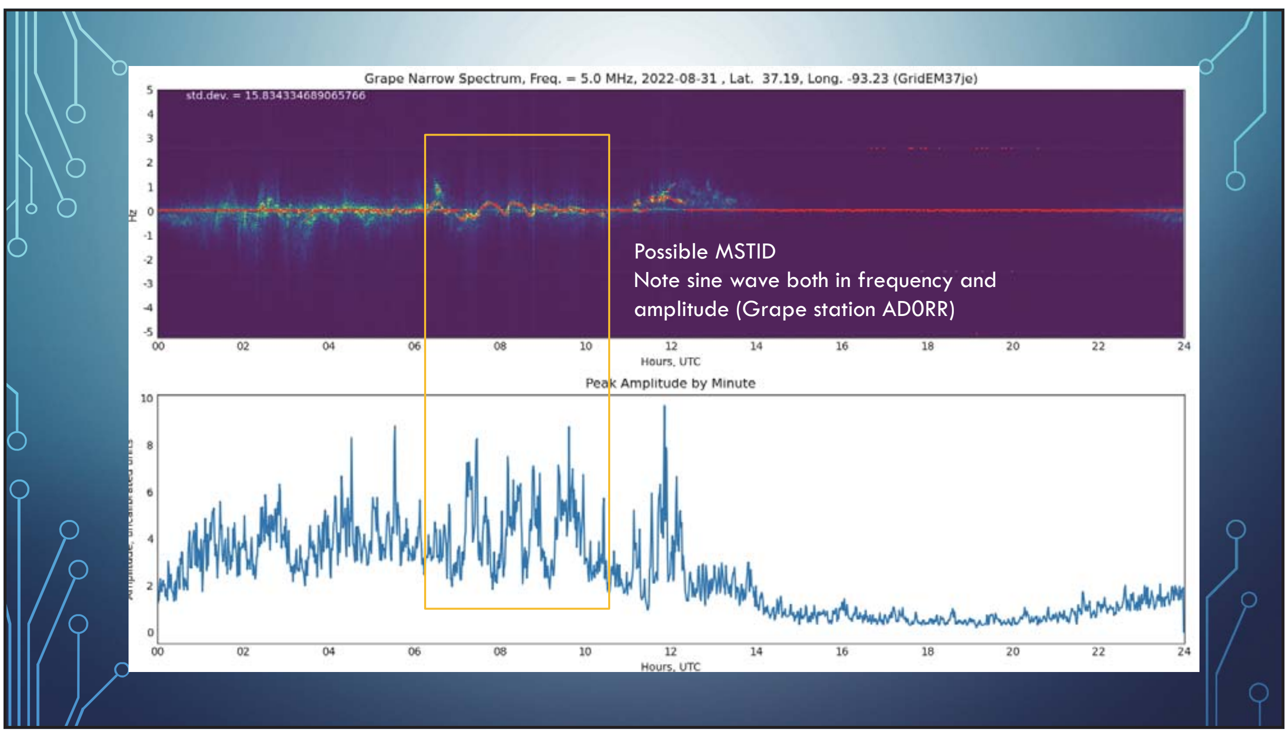


12



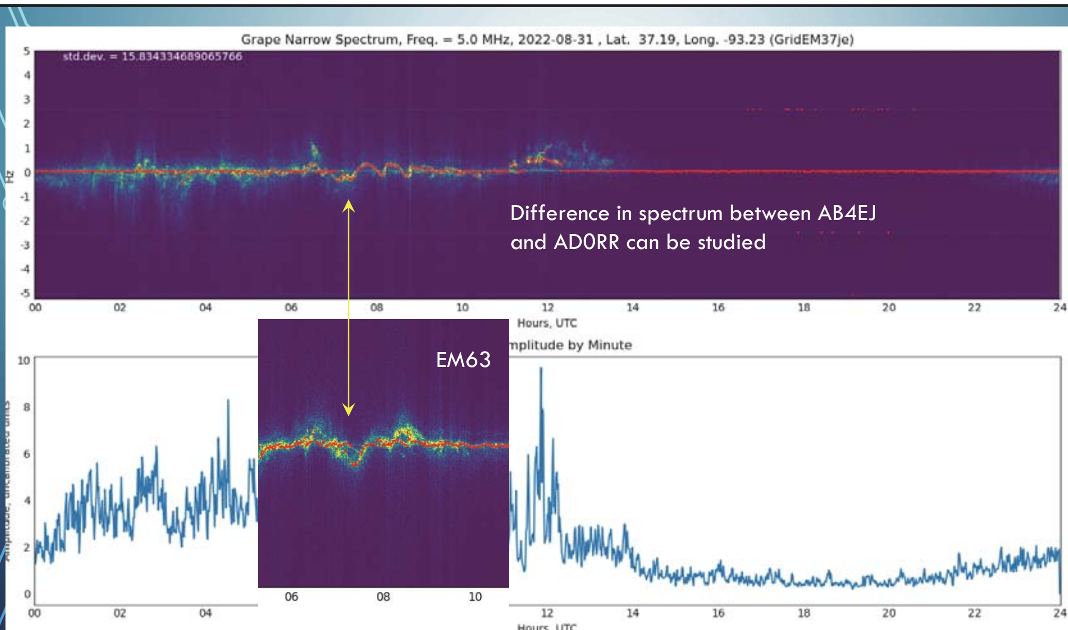
ZOOMING IN – we can see the number of hops; each one magnifies the Doppler shift
Grape Narrow Spectrum, Freq. = 5.0 MHz, 2022-08-17 , Lat. 33.39, Long. -
HOPS
3
4
1
2
Hz
10
12

13



Grape Narrow Spectrum, Freq. = 5.0 MHz, 2022-08-31 , Lat. 37.19, Long. -93.23 (GridEM37je)
std.dev. = 15.834334689065766
Hz
Possible MSTID
Note sine wave both in frequency and
amplitude (Grape station AD0RR)
Hours, UTC
Peak Amplitude by Minute
Hours, UTC

14



Comparing spectrum between stations; eventually interferometry may be possible

15

WHAT NEXT?

- We are looking for more beta testers who can set up a magnetometer and start reporting data to the web site
- Software is planned for producing magnetometer and spectrum plots on web server
  - Today you can download observations & process them at your site
- Final preparations in progress for producing both the Grape V2 and the Tangerine SDR
- Further tweaks and cleanups are planned for web site

16

GET INVOLVED

- www.tangerinesdr.com
- Monday night telco every week at 8 PM Central
- Thursday afternoon telco every other week at 2 PM central for science discussions

17

Q & A

- Bill Engelke, AB4EJ
- bengelke@cs.ua.edu

18

ACKNOWLEDGEMENTS

- NASA Grants 80NSSC21K0002 and 80NSSC21K1772
- Virginia Tech SuperDARN plotting software: https://zenodo.org/record/3727269
- SuperDARN data: https://link.springer.com/article/10.1007/BF00751350
  - Thanks to Virginia Tech SuperDARN team for use of the North America data

19

FST4W on the HF Bands: Why - What to expect - Equipment - Results.

Gwyn Griffiths G3ZIL, Glenn Elmore N6GN, Rob Robinett AI6VN, Lynn Rhymes WB7ABP, John Watrous K6PZB

Corresponding author: Gwyn Griffiths gwyn@autonomousanalytics.com

Summary

A greater use of FST4W on the HF bands could bring several benefits, the marginal sensitivity increase of 1.4 dB over WSPR for the 120-second variant probably being the least valuable. Other potential benefits are greater tolerance to Doppler spread and the significant increases in sensitivity from the longer sequences. While ionospheric Doppler spread will limit the use of the longer sequences, maximum utility of FST4W can come from careful attention to minimising spectral spread within transmitters and receivers. These are areas where the amateur can make improvements. As the protocol measures spectral width it provides the essential tool to both gauge equipment performance and also bring insights into propagation that SNR alone does not provide

1. Introduction - Why HF band FST4W?

The FST4W (and FST4) digital protocols were introduced in WSJT-X 2.3.0 together with a Quick-Start guide[1]. Table 1 summarises the protocol's basic parameters. While designed specifically for the LF and MF bands FST4W has several advantages over WSPR that could prove beneficial on the HF bands. These are:

- Option of four sequence lengths, designated in seconds as FST4W-120, -300, -900 and -1800. While sensitivity increases with sequence length, tolerance to spectral spread decreases.
- Better sensitivity, e.g. the FST4W-120 SNR threshold is about 1.4 dB lower than WSPR.
- Greater tolerance to Doppler spread, e.g. for a signal with a Doppler shift of 2 Hz FST4W-120 will decode at -24 dB SNR while WSPR requires -17 dB SNR.
- Optional measurement of the spectral spread. With WSPR we cannot say whether failure to decode a spot was due to inadequate SNR or excess spectral spread. FST4W gives us insights into spectral spread that lets us determine the likely cause. Spectral spread may also show patterns related to propagation, e.g. at band opening and closing, dusk and dawn, during geomagnetic disturbances and their aftermath that may be of interest.

Despite these potential advantages over WSPR to date the uptake of FST4W on the HF bands has been minimal. Users of WSJT-X on receive have to preselect a single sequence length to decode, thereby missing reception of other sequence lengths, let alone WSPR in the same band. In a substantial upgrade to WsprDaemon[2] 3.0 author Rob Robinett has enabled decode of all requested FST4W sequence lengths as well as WSPR on each selected band. The aim is that by enabling simultaneous multiband, multimode decoding this will encourage more amateurs to transmit FST4W. A real-time list of WsprDaemon users that shows those with FST4W decode enabled is available online[2].

Parameter	**-120**	**-300**	**-900**	**-1800**
SNR threshold (dB)	-32.8	-36.8	-41.7	-44.8
Symbol length (s)	0.683	1.792	5.547	11.200
Tone spacing (Hz)	1.46	0.56	0.18	0.089
Occupied bandwidth (Hz)	5.9	2.2	0.72	0.36
Measured spectral width (mHz)	5.3	2.14	0.74	0.35

Table 1. Basic parameters for the FST4W digital protocol for the four sequence lengths together with the measured baseband spectral width from jt9. SNR threshold is in dB in a 2500 Hz bandwidth. Except for measured spectral width and an SNR threshold of -31.4 dB WSPR-2 has the same parameters as FST4W-120.

This paper is organised as follows. Section 2 outlines what an FST4W user on the HF bands may expect, drawing on the protocol modelling tool *fst4sim* included in the WSJT-X package. Section 3 reinforces the message in the Franke et al. Quick-Start Guide that receiver and transmitter oscillator drift and short-term frequency jitter need to be considered carefully within an overall spectral spread budget for each sequence length. The examples in section 4 concentrate on the 14 MHz band, ranging from surface wave over tens of km to single and multiple hop ionospheric F layer propagation when the earth's geomagnetic field is quiet and active. The paper ends with a discussion of these and other applications for this underutilized digital protocol having clear potential on HF bands as a propagation reporter and investigative tool.

2. What to expect?

Figure 1 summarises two of the advantages of FST4W-120 over WSPR. First, at spectral spreads of less than 0.2 Hz there is 1−1.4 dB better sensitivity. Second, above a spectral spread of 2 Hz the rate of rise of required SNR is less for FST4W-120 than for WSPR. At a spectral spread of 3 Hz WSPR will not decode whatever the SNR.

The potential advantage of much lower decode thresholds for the longer sequences, Figure 2, are realised in practice at LF and MF where spectral spreads due to propagation and equipment are lower than on the HF bands. The practical pointers in Section 3 on equipment requirements and the examples of different propagation conditions in section 4 illustrate what can be achieved on 14 MHz. Suffice to say here that the -1800 sequence length will likely only be of use in few cases at upper HF, e.g. given surface wave propagation and GPSDO oscillators at transmitter and receiver.

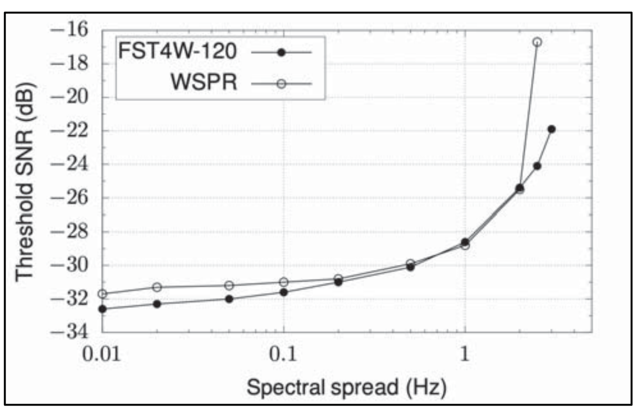


Figure 1. A comparison of the decode threshold with spectral spread for WSPR and FST4W-120. Acknowledgement: Based on figure provided by Steve Franke K9AN.

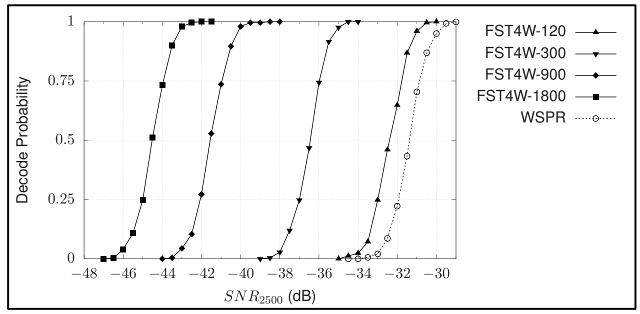


Figure 2. How decode probability varies with SNR near the threshold for FST4W-120 to -1800 in comparison with WSPR. Acknowledgement: figure provided by Steve Franke K9AN.

2.1 Spectral spread and spectral width

It is important to address a matter of likely confusion between spectral spread and spectral width. In this paper spectral spread in an 'input' parameter, e.g. as *fdop* in the fst4sim command line. In contrast spectral width is an 'output' parameter from the *jt9* executable.

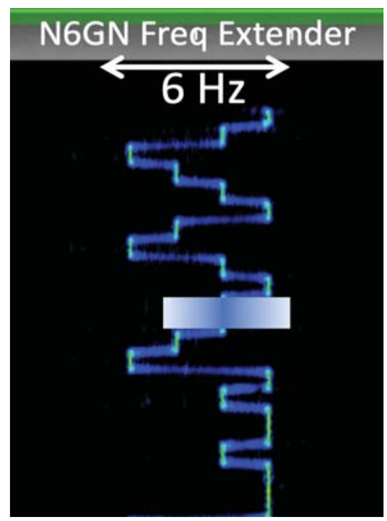


Figure 3. Waterfall display of the spectrum of the 33rd harmonic of a 14 MHz local FST4W-120 transmission from on ANAN SDR at N6GN using a KiwiSDR with GPSDO and a homebrew frequency extender. The image shows the smooth transitions between tones and the fraction of a tone period they span. The rectangle at the centre represents additional spectral spreading.

For a simple pulse of constant amplitude and frequency the bandwidth-time product (BT) is 1, that is, a 2 s long pulse will have a bandwidth of 0.5 Hz. However, for the Gaussian Frequency Shift Keying modulation scheme adopted for FST4W BT is less than 1, set by the bandwidth of the Gaussian filter. For BT=1 the expected spectral width of a 109.3 second transmission (the actual duration for -120) would be 9.15 mHz, i.e. 1/109.3 Hz. From *jt9* the measured spectral width of WSJT-X baseband audio was 5.3

mHz, suggesting a Gaussian filter with a BT~0.6, comparable to that of Bluetooth at 0.5).

The important point is that the *jt9* output spectral width is for a single tone, and not the occupied bandwidth of the received signal spanning four tones. Figure 3, a waterfall spectrum from author Glenn Elmore of the 33rd harmonic of a locally generated 14 MHz FST4W-120 transmission, shows the four-tone G-FSK with smooth (Gaussian) transitions between tones. A rectangle, added to the graphic, illustrates how a spectral spreading increase encroaches on the other frequencies, causing inter-symbol interference while also reducing SNR.

2.2 Uses of the fst4sim simulator

While the information in Figures 1 and 2 give us a good starting point in setting expectations for FST4W the *fst4sim* simulator that is available as a command line tool in WSJT-X lets us explore other potential pitfalls and unexpected outcomes. The command line is:

fst4sim "message" TR f0 DT fdop del nfiles SNR F
where:

message - callsign, locator and power to encode, e.g. "G3ZIL IO90 23"

TR - sequence length in seconds, values for FST4W being 120, 300, 900 and 1800.

f0 - baseband frequency, our examples use 1500 Hz.

DT - time offset in seconds between tx and rx.

fdop - notionally the (ionospheric) Doppler spread, but in this work we take this value as the total *spectral* spread that includes spreading due to the tx and rx. fst4sim uses an embedded version of an ITU channel simulator[3] in turn based on the Watterson[4] model.

del - the differential path (multipath) delay in milliseconds. We have not explored this parameter and use the default of 1 ms; in all cases the differential delay is much shorter than the FST4W symbol lengths.

nfiles - number of simulations to run.

SNR - the input SNR, which we will compare with the model output. In all cases we have used -15 dB.

F - a flag to indicate FST4W mode.

The simulator generates number *nfiles* of wav files of the selected length that the *jt9* command line executable can decode. By including an empty file named *plotspec* in the directory in which *jt9* is run the spectral width will also be calculated and displayed in a WSJT-X window.

2.2.1 Effect of time offset

While rarely a problem these days (a check on the time offset of 1000 WSPR spots on 20 m showed a lower decile of 0.024 s and an upper decile of 1.09 s) larger time offset can occur, e.g. using a remote KiwiSDR via a browser. Figure 4 shows the output SNR and output spectral width when fdop=0 for FST4W-300. Between time offsets of -1.0 s to +2.0 s both SNR and spectral width values are well behaved. However, while still decoding the data, biased values are reported out between -1.5 s and -1.0 s and between +2.0 s and +2.4 s, with SNR biased low and spectral width biased high. Other sequence lengths show the same general behaviour over very similar time offsets.

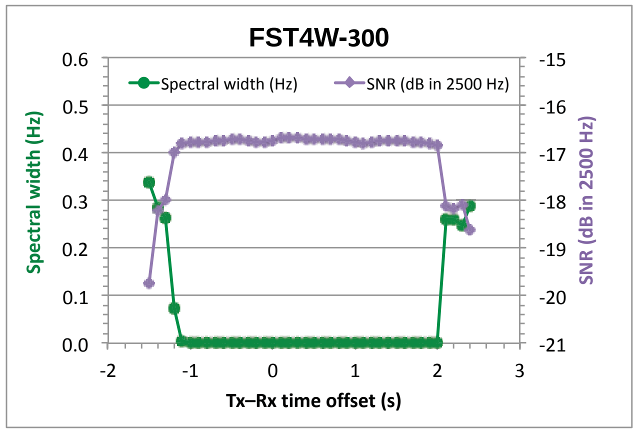


Figure 4. Variation of SNR and spectral with from fst4sim with time offset between transmitter and receiver for -300, showing biased values at each extreme but before no decode is possible.

Closely related to time offset, measurements of the effect of changing a KiwiSDR's *abuf* setting for browser connections are described in section 3.

2.2.2 Calculated spectral width, decode probability and SNR with spectral spread

For these simulations *nfiles* was set at 50 for each value of *fdop* and the mean and standard deviation of the computed set of SNRs and spectral widths calculated. The probability of decode at each *fdop* was estimated from how many of the 50 wav files were decoded. Figure 5 shows the variation of calculated SNR, spectral width and decode probability for the -120 and -300 sequence lengths vary with spectral spread.

We see that:

- At zero *fdop* the decoded SNR is over 1 dB down on the -15 dB input *SNR* for an unknown reason.
- SNR becomes further biased low as *fdop* increases, reaching -2 dB even with probability of decode

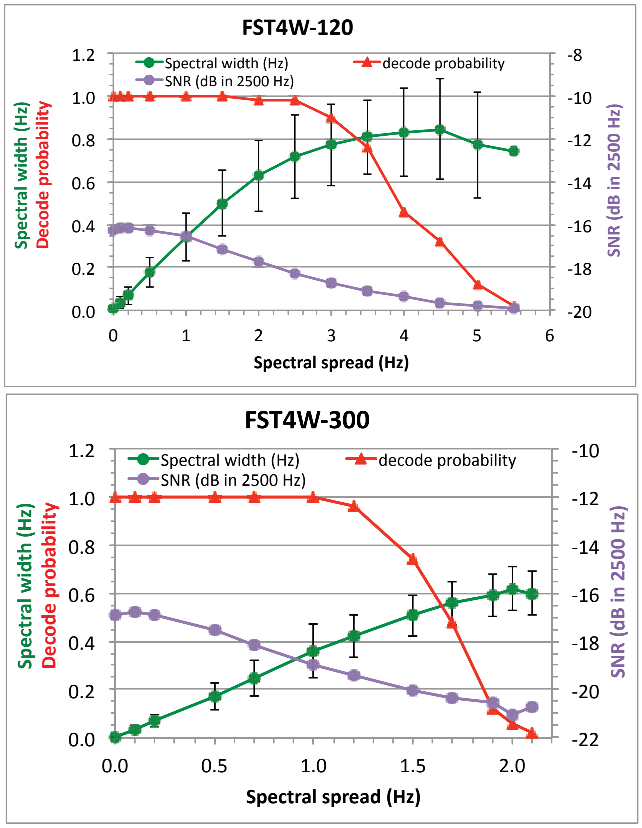


Figure 5. Variation of spectral width, decode probability and SNR with spectral spread from 50 runs of the fst4sim model at each value of Doppler spread for -120 upper, and -300 lower. The bars on spectral width are at +/-2.5 times the standard deviation, that is, 99% of the values lie within the interval.

above 95%. The bias is greater for -300 than for -120.

- The *jt9* output spectral width values show a linear trend with input spectral spread until the decode probability starts to drop. The output spectral width ends up being biased low as it is only those transmissions with (statistically) lower spectral widths that are decoded and therefore included in the measurements.

In summary, these graphs alert us to expect instances of SNR and spectral width biased low as the input spectral spread increases. They also help set expectations for probability of decode.

3. Equipment

While the ionosphere is probably the most variable contributor to spectral spread on the HF bands, as highlighted in the FST4 Quick-Start Guide[1], it may not necessarily be the largest. Spectral spread inevitably arises, to some degree, at a transmitter and receiver from oscillator drift and shorter-term jitter

It is not a trivial task to characterize the spectral spread of transmitters and receivers. A key enabler for this study was to use external GPSDO frequency control of four reference transmitters and receivers, namely:

- Transmitter: ANAN 'Angelia' SDR transceiver at N6GN, Fort Collins, Co.
- Remote receiver: KiwiSDR 21 km from N6GN.
- Transmitter: ANAN-10 at K6PZB, Graton, Ca.
- Receiver: KiwiSDR at WB7ABP/K, Santa Rosa, Ca., 10 km from K6PZB.

These systems, with master oscillators phase-locked to GPS, provided the basis for calculating the spectral spread of other KiwiSDRs using their standard GPS aiding algorithm. This algorithm is best described as intermittent correction rather than frequency-lock, let alone phase-lock.

As FST4W provides us with estimates of overall spectral spread from transmitter to receiver via the propagation path knowing the transmitter and receiver spectral spread estimates we can estimate the remaining spread due to the propagation path, e.g. for one- and two-hop F-layer propagation. Full details of these calculations are in the Annex while example equipment spectral spreads are described below.

3.1 Spectral spread of GPSDO transmitters and receivers.

We could not measure the spectral spreading of a GPSDO receiver or transmitter alone, only in combination over a 21 km surface wave path. The histogram in Figure 6 shows the distribution of spectral width output by *jt9* from 543 spots between 19–23 July 2022 over a completely line-of-sight path. The peak at 0.00625 Hz is almost double the audio baseband width of 0.0035 Hz.

As there were identical GPSDOs at each end, we will assume equal contributions to spreading. This positive-only, unimodal distribution with an extended right tail lends itself a Gamma distribution fit, the red line. From the Gamma distribution we can split equally into spectral width estimates for the GPSDO transmitter and receiver alone (see Annex for the methods used).

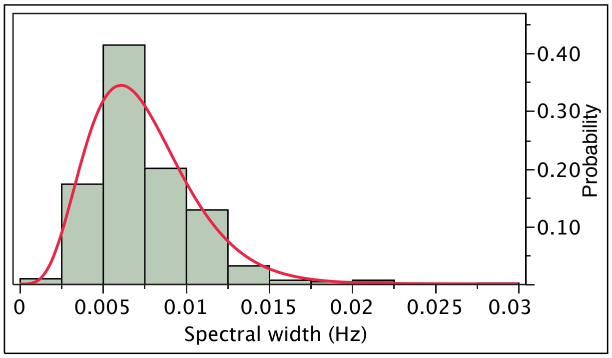


Figure 6. Histogram of the distribution of FST4W-300 spectral width determinations with GPSDO transmitter N6GN and GPSDO receiver N6GN/K over a 21 km line of sight path, from 543 spots 19–23 July 2022. In red is a fitted Gamma distribution, shape=5.62 and scale=0.00132.

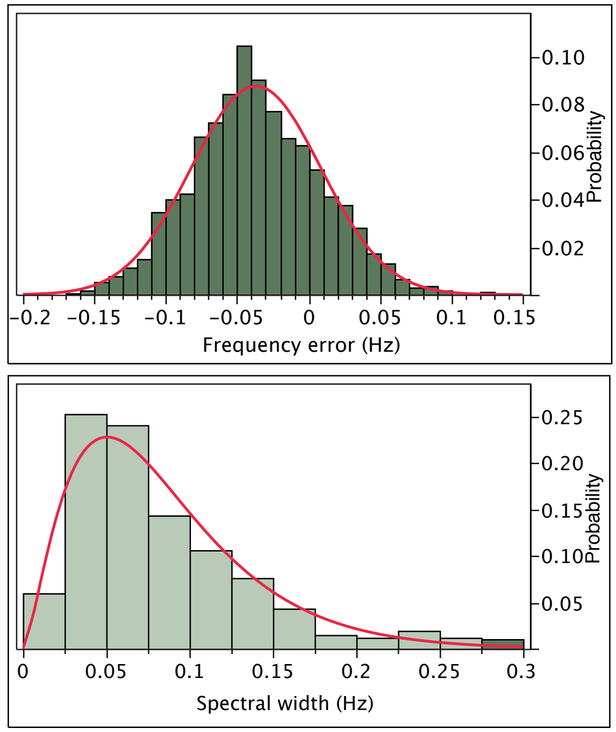


Figure 7. Two measures of short-term frequency variation in two KiwiSDRs. Upper: Histogram and fitted Gaussian distribution at G3ZIL of frequency error at 10 MHz at one-second intervals. Lower: Spectral width from jt9 at KPH from FST4W-120 GPSDO transmissions from K6PZB over a 38 km path.

3.2 Some issues when not using a GPSDO

In practice, many potential users of FST4W on the HF bands may not have access to GPSDOs. Depending on the characteristics of the hardware there may be other spectral spread mechanisms in equipment commonly used for WSPR, such as:

- A transceiver used for sending and receiving FST4W, even at 5 W output, may reach an elevated temperature such that at the end of the transmit cycle it will take many minutes for its (quite likely TCXO) master oscillator to settle to a sufficiently stable frequency for -300 and longer sequences.
- The innovative approach to mapping audio to RF frequency, emulation of the Gaussian slide between tones, and the tone spacing on transmit in the QRP Labs QDX digital modes transceiver[5] merits further evaluation. Although the receive side is well suited for FST4W to -900 the signal generated on transmit appears to be nowhere nearly as good at the time of writing (firmware 1.04).

Because of the ubiquity of KiwiSDRs with standard GPS aiding two spectral spread issues are summarised below followed by an assessment of the low-cost QDX with its TCXO.

3.2.1 KiwiSDR short-term frequency stability

While perfectly acceptable for WSPR and FST4W-120 the short-term frequency stability of the KiwiSDR with its standard GPS aiding is almost certainly a limitation for the longer FST4W sequences at least on the upper HF bands. Figure 7 shows two measures of short-term frequency variations.

The upper histogram is of frequency error at 10 MHz against a 10 MHz GPSDO from measurements one second apart using the frequency analysis tool in fldigi on a captured wav file. The red curve is a fitted Gaussian distribution with a mean error of -0.037 Hz and a standard deviation of 0.046 Hz.

The lower histogram is spectral width from a standard KiwiSDR at KPH, Point Reyes, Ca. of transmissions from a GPSDO transmitter at K6PZB 38 km distant for FST4W-120. While this includes the spectral spreading at the transmitter, from Figure 6 this would be less than 10% of the total. The median spectral spread is 0.068 Hz. This is of the same order as the standard deviation of one-second frequency errors measured at G3ZIL.

3.2.2 KiwiSDR browser connection abuf setting

If accessing a KiwiSDR via a browser reducing the buffer duration by specifying abuf=0.5 in the url, e.g. http://10.0.1.234:8073/?abuf=0.5 will reduce spectral spreading from that with the default value and reduce the time offset, Table 2. Those reductions will increase the margins for spectral spreading and DT for ionospheric paths and transmitters with time offsets. Note that setting a long buffer time is counterproductive; the result is a large spectral spread and a much reduced decode probability likely associated with the higher DT (section 2.2.1).

Parameter	**Default**	**0.5**	**1.0**	**1.75**
Median (Hz)	0.088	0.057	0.085	0.667
% SW/TS	6	4	6	46
% decode	100	100	100	23
DT (s)	1.6	1.1	1.5	1.9

Table 2. Median spectral width in Hz and as a fraction of the Tone Spacing together with the probability of decode and the time offset for FST4W-120 from a KiwiSDR via a browser for different values, in seconds, for the buffer length abuf.

3.2.3 QDX receive short-term stability

The QRP Labs QDX is a low cost digital modes transceiver with a TCXO. On receive it has an impressively low standard deviation of frequency error of 0.008 Hz at 10 MHz against a GPSDO and the -2.04 Hz mean error measured at G3ZIL could easily be accounted for in the user-set calibration. In a test at N6GN the measured spectral width receiving FST4W-300 from a GPSDO transmitter was 0.0198 Hz.

At G3ZIL the receive frequency shift after two cycles of transmitting FST4W-300 was an initial 0.75 Hz, returning toward the pre-transmit value with a time constant of about 230 s, but with a 0.2 Hz shift.

These are very encouraging figures on receive, however, the innovative transmitter does require further study.

4. Results

4.1 Main findings

The main findings of this study are summarised in Figure 8. Each column shows the spectral spread 'budget' out to where the *fst4sim* decode probability has dropped to 10% for each sequence length. Each column has three sources of spectral spread: transmitter, receiver and propagation path. The example transmitter and receiver spectral spreads have been derived from the measurements outlined in section 3.

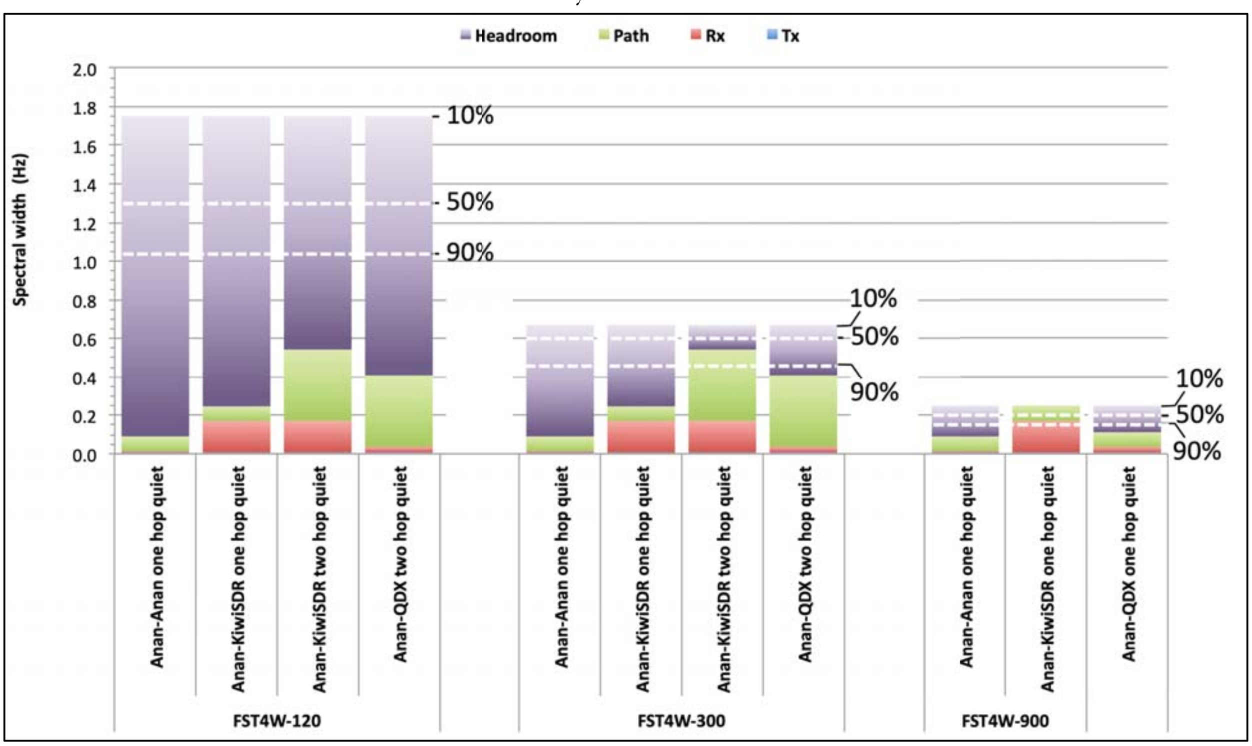


Figure 8. Spectral spread budgets for FST4W-120 to -900 for example combinations of transmitter, receiver and ionospheric propagation on the 20 m band. The upper limit for each column is the spectral spread at which there is only a 10% probability of decoding the signal. Also marked are the limits for 50% and 90% decode probability. While these lines remain in the purple 'headroom' area decode should be possible for quiet conditions (Kp <3).

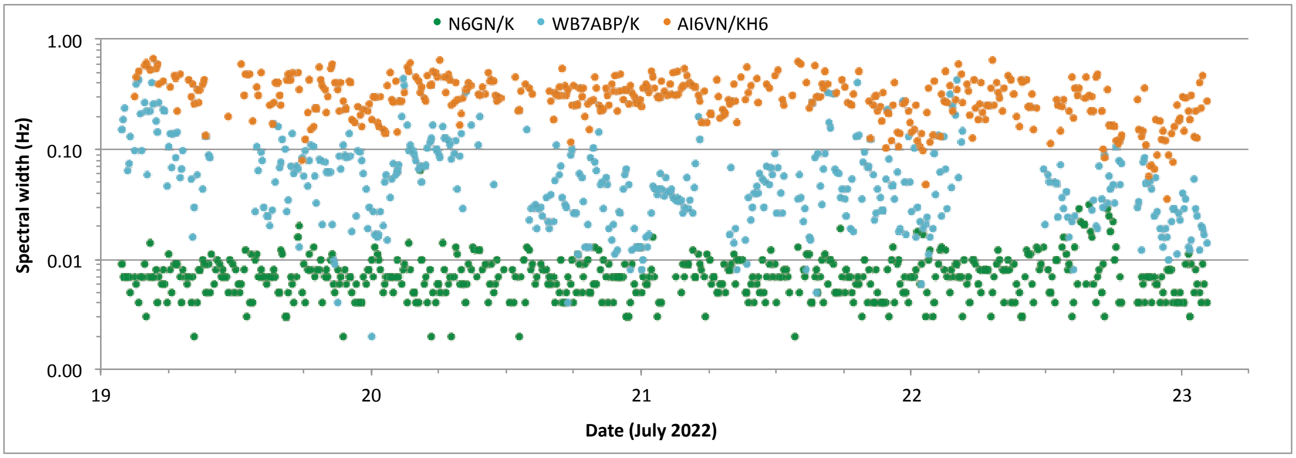


Figure 9. Time series of the spectral width on a log scale as output by jt9 from FST4W-300 transmissions by N6GN for three different propagation paths to: (green) N6GN/K a KiwiSDR with GPSDO at 21 km; (cyan) WB7ABP/K a KiwiSDR with GPSDO at 1558 km; (orange) AI6VN/KH6 a KiwiSDR with standard GPS aiding at 5291 km.

The spectral spreads are at the 90% level, that is, based on the Gamma function modelling, these values would only be exceeded 10% of the time. The propagation path spreads, for one- and two-hop, are from the measurements described later in this section. These were observed on 20 m and an assumption of acceptable extrapolation is made when considered for other bands

In all bar two of the examples (-300 and -900, ANAN and standard KiwiSDR two-hop) there is, to a greater or lesser extent, some headroom between the ionospheric component and the 90% probability level. That is, if the ionospheric spectral spread increases the decode probability will remain above 90%.

On 20 m for -300 with standard KiwiSDR and two-hop, even with a quiet ionosphere, the spectral spread means that the decode probability has already fallen to below 90%. For -900 with standard KiwiSDR and one-hop, the spread budget at 90% probability has already been used by the receiver (and a minor part by the transmitter). Few decodes would be expected. For this reason we have not considered the -1800 mode with half the spectral spread budget of -900. Even fewer decodes would be expected for the higher frequency bands.

From our measurements the QDX receiver with its standard TCXO would be suitable for receiving in -900 one-hop situations on its top band of 20 m. A KiwiSDR with an external TCXO or GPSDO would also be suitable for -900. Arising from this study the KiwiSDR designer has provided an option for updating the oscillator correction only during a gap in FST4W transmissions.

4.2 Spectral widths over contrasting paths

Figure 9 shows a time series of spectral width from *jt9* over four days in July 2022 for transmissions of FST4W-300 from N6GN. The paths were 20 km, 1558 km and 5291 km respectively. All three are at mid-latitudes. The log scale is necessary to show the two orders of magnitude difference between the 20 km and 5291 km paths.

Gaps in reception at WB7ABP/K were, with day-to-day variability, grouped around 0600–1200 UTC when the MUF over the propagation path was at its lowest. The reception pattern at AI6VN/KH6 was different, gaps were more scattered through each day. It is possible that the MUF remained sufficiently high for a range of 5291 km with two-hop propagation.

These spectral width time series include spreading contributions from the transmitter (ANAN with GPSDO, minimal), the receivers (KiwiSDRs with GPSDO at N6GN/K and WB7ABP/K, minimal), and with standard GPS aiding of a KiwiSDR at AI6VN/KH6. Using the method described in the Annex the estimated spectral spread from the ionospheric propagation paths alone were abstracted, with the results shown as model histograms in Figure 10. The spectral spread over the propagation paths from Colorado to California and to Maui are both skewed with a tail to the right, and the mode at the lowest bin. These are characteristics shared by the published[6] Doppler spread over a path from Svalbard

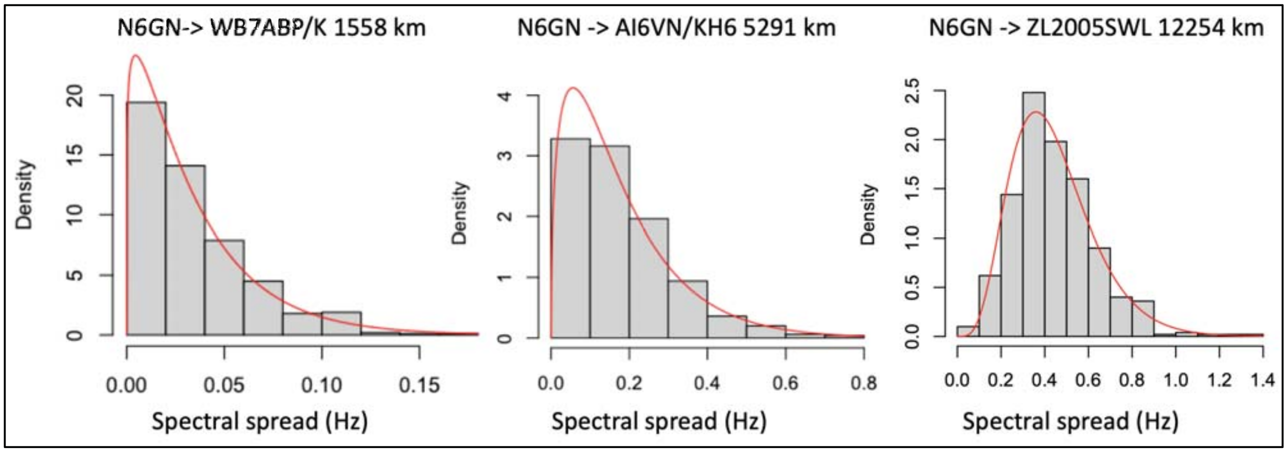


Figure 10. Histograms of spectral spread for the ionospheric path only, derived using the method in the Annex, for three paths from N6GN, Fort Collins, Co., to WB7ABP/K, Santa Rosa, Ca., AI6VN/KH6, Maui, Hi., and ZL2005SWL, North Island, New Zealand. To WB7ABP/K and AI6VN/KH6 transmissions were FST4W-300, and FST4W-120 to ZL2005SWL.

to southern Norway in April 1995 on 9.04 MHz - one of few examples available.

In contrast, the distribution for the 12,254 km trans-equatorial path to short-wave listener ZL2005SWL on the North Island of New Zealand is more symmetrical, with a mode well away from the lowest bin, approaching a Gaussian distribution.

4.3 Some interesting patters of spectral spread

At times, FST4W spectral width estimates appear to contain information related to the physical phenomena underlying the cause of the fluctuations. The WsprDaemon Grafana dashboard for FST4W allows rapid inspection of interesting parts of time series. Figure 11 is one example, from FST4W transmissions from N6GN to WB7ABP/K. There are times where successive, independent measurements, with low scatter, appear to show wave-like behaviour.

