Software Development Stories

Bringing FreeDV Into the Modern Era

Mooneer Salem K6AQ

Who am I?

Why listen to me?

- One of the maintainers of the FreeDV project
  - Been committing to the repos since 2019
- Licensed as a ham for a bit
  - KG6AOV initially, then K6AQ
- Do a lot of C++ stuff for my day job too

What we’re going to talk about today

What you’ll get out of listening

- How FreeDV came to be
- Design decisions on the way
- How you can join in its future

What is FreeDV?

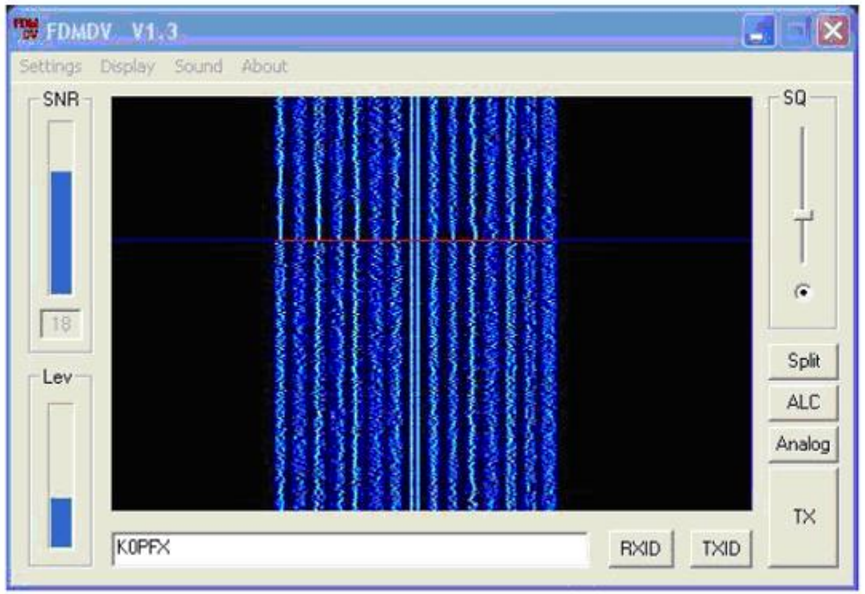
What's in it for me?

- Digital voice mode designed for HF
  - Most ham radio digital voice options are primarily VHF/UHF+
- Free and open source
  - No patents/licensing concerns
- No special hardware needed
  - If you're already doing FT8 and Zoom, you're set!

The Early Days

How did FreeDV start?

FDMDV...Digital Voice for HF
Version 1.3



1.125 KHz Bandwidth FDMDV Spectrum

(Thanks to https://www.qsl.net/py3cnq/FDMDV/fdmdv.html for screenshot)

FDMDV Struck Down

The same patent/commercial codec story

- The commercial codec FDMDV used was “okay” for hams
  - Operated in a gray area since the owner didn’t explicitly give permission
- …until it wasn’t
  - The owner changed their minds one day without notice

A new Codec2 emerges

No more patents/licensing to worry about

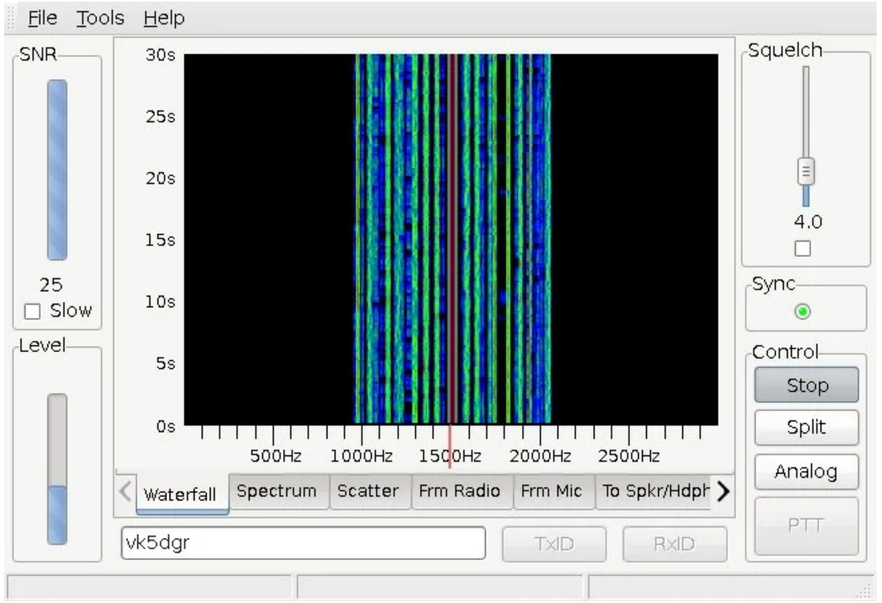
- Developed by David Rowe VK5DGR in 2012 (1)
  - No truly open digital voice solution for HF existed before now
  - LGPL licensed
- Originally contained one mode (“1600”)
- Written in the C language (not C++)