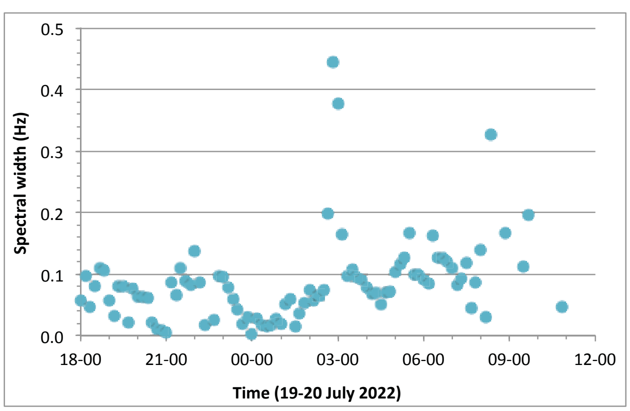


Figure 11. Time series of the spectral width of FST4W-300 transmissions from N6GN received at WB7ABP/K with suggestion of periodic variations, times with very low scatter, an event around 03:00 UTC and a slow rise toward the band closing.

There was also a steady increase from 00:00 UTC on 20 July, and, from 02:40 to 03:10 an event where four measurements were well above the slow, upward trend. As the MUF dropped, and the band approached closing, there were fewer decodes, the SNR had dropped, and spectral widths were high and scattered.

Finally, there remains a puzzle. The mode of spectral widths of FST4W-120 from the GPSDO transmitter at K6PZB at KPH, 38 km distant on 14 MHz was at 0.085 Hz. At KFS, Half Moon Bay, Ca., 124 km distant, over the example time series in Figure 12, there was just one K6PZB decode with a spectral width below 0.1 Hz. Yet the mode of the spectral width at KFS for FST4W-120 from N6GN 1535 km distant, via the ionosphere, was far lower at 0.06 Hz. What form of propagation was there between K6PZB and KFS to produce such high spectral spreading,

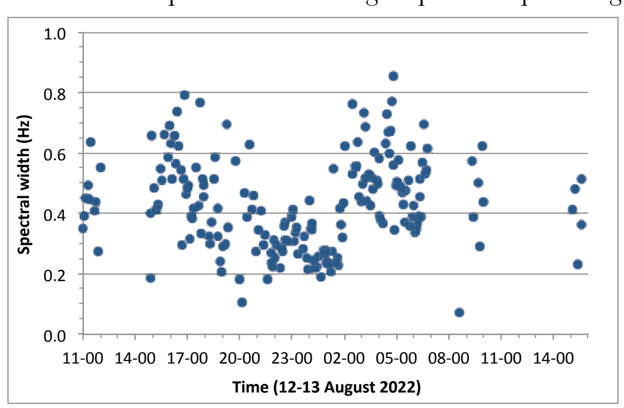


Figure 12. Time series of the spectral width of FST4W-120 transmissions from K6PZB received at KFS using its SW antenna. The separation is 124 km - why the large spectral widths? They were much greater than those from N6GN at 1535 km via the ionosphere at this time.

with SNR between -23 and -10 dB in 2500 Hz bandwidth, and with this temporal pattern? With only SNR, e.g. from WSPR, the question may well not have arisen. As it stands, there may be interesting aspects of surface wave HF propagation yet to be better understood.

4. Discussion and future work

It would seem a missed opportunity if FST4W were to be considered a protocol only suitable for the LF and HF bands. The -120 sequence length has no additional limitations over WSPR; instead it brings the minor benefit of a small increase in sensitivity and a large benefit in its measurement of spectral width. Not only can these measurements spur technical innovation in reducing the spectral spread within transmitters and receivers it can provide additional insight into propagation. It is important to note that sensitivity and spectral width are interlinked - near threshold sensitivities will only be achieved if the spectral spread is low. As we show, equipment-induced spectral spread can exceed that of the ionosphere on HF for one-hop paths.

We have outlined these advantages in this paper together with examples of use and several limitations. In particular, it is likely that the -1800 sequence length will only be of very specialised use on non-ionospheric paths or on lower the HF bands. Furthermore, the -900 sequence length may only be of practical value for one-hop ionospheric paths.

Users of KiwiSDRs and WsprDaemon (who currently report some 26% of all WSPR spots) are well placed to report FST4W transmissions following the WD3.0 upgrade by Rob Robinett. We encourage additional transmissions of FST4W-120 and -300 on the HF bands given this receiver base.

Further studies in the near-term will include:

- Evaluation of the new option on the KiwiSDR to have it alter its clock frequency only during non-data gaps in WSPR or FST4W transmissions.
- In-depth testing to identify the causes of, and potential remediation measures for, significant spectral spreading on transmit of the QDX digital transceiver.
- Adding to the fst4_decode.f90 module within *jt9* to output spectral width for candidate spots that were not decoded. This will help identify where excessive spectral width rather than low SNR was the cause of failure to decode, and by what margin the allowable spectral spread was exceeded.
- What is the propagation mechanism over the 124 km path between K6PZB and KFS and why the large spectral spread compared with a similar but shorter path?
- FST4W into and through the Auroral Oval, working with KiwiSDR installations at Inuvik, Northwest Territories, Canada, and VY0ERC, Eureka, Nunavut, Canada.
- Comparing SNR values for WSPR and FST4W as we have seen significant differences and we do not understand the use of a 12.5*Log10 (sequence length) adjustment in the fst4_decode.f90 program.

Acknowledgement

We are grateful to the Maritime Radio Historical Society (KPH) and Craig McCartney W6DRZ, Globe Wireless Radio Services and KFS Radio Club (KFS) for hosting the KiwiSDR systems, the supporting infrastructure, and the splendid antennas and to Chris Mackerell, ZL2005SWL for his spots from NZ.

Annex. Separating out contributions to spectral spreading from the transmitter, receiver and propagation path using Gamma functions.

This annex describes a practical, statistical method for separating out the spectral spreading from transmitter, receiver and propagation path. It does not attempt to separate out the contributions for a single transmission. The method is applicable where a number of decodes are available over a path and there is some knowledge of the equipment at each end. To begin, experiments over line-of-sight or at least surface wave paths of no more than a few tens of km are needed. The statistical analysis can be done with the R free software for statistical computing.

A Gamma function is an appropriate fit to the data sets observed to date, in that they are positive only, essentially unimodal and with a long right-hand tail. If a distribution is not unimodal there may be reasons, e.g. the data set spans distinctively different propagation conditions. In that case one should attempt to separate, perhaps into two or more time periods, and analyse separately.

Using the equipment and paths in this paper to illustrate the method, the key steps are:

1. Over a line-of-sight or surface wave path of a few 10s of km use a transmitter and receiver with GPDSO master oscillators with WsprDaemon or WSJT-X with empty file plotspec added to obtain spectral widths. Using R or another package plot spectral widths histogram, fit a Gamma distribution, and obtain two parameters: Shape and Scale (note that the mode is at Shape*Scale). Figure A1 (upper) is for N6GN to N6GN/K over a 21 km line of sight path, the data and a best-fit Gamma distribution with Shape=5.62, Scale=0.00132. Next, in R use those parameters to generate a set of random (spectral width) values conforming to that Gamma distribution, Figure A1 (lower), using:

```
synth_n6gn_total<-rgamma(n=500, shape=5.62, scale=0.00132)

fit_synth_n6gn_total<-fitdist(synth_n6gn_total, distr = "gamma", method = "mle")

summary(fit_synth_n6gn_total)      # print summary statistics including Shape and Scale

plot(fit_synth_n6gn_total)         # plot the distribution and Gamma fit
```

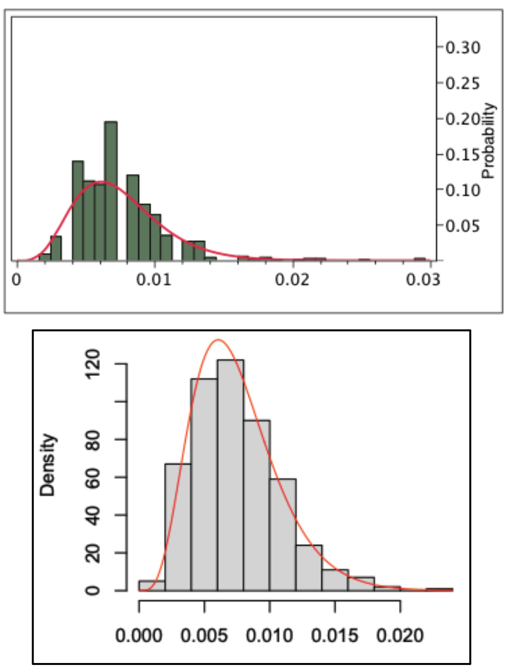


Figure A1. (upper) Histogram of actual spectral width values in Hz over a 21 km line-of-sight path from N6GN to N6GN/K with identical GPSDO master oscillators at the transmitter and receiver with a best-fit Gamma distribution. (lower) A synthesised histogram of a Gamma distribution of 500 values with the same Scale and Shape parameters as the observations.

2. Next, assume we can neglect any spectral spreading over this line-of-sight path. We further assume that the measured spectral spreading was equal at the transmitter and receiver; a reasonable assumption as the GPSDOs were identical. It is a property of Gamma distributions that if we halve the Shape parameter, keeping the same Scale parameter, and add the independent, individual random variables drawn from those two half-Shape distributions we end up with the original distribution. Therefore, we can synthesise individual GPSDO transmitter and receiver spectral spreads from:

```
n6gn_tx<-rgamma(n=500, shape=2.81, scale=0.00132)

n6gn_rx<-rgamma(n=500, shape=2.81, scale=0.00132)
```

These will be distributions of random spectral width values, that of the receiver shown in Figure A2 (upper). In Figure A2 (lower) we have added the individual values from the independent transmitter and receiver distributions and plotted using the commands in step 1. Note that individual columns differ slightly as different random values from distributions with the same parameters have been used. The Shape and Scale are also slightly different at 5.105 and 0.00146. Using more than 500 points would likely bring closer agreement.

```
synth_n6gn_tx_rx<-n6gn_tx+n6gn_rx
```

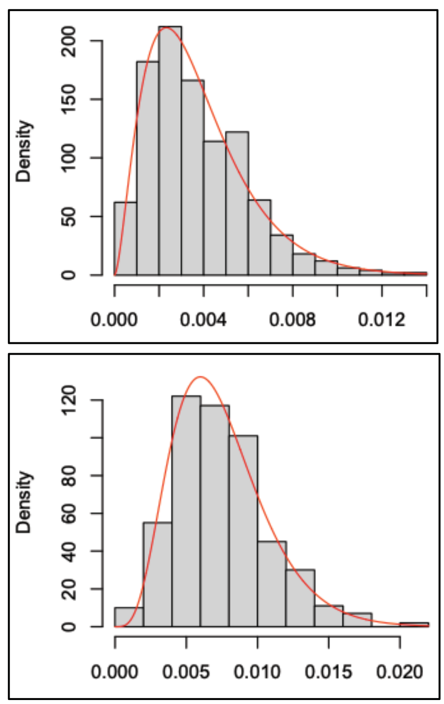


Figure A2. (upper) Histogram with fitted Gamma distribution for the N6GN/K GPSDO receiver only, formed by halving the Shape parameter from Figure A1 keeping the same Scale. (lower) Synthesised distribution for the transmitter and receiver to compare with Figure A1 (lower).

3. Next we take the measured spectral width distribution from the same transmitter, N6GN, to WB7ABP/K, with a GPSDO receiver, over a single-hop ionospheric path of 1558 km, Figure A3 (upper), fit a Gamma distribution

with Shape=1.878 and Scale=0.0215, and synthesise a distribution of 500 random values, Figure A3 (lower).

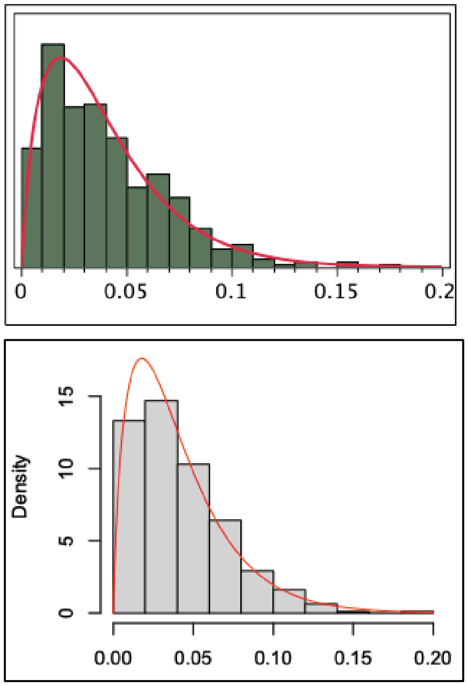


Figure A3. (upper) Histogram with fitted Gamma distribution for the total path from N6GN to WB7ABP/K including contributions from the transmitter and receiver. (lower) Synthesised distribution for the total path to compare with Figure A3 (upper).

4. To arrive at the propagation path only spectral spread we subtract the individual random values generated for n6gn_tx and n6gn_rx (as the equipment at WB7ABP was the same) from the individual random values forming the distribution in Figure A3 (lower). The absolute function is needed as the subtraction produces some small negative values. Next we fit a Gamma distribution to the path only set of values. The resulting distribution, Figure A4, has Shape=1.151 and Scale-0.0297.

```
path_n6gn_to_wb7abp<-abs(n6gn_at_wb7abp - n6gn_tx - n6gn_rx)
fit_path_n6gn_to_wb7abp<-fitdist(path_n6gn_to_wb7abp, distr = "gamma", method = "mle")
```

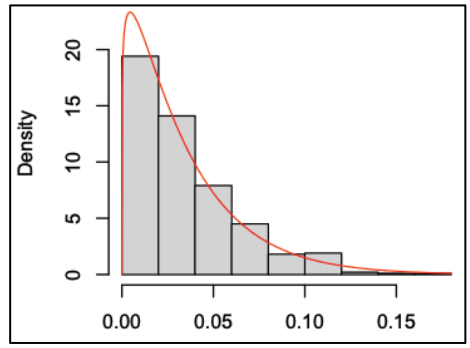


Figure A4. Synthesised distribution with fitted Gamma curve for the path-only spectral spreading from N6GN to WB7ABP/K.

5. From the Shape and the Scale parameters for the transmitter, receiver and path separately we calculate spectral spread values at a cumulative probability of 90% - that is, the value likely to not be exceeded 90% of the time. [This is easiest in Excel using gamma.inv (0.9, Shape, Scale)]. Those values are used as the individual component spectral spreads in Figure 8.
6. The same method is used for the other examples. The estimate for the KiwiSDR receiver with GPS aiding is made over the 38 km path from K6PZB to KPH knowing the K6PZB GPSDO transmitter spread. Knowing the KiwiSDR spread the method in step 4 is used for the two-hop path from N6GN to AI6VN/KH6, whose receiver is a standard GPS aided KiwiSDR.

---

[1] Franke, S., Somerville, B. and Taylor, J., Quick-Start Guide to FST4 and FST4W, available at https://physics.princeton.edu/pulsar/k1jt/FST4_Quick_Start.pdf checked August 2022

[2] Website at http://wsprdaemon.org/ code available at https://github.com/rrobinett/wsprdaemon and real-time list of WsprDaemon listeners with FST4W shown as F* is at http://wsprdaemon.org/fst4w

[3] Testing of HF Modems with Bandwidths of up to about 12 kHz Using Ionospheric Channel Simulators, Recommendation ITU-R F.1487, International Telecommunications Union, 2000.

[4] Watterson, C.C., Juroshek, J. and Bensema, W., 1970. Experimental confirmation of an HF channel model. *IEEE Transactions on communication technology*, *18*(6), pp.792-803.

[5] See https://qrp-labs.com/qdx.html

[6] Angling, M.J., Cannon, P.S., Davies, N.C., Lundborg, B., Jodalen, V. and Moreland, K.W., 1995, September. Measurements of Doppler spread on high latitude HF paths. In Proceedings of the AGARD Sensor and Propagation Panel Symposium on Digital Communications Systems: Propagation Effects, Technical Solutions, Systems Designs.
Available at https://apps.dtic.mil/sti/pdfs/ADA310824.pdf#page=135

VARAC

Walter Holmes K5WH

September 2022

1

Areas of Interest

- What is VarAC?
- Software required
- Hardware required
- Vara-HF settings
- VarAC settings
- Operational Choices
- Operational Choices while in QSO
- Other Features
- Further Assistance

2

What is VarAC?

- A newer digital mode that allows for keyboard conversations even in weak or low band conditions, typically over HF
- If you remember the old Packet Radio days of the 80's, you may find this similar
- The ability to chat, leave messages, transfer files, check signal reports, and see what stations others have heard recently

3

Software Required

- If you use WinLink on HF today, you already have half of the software now

- Vara-HF by EA5HVK
  - Vara v4.6.3 is the latest version
  - https://rosmodem.wordpress.com/

- VarAC by 4Z1AC
  - VarAC adds NEW features frequently
  - https://www.varac-hamradio.com/

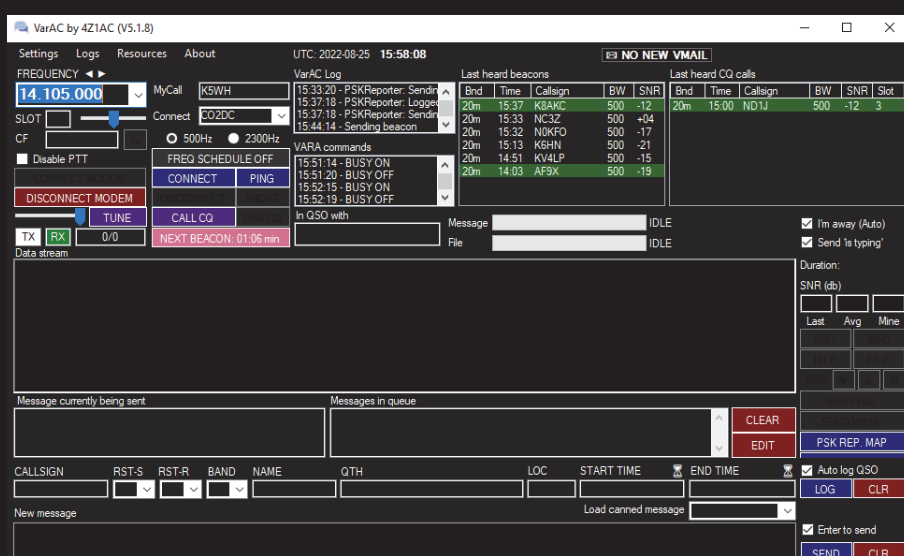
4

Hardware Required

- Like most digital mode software, VarAC requires a sound card digital interface
- Most current radios have this built in
- Legacy or older radios may require a SignaLink, RigBlaster, RigExpert, just to name a few, or any of the similar type interfaces

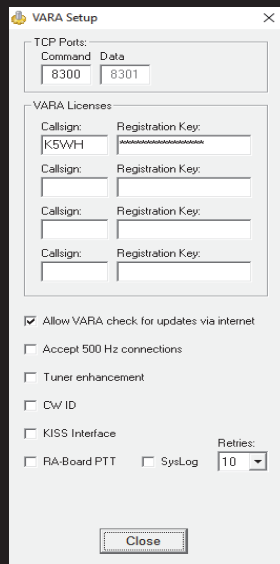
5

Install Vara-HF &VarAC, then launch VarAC



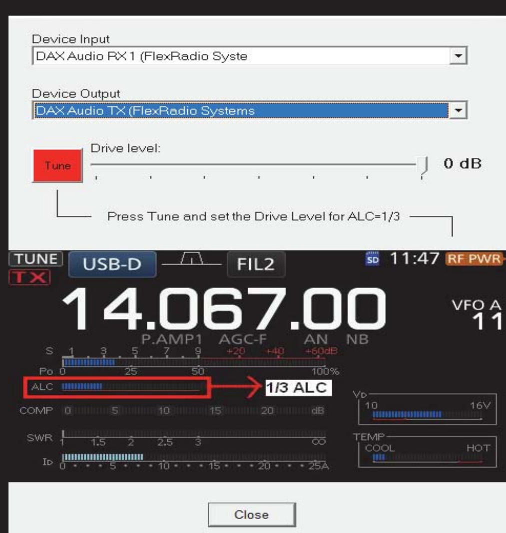
6

Vara-HF Settings, VARA Setup

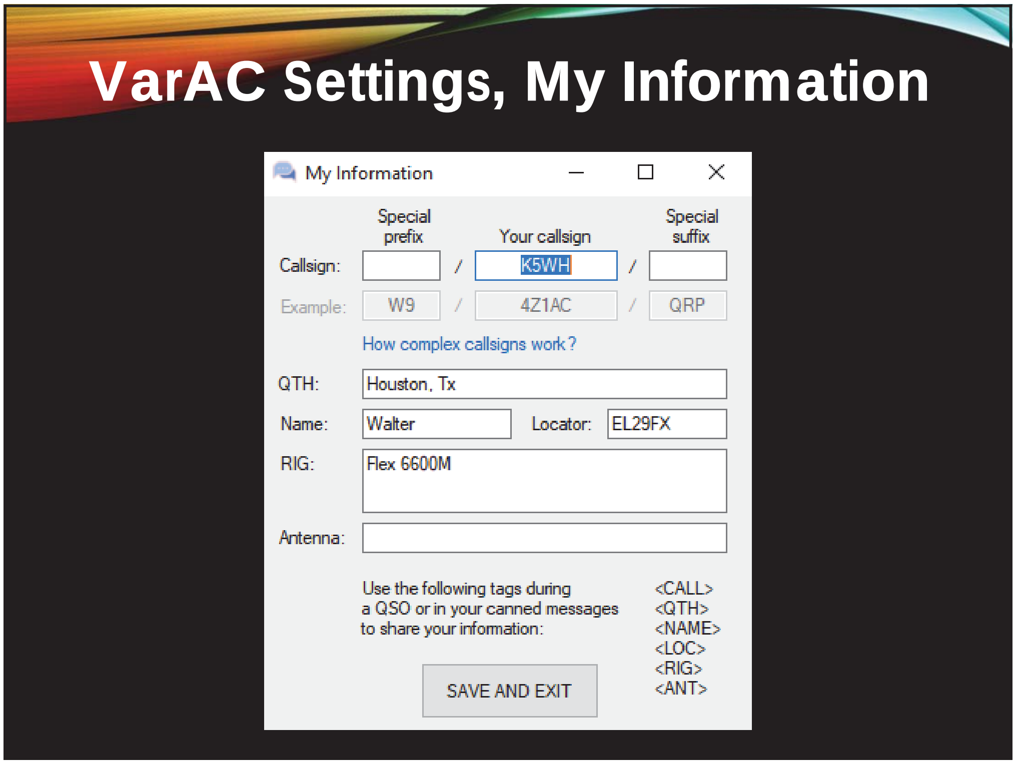


7

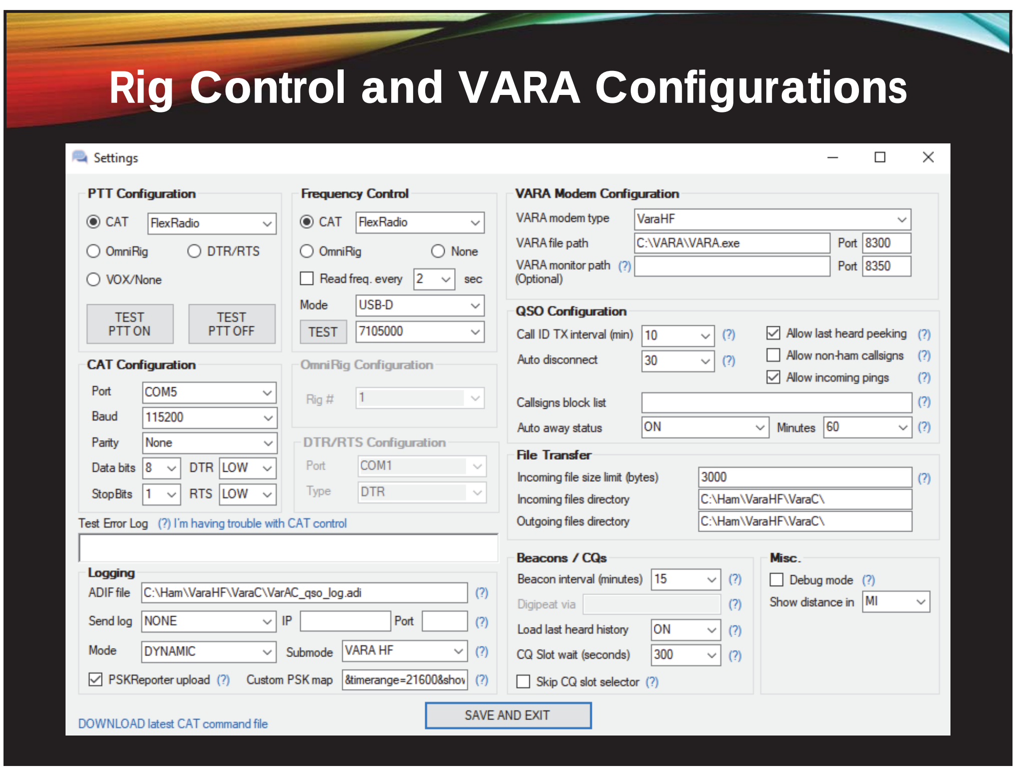
Vara-HF Settings, Soundcard



8


VarAC Settings, My Information
My Information
Special prefix
Your callsign
Special suffix
Callsign:
K5WH
Example:
W9
4Z1AC
QRP
How complex callsigns work?
QTH:
Houston, Tx
Name:
Walter
Locator:
EL29FX
RIG:
Flex 6600M
Antenna:
Use the following tags during a QSO or in your canned messages to share your information:
<CALL>
<QTH>
<NAME>
<LOC>
<RIG>
<ANT>
SAVE AND EXIT

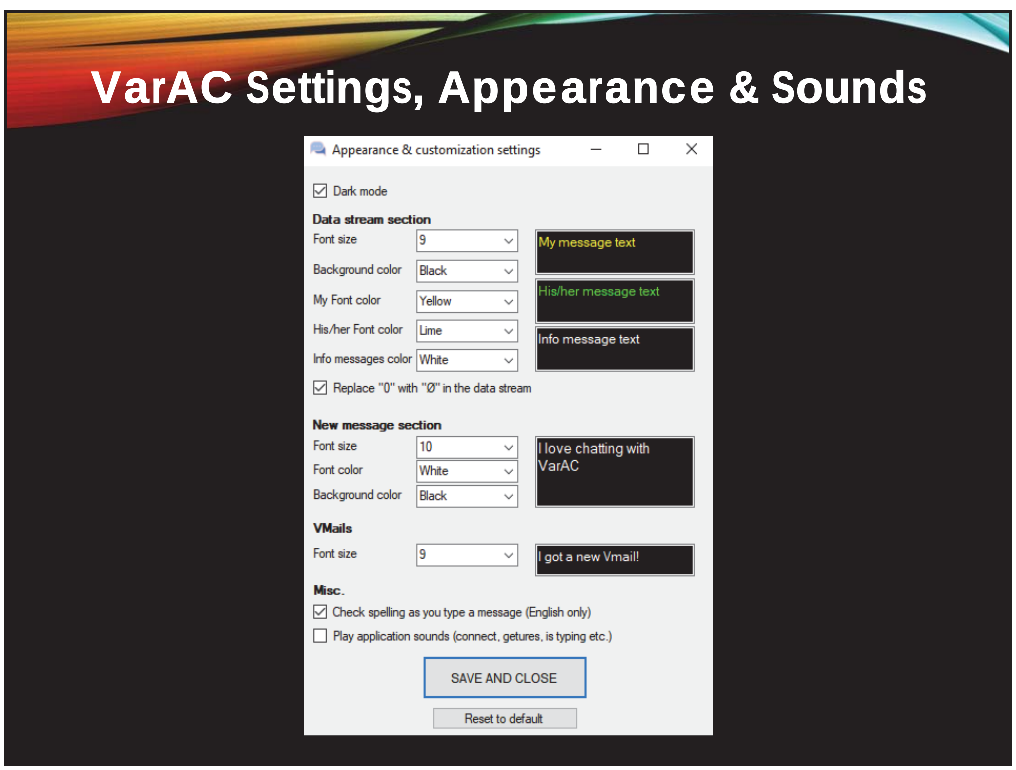
9


Rig Control and VARA Configurations
Settings
PTT Configuration
CAT
FlexRadio
OmniRig
DTR/RTS
VOX/None
TEST PTT ON
TEST PTT OFF
CAT Configuration
Port
COM5
Baud
115200
Parity
None
Data bits
8
DTR
LOW
StopBits
1
RTS
LOW
Test Error Log
(?) I'm having trouble with CAT control
Frequency Control
CAT
FlexRadio
OmniRig
None
Read freq. every
2
sec
Mode
USB-D
TEST
7105000
OmniRig Configuration
Rig #
1
DTR/RTS Configuration
Port
COM1
Type
DTR
Logging
ADIF file
C:\Ham\VaraHF\VaraC\VarAC_qso_log.adi
Send log
NONE
IP
Port
Mode
DYNAMIC
Submode
VARA HF
PSKReporter upload
Custom PSK map
&timerange=21600&show
VARA Modem Configuration
VARA modem type
VaraHF
VARA file path
C:\VARA\VARA.exe
Port
8300
VARA monitor path (Optional)
Port
8350
QSO Configuration
Call ID TX interval (min)
10
Auto disconnect
30
Allow last heard peeking
Allow non-ham callsigns
Allow incoming pings
Callsigns block list
Auto away status
ON
Minutes
60
File Transfer
Incoming file size limit (bytes)
3000
Incoming files directory
C:\Ham\VaraHF\VaraC\
Outgoing files directory
C:\Ham\VaraHF\VaraC\
Beacons / CQs
Beacon interval (minutes)
15
Digipeat via
Load last heard history
ON
CQ Slot wait (seconds)
300
Skip CQ slot selector
Misc.
Debug mode
Show distance in
MI
DOWNLOAD latest CAT command file
SAVE AND EXIT

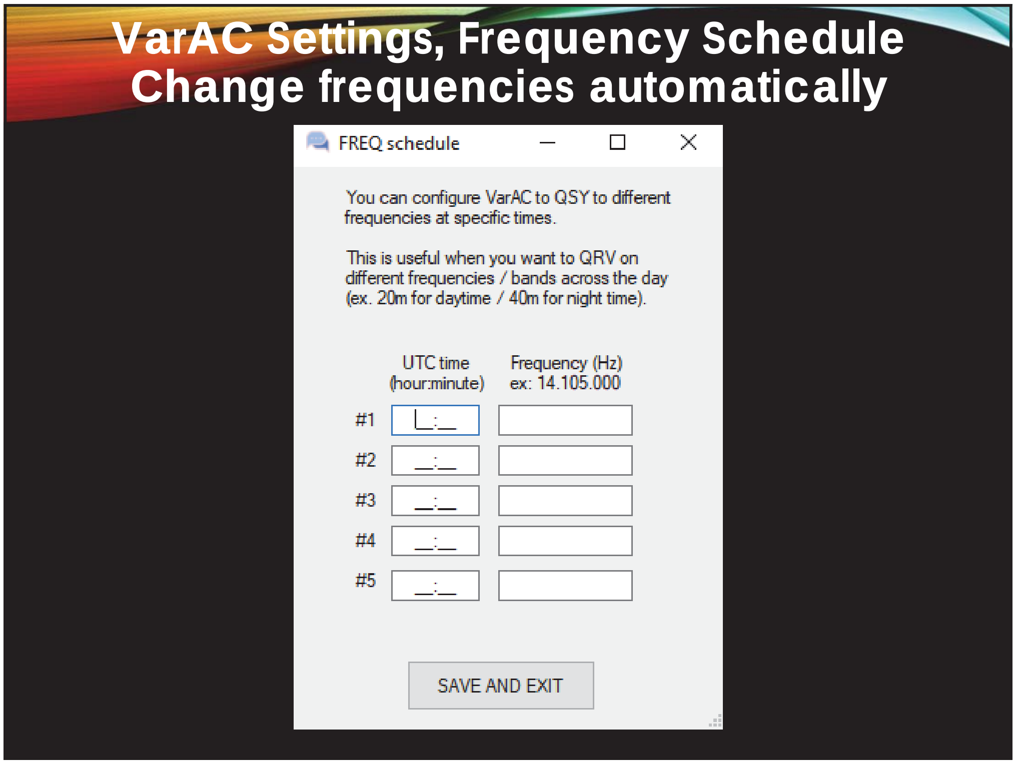
10


VarAC Settings, Canned Messages
Canned messages
Welcome message
Hello from Walter.
18/100
I'm Away message
I'm away. PSE leave a message. TU!
Actually, I'm probably here, so hope to catch you soon.
91/100
Tags
<SND> Send message
<DISC> Disconnect
<CALL> My Callsign
<NAME> My Name
<QTH> My QTH
<LOC> My Locator
<RIG> My RIG
<ANT> My Antenna
<HCALL> His/her Callsign
<HNAME> His/her Name
<HQTH> His/her QTH
<HLOC> His/her Locator
Canned message #1
<NAME:Walter>
<QTH:Houston, Texas>
<LOC:EL29fx><SND>
Name MY INFO
Shortcut key: F1
Canned message #2
My working conditions:
<RIG:Flex-6600M>
<ANT:Cushcraft A3, EndFed, GAP Vertical>
Name RIG
Shortcut key: F2
Canned message #3
73's and all the best!
DE <CALL> Disconnecting...<SND><DISC>
Name 73s
Shortcut key: F3
Canned message #4
VarAC QSO
Name QSO Truck
Shortcut key: F4
Canned message #5
Name Thumb
Shortcut key: F5
Canned message #6
Join me on Zoom at http://www.k5wh.net/zoom/
Name ZOOM Link
Shortcut key: F6
Canned message #7
Name
Shortcut key: F7
Canned message #8
Name
Shortcut key: F8
SAVE AND CLOSE

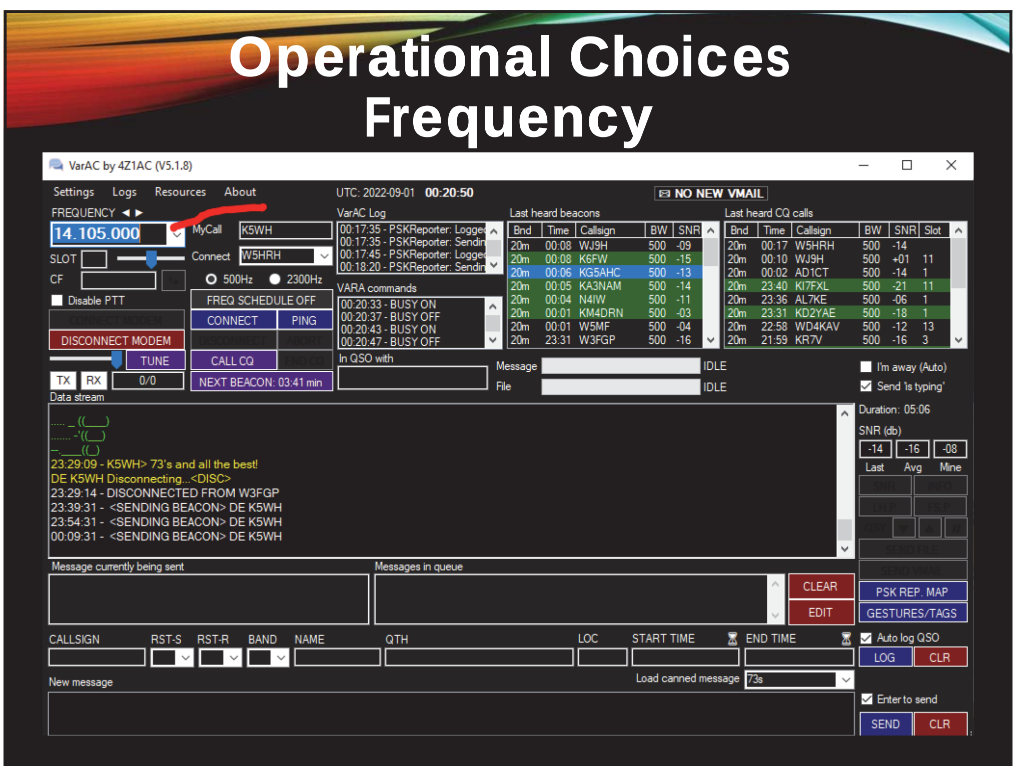
11


VarAC Settings, Appearance & Sounds
Appearance & customization settings
Dark mode
Data stream section
Font size 9
Background color Black
My Font color Yellow
His/her Font color Lime
Info messages color White
My message text
His/her message text
Info message text
Replace "0" with "Ø" in the data stream
New message section
Font size 10
Font color White
Background color Black
I love chatting with VarAC
VMails
Font size 9
I got a new Vmail!
Misc.
Check spelling as you type a message (English only)
Play application sounds (connect, getures, is typing etc.)
SAVE AND CLOSE
Reset to default

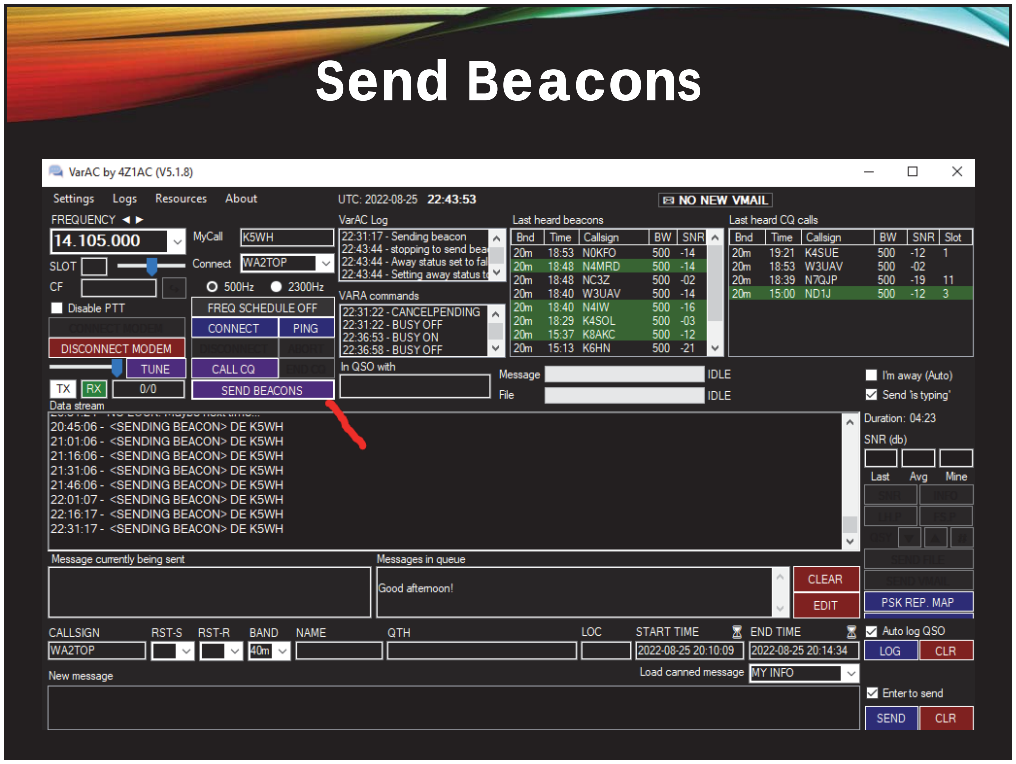
12


VarAC Settings, Frequency Schedule
Change frequencies automatically
FREQ schedule
You can configure VarAC to QSY to different frequencies at specific times.
This is useful when you want to QRV on different frequencies / bands across the day (ex. 20m for daytime / 40m for night time).
UTC time (hour:minute)
Frequency (Hz) ex: 14.105.000
#1
#2
#3
#4
#5
SAVE AND EXIT

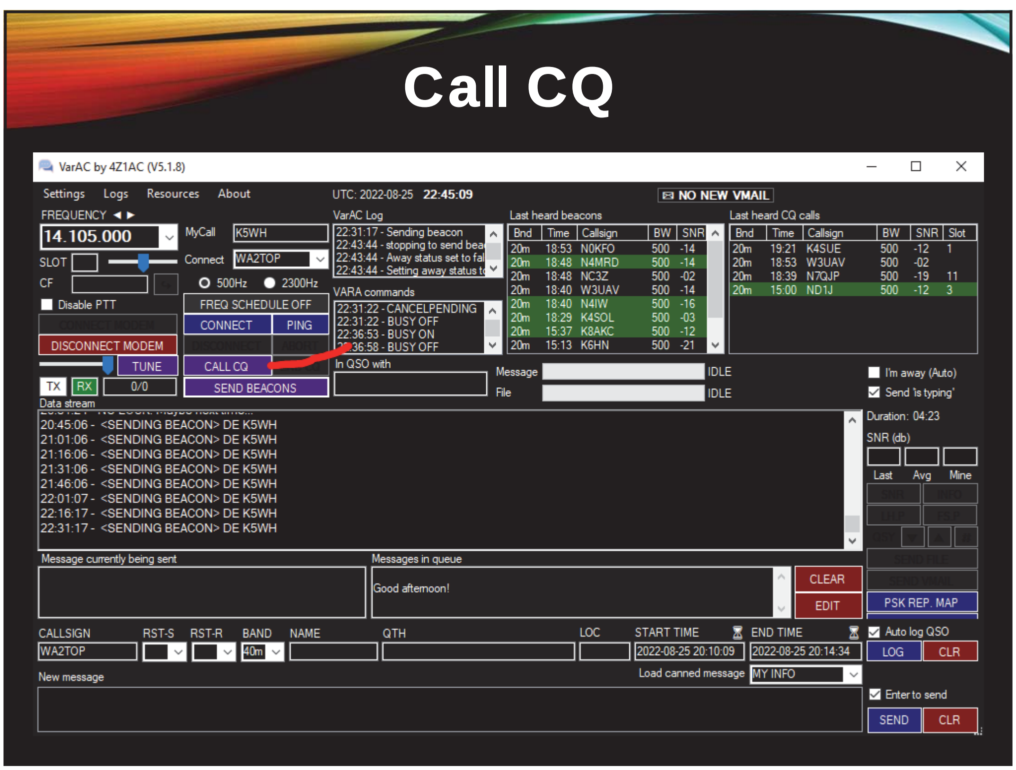
13


Operational Choices
Frequency
VarAC by 4Z1AC (V5.1.8)
Settings
Logs
Resources
About
UTC: 2022-09-01 00:20:50
NO NEW VMAIL
FREQUENCY
14.105.000
MyCall K5WH
Connect W5HRH
SLOT
CF
500Hz
2300Hz
Disable PTT
DISCONNECT MODEM
TUNE
TX
RX
0/0
Data stream
FREQ SCHEDULE OFF
CONNECT
PING
CALL CQ
NEXT BEACON: 03:41 min
VarAC Log
VARA commands
In QSO with
Last heard beacons
Last heard CQ calls
Message
File
IDLE
I'm away (Auto)
Send 'is typing'
Duration: 05:06
SNR (db)
23:29:09 - K5WH> 73's and all the best!
DE K5WH Disconnecting...<DISC>
23:29:14 - DISCONNECTED FROM W3FGP
23:39:31 - <SENDING BEACON> DE K5WH
23:54:31 - <SENDING BEACON> DE K5WH
00:09:31 - <SENDING BEACON> DE K5WH
Message currently being sent
Messages in queue
CLEAR
EDIT
PSK REP. MAP
GESTURES/TAGS
CALLSIGN
RST-S
RST-R
BAND
NAME
QTH
LOC
START TIME
END TIME
Auto log QSO
LOG
CLR
New message
Load canned message
Enter to send
SEND
CLR

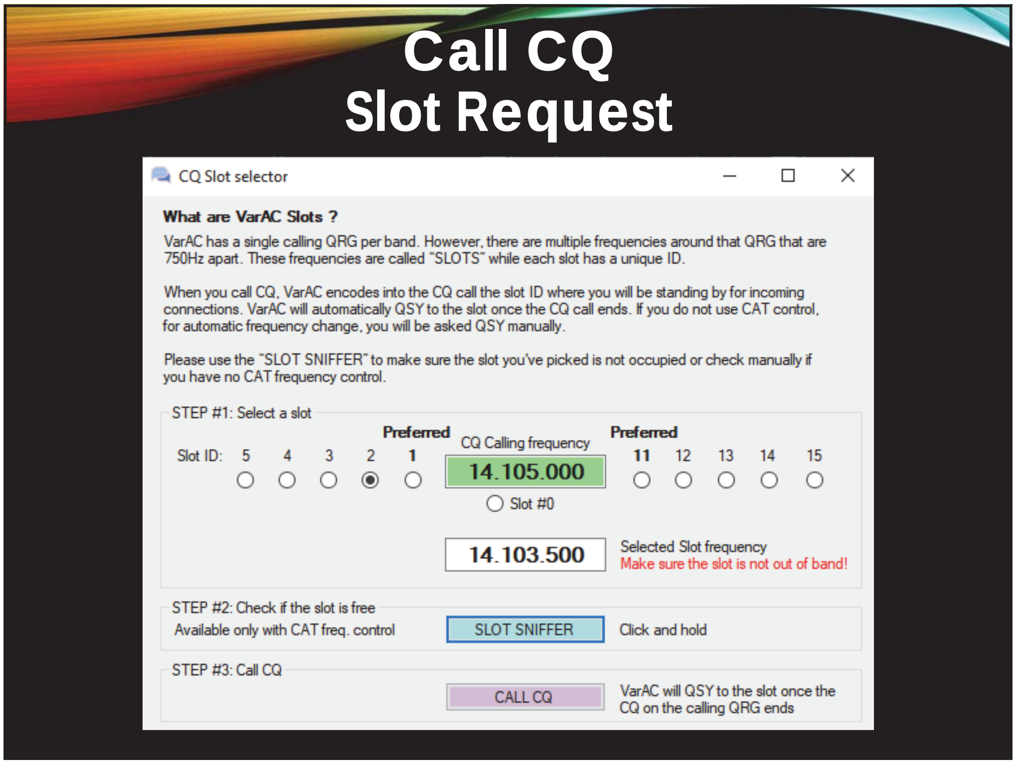
14


Send Beacons
VarAC by 4Z1AC (V5.1.8)
Settings Logs Resources About
UTC: 2022-08-25 22:43:53
NO NEW VMAIL
FREQUENCY
14.105.000
MyCall K5WH
Connect WA2TOP
SLOT
CF
500Hz 2300Hz
Disable PTT
FREQ SCHEDULE OFF
CONNECT
PING
DISCONNECT MODEM
TUNE
CALL CQ
SEND BEACONS
TX RX 0/0
VarAC Log
22:31:17 - Sending beacon
22:43:44 - stopping to send bea
22:43:44 - Away status set to fal
22:43:44 - Setting away status t
VARA commands
22:31:22 - CANCELPENDING
22:31:22 - BUSY OFF
22:36:53 - BUSY ON
22:36:58 - BUSY OFF
In QSO with
Last heard beacons
Last heard CQ calls
Message IDLE
File IDLE
I'm away (Auto)
Send 'is typing'
Data stream
20:45:06 - <SENDING BEACON> DE K5WH
21:01:06 - <SENDING BEACON> DE K5WH
21:16:06 - <SENDING BEACON> DE K5WH
21:31:06 - <SENDING BEACON> DE K5WH
21:46:06 - <SENDING BEACON> DE K5WH
22:01:07 - <SENDING BEACON> DE K5WH
22:16:17 - <SENDING BEACON> DE K5WH
22:31:17 - <SENDING BEACON> DE K5WH
Duration: 04:23
SNR (db)
Last Avg Mine
Message currently being sent
Messages in queue
Good afternoon!
CLEAR
EDIT
PSK REP. MAP
CALLSIGN RST-S RST-R BAND NAME QTH LOC START TIME END TIME
WA2TOP 40m 2022-08-25 20:10:09 2022-08-25 20:14:34
Auto log QSO
LOG CLR
New message
Load canned message MY INFO
Enter to send
SEND CLR

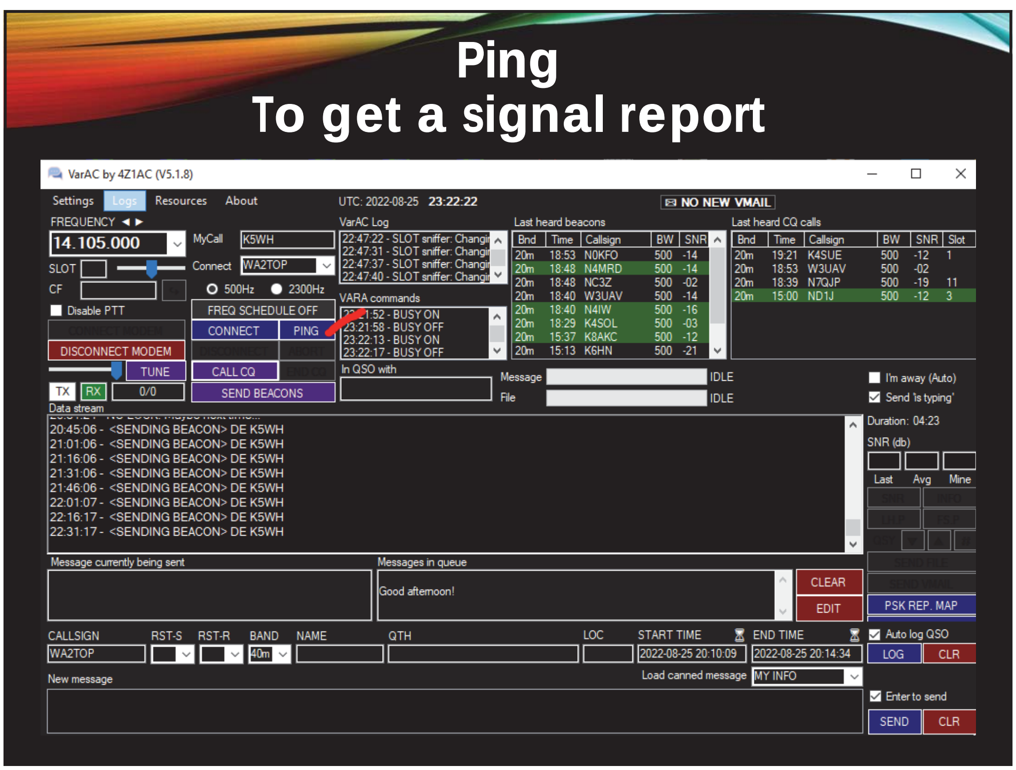
15


Call CQ
VarAC by 4Z1AC (V5.1.8)
Settings Logs Resources About
UTC: 2022-08-25 22:45:09
NO NEW VMAIL
FREQUENCY
14.105.000
MyCall K5WH
Connect WA2TOP
FREQ SCHEDULE OFF
CONNECT
PING
DISCONNECT MODEM
TUNE
CALL CQ
SEND BEACONS
Data stream
20:45:06 - <SENDING BEACON> DE K5WH
21:01:06 - <SENDING BEACON> DE K5WH
21:16:06 - <SENDING BEACON> DE K5WH
21:31:06 - <SENDING BEACON> DE K5WH
21:46:06 - <SENDING BEACON> DE K5WH
22:01:07 - <SENDING BEACON> DE K5WH
22:16:17 - <SENDING BEACON> DE K5WH
22:31:17 - <SENDING BEACON> DE K5WH
Message currently being sent
Messages in queue
Good afternoon!
CLEAR
EDIT
PSK REP. MAP
New message
SEND CLR

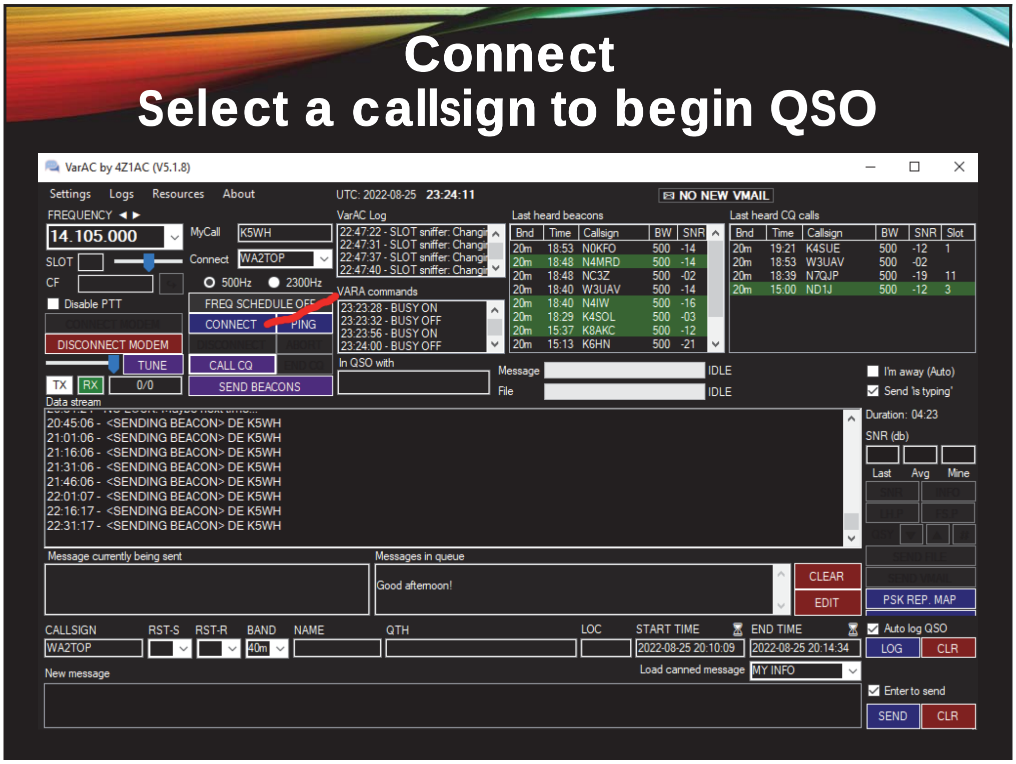
16


Call CQ
Slot Request
CQ Slot selector
What are VarAC Slots ?
VarAC has a single calling QRG per band. However, there are multiple frequencies around that QRG that are 750Hz apart. These frequencies are called "SLOTS" while each slot has a unique ID.
When you call CQ, VarAC encodes into the CQ call the slot ID where you will be standing by for incoming connections. VarAC will automatically QSY to the slot once the CQ call ends. If you do not use CAT control, for automatic frequency change, you will be asked QSY manually.
Please use the "SLOT SNIFFER" to make sure the slot you've picked is not occupied or check manually if you have no CAT frequency control.
STEP #1: Select a slot
Preferred
CQ Calling frequency
Preferred
Slot ID: 5 4 3 2 1
14.105.000
11 12 13 14 15
Slot #0
14.103.500
Selected Slot frequency
Make sure the slot is not out of band!
STEP #2: Check if the slot is free
Available only with CAT freq. control
SLOT SNIFFER
Click and hold
STEP #3: Call CQ
CALL CQ
VarAC will QSY to the slot once the CQ on the calling QRG ends

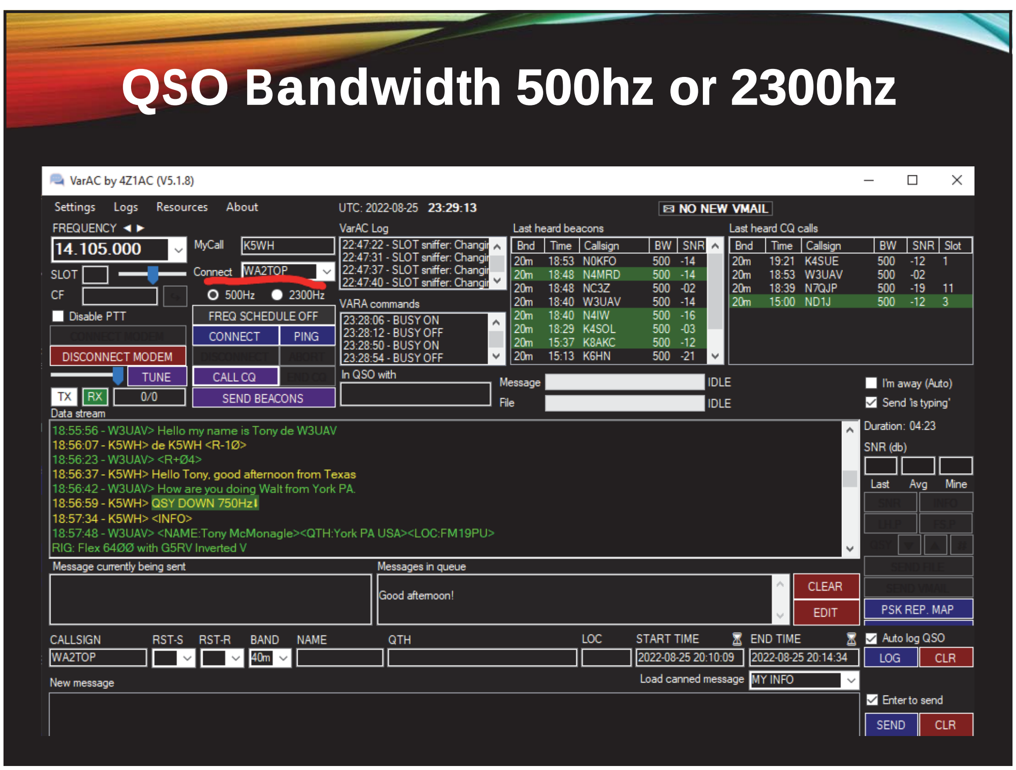
17


Ping
To get a signal report
VarAC by 4Z1AC (V5.1.8)
Settings Logs Resources About
UTC: 2022-08-25 23:22:22
NO NEW VMAIL
FREQUENCY
14.105.000
MyCall K5WH
Connect WA2TOP
VarAC Log
Last heard beacons
Last heard CQ calls
FREQ SCHEDULE OFF
CONNECT
PING
DISCONNECT MODEM
TUNE
CALL CQ
SEND BEACONS
VARA commands
In QSO with
Message IDLE
File IDLE
Data stream
20:45:06 - <SENDING BEACON> DE K5WH
21:01:06 - <SENDING BEACON> DE K5WH
21:16:06 - <SENDING BEACON> DE K5WH
21:31:06 - <SENDING BEACON> DE K5WH
21:46:06 - <SENDING BEACON> DE K5WH
22:01:07 - <SENDING BEACON> DE K5WH
22:16:17 - <SENDING BEACON> DE K5WH
22:31:17 - <SENDING BEACON> DE K5WH
Message currently being sent
Messages in queue
Good afternoon!
CLEAR
EDIT
PSK REP. MAP
CALLSIGN RST-S RST-R BAND NAME QTH LOC START TIME END TIME
WA2TOP
2022-08-25 20:10:09
2022-08-25 20:14:34
Auto log QSO
LOG CLR
Load canned message MY INFO
New message
Enter to send
SEND CLR

18

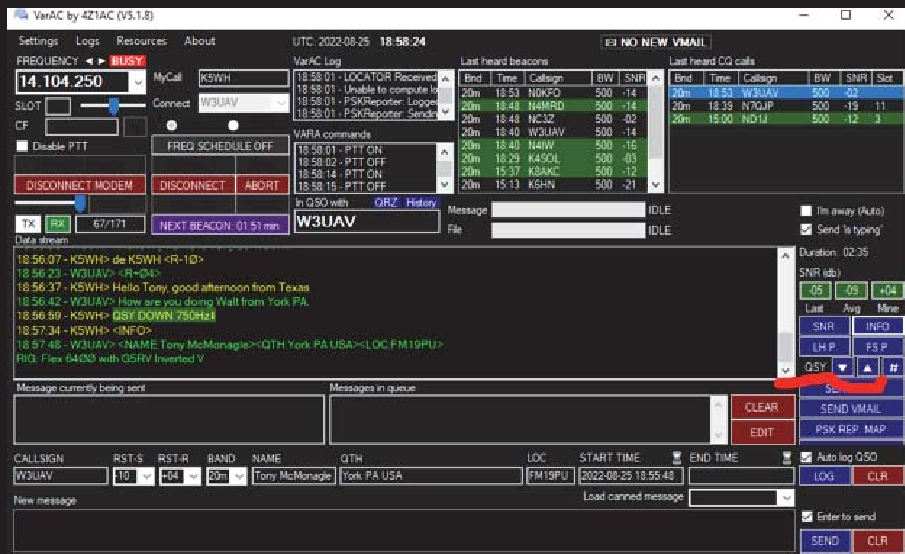

Connect
Select a callsign to begin QSO
VarAC by 4Z1AC (V5.1.8)
UTC: 2022-08-25 23:24:11
NO NEW VMAIL
14.105.000
MyCall K5WH
Connect WA2TOP
500Hz 2300Hz
FREQ SCHEDULE OFF
CONNECT PING
DISCONNECT MODEM
TUNE
CALL CQ
SEND BEACONS
20:45:06 - <SENDING BEACON> DE K5WH
21:01:06 - <SENDING BEACON> DE K5WH
21:16:06 - <SENDING BEACON> DE K5WH
21:31:06 - <SENDING BEACON> DE K5WH
21:46:06 - <SENDING BEACON> DE K5WH
22:01:07 - <SENDING BEACON> DE K5WH
22:16:17 - <SENDING BEACON> DE K5WH
22:31:17 - <SENDING BEACON> DE K5WH
Good afternoon!
CLEAR
EDIT
PSK REP. MAP
LOG CLR
SEND CLR

19


QSO Bandwidth 500hz or 2300hz
VarAC by 4Z1AC (V5.1.8)
UTC: 2022-08-25 23:29:13
NO NEW VMAIL
14.105.000
MyCall K5WH
Connect WA2TOP
500Hz 2300Hz
FREQ SCHEDULE OFF
CONNECT PING
DISCONNECT MODEM
TUNE
CALL CQ
SEND BEACONS
18:55:56 - W3UAV> Hello my name is Tony de W3UAV
18:56:07 - K5WH> de K5WH <R-1Ø>
18:56:23 - W3UAV> <R+Ø4>
18:56:37 - K5WH> Hello Tony, good afternoon from Texas
18:56:42 - W3UAV> How are you doing Walt from York PA.
18:56:59 - K5WH> QSY DOWN 750Hz
18:57:34 - K5WH> <INFO>
18:57:48 - W3UAV> <NAME:Tony McMonagle><QTH:York PA USA><LOC:FM19PU>
RIG: Flex 64ØØ with G5RV Inverted V
Good afternoon!
CLEAR
EDIT
PSK REP. MAP
LOG CLR
SEND CLR

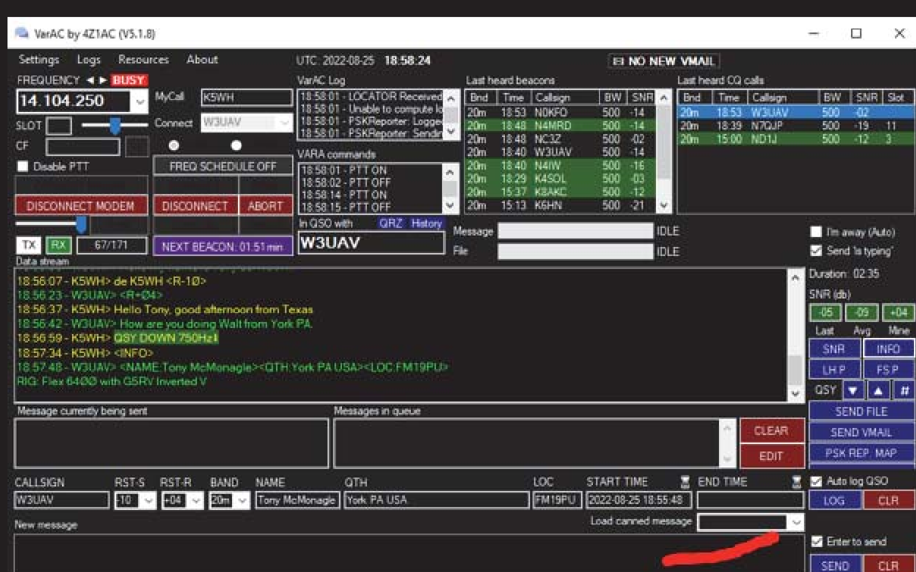
20

Operational Choices while in QSO QSY Down or UP, off calling frequency

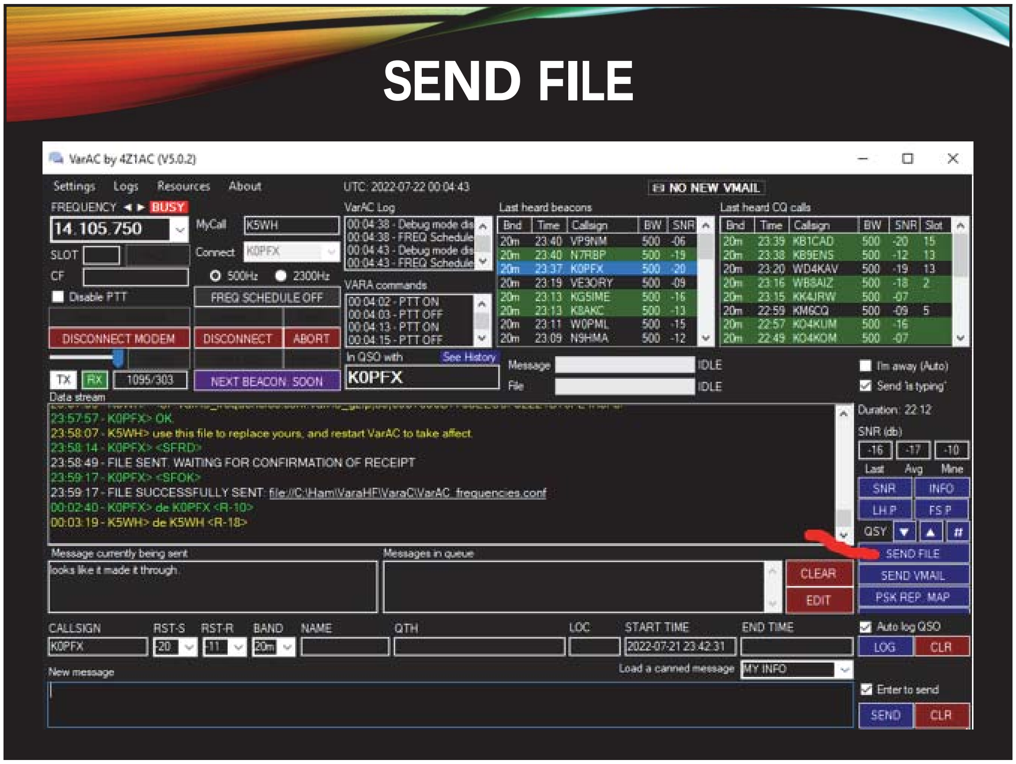


21

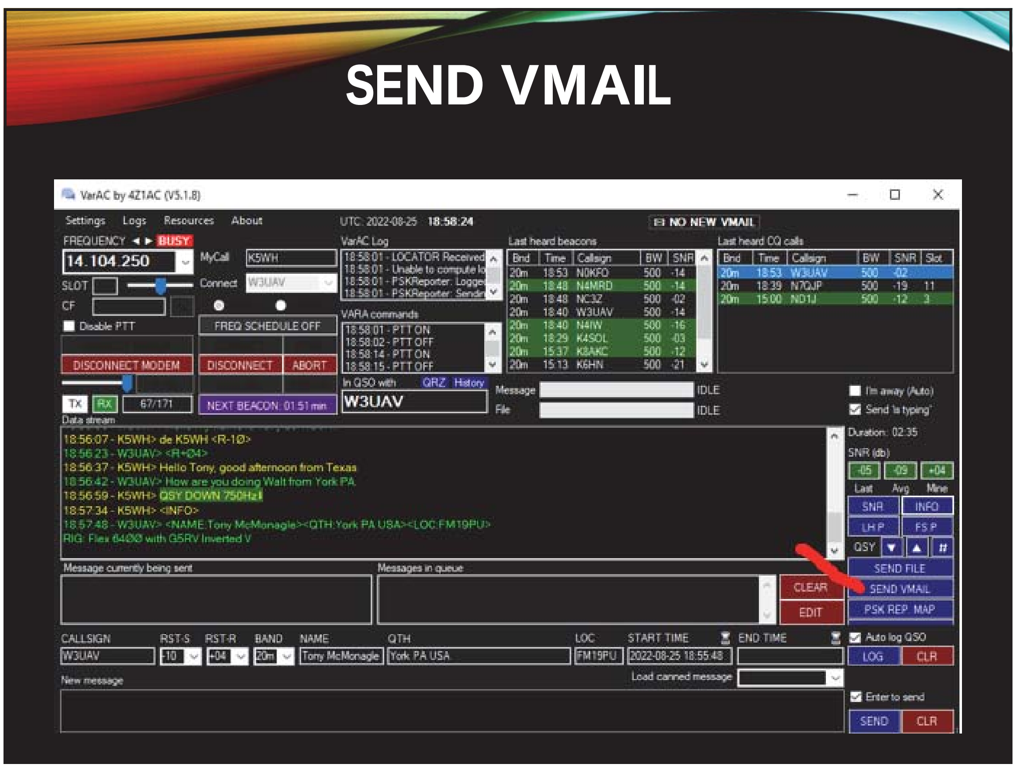
Load canned message



22


SEND FILE
VarAC by 4Z1AC (V5.0.2)
Settings Logs Resources About
UTC: 2022-07-22 00:04:43
NO NEW VMAIL
FREQUENCY BUSY
14.105.750
MyCall K5WH
Connect K0PFX
DISCONNECT MODEM
DISCONNECT
ABORT
FREQ SCHEDULE OFF
NEXT BEACON: SOON
In QSO with
K0PFX
23:57:57 - K0PFX> OK
23:58:07 - K5WH> use this file to replace yours, and restart VarAC to take affect.
23:58:14 - K0PFX> <SFRD>
23:58:49 - FILE SENT. WAITING FOR CONFIRMATION OF RECEIPT
23:59:17 - K0PFX> <SFOK>
23:59:17 - FILE SUCCESSFULLY SENT: file://C:\Ham\VaraHF\VaraC\VarAC_frequencies.conf
00:02:40 - K0PFX> de K0PFX <R-10>
00:03:19 - K5WH> de K5WH <R-18>
Message currently being sent
looks like it made it through
Messages in queue
CLEAR
EDIT
SEND FILE
SEND VMAIL
PSK REP MAP
CALLSIGN RST-S RST-R BAND NAME QTH LOC START TIME END TIME
K0PFX -20 -11 20m 2022-07-21 23:42:31
New message
SEND CLR

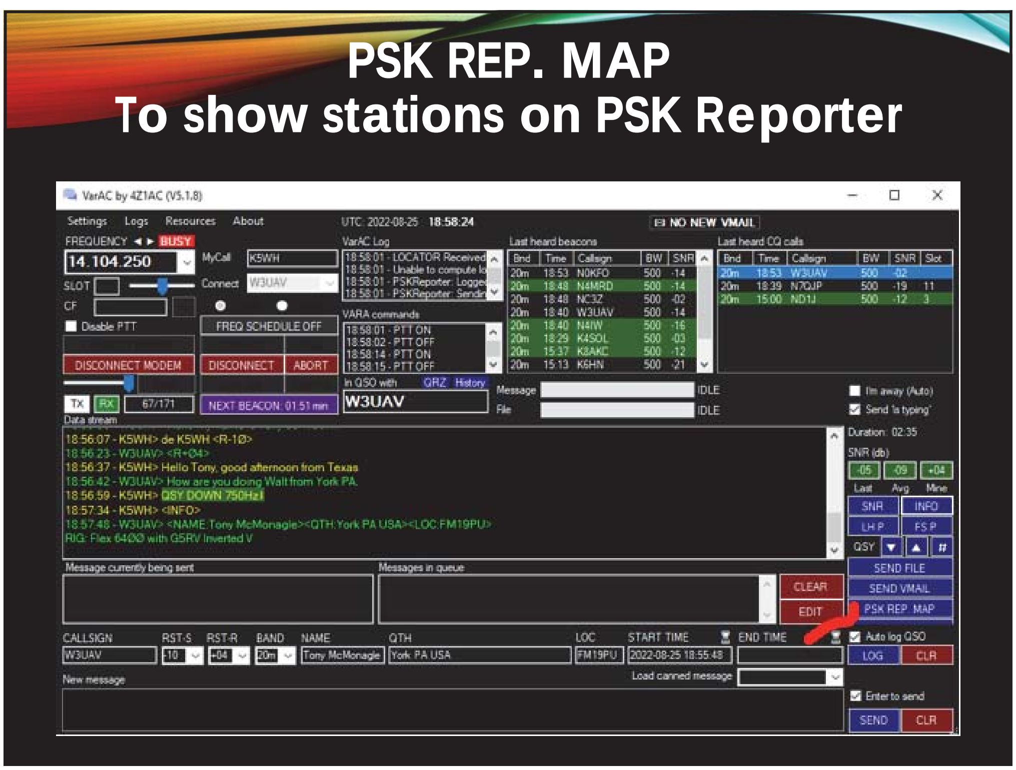
23


SEND VMAIL
VarAC by 4Z1AC (V5.1.8)
Settings Logs Resources About
UTC: 2022-08-25 18:58:24
NO NEW VMAIL
FREQUENCY BUSY
14.104.250
MyCall K5WH
Connect W3UAV
DISCONNECT MODEM
DISCONNECT
ABORT
FREQ SCHEDULE OFF
NEXT BEACON: 01:51 min
In QSO with
W3UAV
18:56:07 - K5WH> de K5WH <R-10>
18:56:23 - W3UAV> <R+04>
18:56:37 - K5WH> Hello Tony, good afternoon from Texas
18:56:42 - W3UAV> How are you doing Walt from York PA
18:56:59 - K5WH> QSY DOWN 750Hz!
18:57:34 - K5WH> <INFO>
18:57:48 - W3UAV> <NAME:Tony McMonagle><QTH:York PA USA><LOC:FM19PU>
RIG: Flex 6400 with G5RV Inverted V
Message currently being sent
Messages in queue
CLEAR
EDIT
SEND FILE
SEND VMAIL
PSK REP MAP
CALLSIGN RST-S RST-R BAND NAME QTH LOC START TIME END TIME
W3UAV -10 +04 20m Tony McMonagle York PA USA FM19PU 2022-08-25 18:55:48
New message
SEND CLR

24

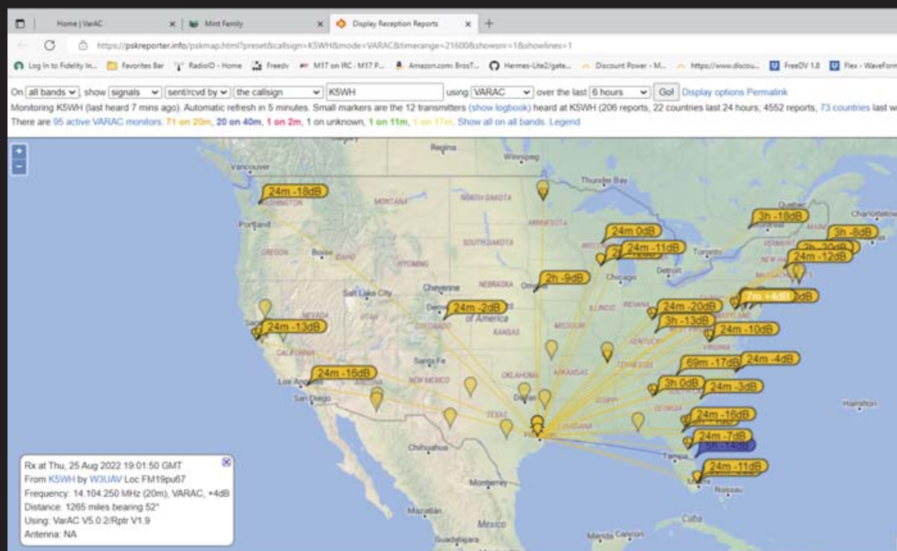

Viewing the Vmail
VarAC Mailbox
DELETE
CLOSE

25


PSK REP. MAP
To show stations on PSK Reporter
VarAC by 4Z1AC (V5.1.8)
NO NEW VMAIL
DISCONNECT MODEM
DISCONNECT
ABORT
SEND FILE
SEND VMAIL
PSK REP. MAP
SEND
CLR

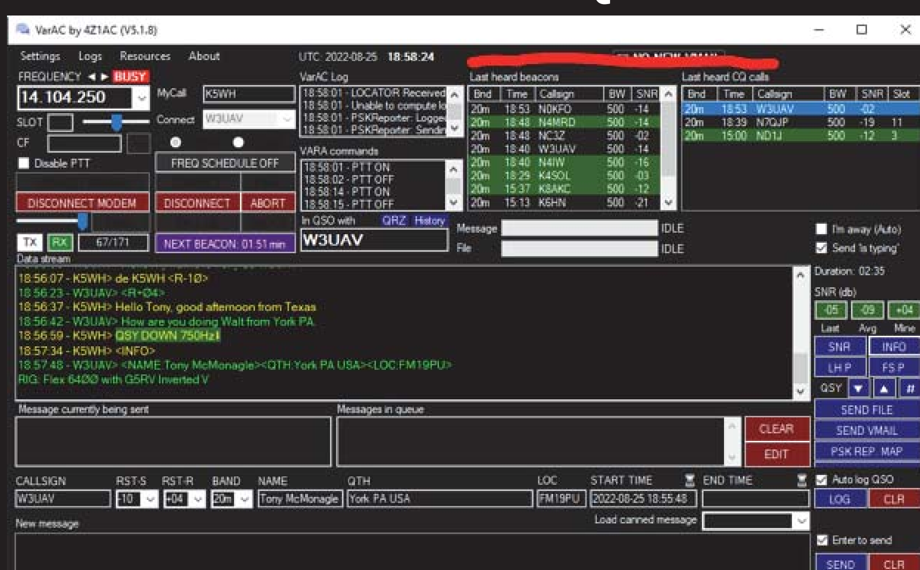
26

PSK REP. MAP
To show stations on PSK Reporter



27

Other Features
Notice the list of Last Heard beacons and Last heard CQ calls



28

Right Click on a callsign for options

- Clear
- Copy All
- Ping (get report)
- QRZ.COM lookup
- PSK Reporter lookup
- Callsign History
- QSY to slot

29

Logs

- QSO Log
- Chat History
- Last Heard Log
- VarAC Log
- Open Logs Directory

30

Resources

- Quick Start Guide
- User Manuals (EN)
- User Manuals (Non EN)
- VarAC Facebook Group
- VarAC Forum
- Troubleshooting
- FAQ

31

Further Assistance

- For any follow-up information or configuration assistance, please contact Walter-K5WH
  - walterh@k5wh.net
- Or join us on ZOOM anytime around the clock
  - http://www.k5wh.net/zoom/

32

HTTP Authorization Implementation for APRS-IS Servers

Peter S. Loveall

ARRL, TAPR, (ISC)²

Author Note

This paper describes the implementation of Base64 encoding in javAPRSSrvr.

Abstract

Implementation of HTTP Authorization using type APRS-IS and Base64 encoding of the login line in the HTTP Authorization header similar to Basic authorization. Base64 encoding is also supported in any authenticated login mechanism on javAPRSSrvr.

Keywords: APRS,Base64,HTTP

HTTP Authorization Implementation for APRS-IS Servers

Requirements

HTTP 1.1 specifies an authorization mechanism which must be supported anytime the 401 response code is used. In the past, javAPRSSrvr and aprsc have not properly provided a WWW-Authenticate header as part of the response because they did not recognize the Authorization header in the request. To remedy this situation, I focused on the most straightforward way to implement authorization in the Authorization header while being compliant yet unique to the HTTP specification. In the process, the same authorization mechanism can be used for non-HTTP logins if implemented properly. As with all new server features, all current and prior clients must be supported by the server and implementation in the client is not required.

Project Scope

javAPRSSrvr is a full APRS-IS server implemented 100% in Java and has been in use since 2003. Through the ensuing years, it has constantly been maintained and available to amateur radio operators worldwide. It was the cornerstone in stabilizing APRS-IS by implementing the q algorithm and other loop detection. javAPRSFilter, an adjunct to javAPRSSrvr created by Roger Bille SM5NRK, is the basis for all server filter commands allowing a client to connect to a server and only receive requested packets and any packets necessary to properly operate an IGate. It has been **critical** that any enhancements to javAPRSSrvr do not require any changes to existing APRS-IS communications and user connectivity.

This project scope specified implementing support for proper use of the 401 response code when responding to HTTP/HTTPS requests with an invalid or missing login line. The HTTP specification states a server MUST provide a WWW-Authenticate header to tell the client what is missing.

If the implementation of the HTTP authorization protocol is done within a broader scope, all login lines could benefit. In fact, by implementing most in the login line parser, this benefit was realized.

HTTP Listener Port

RFC 7235 specifies the HTTP Authentication Framework. It states “The 401 (Unauthorized) status code indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. The server generating a 401 response MUST send a WWW-Authenticate header field (Section 4.1) containing at least one challenge applicable to the target resource.” The format of the WWW-Authenticate header is:

WWW-Authenticate: type realm=”Text”

Since this mechanism is to use existing APRS-IS authentication, I determined the “type” should be APRS-IS. Since the “realm” is simply a textual “hint”, I made the implementation in javAPRSSrvr to be:

WWW-Authenticate: APRS-IS realm="APRS-IS Valid Login"

The next piece, according to the RFC, is for the client to place an Authorization header in the request. For this, I took the implementation for Basic authentication and adapted it for APRS-IS. To do this, the Authorization header is formatted:

Authorization: APRS-IS Base64-encoded-login-line

The Base64 encoding is easy to implement with the myriad of available libraries (I use the java.util.Base64 class), obscures the login line yet accommodates complete encoding without white spaces in the encoded text. The login line that is encoded does not include CR or LF and starts with “user “ (no leading white space). If this is seen within the HTTP headers, it will be used as the login line and any in-stream login lines will be ignored.

I placed the Base64 encoding/decoding in the APRSISLogin class in javAPRSSrvr. By doing so and keying on the mandatory start of the encoded line (dXNlci), I was also able to detect/send a Base64 encoded login line anywhere, on any port.

Upstream Dialer

The upstream dialer and send-only upstream dialer allow for Base64 encoding. On the send-only upstream dialer, if Base64 has been selected and it is an HTTP upstream dialer, the Base64 login line will be sent in an Authorization header. All other dialers will replace the login line with the encoded line in the data stream.

Summary

Because of the modular design of javAPRSSrvr, adding Base64 encoded login line support was relatively simple and easily maintained. RFC 7235 requirements have been met with this implementation as well as adding Base64 encoding/decoding availability to non-HTTP ports and connections.

The implementation meets the requirements of being additive without affecting current software while providing developers another standards-based (and therefore publicly supported) method of logging the client into APRS-IS. It is **NOT** secure but the login line can now be sent encoded instead of in plain text.