(1) https://github.com/drowe67/codec2/blob/main/README_freedv.md

FreeDV and Codec2

Leading up to first FreeDV release

- First commit to freedv-gui repo: 2015-08-13 (1)
  - “Rename fdmdv to freedv”
  - First mention of Codec2 in the repo as well
- FreeDV 1.4 released on 2019-11-11



(1) https://github.com/drowe67/freedv-gui/commit/a362d8ebe7efd9604edf537d7d81af97a8176baa

(2) Thanks to https://www.reddit.com/r/RTLSDR/comments/35n7tc/freedv_digital_voice_for_hf/ for screenshot!

Cross Platform Development

Linux *AND* Windows?

- Codebase was already fairly cross platform
  - wxWidgets for UI
  - Hamlib for rig control
- MinGW gives us a gcc-like experience but for Windows

(1) https://github.com/drowe67/freedv-gui/pull/26

FreeDV and macOS

Challenges Ahead

- Initial work on producing builds that run in 2019 (1)
- macOS is special
  - Supports “universal” builds (aarch64 + x86_64) in the same binary
  - Its Clang is slightly different than the official one
  - Some third-party dependencies had issues with macOS

(1) https://github.com/drowe67/freedv-gui/pull/26

FreeDV and macOS

Challenges Ahead

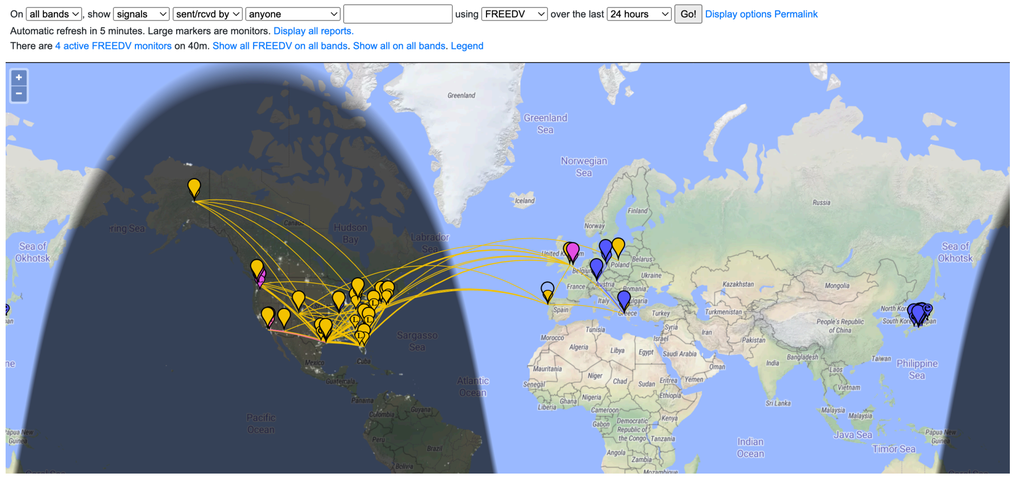
- By default, compiled for running version of the OS and newer
  - Need to force compilation for earlier macOS: -mmacosx-version-min=…
  - Otherwise out of luck
- Initially supported macOS 10.9 (“Mavericks”), now 11.0+ (“Big Sur”)

(1) https://github.com/drowe67/freedv-gui/pull/26

FreeDV Spotting

How will anyone know I’m on the air?