I have reached out to the owner of the aprsc software to encourage inclusion of this login method on that software as well.

References

Overview (Java Platform SE 8) (oracle.com)

RFC 7235 - Hypertext Transfer Protocol (HTTP/1.1): Authentication

Connecting to APRS-IS

TLS Implementation for APRS-IS Servers

Peter S. Loveall

ARRL, TAPR, (ISC)²

Author Note

This paper describes the implementation of TLS on javAPRSSrvr.

Abstract

Implementation of TLS encryption in javAPRSSrvr to support HTTPS and Secure WebSocket (WSS) connections. This paper does not address other TCP encryption methods and certificate exchanges experimentally created by other authors for non-HTTPS connections.

Keywords: APRS,TLS,WebSocket,HTTPS

TLS Implementation for APRS-IS Servers

Requirements

HTTP and WebSocket support have been implemented in javAPRSSrvr for over 5 years. HTTP was implemented first to provide an easily implemented interface for submitting single or multiple packets in a send-only format that would be able to utilize HTTP proxies and make use of lower cost HTTP telecommunications available at the time.

WebSocket support was added in 2017 to the standard HTTP send-only listener port to allow clients to connect bidirectionally using proxies and telecommunications facilities which charged less for HTTP communications.

With the advent of public Access Points and man-in-the-middle attacks on non-encrypted data, the need for HTTPS support has become a requirement for almost all websites. This is such a critical piece of security that there are now free SSL certificate vendors and most web hosts support SNI (Server Name Indication) to allow HTTPS to a single IP address with multiple web sites. All modern browsers are moving to HTTPS as the preferred protocol when connecting to a web site and these same browsers refuse to make a non-secure WebSocket connection from a page received via HTTPS. This last fact caused me to consider implementing Secure WebSocket in javAPRSSrvr which, by default, also implements HTTPS.

Project Scope

javAPRSSrvr is a full APRS-IS server implemented 100% in Java and has been in use since 2003. Through the ensuing years, it has constantly been maintained and available to amateur radio operators worldwide. It was the cornerstone in stabilizing APRS-IS by implementing the q algorithm and other loop detection. javAPRSFilter, an adjunct to javAPRSSrvr created by Roger Bille SM5NRK, is the basis for all server filter commands allowing a client to connect to a server and only receive requested packets and any packets necessary to properly operate an IGate. It has been **critical** that any enhancements to javAPRSSrvr do not require any changes to existing APRS-IS communications and user connectivity. This was the basis for implementing WebSockets as well.

This project scope specified implementing a new TCP listener port for HTTPS and implementing a new upstream dialer (connection type) for Secure WebSocket while reusing as much existing code as possible and using Java 8 and later standard constructs and classes. Because javAPRSSrvr was properly developed with little overlap between classes in functionality, this proved to be an addition of a very few lines of code.

A consideration while scoping the project was whether to implement HTTP/2 in addition to the TLS implementations. I determined that HTTP/2 is 100% backward compatible with HTTP/1.1 because its first communication is using HTTP/1.1 to request upgrading to HTTP/2 using the same upgrade mechanism as the WebSocket upgrade request. One of the major advantages of HTTP/2 is using a single connection for multiple requests. Because HTTP send-only ports only support a single request per connection, I determined that it was best to not implement HTTP/2 at this time.

HTTPS Listener Port

Implementation was done by adding code to the existing TCPListenerPort class to create the server socket using javax.net.ssl.SSLServerSocket. Using standard java.net.security classes, a PKCS12 pfx file is used for the server's certificate and the SSLContext created from that file is used to create the SSLServerSocket. I created a properties file which is included in the JAR file that defines the supported TLS protocols and ciphers. This properties file can be overridden with an external file. The included file includes TLSv1.2 (sans CBC algorithm and RSA handshake ciphers) and TLSv1.3. These are supported by all current JVMs 8.0 and later but are nonetheless used only as a filter to already enabled protocols and cipher suites on the server socket making the implementation cross-platform compatible. Because SSLServerSocket is a subclass of ServerSocket, it is used just like any other socket in the TCPListenerPort class, thereby reusing all existing code for actual client connection handling.

The implementation allows a software developer to connect to a HTTPS port using standard HTTPS connection mechanisms available in the software they are using with needing to create extra code. If they were already connecting using HTTP, HTTPS should be a simple change from http:// or ws:// to https:// or wss://.

Upstream Dialer

The upstream dialer uses a class I created called WebSocketClient. When the WebSocket protocol was added in 2017, there was no off-the-shelf classes available so I created both the WebSocketServer and WebSocketClient classes in javAPRSSrvr. This gave me an advantage to implement WSS support by adding it as simply a connection mechanism to existing code. I again went to the javax.net.ssl package available on any Java 8.0 and later JVM and implemented the upstream connection using existing code which can use a proxy (ProxySelector is passed “https” for a WSS connection instead of “http”) if defined to Java and attached a SSLSocket to the socket returned by ProxySelector. That SSLSocket is also restricted to the secure TLSv1.2 ciphers and TLSv1.3 in the same manner as the listener port and it does host name validation on the certificate. Java also supports SNI by default making this upstream dialer compatible with reverse proxies hosted on shared websites.

Summary

Because of the modular design of javAPRSSrvr, upgrading its HTTP and WebSocket support to also support HTTPS and Secure WebSocket using TLSv1.2 and TLSv1.3 was a relatively short requirement to implementation cycle. In addition to testing directly against remote javAPRSSrvr implementations, the upstream dialer has also been successfully tested against a reverse proxy, ARR by Microsoft. As with all development of javAPRSSrvr, it can be used across platforms including its Android derivative, javAPRSSrvrIGate.

The implementation meets the requirements of being additive without affecting current software while providing developers a standards-based (and therefore publicly supported) method of securely connecting to APRS-IS.

References

Overview (Java Platform SE 8) (oracle.com)

RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3 (ietf.org)

RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 (ietf.org)

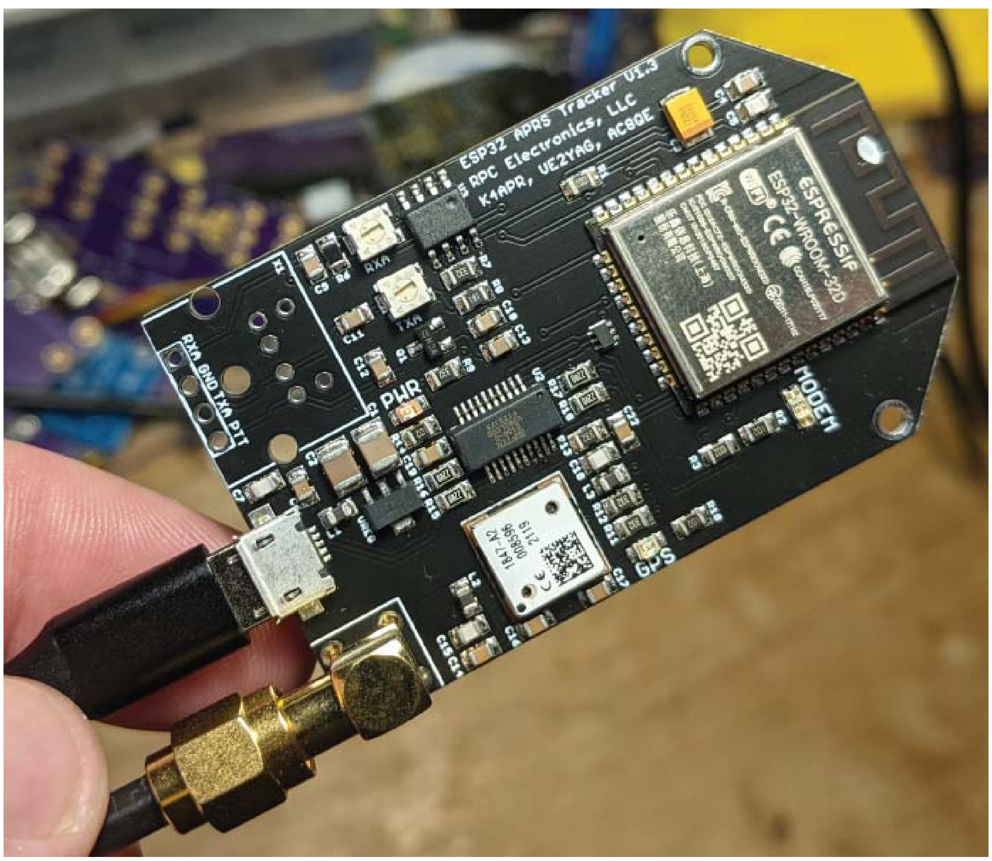
RFC 6455 - The WebSocket Protocol (ietf.org)

Connecting to APRS-IS

ESP32 Packet/APRS

Creating a Low Cost Tracker

Written By:
Jason Rausch K4APR
Remí Bilodeau VE2YAG



In Memory of Robert Bruninga WB4APR 1948-2022
"The Father of APRS"

Introduction

Nearly ten years ago, Remí VE2YAG and I crossed paths via a short YouTube clip I saw of a handmade APRS display device. I contacted the person who uploaded the video and told him I would love to work with him to create some kind of product around the idea. He contacted me back and said that the device was the work of his friend Remí and gave me an email address to contact him. I emailed Remí and we quickly became friends around our common love for packet and APRS. Fast forward, Remí and I have created several APRS related hardware devices. Some have become formal products that we sold and others have been experiments that either were held internally for our own use or we shelved because we got excited about another idea.

When the ESP32 came around, I was still playing with PIC's and had dabbled in Arduino. I should point out, I am by no stretch of the imagination an embedded programmer. My code is embarrassingly cobbled together from my own badly written routines that take me forever to write and snippets I find online. Remí, on the other hand, is a code wizard. I can throw out an idea and he'll have it written and ready for testing in no time. Anyways, the ESP32 came around and we immediately saw the potential to apply it to an amateur radio data application. Since we had previously focused on APRS, it was only natural that we do the same with the ESP32. This paper is based on that work, hopefully shows what we have accomplished and what we are still working on.

Design Goals

We set out to build an APRS modem/tracker with the following design goals:

- Centered around the ESP32's rich feature set
- Integrated 1200 baud AX.25 modem
- USB Serial Interface (FTDI Preferred)
- KISS Support through USB and Bluetooth
- Integrated GPS Receiver
- Op-Amp Audio Design, Avoid Modem IC
- Operates as an APRS tracker autonomously and with a tethered device
- Small, Lightweight and low cost to produce
- Minimal LED Indication (reduce draw when battery operated)
- Easy audio adjustment
- Web Interface Configuration

Most of these design goals are well within reach and most have already been implemented. In this paper, I will do my best to point out where we have achieved a goal, which are yet to be implemented and those that are still being improved.

Choosing a Microprocessor

For years I have been working with PIC microprocessors. They were introduced to me by my involvement in the HamHUD APRS project and I had no idea there was such a powerful device available in such a tiny package. These days, there are many DSPic devices that with some add-on peripherals could do what we needed, but the parts count starts to really add up. Not to mention, debugging any of the sub-systems. Remi and I had used the CORTEX LM3S800 in the original YagTracker. Later we moved onto the LPC1343 and LPC1347 in the ExpressTracker models. While these ARM based processors all worked well at the time for our applications, they too were lacking something…wireless connectivity.

Enter the ESP32. It's truly a marvel of modern microprocessor technology. I'll spell out the full specifications in the next section, but the major advantage the ESP32 had over the other guys was the built in wireless connectivity options. Bluetooth, Bluetooth LE and WIFI 802.11b/g/n. This opens up the endless possibilities for connecting this device to other devices. We're not talking about concentration on internet, as many APRS users have gone to. We're talking about connecting to a modern device, while still using a real radio. Put the radio back into amateur radio.

Not only does the ESP32 have wireless connectivity, but it has a whole host of interfaces and supported protocols making adding on the hardware we need easy!

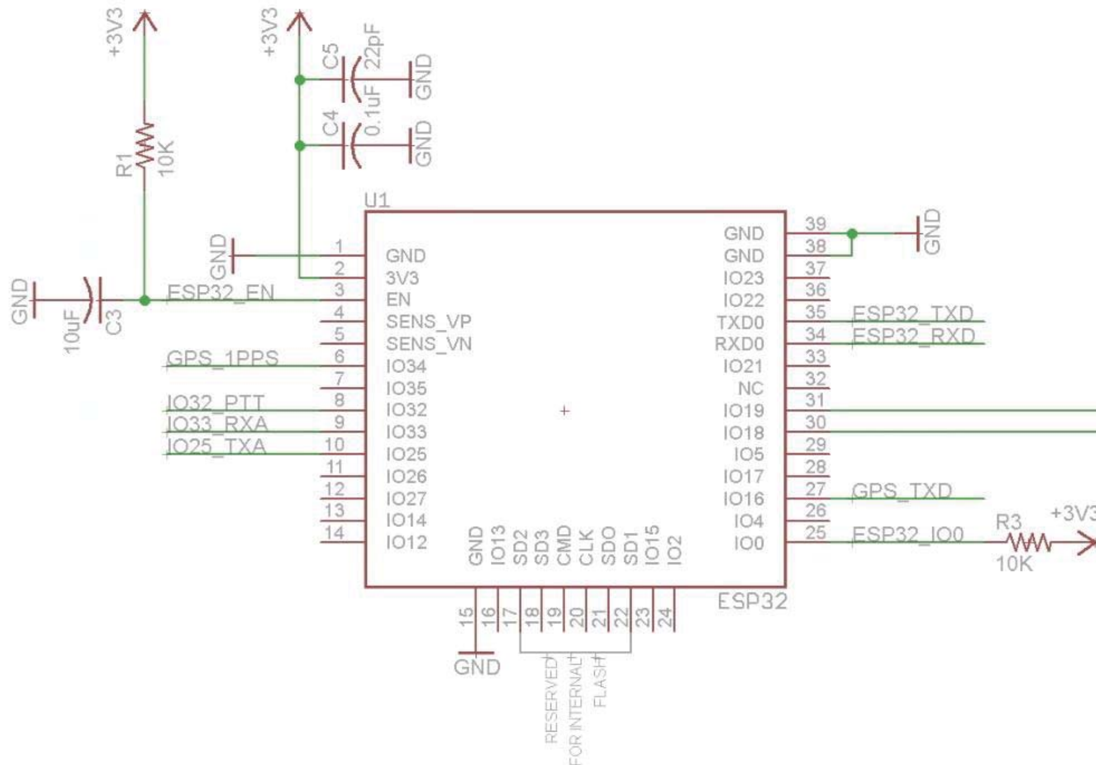
ESP32 Hardware Overview

Let’s get right into it and talk about the ESP32 hardware itself. The ESP32 is an incredible amount of features packed into a tiny package that averages less than $5 each, single quantity. Here are the specifications:

- 32 Bit Microprocessor
- Two CPU Cores
- 80-240 MHz Clock Speed
- Three Hardware UARTs
- 12 Bit ADC, Up to Eighteen Channels
- Two 8 Bit DAC Channels
- I2C, SPI, SD Card Interface, PWM and SDIO
- 22 GPIO Pins
- 32 MB of On-Board SPI Flash
- Built-In Bluetooth and Bluetooth LE
- Built-In WIFI 802.11b/g/n
- +3.3V Operating Voltage
- Can be programmed using Arduino IDE and ESP32 Support Plugin

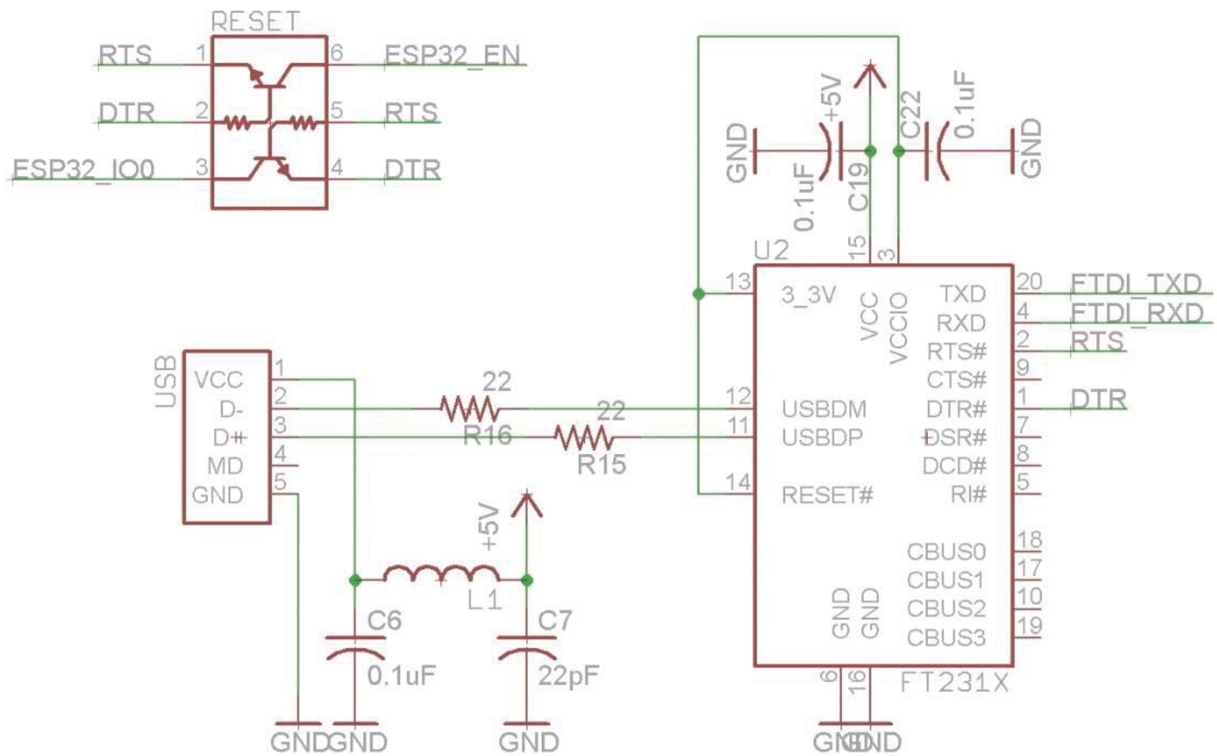
ESP32 and FTDI USB Interface

The ESP32 doesn't require a lot of external components to make it work. It is somewhat sensitive to power fluctuations and we found good power filtering with large electrolytic capacitors worked the best to prevent problems. Notably, the 10 uF capacitor on the EN (Enable) line of the ESP32. The 3V3 line also benefited from 0.1uF and 22pF capacitors to handle high frequency transient noise on the power rail.



The ESP32 is a bit finicky about how it is programmed. We used an FTDI FT231X USB-Serial IC that has a full UART, plus hardware handshaking pins. Most notably RTS and DTR lines needed for the self-resetting circuit. This not only prevents the user from having to manually reset the tracker after a firmware upload, but performs the somewhat complicated and time critical line pulsing "dance" it takes to put the ESP32 into a bootload mode for flashing. We started off by using a two discrete transistor setup for doing this, but found that it didn't always work and we would have to reset and start again. In comes the ROHM UMH3NTN single IC with dual NPN transistors and built in base resistors. By switching to this single part, all of the bootload/flashing issues were solved.

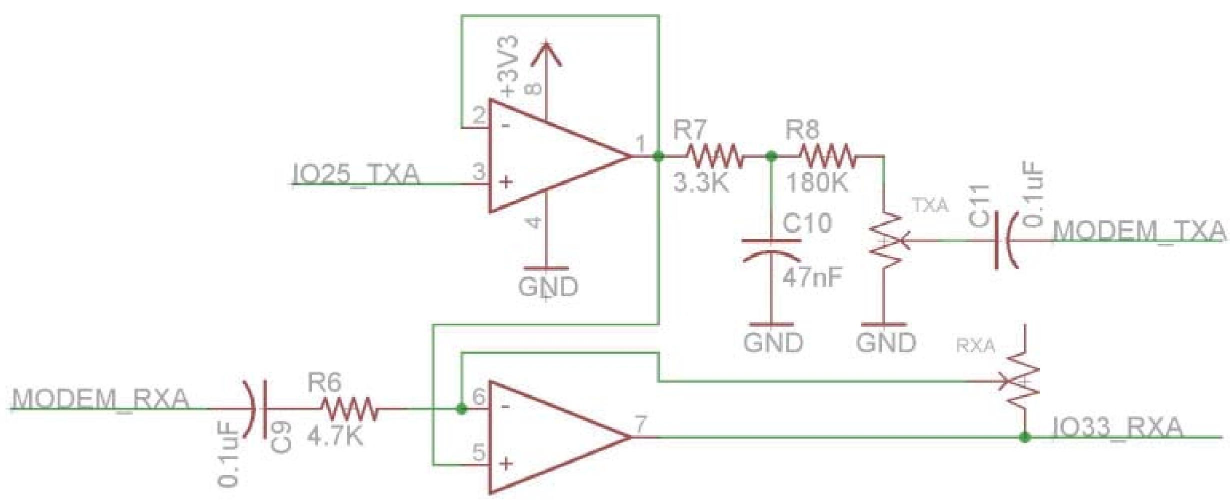
In this section of the schematic you can see the FTDI FT231X single IC solution to USB-Serial communications. The FT231X takes care of the USB heavy lifting, leaving you a TTL serial interface. FTDI drivers are included in just about every major OS now and you rarely have to actually go download them to get things talking. Just plug n' play!



Op-Amp Audio Section

When I went to Remi about the idea to build this, he already had a working op-amp based modem design he was playing with on a couple of other projects. It seemed like the perfect fit for our new APRS tracker. The part we started off with was a Microchip MCP6002. With the global shortage, the MCP6002 soon became nearly impossible to obtain, so we had to look for a replacement. We found the Microchip MCP6402T was close to the specs of the MCP6002 and a perfect pin-for-pin replacement. The part is a dual op-amp, reducing down to a single part, with some additional caps and resistors to form the output low-pass filter on the transmit side and a simple DC blocking circuit on the receive side.

Two 10K single turn trim potentiometers are used to control the transmit level and limit overloading to the receive section. IO25_TXA is a direct connection from the ESP32 DAC to the op-amp. MODEM_RXA is a direct connection from the radio port to the op-amp.



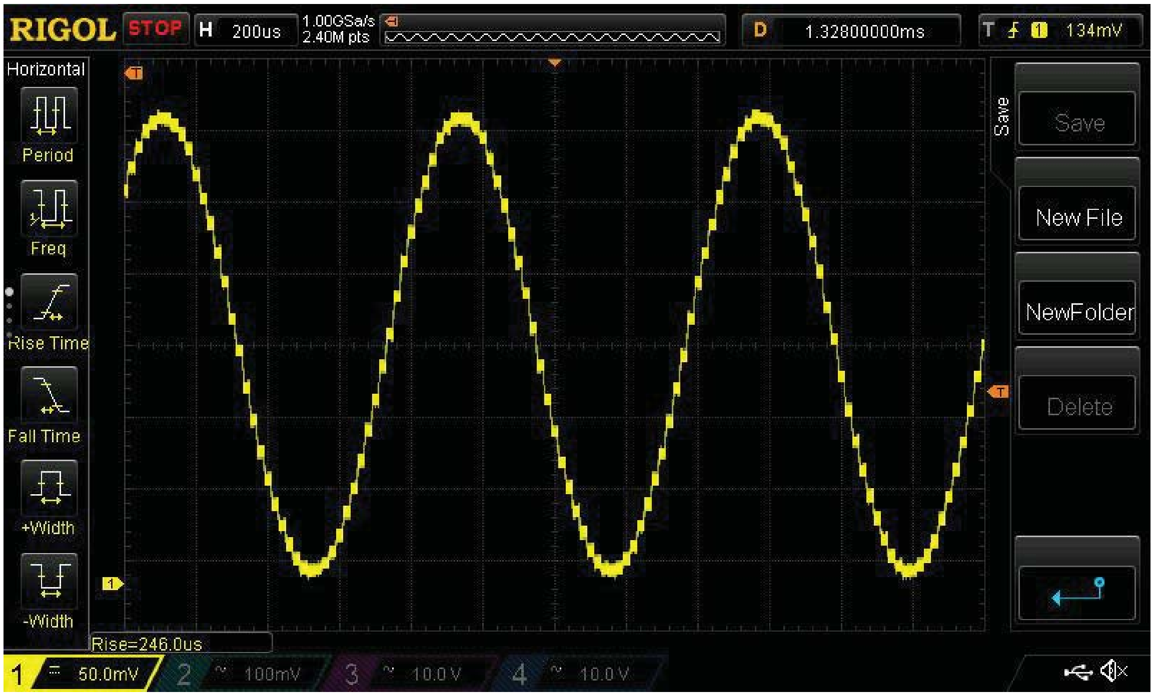
This design along with Remi's method has yielded a phenomenal decode rate of even poorly encoded packets. Remi has four individual decoding algorithms operating simultaneously to potentially catch any scenario and hopefully pull out a CRC-passed packet. The first two algorithms are correlator methods and the second two are filter methods. The correlator algorithms can be described as self-multplicative correlator with a flat and high frequency boost filter. The filter algorithms use a comparison of sine, cosine/sine of the 1200 Hz and 2200 Hz tones to detect which is which.

We Are Having Some Problems

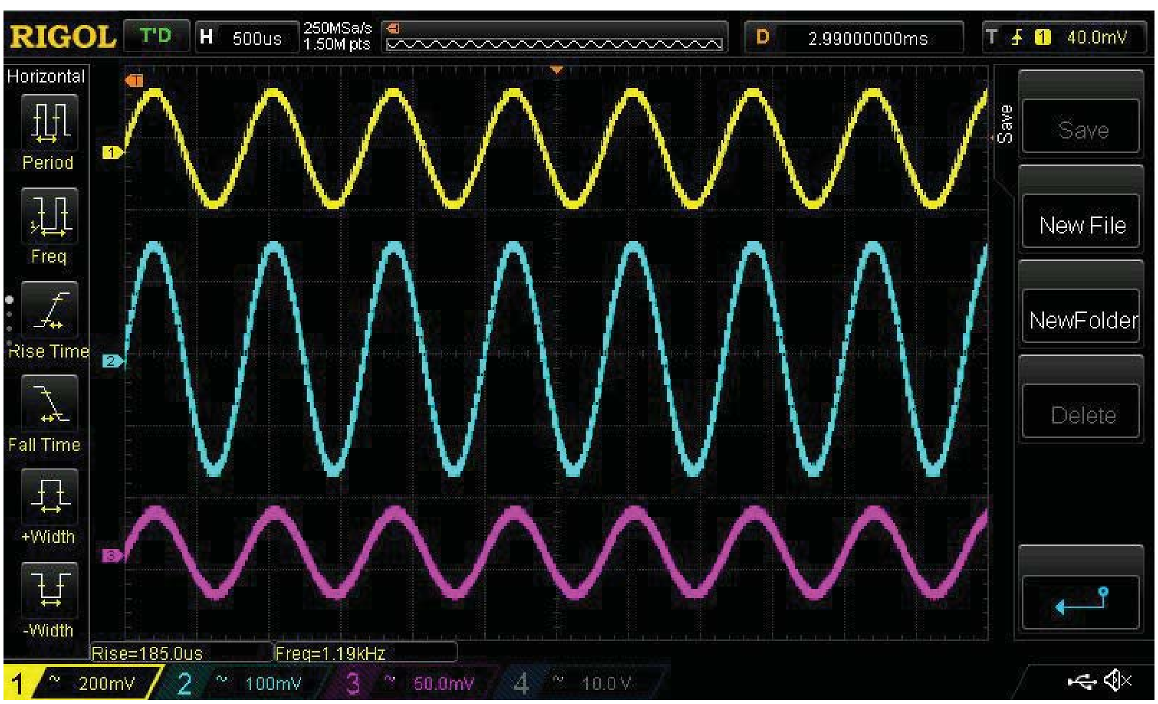
For the transmit side, we are having problems with the audio being decoded by just about any hardware I test with. This includes standards like the Kenwood D7/72/710 radios, Kantronics KPC-3, Argent Data Systems Tracker 2 and the NinoTNC. Some initial tests showed that the waveform might be the issue, having a rough "raspy" sound to it. Remi was able to tweak a few things in his code to clean this up, but yet decode of the transmitted packets have not been successful. I should note that Remi has a hand-built version of this circuit using through hole components and doesn't seem to have this problem. We can't nail down if the issue is with hardware, firmware or a combination of the two.

This o-scope shot shows the audio stepping out of phase. It was determined that the DAC was inverting some of the waveform, causing this. Some quick changes to the code fixed the problem.

The fix resulted in this o-scope capture.



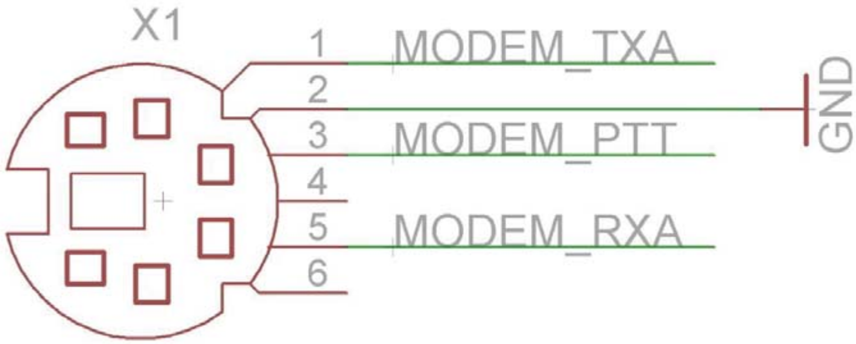
This is a comparison scope view of the raw output from the DAC, output of the op-amp section and just past the low-pass filter before going through the 10K transmit level potentiometer.



In Memory of Robert Bruninga WB4APR 1948-2022
“The Father of APRS”

Radio Interface

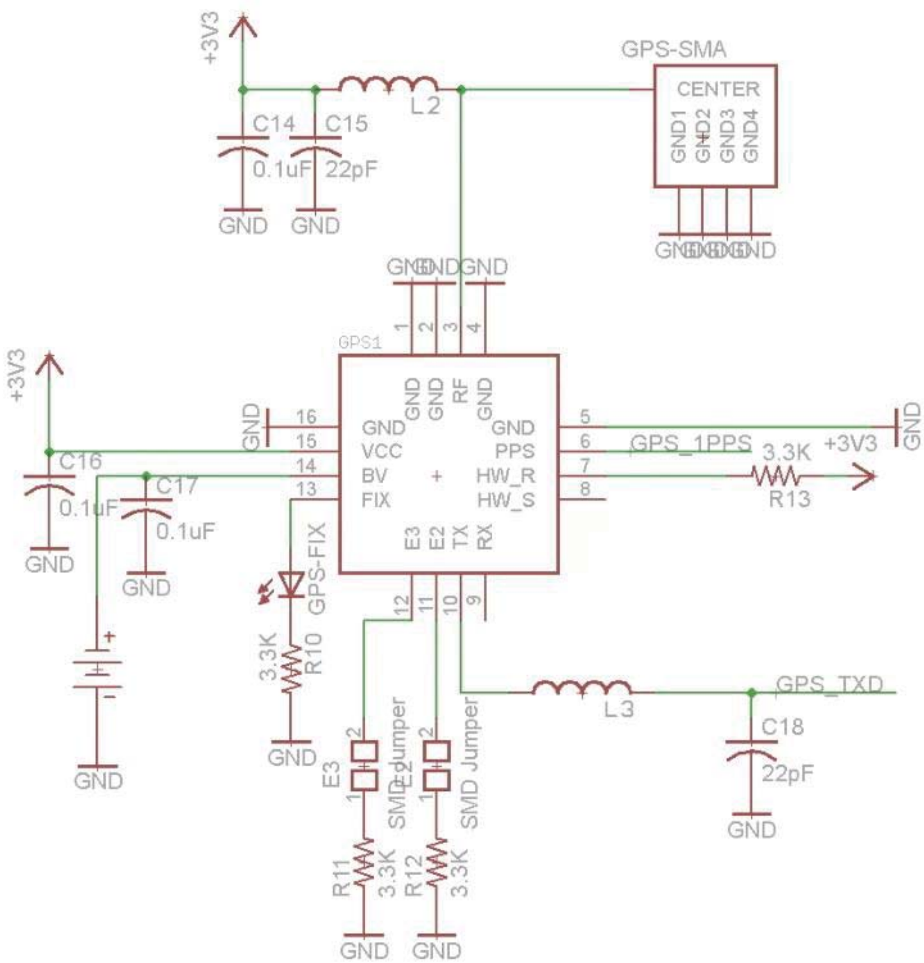
We employed the Mini DIN interface standard as the interface to the radio. There is also a four pin header on the PCB, for applications where the Mini DIN might not be optimal.



On-Board GPS Receiver

When looking for a low-cost, high performance GPS for the tracker, there were several options. While not cheap, the Trimble Copernicus II was an excellent option and proven. I also had a lot of experience with this GPS chipset when I put it in my RTrak all-in-one APRS tracker. It was a great option, but just too expensive. Another popular option is the UBlox line of receivers. Low cost, decent performance, but sometimes hard to source.

No, we needed a better option. I decided to fall back to a gem I had found several years ago, but had mixed results with. The Antenova M10578-A3. At just over $20 USD, it's an excellent performing chipset, small, 3.3V power capable and perfect for our needs. The only real configuration is making sure the E2 and E3 lines are at the correct levels to set the baud rate to 4800 baud. In the future, we could switch up to 9600 baud. A simple change to the solder jumpers on the back of the board and we're talking faster!



Configuration

As of the time writing this, configuration is done in one of two ways:

Command Line

Much like a DOS or Linux terminal interface. Simple text commands that return current values or when accompanied by arguments can commit changes to settings.

Text Configuration File

Remi has implemented an in-terminal text editor that allows you to create a complete configuration file that is loaded on power-up/boot. The structure is much like a DOS batch file with section [xxx] headers and simple x=y settings under the header.

Example (not complete):

```
[tnc]
callsign=k4apr
tz=0
verbose=0

[gps]
baud=4800

[kiss]
enable=1
persist=63
slottime=1
txdelay=35
dac_zero=128
```

Web Configuration

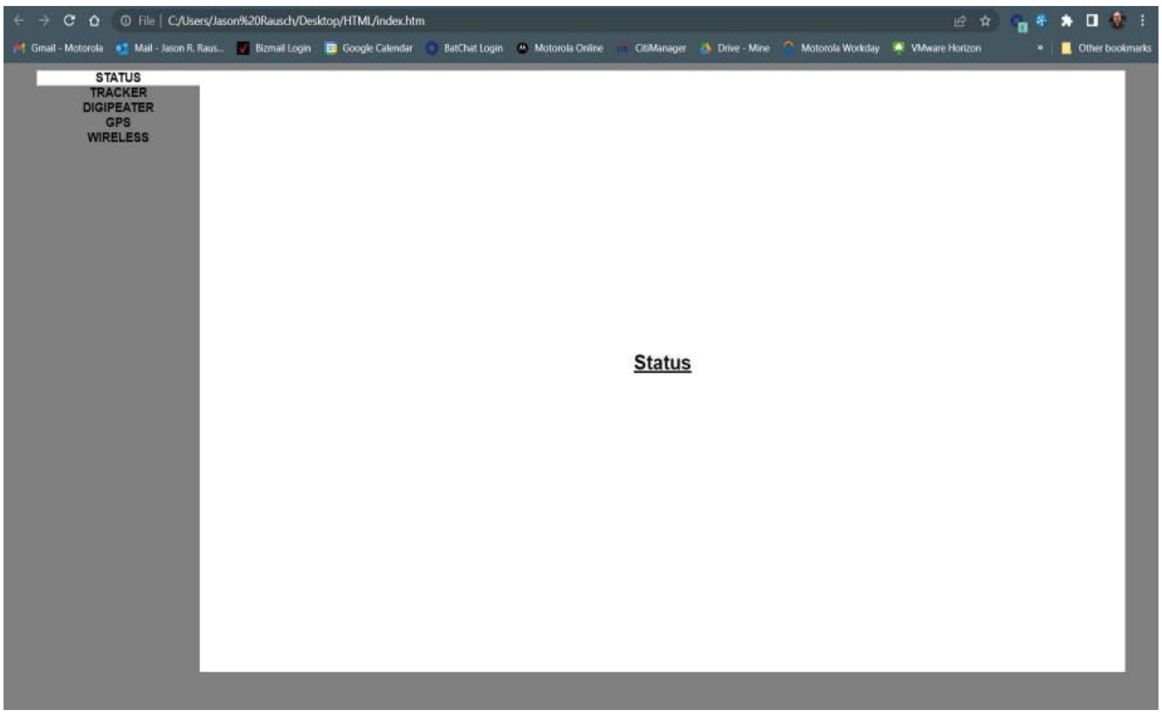
A future option will be the ability to connect to the device via WIFI and using a web browser navigate to 192.168.10.1 to be presented with a web interface for configuration of all settings.

What's Left to Do?

1. We have to get to the bottom of the transmit audio issue. We have been fighting this for a while. Numerous hardware revisions, hacking up older hardware to try "simple fixes". We're open to any suggestions!

2. Web Interface for configuration. We know that if this is going to be a product that people buy, they are going to want an easy way to make configuration changes, especially when in the field/on the go. The web interface is really the only solution for this.

   This is a quick mock-up site written in HTML.



Conclusion

We have made a lot of progress, but there is still work to do. Some might ask "why?" when APRS seems to be waning. Our argument is, APRS is coming back. We're seeing an increase in activity all over the place and we think there is a need for a product like this.

ezDV: Low Cost Digital Voice on the ESP32

Mooneer Salem, K6AQ (mooneer@gmail.com)
August 2022

Abstract

FreeDV is a series of digital voice modes optimized for use on the HF portion of the radio spectrum that uses the Codec2 open source voice codec library. Traditionally, this has taken the form of a desktop application running on a Windows, Mac or Linux PC. However, there are challenges with this approach, especially for portable operation. This paper will discuss a low cost approach with high ease of use that eliminates the need for a computer for encoding and decoding digital voice.

Introduction

Codec2 is a library originally created by David Rowe (VK5DGR)[1] in 2009 to provide an open source, patent unencumbered digital voice codec for use on amateur radio. Since that time, the library has been extended to also support the various FreeDV modems required to use Codec2 on the air, as well as created a GUI based application[2] to enable their use.

While usage has gradually increased, there are several current pain points surrounding initial configuration as well as more practical concerns with the need for a PC to encode and decode FreeDV. Unlike with most popular ham radio applications (for example, WSJT-X), FreeDV requires two sound cards in order to be able to fully use the mode: one for the radio interface and the other for analog audio input and output. This is on top of the typical digital mode setup (e.g. adjusting transmit audio to ensure that ALC does not show any indication on the radio) and is different enough that it has caused confusion for some operators in the past.

In addition, unless one has a laptop that can be carried out into the field for portable use, the requirement that one runs FreeDV on a PC inherently limits it to stationary, home use. Even if one owns a laptop and is willing to take it portable, the decision to do so inherently results in having to rethink one's power budget and other constraints, which could result in having to make other changes to the portable station to stay within those constraints. Depending on one's priorities during portable operation, this could mean that FreeDV is left at home anyway.

---

[1] Rowe, David. "Open Source Low Rate Speech Codec Part 1." *Rowetel*, 26 Aug. 2010, https://www.rowetel.com/?p=128.
[2] "FreeDV: Open Source Amateur Digital Voice." *FreeDV*, https://freedv.org/.

An alternative to a full-featured laptop is an embedded device that can perform basic FreeDV modulation and demodulation. One example of such a device is the SM1000[3]. The SM1000 uses an STM32F4 microcontroller and is able to modulate and demodulate three FreeDV modes: 700D, 700E and 1600. However, due to the ongoing chip shortage, the SM1000 has been sold out with no ETA for return as of the time of this writing. The relatively high price prior to it selling out (~US$200) also made it less accessible to hams that could not easily afford to purchase the device without already being heavy users of FreeDV.

The ESP32 Microcontroller

The ESP32 was originally brought to market by a company called Espressif in 2013 in the form of the ESP8266[4]. This device was the first product to achieve wide adoption by the maker community due to its low cost and the inclusion of wireless functionality. Initially, documentation was solely in Chinese, but English language documentation along with a Software Development Kit (SDK) and libraries came about in short order. The wide adoption and wireless functionality made it a natural candidate for an embedded amateur radio device, especially as many newer radios are coming with network connectivity as a standard (or at least reasonably priced[5]) option.

As of this writing, there were a few variants[6] of the ESP32 that were found to possibly be suitable: the original ESP32 and the ESP32-S3. Unfortunately, the -C3 and -S2 variants are not suitable for FreeDV at this time due to the heavy usage of floating point operations in the library; previous tests by this author on the RP2040 (from the Raspberry Pi project)--which also has no floating point unit–resulted in performance significantly slower than real time[7]. Upon further investigation, it was found that the -S3 variant also has support for SIMD instructions[8] (similar to SSE on the Intel x86 architecture). There was previous successful experience optimizing portions of the Codec2 library to take advantage of those instructions on the PC[9], so the ESP32-S3 was ultimately chosen to enable this possibility during development.

---

[3] Rowe, David. "SM1000." *Rowetel*, http://rowetel.com/sm1000.html.
[4] Cording, Stuart. "What Is the ESP32? Its Brief History and How to Get Started." *Elektor*, 10 Mar. 2022, https://www.elektormagazine.com/articles/what-is-the-esp32.
[5] "Welcome to Wfview.org." *Wfview*, https://wfview.org/.
[6] "Chip Series Comparison." *ESP-IDF Programming Guide*, https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/chip-series-comparison.html.
[7] Salem, Mooneer. "Fixed Point/FPGA Implementation of Codec2 · Discussion #223 · drowe67/codec2." *GitHub*, 31 Oct. 2021, https://github.com/drowe67/codec2/discussions/223#discussioncomment-1565289.
[8] *ESP32S3 Series - Espressif*. https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf.
[9] Salem, Mooneer. "700C: Use Vector Ops for rx_filter_coh() When Possible. by Tmiw · Pull Request #200 · drowe67/codec2." *GitHub*, 24 July 2021, https://github.com/drowe67/codec2/pull/200.

Porting to ESP32

The next step after deciding on the ESP32-S3 was to buy a development device to begin testing the FreeDV codebase. The nanoESP32-S3 was discovered during a search of Tindie[10] and quickly purchased. For futureproofing, the “N8R8” variant with 8MB of PSRAM and flash was selected.

Upon receipt of the nanoESP32-S3, work was undertaken to attempt to compile the Codec2 library using the ESP-IDF development environment. As the ESP-IDF development environment natively uses CMake (much like FreeDV and Codec2), integration with a sample project using the “__EMBEDDED__” compiler definition (which enables various memory and space optimizations to allow it to fit onto the STM32F4) was straightforward. However, upon initially attempting to compile the test application, errors related to the CMSIS library[11] appeared on the console.

Investigating further, it was discovered that the usage of CMSIS was limited to a single file called “ofdm.c” inside the Codec2 library. These references were to complex value and real valued dot product operations provided by CMSIS (namely arm_dot_prod_f32()[12] and arm_cmplx_dot_prod_f32()[13]). Fortunately, Espressif’s ESP-DSP library also contained similar functions for performing dot product operations[14] and used the ESP32 SMID instructions when available. An initial pass at updating the ofdm.c file to use the ESP-DSP functions instead of CMSIS proved to be successful.

Before submitting the pull request to the Codec2 project, the changes made to use ESP-DSP were generalized to support architectures other than ARM and ESP32. This was done through the use of a known wrapper interface that embedded device designers would use to implement the required dot product operations. The SM1000 code was also updated to implement these wrapper functions using the CMSIS library. After testing these changes, a pull request was created in the Codec2 project to merge them into the codebase[15].

---

[10] Wu, Johnny. “NanoESP32-S3 Development Board by MuseLab on Tindie.” *Tindie*, https://www.tindie.com/products/johnnywu/nanoesp32-s3-development-board/?pt=ac_prod_search.
[11] “CMSIS.” *Arm Developer*, Arm Ltd., https://developer.arm.com/tools-and-software/embedded/cmsis.
[12] “Vector Dot Product.” *CMSIS-DSP Software Library*, https://www.keil.com/pack/doc/CMSIS_Dev/DSP/html/group__BasicDotProd.html#gadf26f6bc62d6416528663ad3e46fbf67.
[13] “Complex Dot Product.” *CMSIS-DSP Software Library*, https://www.keil.com/pack/doc/CMSIS_Dev/DSP/html/group__cmplx__dot__prod.html#ga93796e73f02771cf6fe13de016e296ed.
[14] “Espressif DSP Library API Reference.” *Espressif DSP Library v1.2.0-3-g2415e61 Documentation*, https://docs.espressif.com/projects/esp-dsp/en/latest/esp-dsp-apis.html#dot-product.
[15] Salem, Mooneer. “Allow Use of __EMBEDDED__ on Non-ARM Systems. by Tmiw · Pull Request #317 · drowe67/codec2.” *GitHub*, 3 Apr. 2022, https://github.com/drowe67/codec2/pull/317.

Audio Input and Output

The next step after confirming that the FreeDV codebase was able to compile on the ESP32 was to set up audio I/O. Unfortunately, while the ESP32-S3 does have an analog to digital converter (ADC), it does not have the functionality to emit an analog signal again. Additionally, the ADC that did exist on the -S3 had 12 bits of resolution[16]. It was decided to investigate additional audio ADC and DAC chips that could be integrated onto the board.

One product that particularly stood out was the TLV320 series audio codec chip by Texas Instruments. In particular, the AIC3254 variant had two ADC and two DAC channels[17], which made it ideal for processing both the radio audio and the analog headset audio. The chip also supported various audio routing modes (for example, the left channel of the audio going to the ESP32 could be configured to be coming from input #1 while the right channel could be configured as coming from input #2). As FreeDV solely works with mono audio signals and ignores the second channel, this made it possible to only need one TLV320 on the board to handle all the audio requirements of the device. Its cost was also fairly low at around $10 each in small quantities[18].

To make sure that the TLV320 would be satisfactory for the device, investigation was done into purchasing an EVM (also known as a development board) containing the TLV320. However, the TLV320 EVM was US$250 when purchased directly from Texas Instruments[19], a significant cost for effort that may possibly need to be thrown away. Instead, it was determined that a first pass at building a development board containing the TLV320 would be done instead.

Building the Development Board

To build the development board, a new KiCad project was created to hold the schematic and PC board layout. Pin headers were then placed in the schematic to represent the nanoESP32-S3, followed by the TLV320 and its required passive components as per its datasheet and technical reference documentation. Two TRRS 3.5mm jacks were also added, one for radio audio and one for a wired headset. Buttons and LEDs were also added to represent the most common operations (PTT, volume up/down and selecting FreeDV modes) and system status (PTT on, network connection, sync and audio overload).

Once these components were routed on the PCB, some thought was taken as to how it would be assembled once the boards arrived. While this author does have some SMD soldering

---

[16] "Analog to Digital Converter (ADC)." *ESP-IDF Programming Guide*, https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32s3/api-reference/peripherals/adc.html.
[17] "TLV320AIC3254." *TLV320AIC3254 Data Sheet, Product Information and Support | TI.com*, Texas Instruments, https://www.ti.com/product/TLV320AIC3254.
[18] "TLV320AIC3254IRHBT." *JLCPCB*, https://jlcpcb.com/partdetail/TexasInstruments-TLV320AIC3254IRHBT/C182339.
[19] "TLV320AIC3254EVM-K." *TLV320AIC3254EVM-K Evaluation Board | TI.com*, Texas Instruments, https://www.ti.com/tool/TLV320AIC3254EVM-K#order-start-development.

experience, the experience was lacking for components as small as the TLV320. Thus, it was decided to take advantage of JLCPCB's SMD assembly service. This service was reasonably priced for prototyping (generally part cost plus a fee per part for what they call "extended parts"). After some time spent in associating parts on the schematic with parts that were in stock at JLCPCB, the Gerber files were submitted to them and payment was made.

While waiting for the initial boards to arrive, development of the firmware for the ESP32 was undertaken. This involved a first pass at writing the initialization and audio I/O code (using the I^2C and I^2S functions available as part of ESP-IDF) as well as a basic loop for sending audio through the Codec2 library. The initial effort completed in time for the arrival of the development boards.

Debugging the Firmware

Unfortunately, it was discovered that one of the pins on the nanoESP32-S3 that was initially intended for use with the TLV320 was shared by the multi-color LED. As a workaround, the pin on the board's pin header was cut and the female side bridged to a neighboring pin. The change was also reflected in the schematic and PCB for the next revision of the board.

Additionally, the initial pass had significant issues with configuring the TLV320. During development, it was decided to configure the TLV320 such that it used an 8 KHz sampling rate; this sample rate is the native sample rate as used in the FreeDV codebase. However, much of the sample code available for the TLV320 assumed a 48 KHz or 44.1 KHz sample rate, so a guess as to the data to send over the I^2C bus was made. Significant time was spent performing modification/flash/test cycles trying different commands and command orderings in an attempt to have working audio, but eventually a series of commands was found that properly configured the chip for 8 KHz audio.

Because of the way that audio was being fed into the Codec2 library, however, part of one of the input audio channels did not work properly. Initially this was thought to be a hardware issue for two reasons: seeing pulsing on the offending channel with an oscilloscope and due to the issue only affecting one channel and not the other. A second revision of the board was spun that did the following:

- Avoided use of the GPIO pin that was connected to the multi-color LED on the nanoESP32-S3,
- Rerouted all traces to the TLV320 to shorten them as much as possible,
- Added additional vias connecting ground planes (especially surrounding the sensitive audio traces coming from the TLV320) to improve noise immunity,
- Tweaked the values of various components on the schematic/PCB, and
- Added mounting holes for future mounting in an enclosure.

These resulted in the following schematic and PCB design:

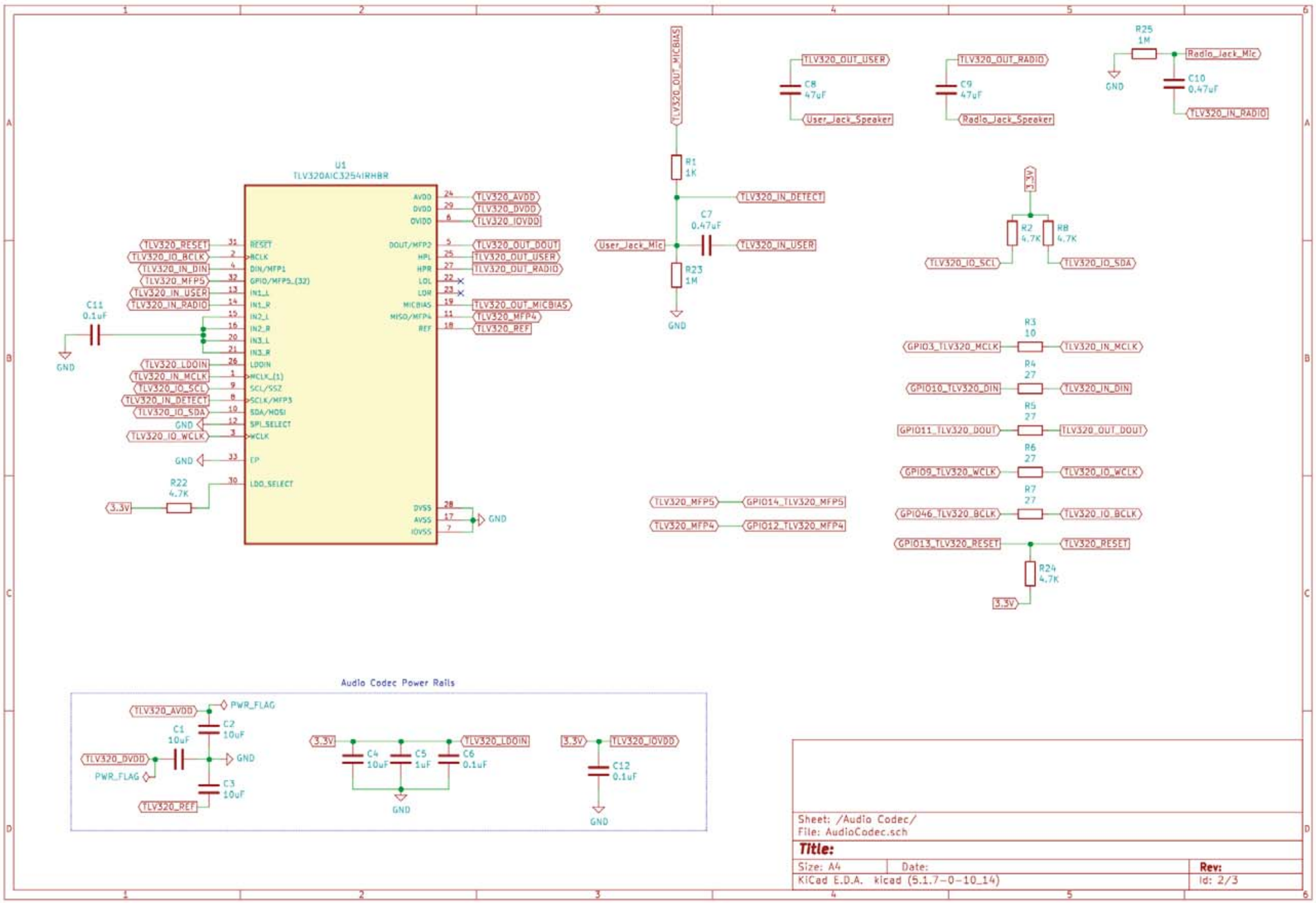


Figure 1: The TLV320 and supporting components on the ezDV schematic.

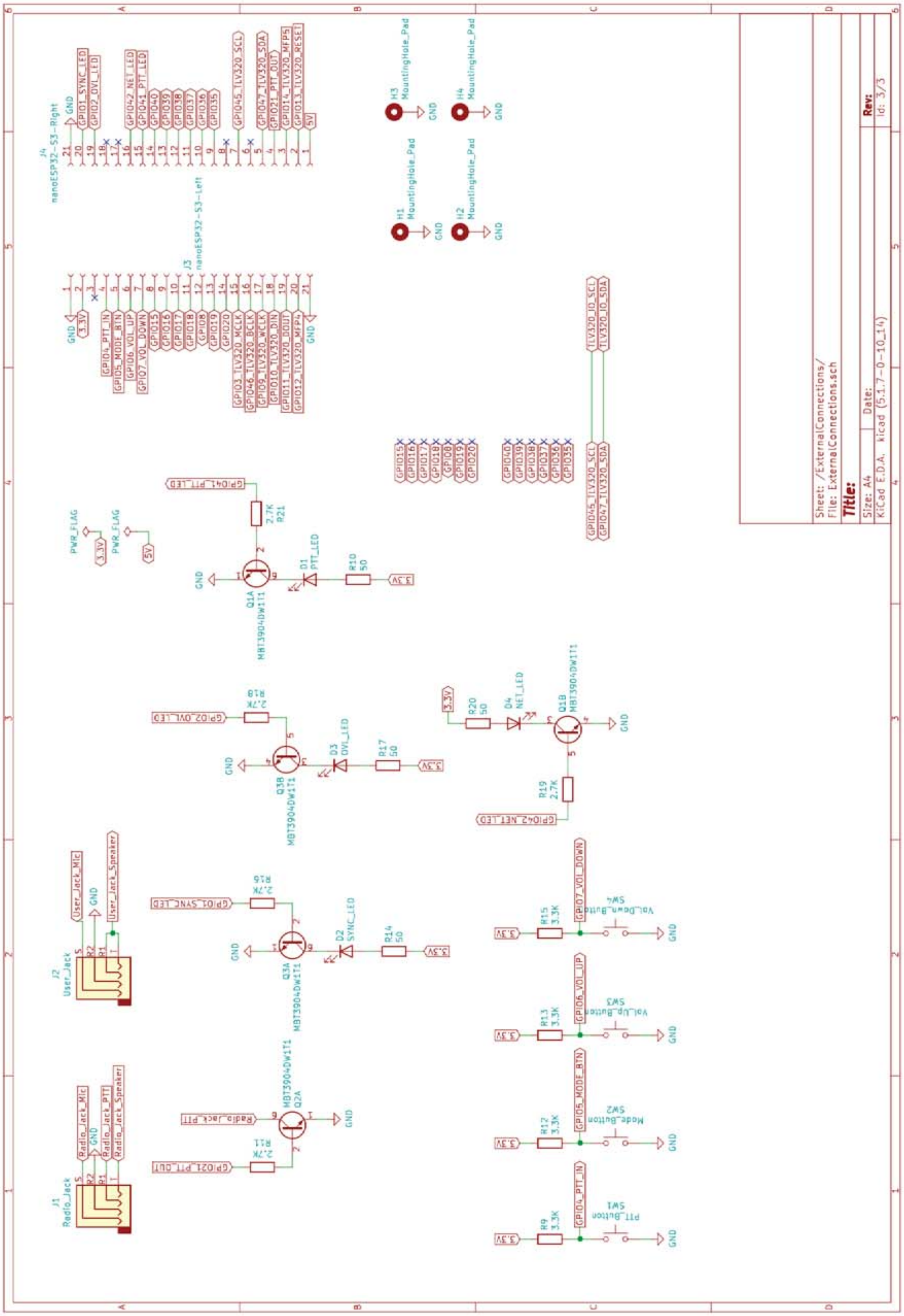


Figure 2: The LEDs and other external interface components on the ezDV schematic.

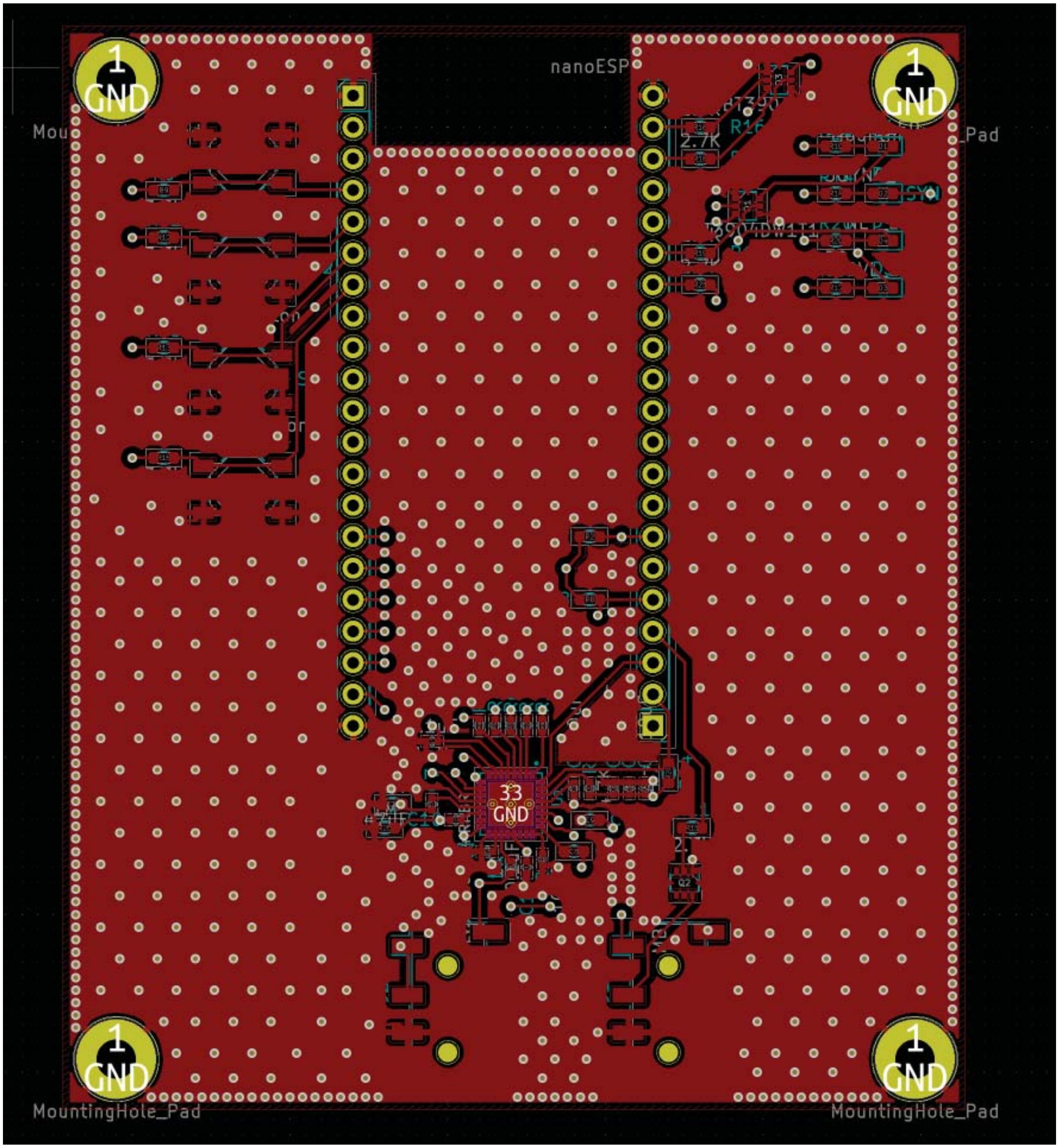


Figure 3: The top signal layer of the ezDV PCB including ground pours.

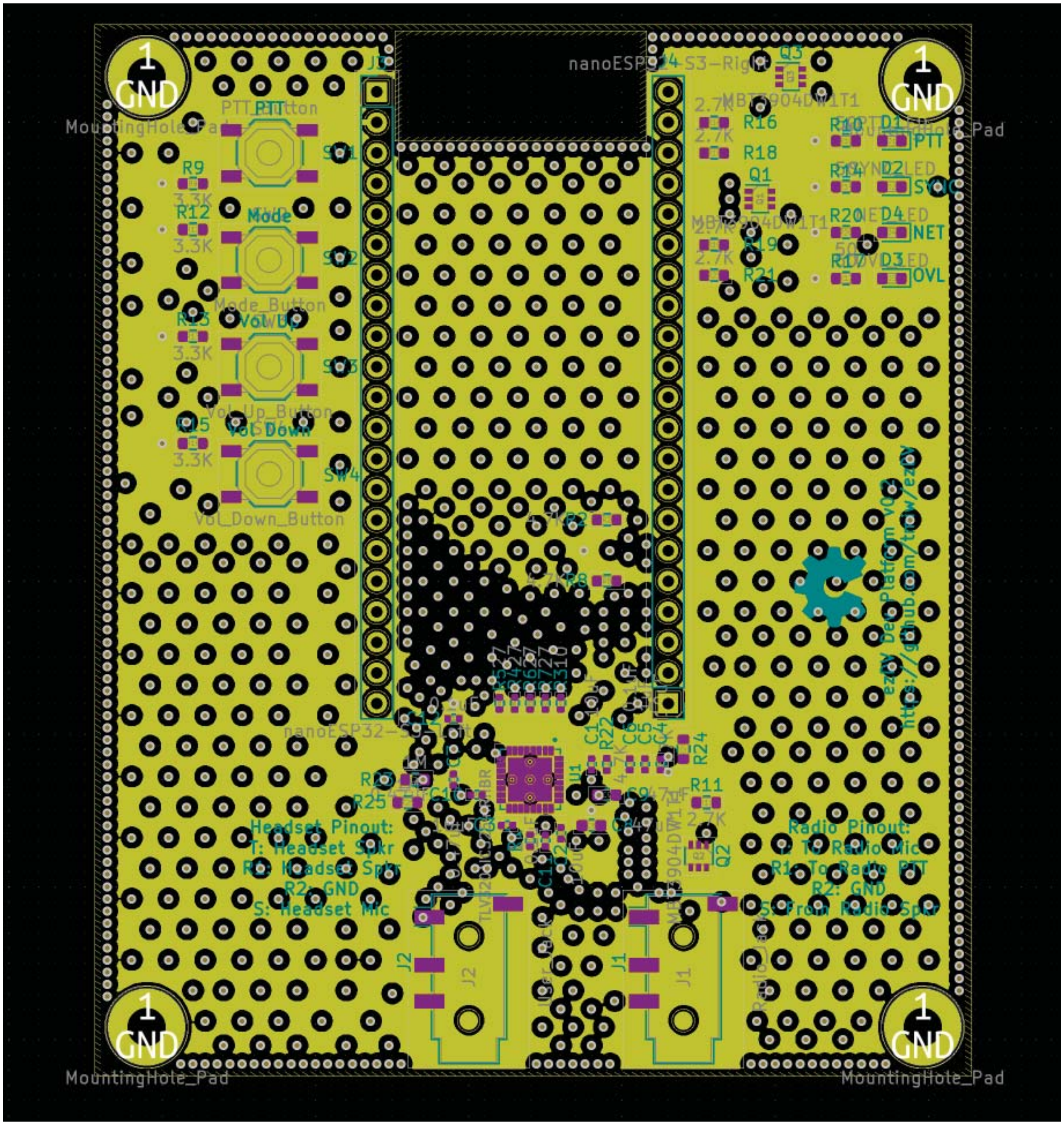


Figure 4: The 3.3V power layer of the ezDV PCB.

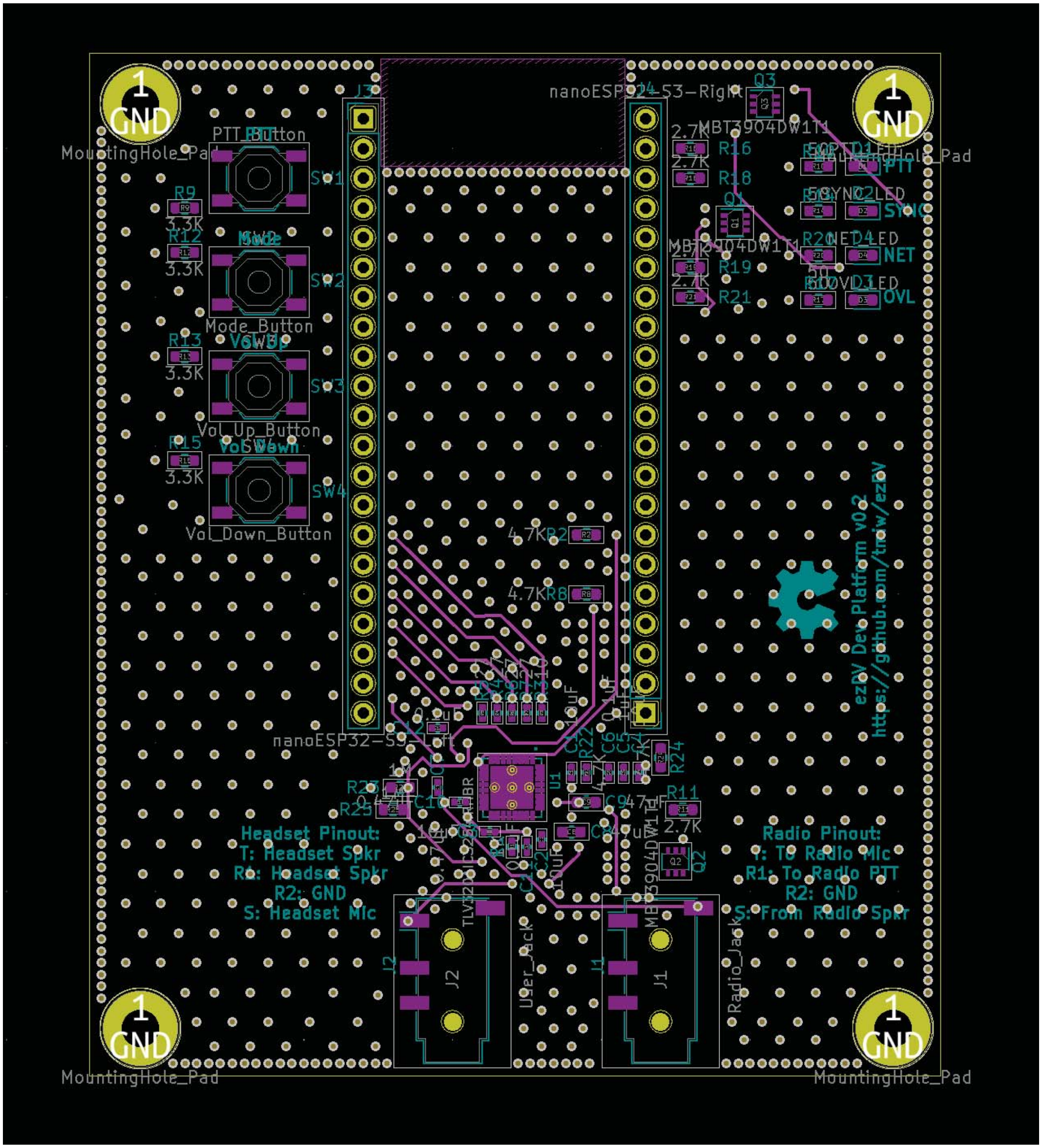


Figure 5: The bottom signal layer of the ezDV PCB.

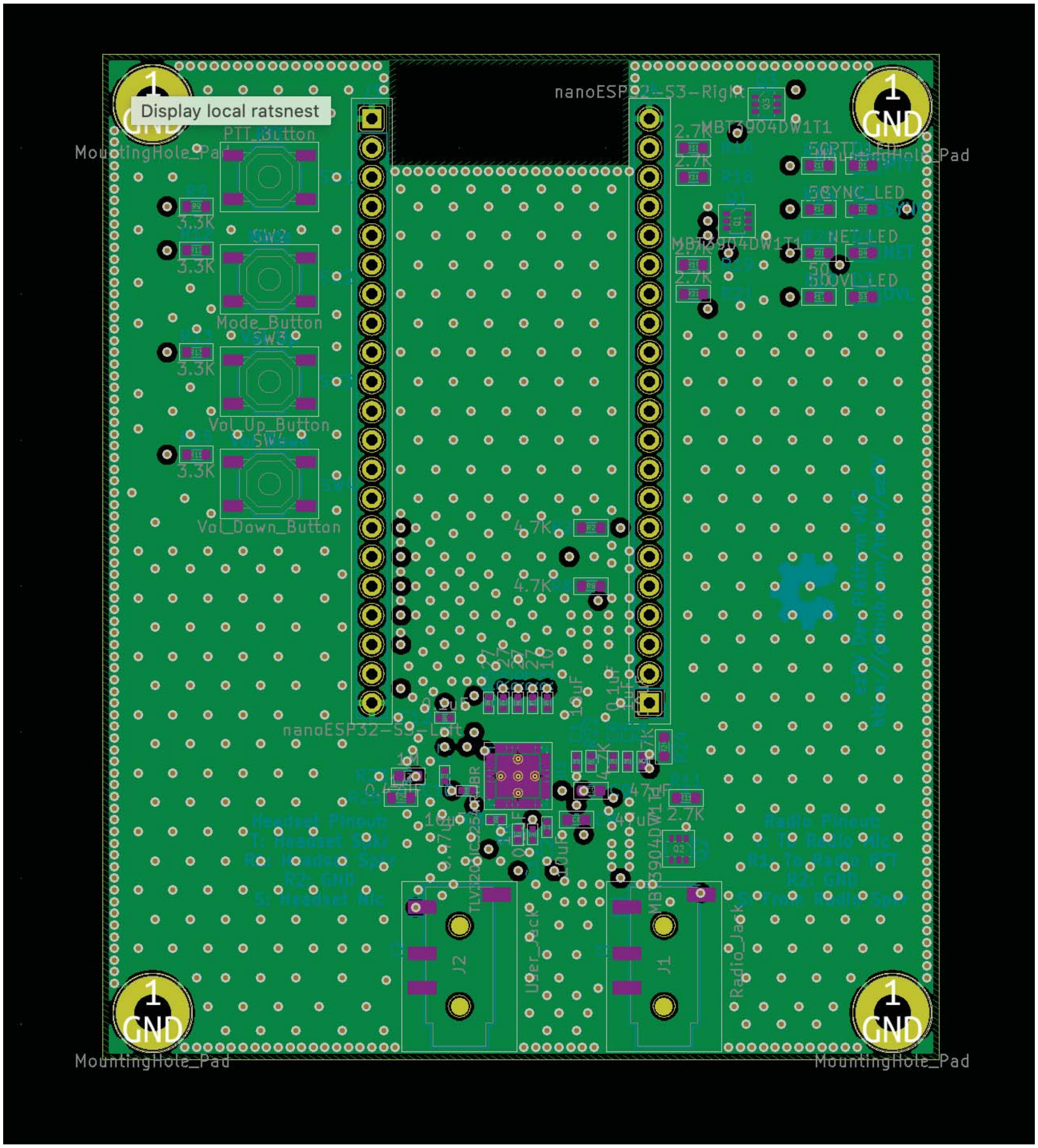


Figure 6: The bottom ground layer of the ezDV PCB.

Upon receiving the new revision boards, it was discovered that performing these changes did not resolve the audio problem. After further research into the TLV320 technical documentation, it was discovered that it had a loopback mode that bypassed the firmware entirely. Enabling this mode demonstrated that audio was being decoded and re-encoded successfully on both channels, indicating a problem in the firmware. Additional debugging of the firmware then found the issue causing the audio failures and thus allowed audio input and output on both jacks to work properly.

Building the Enclosure

The next step after producing a basic working prototype of the firmware and hardware was to build an enclosure to protect it during transport and use. This author purchased a Creality Ender 2 Pro 3D printer to print out prototypes of the enclosure due to its small size (which was good for storage and use in an apartment) and low cost. After some time learning how to use the printer for pre-made designs on websites such as Thingiverse, it was time to learn how to create 3D models. OpenSCAD was especially ideal for someone with a software development background as it allowed designs to be represented in the form of code[20].

One challenge was in determining how to produce buttons that stay with the enclosure (that is, those that don't easily fall out). Building the enclosure such that the buttons were larger on the top and bottom and had a narrow area to allow movement seemed like the obvious choice to enable this to happen (Figures 7 and 8). However, it was difficult to adjust the sizing properly to allow the buttons to freely move yet not fall out.

[20] "OpenSCAD CheatSheet." *OpenSCAD*, https://openscad.org/cheatsheet/.

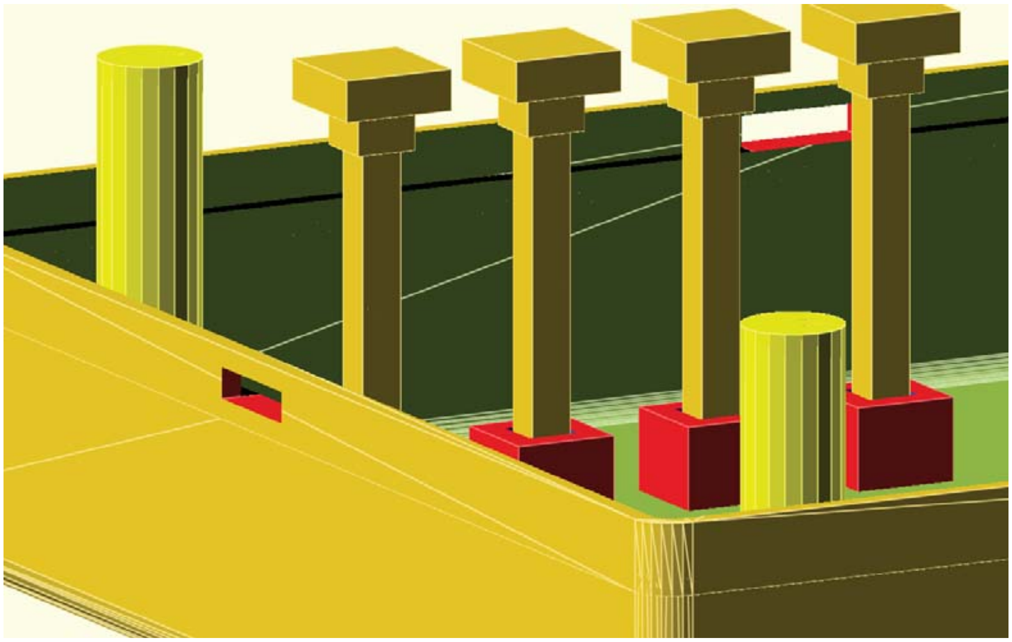


Figure 7: The ezDV enclosure buttons viewed from inside the enclosure.

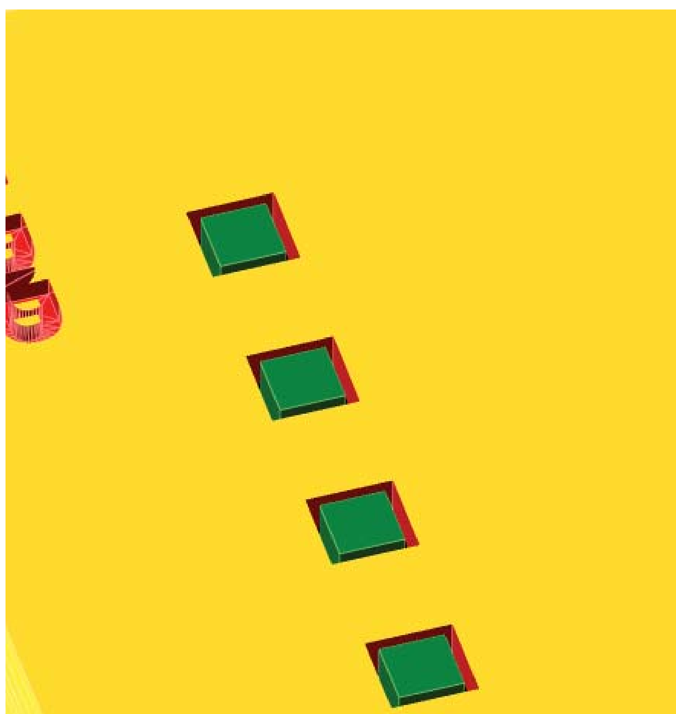


Figure 8: ezDV enclosure buttons viewed from the top of the enclosure.

After updating and printing many iterations of the design (to test various measurements for the buttons), a working version of the ezDV enclosure that allowed the buttons to move up and down and “click” on the PCB buttons resulted. This enclosure was printed using a transparent PETG filament (for added strength) and is shown below:

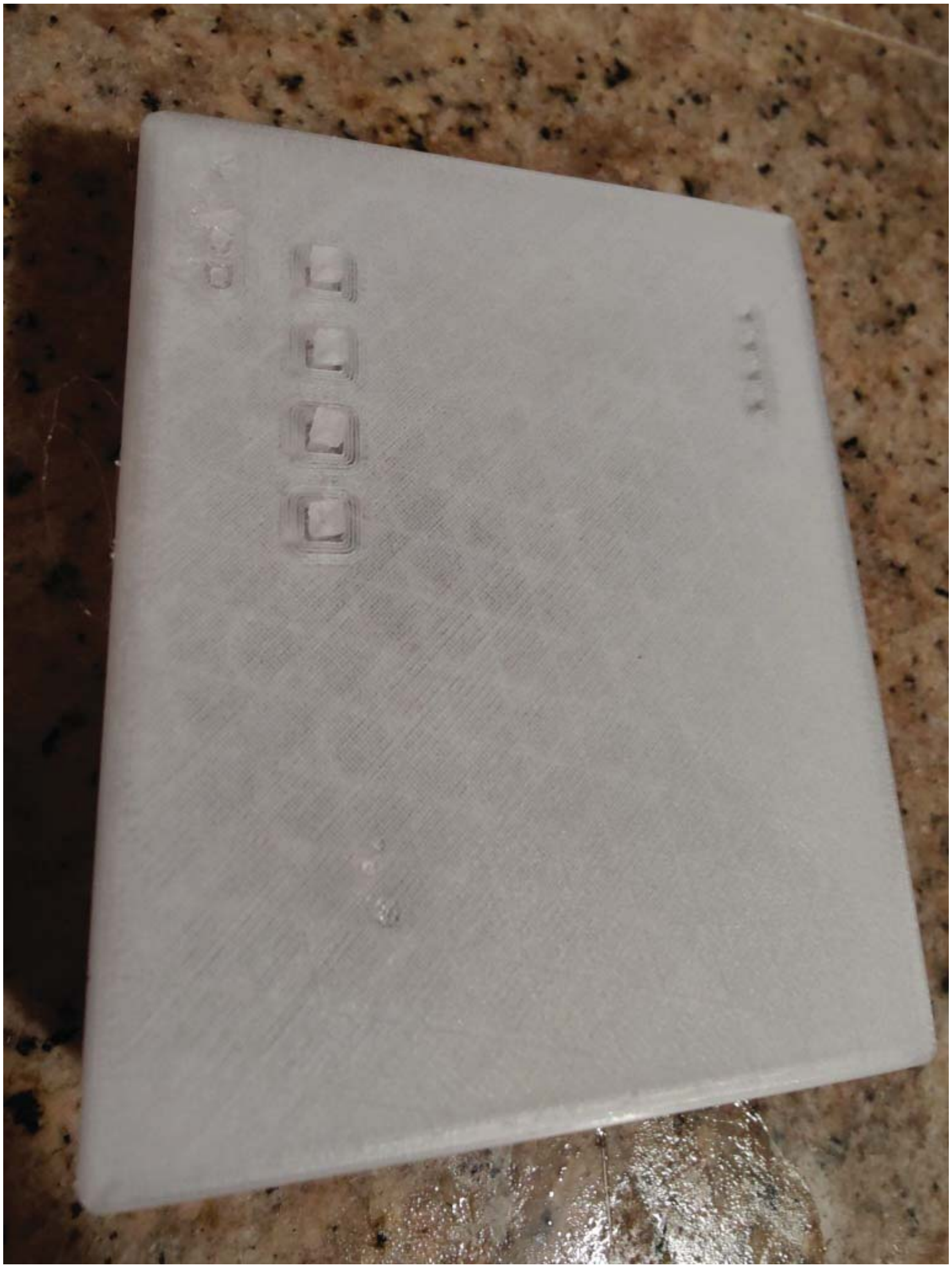


Figure 9: Top view of the 3D printed ezDV enclosure.

Lessons Learned and Next Steps

During the initial development of ezDV, a few things were learned:

1. Hardware development is more accessible than it has been, especially with online PCB assembly houses (such as JLCPCB and PCBWay) that only need a credit card and your Gerber files to produce small runs of boards. In the past, PCB assembly houses only worked with developers if they were willing to produce large runs, which would be problematic if an issue was found in the PCB design after the fact.
2. It's important to definitively rule out firmware bugs before jumping to hardware related issues. While having one audio channel with problems was suggestive of a hardware problem, it wasn't guaranteed to be one in retrospect, especially if the firmware does additional processing of the input or output.
3. The Codec2 library (and FreeDV more generally) are well suited for cross platform use due to the relatively low usage of platform specific code. Even on the PC, audio I/O is done through third party libraries (such as PortAudio and PulseAudio/pipewire) instead of being done directly.