- Originally supported PSK Reporter (https://pskreporter.info/)
  - Uses IPFIX protocol over UDP for sending packets
  - Windows DLL provided by author, but we needed something cross platform



FreeDV QSO Finder

IRC, but for ham radio

- QSO Finder was the initial spotting service for FreeDV
  - Chat interface let you coordinate contacts over the internet
  - Frequency buttons/text box let you tell people where you were on the air
- HF D-Star also used something similar

Thanks to https://www.slideserve.com/ada/north-shore-radio-club-what-s-new-in-digital-modes-november-12-2013 for screenshot!

FreeDV QSO Finder

IRC, but for ham radio

- A few problems with the site
  - The maintainer (John Hays K7VE) went SK :(
  - Need to manually choose the frequency you’re on
  - Was a separate webpage you had to have open (and be paying attention to)

FreeDV Gets ARDC Grant

Now we're really cooking

- FreeDV awarded grant from ARDC in 2023
- Grant objectives:
  - Improve the audio quality of FreeDV
  - Get more people using FreeDV (marketing, radio integration)

Enter FreeDV Reporter

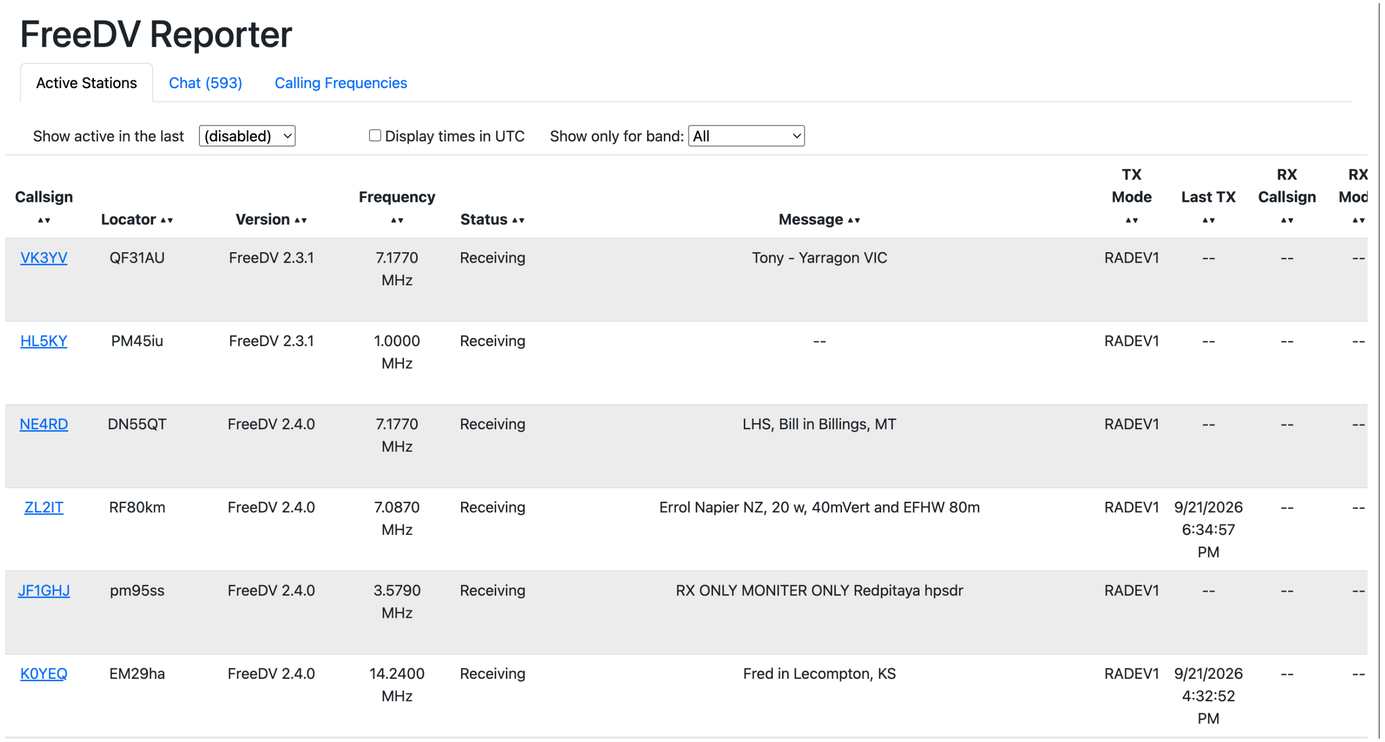
Actual live spotting

- Live view of current FreeDV activity
  - Server updates within 1-2 seconds of an event
  - Example events: frequency change, RX<->TX switching, RX signal
- JSON over Socket.IO link to Python based server
  - Server basically rebroadcasts received events to clients
- Started becoming a thing in version 1.8.10

Enter FreeDV Reporter

Actual live spotting

- Initially website only; application simply sent stuff to the server



FreeDV Reporter

Active Stations | Chat (593) | Calling Frequencies

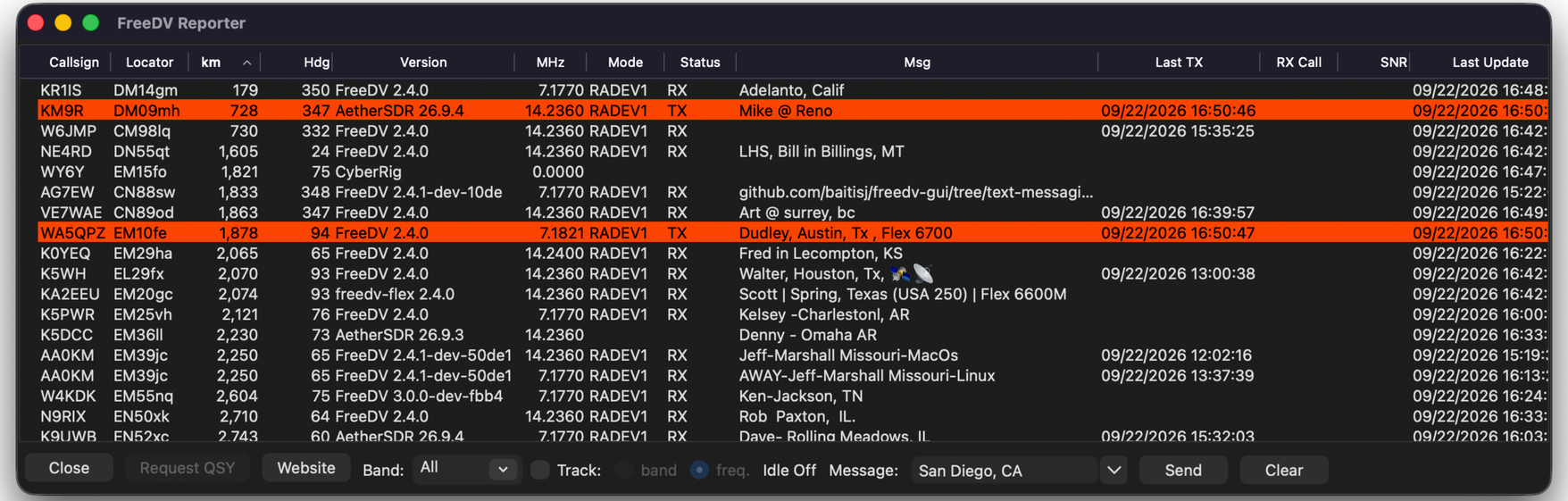
Show active in the last (disabled) Display times in UTC Show only for band: All