There are also some things that would be good to add in future iterations:

1. Removing the dependency on the nanoESP32-S3 by integrating the ESP32-S3 directly onto the board. Doing so would result in a decrease in board size and cost as well as enable other functionality to be included in the space that was saved (for example, a built-in speaker/mic for analog audio or battery charging).
2. Perform testing using the original ESP32. If FreeDV performance is still acceptable on that part, that would enable use of standard Bluetooth instead of requiring a wired headset, resulting in only needing power to run the board (if also used with a radio that supports wireless connectivity such as the Icom IC-705).
3. Enable wireless connectivity. Some basic code to communicate with the IC-705 was prototyped, but it didn't have adequate reliability. Part of this effort would be to improve said reliability as well as possibly lower memory consumption (as it currently requires use of the PSRAM to run without crashing the ESP32). A Web based configuration interface could also be created to configure various components in the firmware, such as Wi-Fi network information and callsign (for PSK Reporter support).

Additional Information and Resources

ezDV is open source and available on GitHub at https://github.com/tmiw/ezDV for those interested in having their own embedded FreeDV solution. Pull requests are always welcome there, as well as the main FreeDV projects (https://github.com/drowe67/freedv-gui and https://github.com/drowe67/codec2).

Detecting field lines, as propagation, at fault lines with an Amateur HF-Radio

Goups.io user group: https://groups.io/g/MDSRadio

MDSR website: www.rf-seismograph.org

Alex Schwarz
VE7DXW
alexschwarz@telus.net

Detecting field lines as propagation at fault lines
A discovery that belongs to all Amateur radio operators

Radio propagation at fault-lines is different, more localized and usually better then the solar model of propagation indicates, especially in the lower bands. Amateur Radio Operators (HAMs) have flocked all over the world to live near areas that have fault lines. DX stations that are located in volcanic and seismically active earthquake zones are frequently visited by HAMs. They know that propagation is different, but they do not know why. This discovery might just answer a lot of questions, but it also creates more questions…

Types of faults and their effect on propagation:

a. Deep fault lines are weak points of earth's outer crust and are mostly located under the ocean. They allow internal energy and processes to leak out. This mostly affects the atmospheric D layer and makes fissures in the ionosphere. Released energy also pushes the plasma further away from the planet's surface. Deep faults create big earthquake events that are usually attenuating, but smaller aftershocks will still create propagation and can last for months afterward.

b. Fault lines in mountainous areas provide a break in the rock and allow the edges of the rock to vibrate. The deeper and longer the cracks, the more area of rock can vibrate. This event creates electricity and magnetic fields when the rock is dry, through piezoelectricity. These are very long, slow vibrations and are caused by the moon as well as by earthquakes. The earthquake depth data provided by USGS seem to indicate that quakes above sea level are more likely to create propagation.

c. The energy is detected by subtle propagation changes in the shortwave radio bands. Interestingly, only the small local quakes within 1000 km and below M4 create short (up to 1 hour) openings in the lower bands when the solar flux is below 100. As the solar flux increases the band openings shift to the higher bands, but they are always triggered by a quake event, which are occurring all the time.

d. Since the RF-Seismograph is located on a small mountainous deep fault, the measurement and earth quakes detected are the result of the energy released by this ground feature. The measured impact is very localized and specific to the fault line. Moving just 10 m away is measurable as a significant drop in propagation. Active faults are narrow and deep, which has a focusing effect to the electro-magnetic waves and they shoot straight up into the sky. The radiation is weak, but fault lines are long and have a huge underground surface area for the energy to escape.

What are the best fault lines for propagation?

The best fault is dry and deep and narrow. They can be found in mountainous terrain and deserts. Still, most faults are water saturated rock, which does not create electricity; the dryer the rock the more prominent the effect is. If the fissure is filled with water the attenuating effect stops the radiation from leaking out as well, which is why the effect is reduced during the raining season. Fault lines cover the planet like a spider web and they also become a conduit for seismic events to dissipate energy.

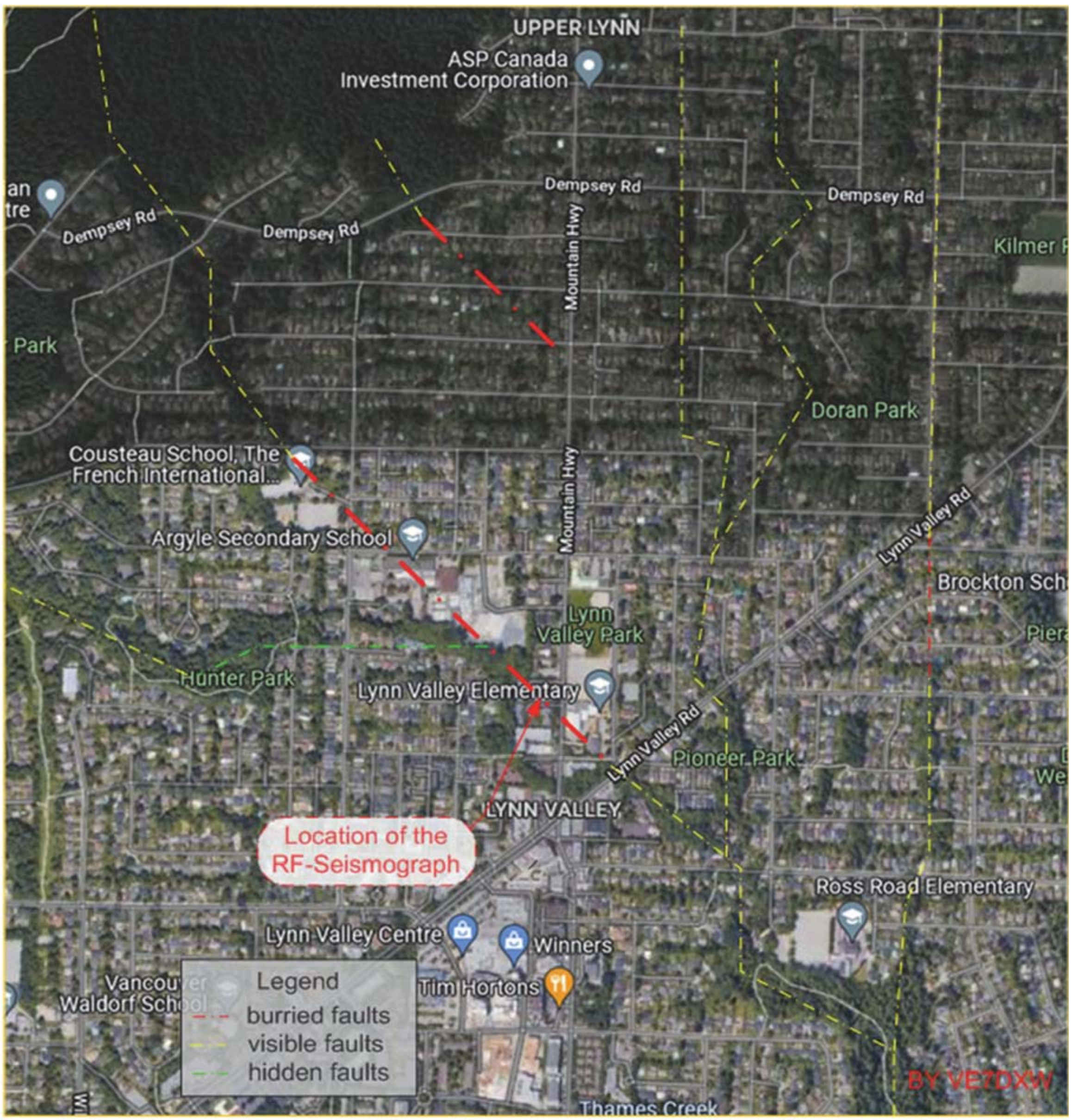
Weak spots in the ground can be measured, even if they are not visible with the naked eye, by using the EMF-390 tester.

Fault-lines are everywhere

Fault lines are misunderstood and a nuisance for most, but if we want to get the best propagation we need to find them and monitor them.

There are a lot of fault lines that are hidden and only when there is a quake within 500 km the fault becomes active. In order to see if a fault is active an EMF meter can be used that has a low cutoff frequency of 1 Hz or lower. I have been using the EMF-390 from GQ Electronics which is surprisingly affordable with great success.

Below is a map of the area with the RF-Seismograph in the center. It is located by sheer luck over a deep local fault that was hidden by development. Only if the antenna is over the fault the propagation data becomes interesting and will display quake activity.



Fault-lines found in the Lower Mainland

a.) Mount Seymour – near the RF-Seismograph -- has several fault-lines that are active. This is a fault that was listed by Earthquake Canada and it was active as the EMF-390 was indicating EMF radiation, while we were there.

b.) Small fault-line at Horseshoe Bay and White Cliff

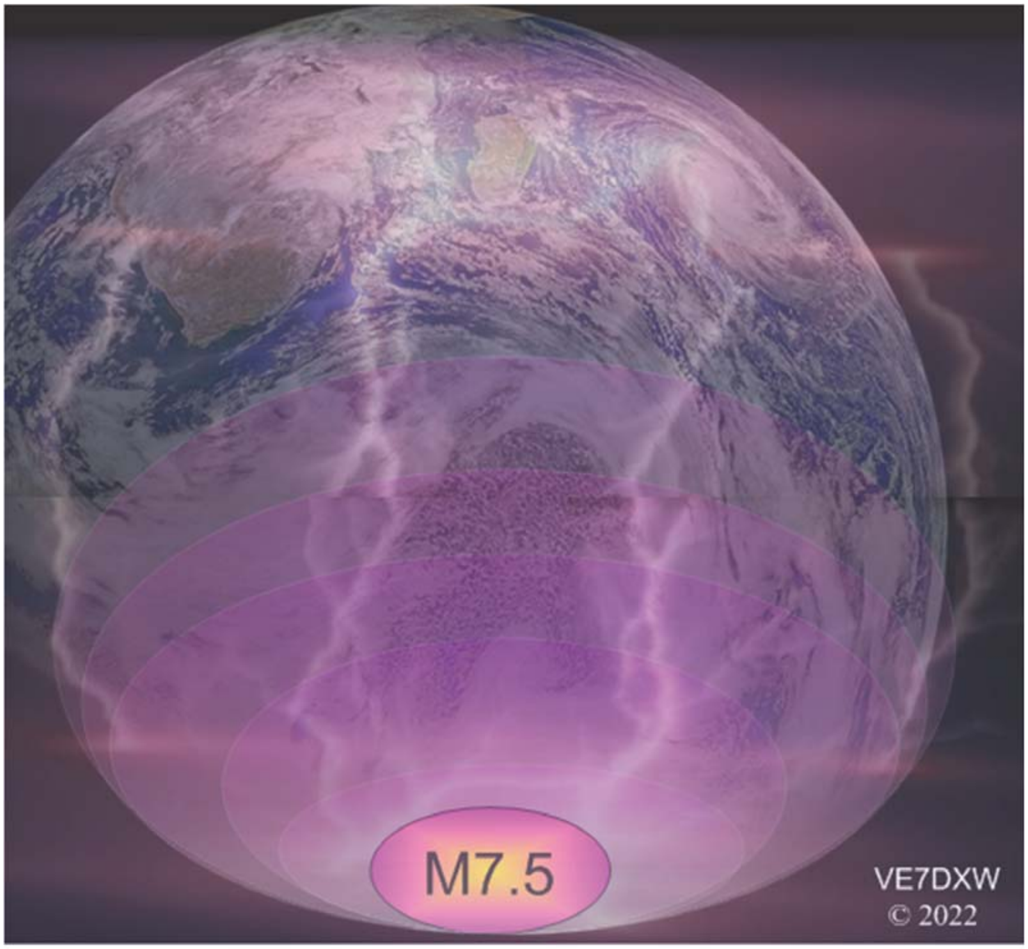
c.) Fault in upper Lynn Valley, which is at least 50 m wide. If there is plant life, it is always disturbed not only by the flow of the river, but also by the widening gap in perpendicular motion at the sides of the rift.

Where does the energy come from – is the moon supercharging the planet?

Energy from Quakes

Even if we do not include the electric effects of quakes, the mechanical shock wave of a quake > M6 nullifies any reflective layer and that is easily measurable with the RF-Seismograph or an Ionosonde. That was most visible with the M7.1 from Searles Valley in California at July 5th2019. The aftershocks though, changed propagation for month after and that was because of the vibrations of small aftershocks that fissure the rock. That rattles the fault lines all over the planet and in this process converts mechanical energy into electricity. Field lines pop out along all fault lines along the ground and as a result shortwave radio propagation changes.

M7.5 in the Falkland Islands shakes the planet creating an electrical tsunami (fictional)



When we convert the quake energy to a metric value there is not only a big number, but it also becomes something that can be integrated into the bigger picture. These are big numbers and the reason we have not integrated these effects, mostly located in the D-Layer is because it is measured in Richter scale – a logarithmic value and not metric. Because the M numbers are small, it leads us to underestimate the total amount of energy released. According to the Alabama University website M7.5 is really 1.122018 10^{+16} J of radiated waves power and 2.192895 10^{+20} J of Seismic Moment energy! Quakes or volcanic events that size do not occur very often so they do not add a lot of energy into the overall global energy budget.

<u>Data provided by Alabama University website:</u>

Earthquake Magnitude:	7.5	Enter Value
Seismic Energy in Waves Radiated from Earthquake Source:	1.122018e+16	Joules
Total "Seismic Moment Energy" (M_O):	2.192805e+20	Joules

Energy from the gravitational pull of the moon

Energy from tidal movements of water

It was estimated by the Ingenium Institute that 1 km of coast line can harness the tidal power of 21 GW of power at the highest tide area. If you add all the coastlines together and estimate on average that the tides only create 10 GW you get an estimate on the power of the global tidal cycle. There are 620000 km of coast line at 10 GW each, which ads 620 TW of power and It is generated by the change of the tides worldwide twice a day. The weight of water is bending the continental shelf and electricity is created by piezo electric energy, providing a shock absorber effect which keeps the oceans form eroding coast lines.

Energy from tidal movements of continents

The liquid waters most obvious effect of the lunar attraction pulls the water over coastal areas bending the coastal plates. Further, the gravitational pull of the moon raises the continents out of the lava bed they are sitting in and bends them, while the moon is overhead. Both processes create electricity and electromagnetic fields twice day in a very consistent way. If there is any energy build up in the crust small earthquakes are triggered, leveling the ground.
The moon is moving the continents up and down like a piston engine! This is supercharging the planet, creating electromagnetic energy that raises the elevation of the ionosphere.

Calculating the energy required to move the continents

The continents are floating on a sea of lava. As the moon moves above, measurements show that it lifts the continents between 11.4 cm to 35.6 cm away from the center of the earth and closer to the moon. If we can calculate the weight of the continents we can come up with the energy needed for this task.

- Lift of plates by the moon: 11.4 to 35.6 centimeters – average 25 cm
- Average depth of the continents is 40 km
- Average crustal density of 2700 kg/m^3
- Area of 150 x 10^9 m^2

Total weight of all the continents: M = area x crust thickness x density
150 x 10^9 m^2 x 40 x 10^3 m x 2700 kgm^{-3} = **1.62 x 10^{19} kg**

We have to use the ISO definition of mass; 1 kg is 9.8 N. Now we are able to calculate the work needed to move the continents by 25 cm; measured in Joules.

Work required moving the continents by 25 cm
W = 1.62 x 10^{19} x 9.8 ms^{-2} x 0.25 m = **4 x 10^{19} J = 40 EJ (exa = 10^{18})**

Process runs every 12 h and it creates: P = W / 43200 s = **9.26 x 10^{14} W of power**

Power from external sources mostly solar

Thermosphere Climate Index: 11.53 x 10^{10} W

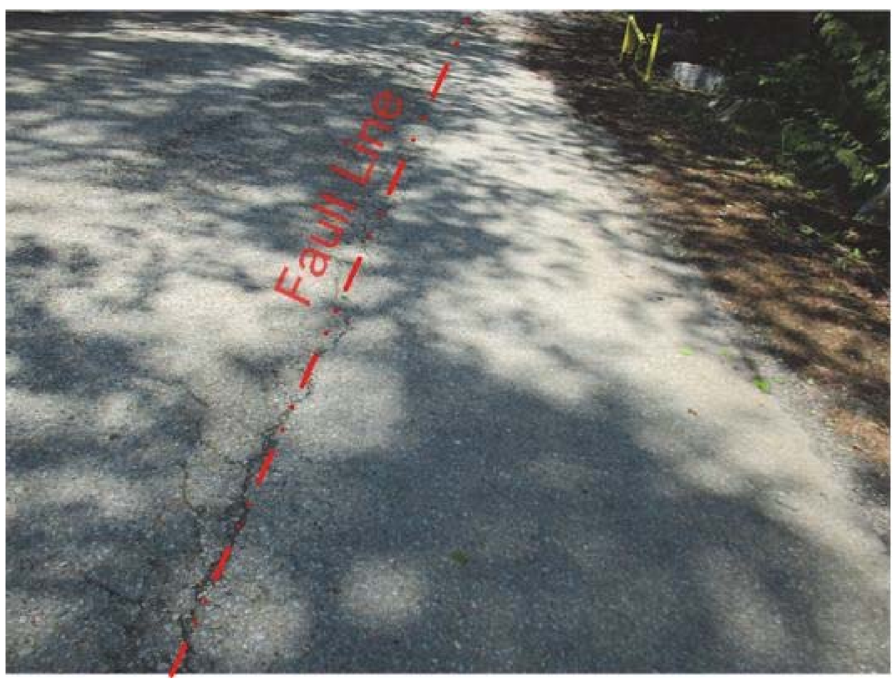
Earth solar received solar radiation: 1365.4 W/m2
Surface of earth: 510 x 10^9 m^2, half of it receives radiation amounts to 3.48 x 10^{14} W

Aurora energy HPI ranges from 5 GW to 200 GW

Field Day Experiment 2022

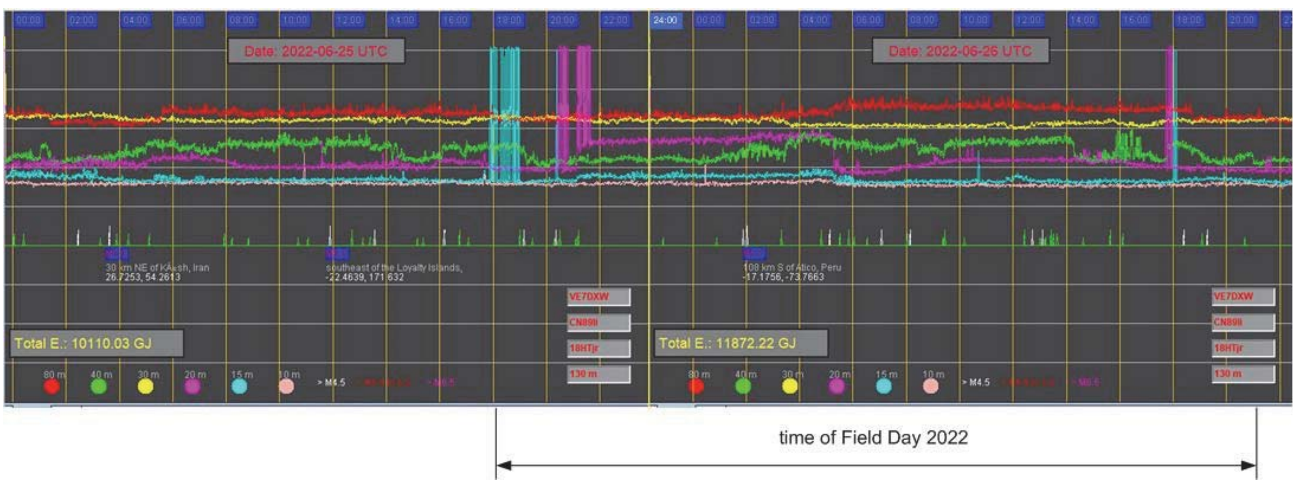
The statement that fault lines are everywhere holds really true, considering that we also found a sizable fault crossing our Field Day site! Visible clues are straight lines and cracks on pavement that break apart. As the fault widens the under-layer support material fills in the expanding gap, weakening the support and then the pavement cracks. The wide area view from Google Earth also reveals the fault coming down the mountain and across the field.

We have been using this site for many years, unaware of the fault and the potential danger during an earthquake.



The areal picture that was obtained using Google Earth also revealed the existing fault

Readings from the RF-Seismograph in Lynn Valley at a distance of 13 km



While we were trying to assess where the best place for the antenna for Field Day is, I was also using the EMF-390 to see if I could get a reading. As mentioned earlier, it was the same as the EMF reading that we were getting at the Seymour fault. Unfortunately we did not see a consistent indication and this can be contributed to two factors. First we had a very unusually cold spring and the ground was still wet. This keeps the underlying rock moist, inhibiting the piezoelectric effect. Secondly, if we look at the RF-Seismograph measurements, propagation was poor overall. Earthquake activity was at a minimum as well. Solar flux was at 130, which would have indicated better propagation overall. Earlier in 2022 we measured better propagation when the SF was below 100, but the quake activity was higher.

Since the NSARC has this field day site every year and it is only about 15 min away, we will keep on monitoring and if something worthwhile changes we will revisit the site and will provide an update.

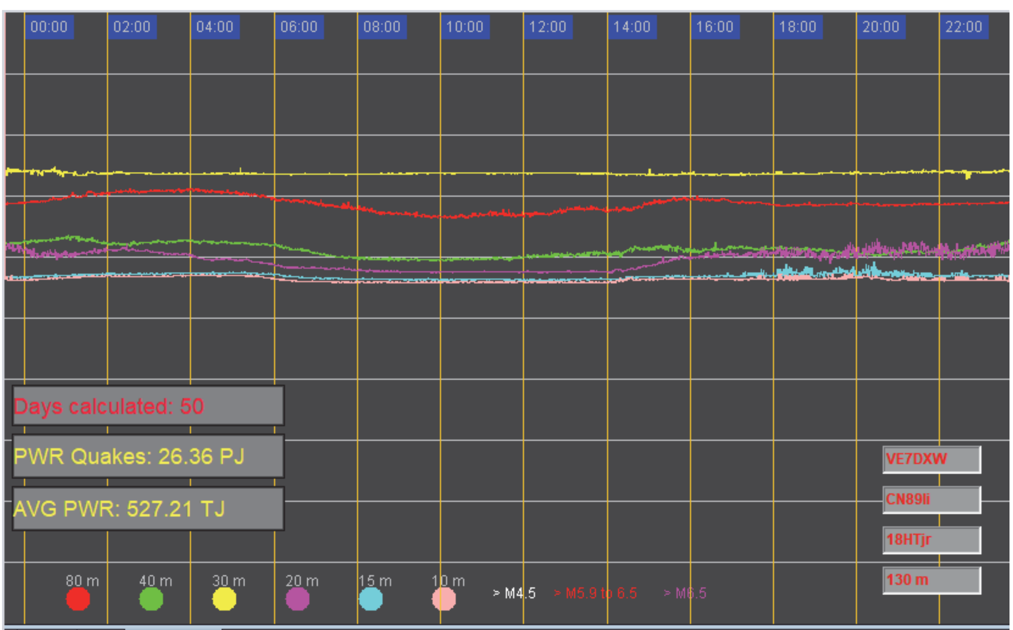
Conclusion of Field Day Experiment (25 % improvement of contacts)
The overall score was higher by 25 %, but Solar Flux also temporally higher topping out at 133 for the day. There were no local quakes and the total energy released by quakes was 11.872 GJ for the day, way below the 50 day running average of 527 TJ.

What are we planning next

Below is a preview of the new 50-day running average graph that will be released soon and be hosted on our website.

<u>Scientific goal:</u> getting a 50 running average allows us to estimate the total quake output and to estimate the average daily energy output of the quakes. Further, we can estimate if the risk of a big quake is higher, because of built up energy in the crust.

50 day running average created by new addition of the RF-Viewer, to be released shortly



Conclusion

When all the energy sources are added together, the energy transferred from moon to earth outnumbers the solar energy by at least a factor of 2! These energy transfers are also occurring on Jupiter and Saturn with their unusually warm moons.
Because earth and the moon are very unique, the combination has unforeseen consequences via gravitational energy transfer, and this has to be investigated further. If the fault lines are starting to radiate more and worldwide communication is easily obtained, could this mean a golden age for shortwave radio?

References

Article; using an airborne fault line detector in Salt Lake City.
https://www.abc4.com/news/top-stories/earthquake-fault-lines-new-study-pinpoints-wasatch-fault-zones/

Website for EMF-390: www.GQElectronics.com

The nature of fault lines
https://www.youtube.com/watch?v=qlk7IfYMufs

Convert the M Richter scale to metric values into moment and wave energy; ISO values of Joules
https://earthalabama.com/energy.html#/

Connection between small quakes and the moon
https://swisscows.com/video/watch?query=is%20there%20a%20cfonnection%20between%20the%20moon%20and%20earthquakes&region=iv&id=34670A7D6C8F8554937434670A7D6C8F85549374

Tidal energy
https://energy.techno-science.ca/en/energy101/tidal.php

The lifting of the continents
https://www.papertrell.com/apps/preview/The-Handy-Science-Answer-Book/Handy%20Answer%20book/Do-the-continents-move/001137021/content/SC/52cb009082fad14abfa5c2e0_Default.html

How much do all the continents weigh?
https://www.quora.com/How-much-does-the-continent-of-Asia-weigh

Total land mass of planet earth
https://hypertextbook.com/facts/2001/DanielChen.shtml

Thermosphere Climate Index
https://www.spaceweather.com/

https://earth.gsfc.nasa.gov/climate/research/solar-radiation

https://ourworldindata.org/energy-production-consumption#how-much-energy-does-the-world-consume

https://www.energy.gov/eere/solar/solar-radiation-basics

Received radiation
https://earth.gsfc.nasa.gov/climate/research/solar-radiation

Google Earth: https://earth.google.com/

A Brief Survey of Technological Innovation in Amateur Radio

By Steve Stroh N8GNJ[1]

ARRL and TAPR 2022 Digital Communications Conference
Charlotte, North Carolina, USA 2022-09-16 thru 18

Abstract

In recent decades, the perception of Amateur Radio within the general public has shifted from Amateur Radio being useful, innovative, and an interesting technical activity, to Amateur Radio being perceived as an anachronism and largely irrelevant (except in the direst of communications emergencies). Summarized: “Ham Radio – that’s still around?”

Amateur Radio’s service to the public for emergency communications is being supplanted by improved commercial and government communications capabilities such as improved Iridium[2] satellite phones, the FirstNET[3] public safety cellular system, and most recently, the nomadic capability of the Starlink[4] broadband satellite system.

Amateur Radio has continuously developed unique technological innovations in radio technology, and that has not only continued in the modern era but has *accelerated*. However, that ongoing, unique contribution to technological society is, increasingly, unrecognized. That is unfortunate. *If* regulators, lawmakers, industry, the general public… and the Amateur Radio community itself understood the unique contributions to technological innovations in radio technology that Amateur Radio continues to develop, perhaps such recognition might improve Amateur Radio’s perception that it remains a valuable part of society, worthy of continued access to portions of the electromagnetic spectrum.

Keywords

Amateur, Radio, Operator, Ham, Wireless, Technology, Innovation, Spectrum, Digital, VHF, UHF, SHF, Microwave, Communications, ARDC, Techies, Makers, Hackers, Zero Retries Newsletter, Experimentation, Research and Development, FlexRadio, Steve Stroh N8GNJ

Background

For decades, I have been an admirer of technological innovation in Amateur Radio. Not just new technologies like Packet Radio emerging in the 1980s, but new techniques for old problems such as digital techniques enabling reliable communications via unreliable mediums such as the High Frequency (HF)[5] (aka Shortwave) portions of the electromagnetic spectrum.

Amateur Radio’s unique culture, the varying characteristics of various portions of spectrum allocated to (or shared with) Amateur Radio operations, and the many highly capable and skilled Amateur Radio Operators, have resulted in a fertile, and welcoming “experimental zone” for technological innovation in radio technologies. Until recent decades, that culture of technological innovation was widely recognized, and encouraged. In the last few decades, the recognition of

---

[1] Email – stevestroh@gmail.com
[2] https://www.iridium.com/network/
[3] https://firstnet.gov/network
[4] https://www.starlink.com/rv
[5] https://en.wikipedia.org/wiki/High_frequency

Amateur Radio's utility and contributions to technological innovation have been deprecated to near irrelevence… at least in popular perception… by ubiquitous Internet access, mobile phones, caricatures of Amateur Radio as "Grandpa sitting in the basement tapping on a Morse Code key", and most notably, the removal of old barriers to individuals communicating across international borders.

A primary reason that this is a concern for society is that it has become irrevocably dependent on radio technology as the primary method of communications for mobile devices, most notably cellular technology, wireless local area networks (Wi-Fi), and most recently, direct-to-user satellite communications. For many people, their mobile phone is their only method of communications and media consumption. Much of that technology has been developed and manufactured in China. Dependence on China for such a critical infrastructure function is proving to be fraught with peril. To counter that peril, the US and other Western nations must quickly develop additional expertise, and personnel, "in nation" to better develop and support this now-critical wireless infrastructure. Amateur Radio can be a "training ground" for developing familiarity and expertise with radio technology, leading to careers in developing and supporting radio technology… but only if Amateur Radio is recognized as a useful and interesting.

The rise of technology specialists, especially those trained in Information Technology (IT), the "Maker culture"[6], and the "Hacking Culture"[7] have breathed new life into Amateur Radio. "Techies" have discovered Amateur Radio as an enabling technology for supporting experimentation with Information Technologies (such as building hobbyist / not-for-profit wide-area microwave networks). Makers have discovered that there are incredibly interesting things that they can add to their personal knowledge base and practical projects based on capabilities Amateur Radio has long taken for granted, such as long-range communications via VHF / UHF repeaters. Hackers have discovered Amateur Radio as a fertile "playground" for their experiments and expansion of knowledge about radio technology, such as Software Defined Receivers… and Transmitters (*with* an Amateur Radio license).

I started the Zero Retries Newsletter[8] in July, 2021 out of frustration that the totality of technological innovation in Amateur Radio wasn't being recognized by the Amateur Radio community, its regulators, and especially the public at large. Specifically, I was worried about the growing public perception that Amateur Radio is irrelevant, or worse, an anachronism. Such a perception, if it is to continue for much longer, may prove catastrophic to Amateur Radio, most notably in the loss of Amateur Radio access to various portions of spectrum. To date I've published more than fifty weekly issues of Zero Retries, and each issue highlights some aspect of technological innovation in Amateur Radio.

Literally, Amateur Radio is a license to experiment with radio technology and a welcoming "innovation zone" to develop new and exciting technological innovations in radio technology. I hope to make that point with the vignettes in this paper.

[6] https://en.wikipedia.org/wiki/Maker_culture
[7] https://en.wikipedia.org/wiki/Hacker_culture
[8] https://zeroretries.substack.com (will eventually migrate to https://zeroretries.org)

Amateur Radio Digital Communications[9]

One of the most significant factors regarding technological innovation in Amateur Radio is the recent emergence of Amateur Radio Digital Communications (ARDC)[10] as a funding source for innovative projects and organizations. Many promising Amateur Radio projects die out before completion because of lack of resources, especially expensive and unavailable expertise in radio frequency engineering, requirements for expensive test equipment, "only professionals can afford it" design software, the expense of prototype manufacturing, etc.

ARDC was formed to manage 44Net[11], the Class A Internet Protocol v4 Internet address block of ~16 million contiguous IPv4 addresses. A few years ago, ARDC sold a contiguous block of ~4 million IPv4 addresses, and with the proceeds of that sale, reorganized itself as a private foundation and created an endowment fund. ARDC invested its endowment prudently, and from that investment it distributes 5% (minimum) annually of its endowment in the form of grants[12], funds the minimal expenses of the organization including staff, contractors, and overhead, and continues to operate 44Net.

By mid-2021, ARDC was fully staffed, including a volunteer Grants Advisory Committee, and since then has funded many grants, large and small[13]. While ARDC isn't the only grant making organization focused on Amateur Radio, ARDC is unique in the size and scope of its grants to Amateur Radio. The grants that ARDC provides can be transformational to organizations and projects. Listed below are a few grants that reflect the technological innovation in Amateur Radio that ARDC grants are empowering:

- ARDC's largest grant to date was $1.6 million to repair and refurbish "**The Big Dish**" at Massachusetts Institute of Technology (MIT)[14]. This dish was originally installed on the roof of the 22-story Green Building on the MIT campus to develop weather RADAR technology. After that project was complete, the dish was turned over to the MIT Radio Society (Amateur Radio club) for experimentation. The dish has been used for Earth Moon Earth (EME) communications, radio astronomy, Amateur Radio VHF / UHF / Microwave contesting, and many other innovative experiments. The Green Building was slated for renovation, including the roof, and MIT planned to remove the dish. The MIT Radio Society was able to convince MIT to give them a chance to raise private funds for the repair and refurbishment of the dish, including a new fiberglass radome. With only a few months before the Green Building work was to commence, approximately $300,000 of the required $1.9 million had been raised. Fortunately, ARDC was able to step in with the balance of funds needed, and with the required funds secured, "The Big Dish" will be returned to the roof of the Green Building including all required structural updates, a new radome, new mechanicals, etc.

---

[9] Disclaimer – In 2021 and 2022, the author is a volunteer member of ARDC's Grants Advisory Committee. The views expressed here are his own, not intended to reflect, or speak for, the views of ARDC. This paper is written entirely independent of ARDC.
[10] https://www.ampr.org
[11] https://wiki.ampr.org/wiki/Main_Page
[12] https://www.ampr.org/grants/
[13] It should be noted that ARDC grants have also funded numerous projects for Amateur Radio clubs (such as new stations, new repeaters, new trailers, professional tower climbing), funded numerous scholarships, and funded research and development projects (unrelated to Amateur Radio), and significant assistance to Open Source work.
[14] https://www.ampr.org/grants/2021-grants/grant-mit-radio-society-radome-renewal/

- **Amateur Radio on the International Space Station (ARISS)**[15] has received several ARDC grants, including an early grant to help fund the ARISS Next Generation Radio[16] and a five-year $1.3 million grant[17] to develop new curriculum material and hands-on experiments for classroom Science, Technology, Engineering, Arts, and Math (STEAM) lessons that can be provided to teachers in conjunction with space studies.
- Although not specific to Amateur Radio, **GNU Radio Project**[18] is a vibrant, Open Source project for Software Defined Radio that is used widely for commercial and academic research, as well as independent experimentation. GNU Radio significantly influences and enhances Amateur Radio and is undeniably a source of technological innovation. It's sometimes said that if you "can't do [a radio technology] in GNU Radio… it can't be done - *yet*". ARDC provided an early grant[19] to GNU Radio Project, but its most recent grant[20] is particularly notable in funding usability improvements for GNU Radio, including improvements to the Windows OS version, and documentation improvements.
- The **M17 Project** is an international coalition of volunteers whose goal is to create an Open Source ecosystem for Digital Voice for radio communications – software, hardware, on-air protocols, networking, etc. Think of D-Star[21] or Digital Mobile Radio (DMR)[22] but with only Open Source technology, including the use of Codec 2[23] vocoder subsystem (that in D-Star and DMR is implemented with a proprietary technology). An ARDC grant[24] enabled the M17 Project to purchase test equipment and other significant expenses incurred in development. One project of the M17 Project is development of Mini17[25], an Open Source low-power portable radio.
- **Rhizomatica** is a not-for-profit organization that develops communications systems applicable for the developing world such as parts of South America that do not have commercial communications infrastructure. Rhizomatica makes interesting use of HF technology and an ARDC grant[26] has enabled it to further develop Open Source solutions for expensive, proprietary systems such as higher speed modems used on HF.
- An ARDC grant funded a first of its kind "Amateur Radio Universal Online Library" within the Internet Archive[27] called **Digital Library of Amateur Radio & Communications**[28]. Such a project is expansive, intended to create a universal, searchable resource on Amateur Radio literature, software, radio information, publications, etc. This is coming just in time as many Amateur Radio Operators are "aging out" and much of their valuable and unique data is disappearing as their estates are liquidated. While it will be years before DLARC will emerge as a useful resource, but DLARC, will soon be a place to donate such information so it can be preserved for posterity.
- Most (all?) Amateur Radio satellites to date have used solar panels bonded to their structure, because moving parts in space are hard to engineer and if it breaks, it cannot

[15] https://www.ariss.org
[16] https://www.ampr.org/grants/2019-grants/grant-ariss-next-generation-radio/
[17] https://www.ampr.org/grants/2021-grants/grant-ariss-usa-program/
[18] https://www.gnuradio.org
[19] https://www.ampr.org/grants/2020-grants/grant-gnu-radio-project/
[20] https://www.ampr.org/grants/2022-grants/grant-gnu-radio-usability-enhancements/
[21] https://en.wikipedia.org/wiki/D-STAR
[22] https://en.wikipedia.org/wiki/Digital_mobile_radio
[23] https://en.wikipedia.org/wiki/Codec_2
[24] https://www.ampr.org/grants/2021-grants/grant-m17-open-protocol/
[25] https://github.com/M17-Project/Mini17
[26] https://www.ampr.org/grants/2021-grants/grant-digital-hf-telecommunications-for-civil-and-amateur-uses/
[27] https://archive.org
[28] https://www.ampr.org/grants/2021-grants/grant-building-the-digital-library-of-amateur-radio-communications/

be repaired. An ARDC grant[29] to **AMSAT**[30] is funding the development of a 3U satellite frame with deployable solar panels that will provide the electrical power required higher power operations in highly elliptical orbits that will be used by future AMSAT satellites.

- In conjunction with Ham Radio Science Citizen Investigation (HamSCI)[31], TAPR[32] is developing the **TangerineSDR**[33] [34] [35] [36], an Open Source modular HF radio that will be used, in part for HamSCI's scientific investigation of High Frequency (HF) radio propagation and other scientific experiments.
- Another new radio system under development is the **RPX-100**[37] being developed by the Austrian Amateur Radio Society (ÖVSV)[38]. The RPX-100 will operate on the 50-54 MHz (6 meter band), 144-148 MHz (2 meter band) and 430-450 MHz (70 centimeter band) using new protocols and high speeds. An ARDC grant[39] will help accelerate this project.

Again, the projects mentioned here are only a few highlights (of *many* grants) to emphasize the transformative and accelerative effect that ARDC grants are enabling for technological innovation in Amateur Radio (and related radio technology fields).

It's hoped that in the coming years, ARDC can apply its resources to support even more technological innovation in Amateur Radio, such as supporting the use of FCC Special Temporary Authority authorizations and Part 5 Experimental licenses, coordinating large scale testing of new paradigms in Amateur Radio for potential regulatory updates, coordinating standards bodies of Amateur Radio vendors, etc.

Amateur Radio Integration with Internet

As mentioned in the previous section, Amateur Radio was granted early access to the Internet, and there have been numerous technological innovations resulting from that long experience and familiarity with synergies that can be applied between radios and Internet, such as Winlink (mentioned in the next section).

Brandmeister

Brandmeister[40] is a decentralized network for Amateur Radio digital voice repeaters that encourages experimentation. A significant feature of Brandmeister is that *multiple digital voice systems* (not just Digital Mobile Radio – DMR) are "co-equal" on Brandmeister. Another feature is that text messaging and position beaconing is not just possible on Brandmeister – it's encouraged. (Text messages and position beaconing are often not allowed on some [more fragile] Amateur Radio digital voice networks.)

Networks of Radios Via Internet

---

[29] https://www.ampr.org/grants/2022-grants/grant-develop-a-3u-open-source-cubesat-space-frame-with-deployable-solar-panels/
[30] https://www.amsat.org
[31] https://hamsci.org
[32] https://tapr.org
[33] https://tangerinesdr.com
[34] https://www.ampr.org/grants/2020-grants/grant-tapr-tangerinesdr-prototype-build/
[35] https://www.ampr.org/grants/2021-grants/grant-tangerinesdr-test-fixtures/
[36] https://www.ampr.org/grants/2022-grants/grant-tangerinesdr-project-advancement/
[37] https://rpx-100.net
[38] https://oevsv.at/home/
[39] https://www.ampr.org/grants/2022-grants/grant-wireless-regional-area-network-in-sub-ghz-bands-as-last-mile-for-hamnet/
[40] https://wiki.brandmeister.network/index.php/What_is_BrandMeister

As discussed earlier in this paper, Amateur Radio was *early to experiment with integrating radios via the Internet* because of the early allocation of 44net. There are *many* Internet-connected networks of Amateur Radio units, including innumerable networks of repeaters such as Brandmeister (mentioned previously). It's notable that such networks are built and maintained on a non-commercial basis, mostly for the use by Amateur Radio Operators:

- **aprs.fi**[41] – Largest and best-known website for displaying aggregated APRS position, weather, and other tactical information from Amateur Radio stations.
- **AREDN** (mentioned in the next section) provides optional Internet connectivity, largely as "training wheels" to help new AREDN users get familiar with AREDN and remain connected to other AREDN users in their area, in preparation for getting an AREDN node on the air.
- **Personal Space Weather Station (PSWS)**[42] – Project of Ham Radio Science Citizen Investigation (HamSCI) to develop a network of radio receivers and other instruments such as magnetometers.
- **PSK Reporter** (Digimode Automatic Propagation Reporter)[43] – Network of receivers monitoring Amateur Radio data transmissions on HF.
- **Receiverbook**[44] – Directory of online Software Defined Receivers available for public use.
- **Reverse Beacon Network**[45] – Transmit on HF and see where your transmission was heard worldwide.
- **SatNOGS**[46] – Network of receivers focused on tracking and receiving low earth orbit (LEO) research satellites, especially those built by students. SatNOGS stations track satellites, download data locally, do some local processing, and then upload the raw and processed data for the researchers. Ground stations can be built relatively inexpensively, including some parts for the tracking hardware that can be 3D printed. SatNOGS is not entirely Amateur Radio.
- **SondeHub**[47] – Network of receivers that monitor weather radiosonde[48] transmitters, and radiosondes repurposed for Amateur Radio use.
- **Weak Signal Propagation Reporter Network (WSPRNet)**[49] – Monitors for WSPR transmissions and displays where your WSPR transmission was heard, worldwide.
- **WebSDRs**[50] – A WebSDR is a Software Defined Receiver connected to the Internet, allowing many listeners to listen and tune it simultaneously. SDR technology makes it possible that all listeners tune independently, and thus listen to different signals on different frequencies. There are *many* SD Receivers available for use on the Internet, findable via this page.

---

41 https://aprs.fi
42 https://hamsci.org/basic-project/personal-space-weather-station
43 https://pskreporter.info
44 https://www.receiverbook.de
45 http://www.reversebeacon.net/
46 https://satnogs.org/
47 https://sondehub.org
48 https://en.wikipedia.org/wiki/Radiosonde
49 http://wsprnet.org/drupal/
50 http://www.websdr.org

Multipurpose Remote Nodes (MRNs)

Several Multipurpose Remote Nodes (MRNs)[51] have been deployed in Whatcom County (Bellingham area), Washington USA. MRNs are unique in that they are located at remote sites, are connected to Internet, and their usage and radio parameters can be reconfigured remotely. While each MRN has a primary function – Winlink Radio Mail Server (RMS) on VHF / UHF[52], APRS Digipeater[53] and Igate[54], or fldigi (fsq mode)[55] relay, they can be remotely reconfigured as needed.

RadioID.net

Sometimes the complicated things can get fixed by applying the right idea. Digital Mobile Radio (DMR) was intended as a radio system for organizations, and thus each DMR *system* is licensed (such as a factory), not individual radios or users. When Amateur Radio began using DMR, one of the first issues with using DMR in Amateur Radio was that DMR radios transmit only “ID numbers”; there was no requirement to transmit a callsign. Thus, a database had to be established to issue, and cross reference DMR ID numbers to Amateur Radio callsigns. But, being modern Amateur Radio Operators, we quickly started to network DMR repeaters, and thus found the second major “Amateur Radio deficiency” of DMR – it was easy to create duplicate ID numbers, which can play havoc with routing within networks of DMR repeaters. As DMR usage became more common in Amateur Radio, various databases of Amateur Radio DMR IDs were established for various communities (such as Motorola DMR users). However, *keeping* those databases consistent and synchronized was problematic and time consuming. Eventually, a single, authoritative database of Amateur Radio (and other) DMR IDs emerged that was universally accessible via the Internet - RadioID.net[56].

Amateur Radio *Radio* Technological Innovation

Some might argue that it’s harder to create new radio technologies in this era, but if so, there’s still ample technological innovation occurring in Amateur Radio in the development of new radio units and systems. Besides the examples of radio development funded by ARDC grants, these are a few examples of *radio* technological innovation.

Amateur Radio Emergency Digital Network (AREDN)

AREDN[57] is an outgrowth of earlier projects to repurpose commercial microwave communications units such as Wi-Fi access points and microwave equipment intended for use by enterprises and Wireless Internet Service Providers (WISPs).

AREDN firmware was originally based on OpenWRT[58] but has diverged significantly from that technology as AREDN has been optimized for use in Amateur Radio. Notably, there have been three significant AREDN firmware releases to date in 2022.

AREDN develops replacement firmware for these units that add features specific for Amateur Radio use, including:

- Use of semi-exclusive portions of spectrum allocated to Amateur Radio,

---

51 https://zeroretries.substack.com/p/zero-retries-0010
52 https://www.winlink.org/content/rms_packet
53 http://www.aprs.net.au/vhf/aprs-digipeaters-101/
54 http://www.aprs-is.net/igating.aspx
55 http://www.w1hkj.com
56 https://www.radioid.net
57 https://www.arednmesh.org
58 https://openwrt.org

- Automatic network and route discovery,
- Automatic configuration of networking parameters such as IP address assignment, gateway configuration, etc.,
- Seamless interoperability between Ethernet connections, radio connections, and Internet connections (tunneling between AREDN nodes).

What is most impressive about AREDN is the “Automatic network and route discovery” feature. There have been innumerable attempts to implement “open” mesh networking, and mesh networking was even codified into an IEEE standard for Wi-Fi – 802.11s[59]. But all such popular, usable implementations of mesh networking have been proprietary: “Brand X” Wi-Fi and “Brand Y” Wi-Fi will not automatically recognize each other’s mesh network capabilities. AREDN solves that dysfunction: any AREDN devices that are on the same frequency (and same channel size – 5, 10, or 20 MHz) will recognize each other and “just mesh up” regardless of hardware manufacturer or model. This provides a unique capability to Amateur Radio Operators – not only access to some portions of spectrum that are semi-exclusive to Amateur Radio, but the “it just works” capability to local area and wide area high speed microwave networking.

AREDN Networks have been formed in many areas of the US and internationally, and it’s been reported that some Information Technology (IT) professionals have gotten their Amateur Radio licenses specifically to work with Amateur Radio Operators to build out and experiment with AREDN networks.

New Packet Radio (NPR)
Despite the name, New Packet Radio[60] has no overlap with classic Amateur Radio Packet Radio: An NPR unit communicates via Ethernet, uses TCP/IP natively over the air, and communicates at speeds up to 500 kbps using a 100 kHz channel. NPR is Open Source, and can be built for < $100, or assembled and tested units (now in their fifth generation) are available for purchase[61].

NPR is notable that upon its debut in 2019, it was a one-person project by F4FDK and is an example of how much innovation can be done in Amateur Radio. NPR can be used as point-to-point, point-to-multipoint, or in a “repeater” configuration. Although it was originally intended as a “feeder” for high-speed microwave networks such as HAMNET[62] and AREDN, it is fast enough and useful enough to operate as a standalone network. The NPR unit does not generate much RF transmit power, but NPR is designed to be able to use commonly available power amplifiers intended for use with Digital Mobile Radio (DMR) portable radios.

Open IP over VHF / UHF
David Rowe VK5DGR[63] is creating a system he calls Open IP[64] that will do native TCP/IP over VHF / UHF frequencies, at a data rate of up to 100 kbps, at a range up to 15 km (urban). The

---

[59] https://en.wikipedia.org/wiki/IEEE_802.11s
[60] https://hackaday.io/project/164092-npr-new-packet-radio
[61] https://elekitsorparts.com/product/npr-70-modem-by-f4hdk-new-packet-radio-over-70cm-band-amateur-radio-packet-radio/
[62] https://hamnet.eu/site/
[63] https://www.rowetel.com
[64] **Part 1** - http://www.rowetel.com/?p=7207, **Part 2** - http://www.rowetel.com/?p=7334, **Part 3** - http://www.rowetel.com/?p=7567, **Part 4** - http://www.rowetel.com/?p=7567, **Part 5** - http://www.rowetel.com/?p=7898.

transmitter is a Raspberry Pi. The receiver is a generic RTL-SDR[65] dongle. This system is *almost entirely software*. It sounds… speculative… but I'm not betting against VK5DGR.

Software Defined Transceivers using Raspberry Pi
RadioBerry[66], TAPR WSPR[67] Boards[68], and most recently, CaribouLite[69] all leverage the abundant compute power and hardware flexibility and the ubiquity[70] of the Raspberry Pi computers[71] to trade hardware development of a standalone radio unit for a simplified design that puts more of the radio function into software and computing power. The TAPR WSPR Boards are notable that they are very simple, acting mostly as a filter to clean up harmonics and other undesirable signals resulting from rapidly toggling an Input / Output pin at high speed to produce a radio signal at HF frequencies. The CaribouLite is notable because it was designed to use the more minimal $15 - $20 Raspberry Pi Zero / Zero 2 instead of the (full size) $35 - $80 Raspberry Pi units.

HAMNET Access Protocol (HNAP)
HNAP[72] is an "precompiled image" for the ADALM-PLUTO[73] Software Defined Transceiver (SDT) for data communications on the Amateur Radio 420-450 MHz band. The "Pluto" is intended for use by students and for evaluation of the vendor's chipsets and is typically used with GNU Radio. In contrast to the complexity of GNU Radio, HNAP is "plug and play" for Amateur Radio data communications. *Amateur Radio needs a lot more of these practical, easy-to-use examples of Software Defined Radio technology.*

Decentralized Amateur Paging Network (DAPNet)
Paging[74] is a radio technology that dates back to the 1950s – broadcasting a signal unique to individual pagers, in continuous sequence. Paging technology evolved from a simple "beepers" to units that could receive text messages, to a few "two-way" paging systems where the pager unit could send back acknowledgement-of-receipt and reply messages. Paging technology has largely been obsoleted by ubiquitous mobile telephony and integrated text messaging.

Paging protocols are a robust method to transmit text messages. DAPNet[75] has adapted text messaging paging technology for Amateur Radio use with low-cost hardware, Open Source software, and multiple Amateur Radio stations via Internet into an *Amateur Radio* paging network.

Repeater Builder Website
There are many unique websites that support Amateur Radio activities, but there are few with the breadth and depth as Repeater Builder (RB)[76]. RB is so extensive I refer to it as an Omnipedia. Not only are there tutorials about how to create an Amateur Radio (or other) repeater, but there is also extensive reference material, including documentation about radios

---

65 https://www.rtl-sdr.com/about-rtl-sdr/
66 https://github.com/pa3gsb/Radioberry-2.x/wiki
67 https://physics.princeton.edu/pulsar/k1jt/wspr.html
68 https://tapr.org/product/wspr/
69 https://www.crowdsupply.com/cariboulabs/cariboulite-rpi-hat
70 Until the "chip shortage" beginning in 2020
71 https://www.raspberrypi.com/products/
72 https://hnap.de/about/
73 https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html
74 https://en.wikipedia.org/wiki/Pager
75 https://hampager.de/#/
76 https://www.repeater-builder.com/rbtip/index.html

that are no longer supported by their manufacturer. I'm unaware of the depth of information for topics like this outside of Amateur Radio.

Developing New *Systems*

Some technological innovation in Amateur Radio requires "Thinking Big" by imagining the *whole system*. The systems in this section illustrate technological innovation in Amateur Radio by imagining the big picture.

KA6M-1 Digipeater
Talk about developing new systems and technological innovation in Amateur Radio!!! In late *1980*, years before the phrase Packet Radio would be familiar to Amateur Radio Operators, the KA6M-1 Digipeater[77] began proving out Packet Radio technology in US Amateur Radio, and especially the capability to use of "repeaters" that used only a single frequency by receiving, buffering, and then transmitting the received data – digipeaters.

The KA6M-1 Digipeater used the Packet Radio protocols developed by the Vancouver Digital Communications Group (VADCG)[78] and presaged the development of the TAPR TNC-1[79] and very popular TNC-2 (with built in digipeating capability), and APRS.

Automatic Packet Reporting System (APRS)
Over four decades, APRS[80] has become so ubiquitous within Amateur Radio that it's easy to forget how technologically innovative APRS was when it debuted in early 1980s. Bob Bruninga WB4APR combined a GPS receiver (then, new technology), a Packet Radio TNC, and an Amateur Radio transmitter to transmit real-time position reports via radio. The receiving station was equally simple – an Amateur Radio receiver, a Packet Radio TNC, and a personal computer with map software to display the position data.

APRS is now an entire ecosystem, embedded into radios, a worldwide network of digipeaters, Internet gateways, inexpensive trackers… *ubiquitous*! APRS technology is so ubiquitous that it's being used to track balloon launches that are classroom experiments using unlicensed radio transmitters. It's probably one of the proudest achievements of WB4APR (now a silent keyboard) that APRS is a permanent presence on the radio stations on the International Space Station. APRS continues to evolve, and some in a position of influence within the APRS developer community have taken the first steps to form an "APRS Foundation"[81]. It's worth remembering that the technological innovation of APRS, and Automatic Identification System (AIS)[82] used on vessels, *began within Amateur Radio*.

Winlink
Like APRS, the Winlink[83] system has become so embedded into Amateur Radio over decades that we forget how technologically innovative it was at the time to be able to reliably send Internet email via Amateur Radio, especially via HF from nearly anywhere on Earth. Like APRS, Winlink now works so well, and has done so for so long, it feels like "a utility" within Amateur Radio. Winlink seamlessly provides a network of Internet servers, and Winlink Radio Mail

---

77 http://www.pprs.org
78 https://tapr.org/pdf/CNC1986-FeaturesOfVadcgTNCplus-VE7APU.pdf
79 https://web.archive.org/web/20090704021623/http://www.tapr.org/history.html
80 https://en.wikipedia.org/wiki/Automatic_Packet_Reporting_System
81 http://lists.tapr.org/pipermail/aprssig_lists.tapr.org/2022-March/049356.html
82 https://en.wikipedia.org/wiki/Automatic_identification_system
83 https://winlink.org

Servers (RMS) are provided by individual Amateur Radio Operators. Winlink has developed client and RMS software that makes it easy to set up a VHF / UHF RMS, and the Winlink client software is stable and sufficiently well-documented that newcomers are easily able to quickly get up to speed on using Winlink at their own Amateur Radio station. Winlink is another example of technological innovation that *began within Amateur Radio*.

Terrestrial Amateur Radio Packet Network (TARPN)

TARPN[84] has re-thought, and re-engineered Amateur Radio Packet Radio networking, identifying weak points in "traditional" packet radio networking and architected TARPN networks to eliminate those weaknesses. For example, for best performance, TAPRN networks do not use Packet Radio Digipeaters[85].

TAPRN uses Amateur Radio networking software written to run on Raspberry Pi Linux and provides copious documentation and a Raspberry Pi image to make it easier to get a TARPN network up and running.

Out of this work, TARPN has created its own hardware: the NinoTNC[86] - a KISS[87] TNC[88] connected via USB and providing 1200 / 2400 / 4800 / 9600 bps data speeds. This unit has gone through a number of evolutions as TARPN has gathered more feedback about its performance in the real world, and feedback on building the unit (it's supplied as a printed circuit board, a programmed processor, and a list of parts that the user procures).

Perhaps more notable than the NinoTNC was the parallel development of a *new Forward Error Correction (FEC)[89] mode* called Improved Layer 2 Protocol (IL2P)[90]. IL2P is an efficient protocol because it does not attempt backwards compatibility with AX.25. Interoperability between AX.25 and IL2P isn't an issue with TARPN networks as all connections are point-to-point, thus AX.25 vs IL2P need only be negotiated between each two endpoints. IL2P was sufficiently well documented that IL2P support has been designed into Dire Wolf Software TNC (see below).

File Distribution via Broadcast - flamp and RadioMirror

Phil Karn KA9Q once stated (paraphrased) "Why do we in Amateur Radio try implement one-to-one communications via radio instead of taking advantage, as much as possible, of *the broadcast nature of radio?*". flamp[91] and RadioMirror[92] are two implementations of that observation that radio communications are inherently a broadcast medium, from one transmitter to any number of receivers within range of the transmitter. With these systems, at the transmitter, each file to be distributed is broken into blocks and given a checksum and sequence number, and all blocks / files are transmitted in turn. When all blocks / files have been transmitted, the process repeats. Any new or changed files are added to the queue.

At the receiver, each block is received, and the checksum verified. Valid blocks are queued for assembly per the sequence number. Once all blocks are received correctly, the file is written. If

---

84 http://tarpn.net/t/packet_radio_networking.html
85 http://www.aprs.net.au/vhf/aprs-digipeaters-101/
86 http://tarpn.net/t/nino-tnc/nino-tnc.html
87 https://en.wikipedia.org/wiki/KISS_(TNC)
88 https://en.wikipedia.org/wiki/Terminal_node_controller
89 https://en.wikipedia.org/wiki/Error_correction_code#Forward_error_correction
90 http://tarpn.net/t/il2p/il2p.html
91 http://www.w1hkj.com/files/manuals/US_English/FLAmp_2.2_Users_Manual.pdf
92 https://www.superpacket.org/2021/03/revisiting-radiomirror.html

a block is missing (discarded because the checksum failed), assembly of that file waits until the next cycle.

The benefit of file distribution via broadcast is that there is no need for a two-way handshake such as a packet radio file transfer. The process can be a background task for populating files that require periodic updating, such as map files, lists of repeaters, and even Amateur Radio bulletin texts. In 2022, the receiver can be simplified to an inexpensive Software Defined Receiver dongle and a Raspberry Pi.

Innovative Use of Inexpensive Computing Power

These projects illustrate technological innovation in Amateur Radio by imagining “what *could* we do to create better radio technology by making use of inexpensive computing power?”. It should be noted that the ubiquity and versatility of the Linux operating system is usually a “silent partner” with inexpensive computing power to enable many technological innovations in Amateur Radio.

WSJT-X
WSJT-X[93] illustrates (humorously) the “dangers of a Nobel Prize laureate with too much time on his hands”. After retiring from a distinguished career as an Astrophysicist, Joe Taylor K1JT[94] applied his extensive knowledge of radio signal processing technology to Amateur Radio weak signal modes. The various WSJT-X modes utilize computing power to apply both forward error correction and “deep down in the noise signal recovery” to provide new capabilities to Amateur Radio using very, very low power, including Earth Moon Earth (EME)[95] communications with modest Amateur Radio stations, meteor burst[96] (meteor scatter) communications, and more. Other WSJT-X modes provide communications when other modes cannot operate due to low signal level or excessive channel noise.

Dire Wolf Software TNC
Dire Wolf[97] (ostensibly) is an acronym for “Decoded Information from Radio Emissions for Windows Or Linux Fans”. Dire Wolf is a software implementation of an Amateur Packet Radio Terminal Node Controller (TNC) and is an actively maintained Open Source software project.

In developing Dire Wolf, John Langner WB2OSZ applied a key insight, that AX.25 packet radio required retransmission of an entire packet if even 1 bit was incorrect and thus the packet’s Cyclic Redundancy Check (CRC) failed. WB2OSZ wondered “what if we flip each individual bit, and see if the CRC is correct? There is ample computing power available for such a test, and that simple innovation resulted in decoding many more packets successfully. Of course, there were cases where the bit flipping resulted in an incorrect packet (despite the CRC), so he applied other techniques to ensure that the ”bit flipping” resulted in a correct packet.

Dire Wolf has evolved to be a “Packet Radio toolkit” – it can act as an APRS digipeater, an APRS Igate[98], a high speed TNC (9600 bps, and many other data speeds other than 1200 bps), and many other capabilities. Dire Wolf works very well on a Raspberry Pi computer.

---

93 https://physics.princeton.edu/pulsar/k1jt/wsjtx.html
94 https://en.wikipedia.org/wiki/Joseph_Hooton_Taylor_Jr.
95 https://en.wikipedia.org/wiki/Earth–Moon–Earth_communication
96 https://en.wikipedia.org/wiki/Meteor_burst_communications
97 https://github.com/wb2osz/direwolf
98 http://www.aprs-is.net/igating.aspx

It's notable that Dire Wolf has implemented two Forward Error Correction systems for Amateur Radio Packet Radio – FX.25[99], which is compatible and interoperable with conventional AX.25[100] (in the stable distribution of Dire Wolf) and Improved Layer 2 Protocol (IL2P) developed for the NinoTNC (in the development branch of Dire Wolf).

VARA FM
VARA[101] is a robust and fast audio interface ("sound card") data mode for reliable file transfers. VARA is most typically used by Winlink user stations and Winlink Remote Mail Servers (RMS) for faster and more reliable file transfers than previous methods such as WINMOR and packet radio.

VARA is not compatible with any other data communications mode (such as packet radio); VARA stations can only communicate with other VARA stations. VARA FM is designed for use on the wide, quiet channels of Amateur Radio VHF / UHF, achieving speeds up to 25 kbps and incorporating Forward Error Correction (FEC) for very fast and reliable file / message transfers compared to conventional packet radio, even at 9600 bps.

VARA achieves its robustness and speed by incorporating a number of techniques:

- Orthogonal Frequency Division Multiplexing (OFDM) generates multiple subcarriers within the audio signal.
- Varying modulation methods (Modulation Index) – FSK through 256 QAM depending on mode and quality of channel between stations.
- Robust handshake between transmitting and receiving station to negotiate best possible speed on each transmission.
- Huffman data compression.
- Turbo Codes Forward Error Correction.

While VARA FM achieves its best performance using radios that provide "flat audio" connections (bypass pre-emphasis and de-emphasis audio circuits), it's quite usable when connected to speaker and microphone connections, even on portable radios. Most notably, VARA FM will "handshake" when establishing a connection and negotiate the best possible common speed between two VARA stations, overcoming the issue with packet radio 1200 bps and 9600 bps stations not being able to communicate with each other.

ka9q-radio
ka9q-radio[102] virtualizes a single Software Defined Receiver into multiple receiver modules. "A single Raspberry Pi 4 can simultaneously demodulate, in real time, every narrowband FM channel on a VHF / UHF band (i.e., several hundred) with plenty of real time left over." The use of IP multicasting[103] "makes it easy for more than one module, on the same computer or on a LAN, to operate on the outputs of other modules…".

Codec 2 – Open Source / non-proprietary Digital Voice
The work on Codec 2[104] began more than a decade ago. Early in the experiments with the use

---

[99] https://en.wikipedia.org/wiki/FX.25_Forward_Error_Correction
[100] https://en.wikipedia.org/wiki/AX.25
[101] https://rosmodem.wordpress.com
[102] https://github.com/ka9q/ka9q-radio
[103] https://en.wikipedia.org/wiki/IP_multicast
[104] https://en.wikipedia.org/wiki/Codec_2

of digital voice in Amateur Radio, a study concluded that there was no practical method of implementing digital voice in Amateur Radio that wouldn't infringe on the numerous digital voice patents at the beginning of conversion of cellular phones from analog voice to digital voice.

In the development of Codec 2, an interesting approach was used: instead of trying to work around patented digital voice methods, patented methods of digital voice *where the patent had expired* were sought out. Codec 2 is the result, a fully Open Source software approach to digital voice that is finally beginning to see widespread use now that it can be applied as "just a bit more software" to Software Defined Radio transceivers. Codec 2 is the digital voice implementation used in M17 Project.

Codec 2 / FreeDV Data for HF
Having been proven robust and spectrally efficient for use on HF, the technology and modems developed for Codec 2 / FreeDV[105] are being adapted as "transport" for text messaging and data over HF in two separate projects – Codec 2 HF Data Modes (Part 1)[106] (Part 2)[107], and FreeDATA[108].

Multi-Mode Digital Voice Modem (MMDVM)
Sometimes talented Amateur Radio Operators see a "problem" as a challenge and take a unique approach to solving the challenge. The genesis of MMDVM[109] developed by Jonathan Naylor G4KLX, was the proliferation of digital voice implementations in Amateur Radio that weren't interoperable:

- D-Star
- Digital Mobile Radio (DMR)
- Next Generation Digital Narrowband (NXDN)[110]
- Project 25 (P25)[111]
- System Fusion[112]

Thus, the MMDVM which handles "all of the above" on an equal basis. MMDVM can also be used to "transcode" one digital voice mode to another, such as linking a D-Star repeater to a DMR repeater. MMDVM can be used to build repeaters which work equally well on all digital voice modes, as well as the basis of "personal hotspots" which can act as "Pico repeaters" for very localized use. Other modes have been added to MMDVM including FM, Packet Radio (AX.25), POCSAG[113] (paging), and M17.

---

105 https://freedv.org
106 http://www.rowetel.com/?p=7167
107 http://www.rowetel.com/?p=7665
108 https://groups.io/g/freedata
109 https://github.com/g4klx/MMDVM
110 https://en.wikipedia.org/wiki/NXDN
111 https://en.wikipedia.org/wiki/Project_25
112 http://systemfusion.yaesu.com/what-is-system-fusion/
113 https://en.wikipedia.org/wiki/Radio-paging_code_No._1

One Last Vignette – FlexRadio Systems Origins in Amateur Radio

The crossovers from Amateur Radio to government and industry are innumerable. That point is made very well in a film[114] made during World War II about the Hallicrafters SCR-299. This unit was designed and manufactured for Amateur Radio Operators, but during World War II it was adapted for mobile use on the battlefield.

This "crossover" continues to the present day, as exemplified by the emergence of FlexRadio Systems.

FlexRadio Systems[115] began as a personal project by Gerald Youngblood K5SDR in 2013. K5SDR's goals were to familiarize himself with (then) new (to Amateur Radio) technology of Digital Signal Processing. He published[116] [117] [118] [119] his project, the SDR-1000, in QEX magazine[120] and it was well-received to the point that K5SDR was asked to provide kits of parts to allow other Amateur Radio Operators to build their own SDR-1000s. On the basis of that kit, FlexRadio Systems was founded in K5SDR's home[121]. FlexRadio quickly outgrew those modest beginnings and FlexRadio's innovative and extremely cost-effective Software Defined Radio technology quickly gained notice in government and industry that needed highly flexible and cost-effective radio systems, which FlexRadio grew rapidly to accommodate.

In March 2022, FlexRadio was awarded a significant government contract[122] to supply radio systems to US Air Force aircraft. Despite the inevitable motivation to change its focus to more lucrative government and industry products, FlexRadio has chosen[123] to remain grounded in Amateur Radio and continue to develop new products for Amateur Radio:

Throughout the [US Government] project, FlexRadio has been asked about our ongoing business and we have continued to inform all of our customers that the Amateur Radio business is strategic for both FlexRadio as well as the long-term benefits to the radio art and communications community. Specifically, FlexRadio has repeatedly asserted that we believe that continuing to invest in Amateur Radio is an investment in the future of communications. There is not a corner of the communications world that FlexRadio has been involved in that we do not see Amateurs making key contributions.

FlexRadio Systems, the new USAF radio, and the stellar example to other Amateur Radio manufacturers in the overwhelming advantages of a *fully Software Defined* radio architecture, *would not have come into existence without Amateur Radio*.

---

[114] https://www.youtube.com/watch?v=A6z18otFPVY
[115] https://www.flexradio.com
[116] http://www.arrl.org/files/file/Technology/tis/info/pdf/020708qex013.pdf
[117] http://www.arrl.org/files/file/Technology/tis/info/pdf/020910qex010.pdf
[118] http://www.arrl.org/files/file/Technology/tis/info/pdf/021112qex027.pdf
[119] http://www.arrl.org/files/file/Technology/tis/info/pdf/030304qex020.pdf
[120] http://www.arrl.org/qex
[121] https://www.flexradio.com/videos/flexradio-history-with-gerald-youngblood/
[122] https://www.flexradio.com/insider/press-releases/flexradio-awarded-contract-in-collaboration-with-bae-systems/
[123] https://zeroretries.substack.com/i/43856023/flexradios-commitment-to-amateur-radio

Conclusion

In “About Zero Retries”[124] (a link that appears near the beginning of every issue), I list a number of poignant quotes. These three seem particularly relevant to mention in this paper:

- Ultimately, amateur radio must prove that it is useful for society - Dr. Karl Meinzer DJ4ZC.
- The Universal Purpose of Ham Radio is to have fun messing around with radios - Bob Witte K0NR.
- Amateur Radio is literally a license to experiment with radio technology! - Steve Stroh N8GNJ

I agree with DJ4ZC – it is imperative that Amateur Radio “prove its worth”.

I also agree with K0NR – much of Amateur Radio is, in the end, having fun with radios.

The last quote, mine, is a fact that is almost completely overlooked – Amateur Radio is… (quite literally) a *license*… to *experiment*… with *radio technology*!

I hope that this paper has helped to illustrate that there *is* an *amazing* amount of technological innovation occurring *now* within Amateur Radio. The scale and scope of that technological innovation isn’t widely recognized in part because of the highly decentralized… and individualistic nature of Amateur Radio and Amateur Radio Operators.

Another part of that lack of recognition is that technological innovation in Amateur Radio isn’t regularly featured in Amateur Radio “media” such as popular magazines, YouTube shows, podcasts, and blogs. It’s no wonder that the popular perception of Amateur Radio is often “old… tired… no longer relevant”. Amateur Radio *really* needs to *change* that perception… *somehow*.

It is critical that the technological innovation occurring now, and in projects, and products, and systems that will extend into future years, and even decades be recognized by the public, but more importantly by industry, regulators, and lawmakers. In my opinion, such wider recognition of technological innovation in Amateur Radio will be a primary justification for Amateur Radio (and its operations in various portions of spectrum) being allowed to continue.

The vignettes of technological innovation mentioned in this paper were selected from a much larger collection of such information on a web page:

Zero Retries 0070 Omnibus of Zero Retries Interesting Information

https://www.superpacket.org/zero-retries-0070-omnibus.html

If you would like to read more about technological innovation in Amateur Radio, subscribe to the Zero Retries Newsletter. It is free (as in beer) and delivered weekly via email. Join the fun at https://zeroretries.substack.com (eventually migrating to https://zeroretries.org).

Steve Stroh N8GNJ
September 2022

[124] https://zeroretries.substack.com/about

Bushwhacking In The Land Of Digital Voice

By David A. Vine, WA1EAW
Aiken, South Carolina, USA

Abstract

Amateur Radio Digital Voice[1] (DV) nets flourish throughout the U.S. and elsewhere however, most are “virtually” inaccessible to all but a few persistent explorers. The main reason for this is the lack of a widely know, central, continuously updated, comprehensive catalog of information to guide interested Radio Amateurs to any net accessible via a myriad of hubs, interconnected repeaters and systems as well as individual networks (Brandmeister, TGIF, etc.) using a variety of DV formats including DMR, D-STAR, YSF, etc.

An efficient way to find nets does not exist. What is lacking is a single convenient way to select a net from a comprehensive, frequently updated database of regularly scheduled nets, especially those with topical discussions. It is difficult and time consuming to create a comprehensive list for personal use.

The Significance of Nets

Nets are Amateur Radio’s version of “gather ‘round the camp fire” sessions. Active Radio Amateurs usually check into a local club or group VHF/UHF net on a daily or weekly basis. Emergency Communications units do this too, primarily as a test and demonstration of capabilities. These local or regional activities are a means of regularly scheduled radio communication for individuals, some of whom may be housebound or have mobility issues.

Perhaps most importantly, these on-air gatherings foster cohesiveness among group members and provide friendly greetings and helpful advice especially important to encourage newly licensed Radio Amateurs. This was especially true during the height of the Covid Pandemic Quarantine from March 2020 through 2021.[2]

The Amateur Radio Club of Augusta (**ARCA**) Georgia Nightly 2-Meter Net kicks off at 8:00 p.m. local time. The ARCA net generally attracts 20 to 40 Radio Amateurs. A session almost always includes three to five check-ins via EchoLink. This feature, recently added to a wide coverage VHF repeater, has enabled traveling club members to maintain their regular check-in schedule. An over the road tractor-trailer driver regularly checks in from various locations throughout the eastern half of the U.S.

Some nights a considerable number of announcements, questions, comments and discussions extend the net up to 80 minutes. Longer net sessions with more conversations go a long way toward building camaraderie among Radio Amateurs in the region.

---

1 **Icom D-STAR General Information**
2 **CDC Covid-19 Timeline**

The following following excerpts from a net control script specify the purpose of the ARCA Nightly Net… We pass traffic and information, list amateur radio equipment for sale or trade and as a public service to demonstrate emergency preparedness and promote Amateur Radio fellowship… All licensed Radio Amateurs are welcome, and encouraged, to check-in and participate in tonight's net… this frequency may be activated for emergency or public service activity. This repeater is also used for the Central Savannah River Area National Weather Service **SKYWARN** program…"

Listeners may include non-licensed persons within range of the repeater who use inexpensive hand-held transceivers or scanner radios. Some Amateur Radio repeaters are available via **Broadcastify** and **Rangecast** systems. Non-licensed people can listen using apps on smartphones, tablets, computers, etc. These listeners are a potential source of recruits for Amateur Radio. Jeffrey Kopcak, K8JTK has a **web page** with more on this subject.

The importance of Nets nets cannot be overstated. Nets contribute to the overall awareness of local, regional and national Amateur Radio activities.

An Amateur Radio license is a passport to a world of learning and adventure. Along the way of this journey we meet like-minded Radio Amateurs as well as those who are subject matter experts. We we may encounter helpful "Elmers[3]" who encourage and assist us in our exploration of the many, many Amateur Radio related activities.

Participation in nets is a way to meet people beyond a local circle of acquaintances, colleagues and friends. Here is a sampling of the **D-STARinfo.com** net listing. These selected nets most likely have more general interest and/or topical orientation rather than nets that are localized or focused on interests of a specific club.

- Amateur Radio Astronomy
- Australian D-STAR
- Backyard Repeater Owners
- Canadian D-STAR
- CERT D-STAR
- Coastal Carolinas D-STAR
- D-STAR HF
- D-STAR Trains and Railroads
- Military Veterans
- Pacific Division D-STAR ARES
- PAPA System D-STAR
- Philadelphia Digital Assoc. D-STAR
- Powersports
- Quarter Century Wireless Assoc.
- RACES/MARC Digital Voice
- Raspberry Pi
- D-STAR Users Group
- Friday Night Round Table
- Ham Nation After Show D-STAR
- Independent Radio Club
- International D-STAR
- Kids in Amateur Radio
- Mid-Atlantic Aux Comm Svc.
- PAPA System D-STAR Net
- Region III Aux Comm Svc.
- Round-the-World QSO
- Rural Radio Preparedness Assoc.
- RVers Digital
- Thursday Night D-STAR Round Table
- Thursday Night Tech Round Table
- W6DHS Global Village
- Young Operators DV

[3] **http://www.arrl.org/elmer-award**

The PAPA System short listing in the D-STAR Net list does not reveal the breadth of offerings. According to the ARRL Net Directory listing for the system, "The PAPA System is a member-supported wide-area amateur radio network of 57 inter-linked analog FM, D-STAR, DMR, and P25 repeaters on 19 hilltops providing extensive coverage of the Southern California region and beyond…." A good description but without a list of the PAPA System's weekly nets:

- Antenna Net
- ARRL SW Division Net
- D-STAR Tech Net
- DMR Roundtable
- New Ham's Net
- Outdoor Net
- P25 Net
- PAPA Tech Roundtable
- Saturday Night Roundtable
- SoCal Boater's Net
- Topanga Disaster Radio Net

Finding Nets

Many different individually created PDF documents listing DV nets are compiled, occasionally updated and freely available. Knowing where those lists reside on Internet sites and how comprehensive or current the listings are presents a challenge to would-be users. Many of these lists usually focus on one mode such as D-STAR or EchoLink.

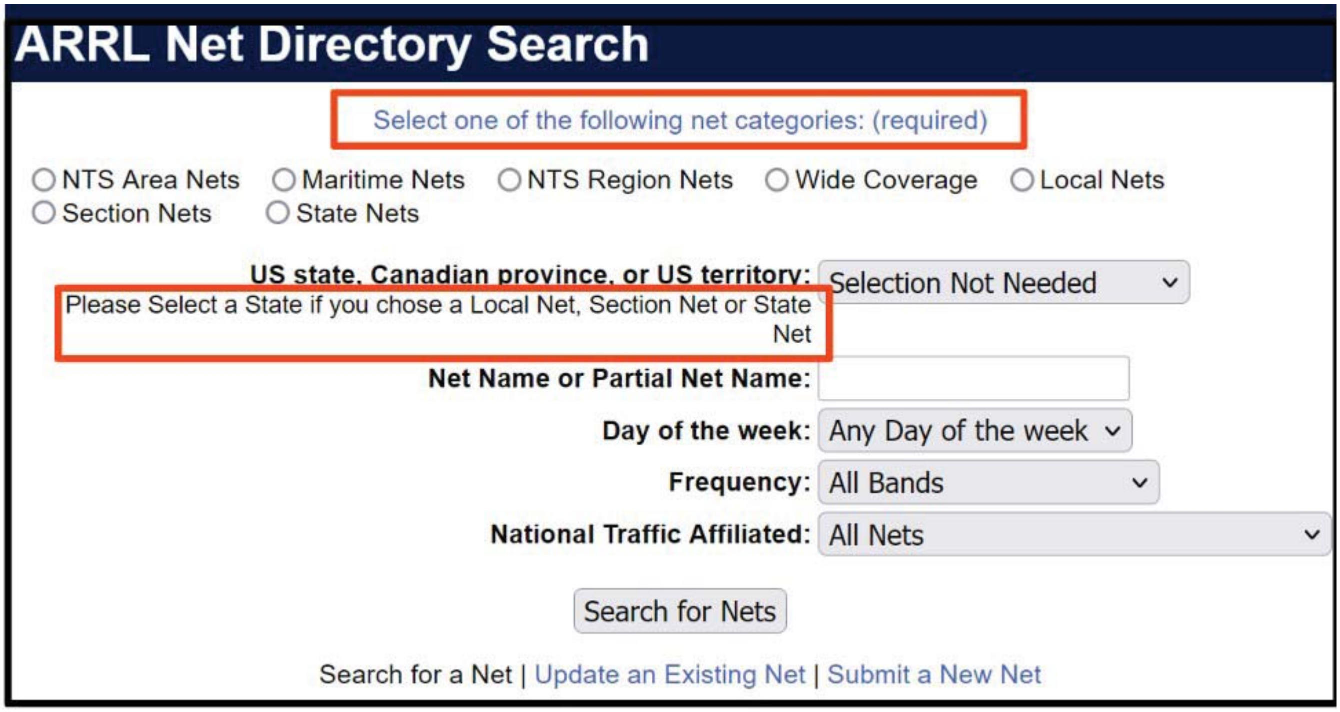
Representative Digital Voice Net Finding Aids

- **AllStar Nets**
- **ARRL Net (Primarily traffic nets)**
- **AugustaHam.net Calendar**
- **BM DMR Nets**
- **BM Hoseline TG & Listen In**
- **Control Center Bronx-TRBO**
- **D-STAR Info Net List**
- **D-STAR Nets (Facebook login)**
- **D-STAR Users Last Heard**
- **DWARN Nets Calendar**
- **EchoLink Conference Status - Live**
- **EchoLink Link Status Geo Search**
- **FreeDMR Most Used Talkgroups**
- **FreeSTAR Net Control**
- **Google Site Search Example**
- **Ham Radio DMR Nets**
- **HamNetList**
- **Hamshack Hotline Bridges**
- **KAPIHAN Network nets**
- **Net Scrapper**
- **NetControl Manager**
- **NetLogger**
- **NETS - DMR, C4FM, D-STAR, P25 (FB)**
- **Network Digital Radio Monitor (YSF)**
- **QuadNet Nets**
- **RepeaterBook Search for "nets"**
- **SkyHub Link System Nets**
- **Southern Tier NY DMR**
- **Telegram - Ham Radio DMR Nets**
- **TGIF Network Active Talkgroups**
- **WIN System Who's Talking**
- **WX4QZ.Net**

Invariably, when the subject of finding "other" nets comes up a Radio Amateur might suggest one or two websites with lists including such sites as NetLogger. This hybrid software and online service has a "get previous nets" function listing more than 3,500 nets (including many duplicates) of all types along with pertinent details about each net.

The ARRL Net Directory search function has a few shortcomings. According to their description of the Directory, "One focus of the directory is toward public-service oriented nets that support the ARRL National Traffic System (NTS) and the Amateur Radio Emergency

Service (ARES).“ The search interface restricts searches to specific types of nets, one state at a time. Also, it relies on updates by individuals so it contains duplicate and/or outdated listings.



There are hundreds of RF and DV net lists available on web pages, social media sites, and in downloadable files. A few dedicated Radio Amateurs periodically create up-to-date lists covering two or three DV modes but none are truly comprehensive. Keeping track of hundreds of nets is impossible for one person to do in their spare time. Having said all that the most comprehensive source for net schedules seems to be **HamNetList.**

Net lists come in a variety of formats including locked (text cannot be copied) and unlocked PDF documents, one-off and regular web or social media posts, spreadsheets and calendars with or without ICS[4] files. There are variations of record layouts or columns in a document or spreadsheet. Different descriptors, field names, times and time zones, day(s) of the week abbreviations or network specifics for each listed net. The information is exceedingly difficult to collate.

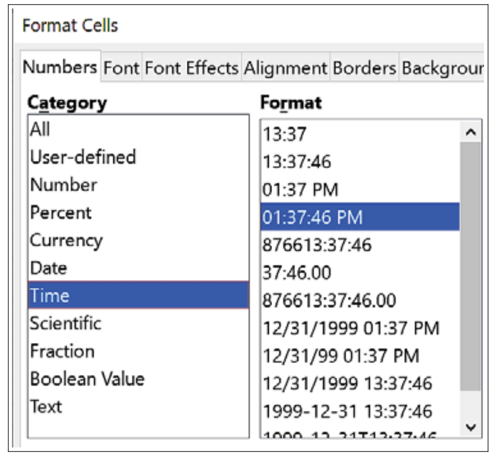
For example, when using a spreadsheet to import or copy and paste net listings one might encounter various difficulties. HTML text that is columnar on the web sometimes doesn’t lend itself to the usual copy and paste convenience. Saving spreadsheet data from the source list in a CSV[5] file strips out embedded Internet links.

---

4 An ICS file is a calendar file saved in a universal calendar format used by several email and calendar programs, including Microsoft Outlook, Google Calendar, and Apple Calendar. It allows users to share calendar information on the web and over email.

5 A CSV (comma-separated values) file is a text file that has a specific format which allows data to be saved in a table structured format.

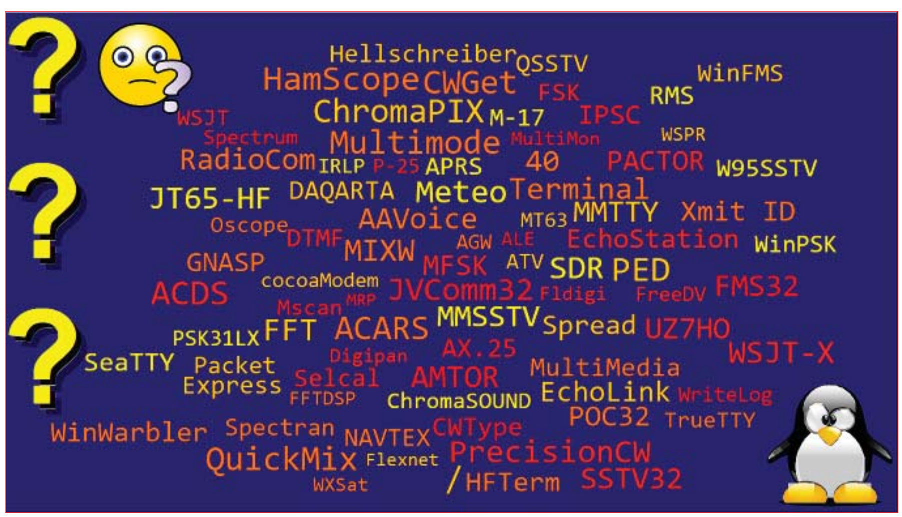
Another significant matter is time and date formats in cells of spreadsheets. For example, OpenOffice and LibreOffice spreadsheet time formatting seems to be a particular problem. Attempting to automatically derive Eastern time from cells containing UTC seems impossible, despite considerable research about the time function and extensive experimentation.



Sorting by day of the week and then starting times of nets presents another problem. Monday through Sunday days of the week can't be sorted alphabetically in a document or spreadsheet and remain in correct order. Using numbers before days to properly sort the list is a way to do it. Considerable search and replace actions are need to standardize the abbreviation of each day and include the number. This LibreOffice Calc spreadsheet shows one approach combining a list of RF and DV nets:

Days	Start	Net Name	Where - Comment/Info
1-Mon	20:00:00	D-Star Net	REF020A
1-Mon	21:00:00	North Carolina D-Star Net	REF054A
1-Mon (1st/Mo.)	14:30:00	FEMA Region 8 RECCWG AuxCom Net	Echolink 3575
1-Mon-4-Thr	19:30:00	The 420 Ragchew	Node 66420
1-Mon-5-Fri	08:00:00	SCARS HF runs to 1:00 PM	7.251 MHz
1-Mon-5-Fri	09:00:00	Breakfast group	KB4SVP-R
1-Mon-5-Fri	14:00:00	Nuts Bolts And Screws HF Net (Pre-net @ 2 p.	7.185
1-Mon-5-Fri	14:30:00	Vagabond Ragchew Net	WB9SZL-R, 420473
1-Mon-5-Fri	20:00:00	Happy Hour Net	Oregon 3141
1-Mon-5-Fri	23:00:00	PAPA DMR Roundtable	PAPA Chat 31077 & DODROPIN
1-Mon-6-Sat	09:00:00	Northern Florida ARES Net	3.950 MHz, 7.242 MHz and 7.247 MHz

Uncomfortable Questions



Is Amateur Radio digital communications too complicated? Diversity of three major DV protocols, by individual manufacturer (DMR generics, Icom D-STAR and Yaesu System Fusion) presents programming challenges.

Would comprehensive database of DV nets be a good thing? Is there a benefit or a liability for the club or Amateur Radio in general to facilitate visits by "out-of-towners" to various local nets? Would a highly publicized central directory attract more DV users, especially the much sought after younger people? Would an updated, comprehensive database making information about nets widely available cause undesirable competition to see which net is most “popular?”

Should the The **Radio Amateur's Code** be expanded to include “netiquette[6]” or does the existing Code cover behavior during net interactions?

Would a sufficient number of net hosts, managers, repeater owners or club officials report updates, additions, corrections, etc. if provided with a simple input form accessible by Internet?

The M17 RF Protocol is “optimized for amateur radio use and simple to understand and implement.” Developers state that M17 must be “capable of doing the things hams expect their digital protocols to do: Voice (eg: DMR, D-STAR, etc); Point to point data (eg: Packet, D-STAR, etc); Broadcast telemetry (eg: APRS, etc) and Extensible, so more capabilities can be added over time.” Is M17 overcome some of the complexities of connecting to Amateur Radio DV nets?

Technical Barriers To Net Participation

For some Radio Amateurs, participating in HF nets can be difficult due to a variety of factors. The growth of real estate developments with private community restrictions on antennas, relatively limited range of VHF/UHF repeaters, expense of radio equipment for HF operations, irregular radio wave propagation[7] and FCC license class restrictions are some of the factors that may impede newly licensed entry-level Radio Amateurs from unrestricted participation in Amateur Radio nets via radio.

Radio programming for DV net participation is complex. There are hundreds of local, regional, national and international interconnect systems. Interconnection examples include such networks as **K8JTK Hub** and **NEDECN**.

Even experienced computer users have difficulty programming their radios or setting up hotspots. Use of specialized software like **Pi-Star** running on a Linux OS device is foreign to many of us. During setup there are many fields to fill and selections to be made when installing Pi-Star on a Raspberry Pi hotspot. If just one of those selections is incorrect the system probably won’t work. Without research or help the potential user might be stymied.

Could specifically designated digital subject matter experts like the DMR/Hotspot Office Hours Net **Panelists** be organized to help in these situations?

Nearly every Radio Amateur owns or has easy access to a smartphone, mobile device, tablet, laptop or desktop computer with Internet connectivity capable of running EchoLink software. The cost of net participation via EchoLink is simply the time it takes to install, learn and use the free program. EchoLink gives technician class license holders access to repeaters worldwide. EchoLink even offers the tech an instant international means of calling CQ[8].

---

6 See the Internet Society RFC 1855 **Netiquette Guidelines**

7 **https://www.swpc.noaa.gov/products/d-region-absorption-predictions-d-rap**

8 **https://www.echolink.org/cq.htm**

Technical barriers can arise even with as simple a program as EchoLink. Understanding Internet proxies and port forwarding[9] could be needed when using EchoLink. The requirement to occasionally alter the direct connection or proxy settings in EchoLink requires awareness of that function and the knowledge of how to deal with connection problems, especially when using smartphone access to cellular Internet connections.

Those who can afford a relatively inexpensive USB **AMBE vocoder** device find a greatly expanded world of Digital Voice nets. Beyond EchoLink, the USB dongle or outboard device and applicable software like BlueDV enable relatively simple access to DMR, D-STAR and YSF Digital Voice modes.

Availability and use of ***a very wide variety of different local, regional and nationally networked repeaters[10] using various linking systems adds to the complexity of use and presents a barrier to the end-user who wants to easily traverse the digital landscape.*** The complexity of different Digital Voice modes and the means to use those modes further complicates the situation even for the tech savvy Radio Amateur. Digital Voice is definitely not a point and click affair like the World Wide Web.

The Hybrid RF/Digital LMARC Example

The Lookout Mountain Amateur Radio Community (**LMARC**) N4LMC and KO4GVX repeaters' RF footprint covers Chattanooga, Tennessee, Northwest Georgia, Northeast Alabama and surrounding areas. However, like most other repeaters with Internet connections, their digital footprint is world wide.

The LMARC system and neighboring repeaters provide a variety of RF features, including: APRS Tx/Rx iGate + Digipeater; C4FM Reflectors, D-STAR Modules; DPlus; DMR (Brandmeister and TGIF networks); EchoLink and HamShack Hotline connections; IRCDDB and IRLP; NXDN; P25; RMS Mail Gateway; access to the SouthEast Link System; and Wires-X Fusion along with emergency power for several of the repeaters.

In addition to the many technical features and benefits available to all classes of licensed Radio Amateurs via LMARC area repeaters, some or all of the repeaters and inter-connected networks host more than **20 daily and weekly** nets.

DMR, D-STAR, YSF, etc. are specialized versions of DV. Inexperienced Radio Amateurs seem to have the most trouble programming DMR radios and talkgroups. Some rely on easy access to downloadable "code plugs[11]." Downloadable codeplugs help neophyte user to avoid extensive programming of transceivers. The result is a lack of understanding programming the radio and perhaps a deficit in understanding how to use DV.

In a paper titled simply "**Mode Overview**" written by Daryl Stout, WX4QZ, the author stated, "As noted in the Net List Spreadsheet file at **http://www.wx4qz.net/elk.htm** -- there are over

9 **Firewall Solutions**
10 **Examples**
11 **Codeplugs**

200 D-STAR, EchoLink, and D-Rats Nets, in the 4 main US time zones that meet weekly, or as little as once a month. Links to the DMR, CQ100, D-STAR HF, Hamsphere, AllStar, FreeStar, WIRES-X, System Fusion, and Christian Nets are noted later in this file...”

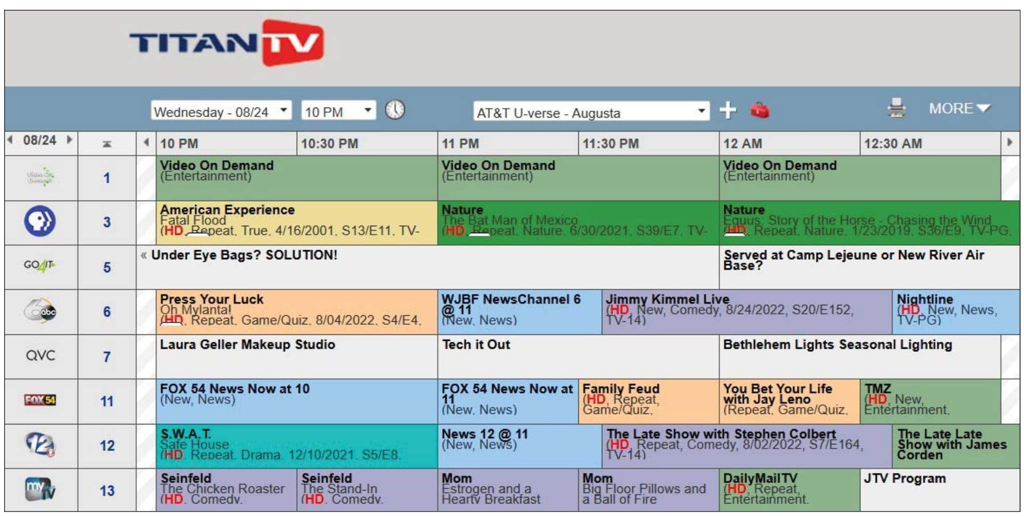
Stout’s 65 page Mode Overview document, updated in July 2022, is a comprehensive overview of getting started with D-STAR, D-Rats, DMR, and the QuadNet Array. It was updated June 8, 2022.

There are hundreds of other regularly scheduled long-running nets combining Digital Voice access primarily with VHF/UHF repeaters. For example, the Amateur Radio Club of Augusta (GA) operates six repeaters including a wide coverage regional repeater. The flagship W4DV repeater transmitting on 145.490 MHz is used by for a nightly net for Radio Amateurs in parts of Georgia, South Carolina and North Carolina. This repeater can be accessed via EchoLink but is not listed anywhere except for its **W4DV.club** web site.

A National Directory of DV Nets

Thorough research of digital voice nets indicates that there may be 1,000 or more Internet accessible nets through one of the many of DV systems in the U.S. alone. A sort of nationwide TV Guide for public nets would go a long way toward facilitating involvement and exploration.

Cable TV system subscribers can go to a scrolling matrix of programs, cable channel and times available. Amateur Radio has no such facility, that I am aware of, with similar information about public nets or regularly scheduled EchoLink conferences. Some examples of television programming displays can be viewed at **TITANTV.com** and **epguides.com**.



Simple reporting formats are familiar to many Radio Amateurs. Examples are Amateur Data Interchange Format (**ADIF**[12]) and **ICS** files widely used in online calendars.

A common reporting format is the use of **Google Calendars** by individuals and groups. Small lists of nets are used by people who want to add someone else’s calendar to their own personal calendar. Examples are: **AugustaHam.net** and the Brandmeister DMR nets calendar compiled by **W0WC**. Google Calendar display options are not useful for many events in one day. A National Directory of DV Nets might utilize the ICS calendar data file format for import and export of net schedules.

12 **ADIF Specifications**

Time and date standardization is required to develop a consistent reporting format. International Standards Organization (**ISO**) **8601** can be used as a standardized way of presenting: Date; Time of day; Coordinated Universal Time (UTC); Local time, etc.

Cooperation & Assistance

To be effective, concept development and promotion needs input and assistance of relevant stakeholders having regular contact with large audiences. This would include: Amateur Radio Digital Communications (**ARDC**); **Frequency Coordination** entities, (relating to repeater operators), ARRL **Affiliated Club Coordinators**, as well as **Section Managers**, DV networks (AllStar, Brandmeister, TGIF, and others) as well as related web sites; Amateur Radio DV radio manufacturers (Icom, Kenwood, Yaesu) also have an interest in growing the DV market.

A team of several qualified Radio Amateurs would be needed to create, develop and manage an Internet accessible National Directory of DV Nets. Skilled volunteers might be recruited from the ranks of stakeholders and other interested persons. Soliciting the input of hosts during a net session also serves to promote the concept and build momentum.

Technically adept volunteers would be needed to build an Internet accessible directory and a mechanism for near-automatic updates. Quality control of listings is important and might be automated to some degree. Promotion would be especially important for the first few years or until the directory is widely known and heavily used.

A web platform for a National DV Net Directory could be a popular web site like **ARRL.org**, **eHam.net**, **QRZ.com** Owners of these sites want to attract new users and the directory would be an attract Radio Amateurs to the host site. For example, eHam.net Vision Statement states “build the largest and most complete Amateur Radio community site on the Internet - a "portal" that hams think of as the first place to go for information, to exchange ideas, and be part of what’s happening with ham radio on the Internet. Establishing a positive working relationship with a host that is motivated by such values seems to be realistic.

Conclusion

A National Directory of DV Nets would foster growth and use of DV. A larger audience for DV nets would drive the creation of new nets and raise the quality of nets overall. This in turn may appeal to younger people who have experience with social media. More users of DV nets would drive demand for DV capable radios, devices and associated equipment.

Growth and development of DV nets supports the basis and purpose of Amateur Radio: Extension of the amateur's proven ability to contribute to the advancement of the radio art.; Advance skills in the communication and technical phases of the art; expansion of the existing reservoir of trained operators, technicians, and electronics experts; Continuation and extension of the amateur's unique ability to enhance international goodwill.

David Vine, WA1EAW
davidvine1999@gmail.com
803-679-3582

Preliminary Analysis of an AI-powered Transcription Bot for FM Transmissions

Zhemin Zhang KD2TAI[1], Brian Robert Callahan AD2BA[1]
1: Rensselaer Polytechnic Institute

Abstract
Amateur Radio communications mediated by Artificial Intelligence (AI) and Machine Learning (ML) provide a wide variety of experimentation. One potential experiment is developing a bot using AI/ML that can read FM voice transmissions, transcribe the audio heard into text, and finally post the transcriptions on social media, all without human intervention [1]. In this paper, we consider some factors that may influence transcription accuracy. While accuracy is important, there are additional considerations for transcriptions, especially those that may be archived on the Internet: these include not posting gibberish and indecent words. We consider these points as improvements to our initial design.

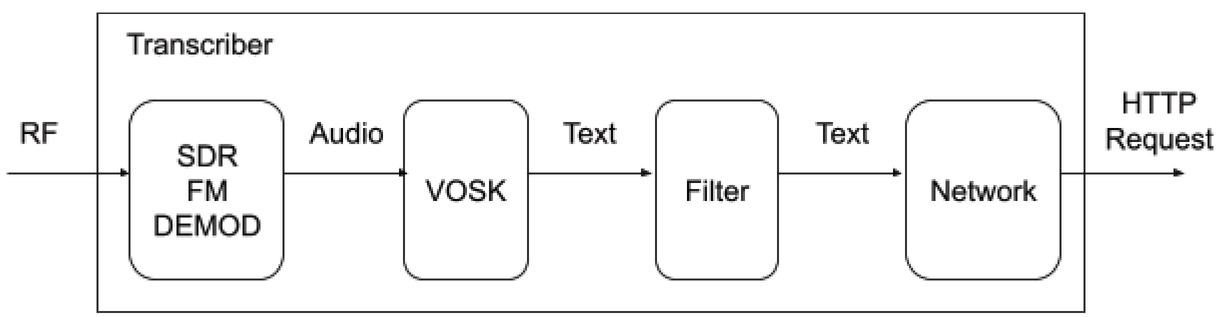


Figure 1. System Block Diagram

Introduction

Artificial Intelligence (AI) and Machine Learning (ML) is an exciting new focus for amateur radio experimentation. In a forthcoming article, we demonstrated a prototype of an AI-powered transcription bot for FM transmissions, named FMBot [1]. The system uses AI/ML technology to transcribe text from audio demodulated from an FM signal using an SDR, then push the transcript to a social media account on the Internet. As our initial prototype was solely a minimum viable product, more questions must be addressed in order to produce a more full-fledged system. Some of the most important questions identified center around accuracy. How accurately can the system transcribe words, and how does it select the messages to be broadcasted? Additionally, as we came to refine our understanding of the problems the AI/ML bot intends to solve, we have come to adopt a broad view of the concept of accuracy to incorporate ideas of unintelligible and inappropriate transmissions.

In this article, we conduct experiments to explore possible factors that may influence the transcription accuracy of FMBot, and measure the degree of such impacts. We also explore methods to filter out meaningless (also known as “gibberish”) or inappropriate messages.

On accuracy

A key takeaway from the minimum viable product was that different models yielded different levels of accuracy. We used VOSK [2] as the open source off-the-shelf AI/ML speech recognition software for transcription. For its English-language models, VOSK offers three: the largest model we learned would only run on quite powerful desktop machines; a moderately powerful laptop was unable to run the largest model. There is a midrange model for such laptops and other similarly midrange-specced machines. Finally, there is a small model for lightweight machines such as the Raspberry Pi.

The size of the model made a noticeable difference in accuracy of the transcribed text. While we didn’t measure the exact accuracy of each model, as conditions between radio and computer setups might skew exact numbers, we felt anecdotally confident that the improved accuracy was noticeable by comparing outputs between different models.

Accuracy in transcription is a common problem for many disciplines. To take just two disparate examples, psychotherapists have argued that increased automatic transcription would increase effectiveness, training, and monitoring [3], and musicians are interested in automatic transcription to aid in capturing music heard accurately into other forms more suitable for analysis [4]. Depending on context, accuracy for our bot might be more or less needed: amateur operators looking for amusement might not require much more accuracy than is needed to enjoy their bot. However, we also envision a bot like the one we are developing acting in part as a response to Covid-19: as clubs turned to Zoom and other Internet services for meeting, our bot can be used to put those meetings squarely back into the amateur radio space over the airwaves without needing to record audio or have someone furiously take notes. For this activity, clubs likely need a higher level of accuracy than those looking for amusement. Moreso, we also envision the bot enabling voice activities such as voice contesting for amateur operators who might not otherwise be able to participate in such voice-based activities.

Not only does accuracy mean the literal English-words-to-English-text that we often think, in our preliminary experiments we noticed oftentimes the bot would post transcripts of gibberish, effectively nonsense and nonsense words. We also noticed that the bot would interpret pockets of silence as the word “the” which led us to eventually implement a filter that would reject all single-word messages. We also think of accuracy in terms of not posting inappropriate transmissions. Such transmissions could include hate speech, offensive words, and other transmissions that may be deemed “not safe for work.” While the FCC does have rules prohibiting the transmission of “obscene or indecent words or language” [5], it does not necessarily follow that all amateur transmissions are free of such language. Amateurs with transcription bots of their own may not want such language transcribed into text and then posted on social media on the open Internet on their accounts. A bot ought to be able to identify inappropriate transmissions and prevent the posting of transmissions including such words to the best of its ability.

Experiments

We have identified two factors that are important to transcription accuracy: signal-to-noise ratio (SNR) and bandwidth. Intuitively, it is harder to understand another person's words with noise in the background or talking over the telephone, which demonstrates the effect of the two aforementioned factors. We previously discussed the impact of SNR on accuracy quantitatively [1], so we will focus solely on bandwidth here.

These are the specifications for the hardware and software we used to conduct our experiments.

a. Hardware specifications
    i. Processor: AMD Ryzen 7 3800X 8-Core Processor 3.90GHz
    ii. Installed RAM: 32 GB
    iii. Software-Defined Radio: RSP1 kit connected via USB
b. Windows specifications
    i. Edition: Windows 10 Home
    ii. Version: 2004
c. Software specifications
    i. Python 3.9.12
    i. VOSK 0.3.42
    ii. PyAudio 0.2.11

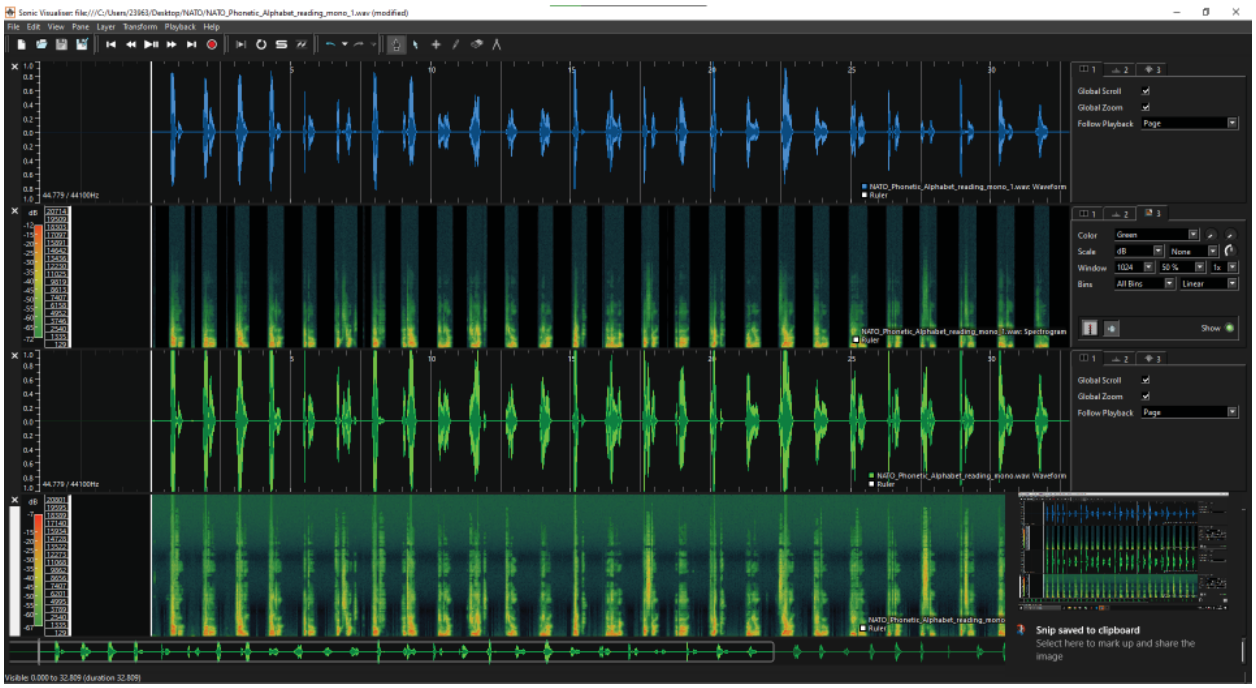


Figure 2. Visualization of the filtered spectrogram (top) vs. the original one (bottom). 300 Hz low cutoff and 2700 Hz high cutoff. Note the intensity difference at higher frequencies.

The bandwidth contains two variables: the low and high cutoffs. First, we test the recognition accuracy of a clear recording (SNR > 64 dB) by adjusting two boundaries of a 4th-order Butterworth filter at 100 Hz or 200 Hz intervals, respectively, for low and high cutoffs. Since the way the filter was implemented prohibits low-cutoff to be zero, we set it to 1 Hz instead.

We then iterate over filtered audio files to perform transcription. Punctuation is included in VOSK output, however we chose to drop all punctuation and then format the output one word per line. Some recognized words are replaced using sed (e.g., “nine” to “niner”), as some words in NATO phonetic alphabets are read differently than in common English. Then we use the command

$ diff -y --suppress-common-lines standard.txt result.txt | *wc -l*

to report the differences between the transcript and ground truth. Another script collects the result and plots a 2D heatmap, as shown in Figure 3, also known as the contour map for discrete values, to demonstrate how two variables together affect the accuracy.

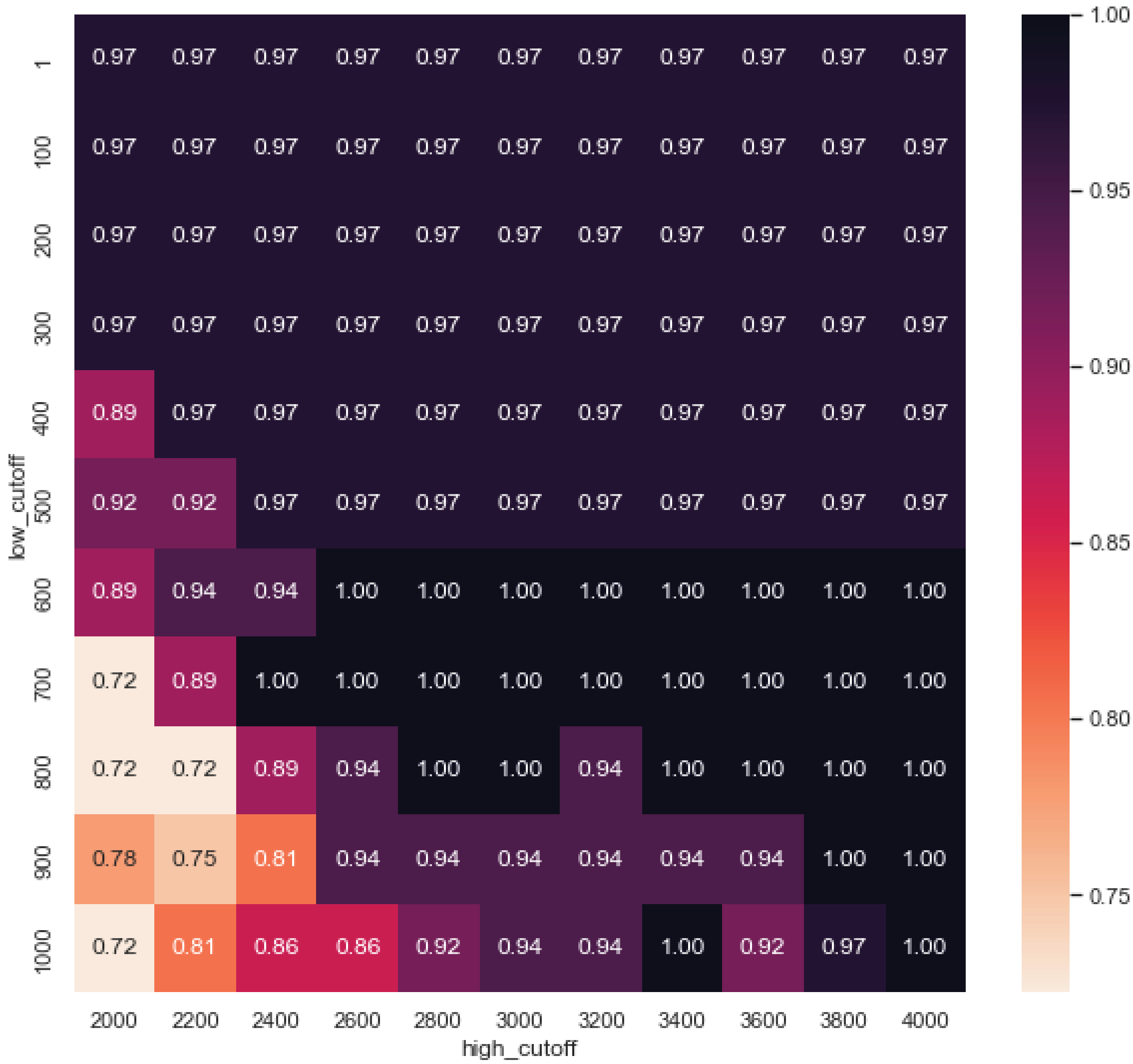


Figure 3. Heatmap for NATO phonetic alphabet recognition accuracy with various low and high cutoffs in Hz.

The accuracy deteriorates as the bandwidth gets narrower, in other words, towards the bottom left of the map. For example, to yield an accuracy greater than 90%, the minimum bandwidth needs to be greater than 1700 Hz, whereas it takes 2800 Hz on average to reach near 100% accuracy.

Note that this does not mean that lower lower-bound and higher upper-bound are always desirable. For certain words, we expect cutoffs to be at the appropriate location to improve recognition. The error most likely happening in this test setup is the pair “kilo” and “kino,” and this is fixed as the low cut-off rises, which is visualized by the bottom right corner being darker than the top.

Like the previous SNR test, human perception outperforms the AI transcriber. Even at the narrowest bandwidth (1000, 2000), it is still understandable. This is probably due to the context that humans can understand. For example, in this NATO alphabet testing, human operators specially trained for it will subconsciously correlate the word heard with one of 36 possibilities in the set, while to achieve the same effect, the programmer needs to specify such information to a machine.

In the above bandwidth test and the previous SNR test, we used NATO phonetic alphabet recording as audio input. Despite being a good starting point as it is a standard format to exchange call signs, it is only a small portion of ham radio traffic. Furthermore, the recording pronounces a single word at a time with extra long spacing; this speech pattern is uncommon in real conversation. As such, we chose an excerpt from the Wikipedia page on artificial intelligence [6], recorded by Zhemin himself in a young male voice, to test the system's performance in a more practical situation. Below is an annotated excerpt; the contents in parentheses were unintelligible to the bot due to their unstressed nature, and the parts in square brackets indicate a human error made when reading. These factors are compensated for in our accuracy calculations.

> *Artificial intelligence is intelligence demonstrated by machines, as opposed to a natural intelligence displayed by animals including humans. AI research has been defined as the field of study of intelligent agents, which refers to any system that perceives its environment and takes actions that maximize (its) chance of achieving its goals.*
>
> *The term "artificial intelligence" had previously been used to describe machine(s) that mimic and display "human" cognitive skills that are associated with (the) human mind[s], such as "learning" and "problem-solving." This definition has since been rejected by major AI researchers who now describe AI in terms of rationality and acting rationally, which does not limit how intelligence can be articulated.*

The microphone, an RØDE NT-USB Mini, was placed 40 centimeters away, with Windows Voice Recorder running on the PC with a default sampling rate of 48000 Hz. The audio is then noise reduced using Audacity 3.0.2 with reference to silence before talking starts. After that, we applied the compressor with default settings to yield the reference audio for this test. Finally, we repeat the above experiment steps to generate the result in Figure 4.

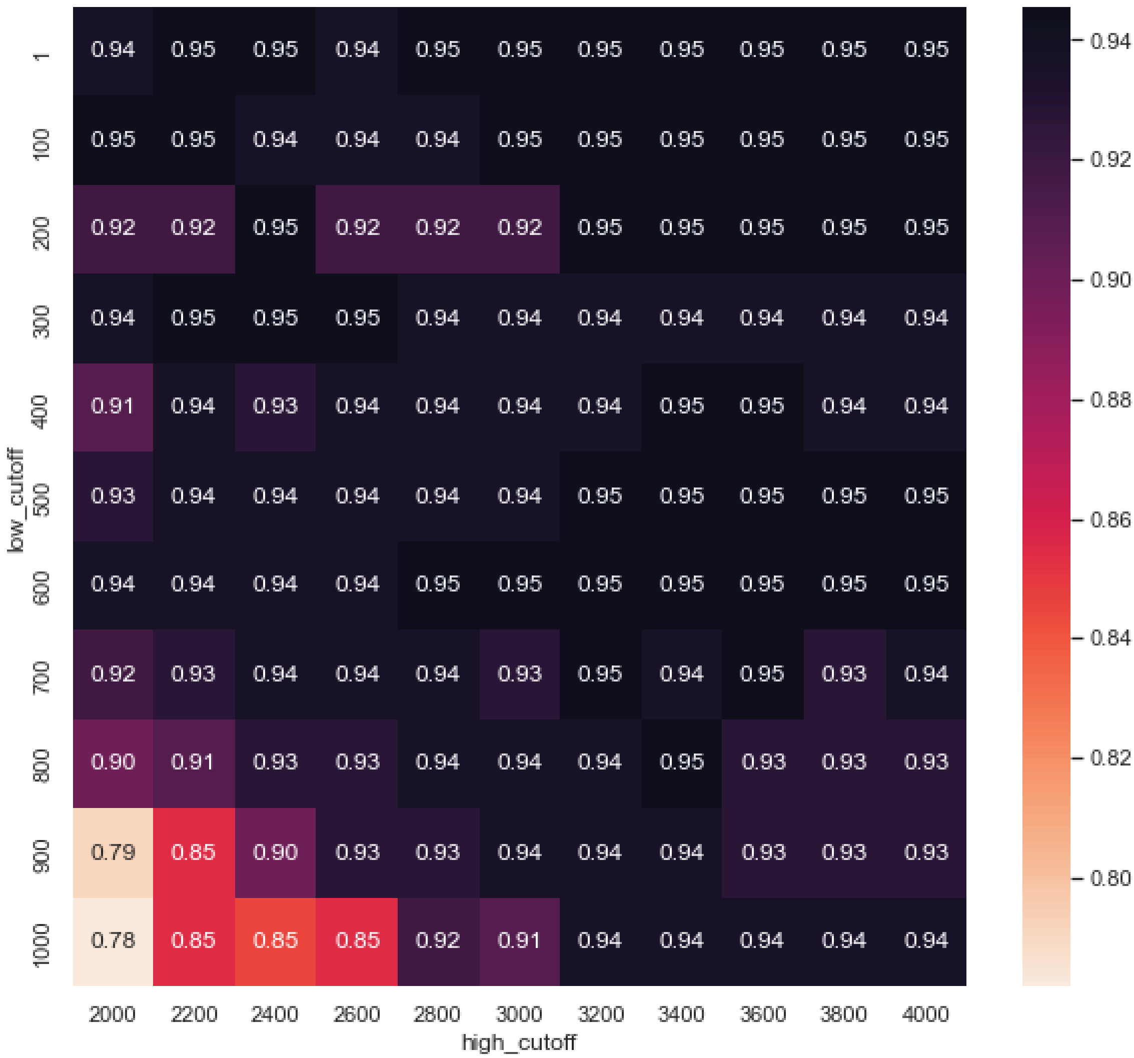


Figure 4. Heatmap for AI Wikipedia article recognition accuracy with various low and high cutoffs in Hz.

As expected, the overall accuracy decreases in this test set. We believe the most plausible explanation lies in the limited vocabularies of the model; accuracy would likely be improved if the model could be trained on a large corpus of actual amateur radio transmissions and conversations. The stressed and unstressed syllables and words found in natural speech patterns are a challenge to the model. The countermeasure for the vocabularies issue is to train the model with an even larger pool of words, although improvement will be limited given the law of diminishing returns; the size of the model grows rapidly as words rarely used in everyday conversation get included. To the stress and unstress issue, a professional announcer may overcome the obscurity, but as the system targets the public, there may be little that can be done.

What is unexpected, however, is that in our experiments the male voice allows a higher low-cutoff frequency before the accuracy deteriorates compared to the female voice in the NATO test set. Intuition suggests that the male voice in general has a lower fundamental frequency. With a higher low-cutoff, we might expect a more severe loss of information, yet the measurement suggests opposing evidence. One possible explanation is that the model extracts information from the harmonics more than the fundamental frequency, but without solid evidence, this phenomenon remains unresolved and is worth further study.

Recently, the editor-in-chief of *QEX* called for more rag chewing [7, 8], and we could not agree more–as that would significantly aid in training an amateur radio-specific model to increase the accuracy of the bot! CQ CQ CQ any audio you would like to share for improving the accuracy of the bot.

Message Filtering

The filtering process involves multiple filters for different purposes, all cascaded together. Each unit specialized in specific topic detections is joined by “OR” logic to assemble a complete filter module. Figure 5 shows an example configuration and its flowchart. An inaccurate RF analogy to this is a series of bandstop filters that reject undesirable information. This flexible design allows more user-defined filters to be added later on.

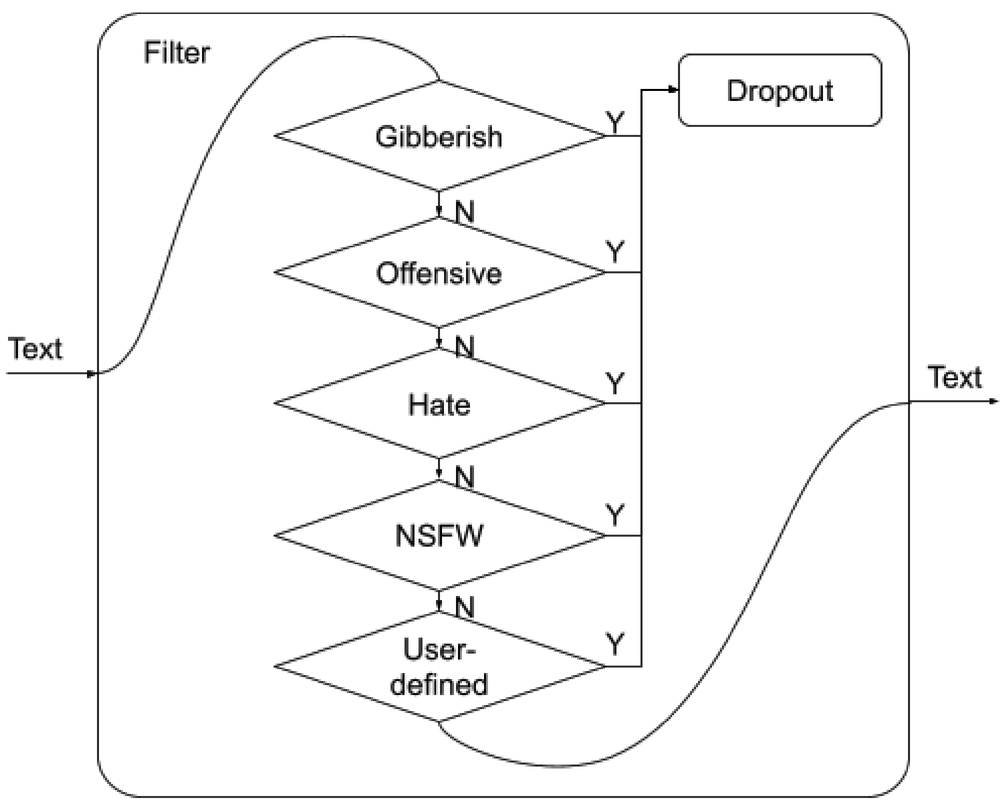


Figure 5. A Closer Look at Filtering Mechanism

There may be some concerns with a filtering mechanism that an amateur operator ought to be aware of. First is the idea of censorship. While it is of course true in the United States that a private person has the right to determine what speech makes it out of the filter, one operator's peace of mind can sometimes be felt as another operator's censorship. Nonetheless, censorship may be an important consideration in other locations.

Additionally, different taboos exist across different cultures; what constitutes an offensive word to one person may be a totally innocuous word to another, even within the same language. Consider the differences in slang among American, British, and Australian English for some notable examples. This means there will never be a one-size-fits-all filter. The filter will always have some degree of cultural context implicit in its design and operation. We consider this to be perfectly acceptable and perhaps even desirable, as it helps to imagine the flexibility of the filter design overall while being able to be highly adaptable to any situation.

Future filtering work

We have identified two important areas of further work that could be added to enhance the power of the filtering mechanism. First, we might consider the informativeness of a message. A message that was not sufficiently informative by any number of metrics might be a good candidate to drop. Second, we might consider messages containing sensitive information, such as passwords to online accounts, and it would be important to drop those messages as well lest an operator's secrets be divulged to the world.

Conclusion

In designing the original AI/ML transcription bot, we discovered several issues with the accuracy of transcription that required further exploration. In this paper, we conducted a number of experiments designed to better understand how we could refine the accuracy, defined broadly, of the bot. Our experiments examined the effects of bandwidth on transcription accuracy. We then experimented with filters for unintelligible and inappropriate transmissions, designing a system that is flexible enough to deal with cultural contexts.

Future work may identify additional variables that affect transcription accuracy. Future work may also explore concepts such as informativeness of a message in determining if it should be posted to social media. The filter may also be extended to not post sensitive information, such as passwords.

The continued development and experimentation of the AI/ML transcription bot demonstrates the viability of research and experimentation at the intersection of amateur radio and AI/ML. It is our hope that this article inspires future work by other amateur operators in this space.

Works Cited

[1] Callahan, B. R., and Z. Zhang. Forthcoming. “An AI-powered transcription bot for FM transmissions.” *QEX*.

[2] https://alphacephei.com/vosk/

[3] Miner, A. S, A. Haque, J. A. Fries, S. L. Fleming, D. E. Wilfley, G. T. Wilson, A. Milstein, D. Jurafsky, B. A. Arnow, W. S. Agras, L. Fei-Fei, and N. H. Shah. 2020. “Accessing the accuracy of automatic speech recognition for psychotherapy.” npj Digital Medicine 3, 82.

[4] Mauch, M., C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon. 2015. “Computer-aided Melody Note Transcription Using the Tony Software: Accuracy and Efficiency.” Available online: https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/7247/tony-paper_preprint.pdf

[5] § 97.113 Prohibited transmissions. Available online: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-D/part-97#97.113

[6] https://en.wikipedia.org/wiki/Artificial_intelligence

[7] Siwiak, K. 2022. “Perspectives.” *QEX* July/August 2022: 2.

[8] Siwiak, K. 2022. “Perspectives.” *QEX* September/October 2022: 2.

Notes

ISBN: 978-1-62595-138-0
90000
9 781625 951380