Callsign	Locator	Version	Frequency	Status	Message	TX Mode	Last TX	RX Callsign	RX Mod
VK3YV	QF31AU	FreeDV 2.3.1	7.1770 MHz	Receiving	Tony - Yarragon VIC	RADEV1	--	--	--
HL5KY	PM45iu	FreeDV 2.3.1	1.0000 MHz	Receiving	--	RADEV1	--	--	--
NE4RD	DN55QT	FreeDV 2.4.0	7.1770 MHz	Receiving	LHS, Bill in Billings, MT	RADEV1	--	--	--
ZL2IT	RF80km	FreeDV 2.4.0	7.0870 MHz	Receiving	Errol Napier NZ, 20 w, 40mVert and EFHW 80m	RADEV1	9/21/2026 6:34:57 PM	--	--
JF1GHJ	pm95ss	FreeDV 2.4.0	3.5790 MHz	Receiving	RX ONLY MONITER ONLY Redpitaya hpsdr	RADEV1	--	--	--
K0YEQ	EM29ha	FreeDV 2.4.0	14.2400 MHz	Receiving	Fred in Lecompton, KS	RADEV1	9/21/2026 4:32:52 PM	--	--

Enter FreeDV Reporter

Actual live spotting

- Added built-in client in 1.8.12



Introducing Radio Autoencoder

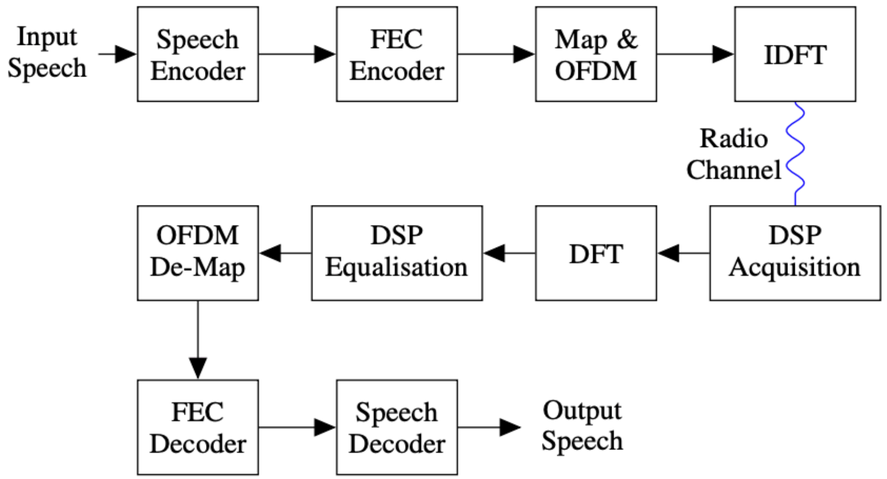
Change in direction

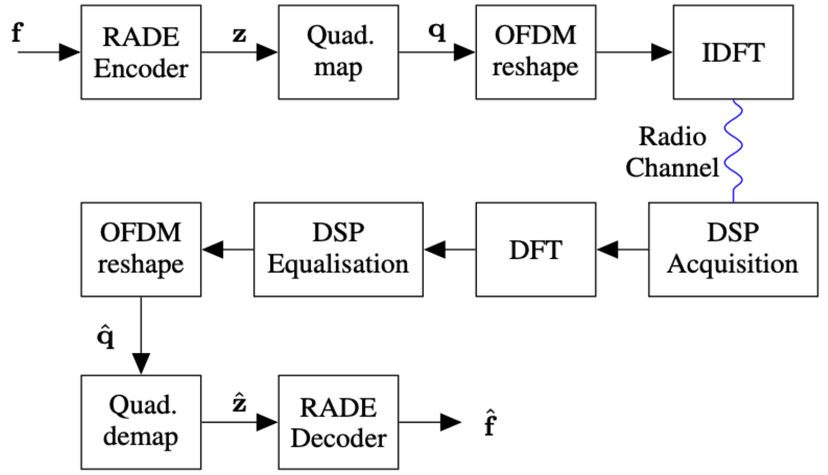
- Originally we were trying to improve the quality of the existing FreeDV modes
  - May have been possible with additional development
  - Unknown how much of an improvement would have happened
- 2024: New developments in machine learning made a new mode possible
  - Radio Autoencoder (“RADE”)

Introducing Radio Autoencoder

Change in direction

- Classical digital voice mostly uses common DSP
  - RADE replaces some of these with machine learning models





Diagrams from RADE paper: https://arxiv.org/pdf/2505.06671

Integrating RADE into FreeDV

Python everywhere

- Initially a mix of Python and C code
  - Dramatically ballooned FreeDV download size (from 10s to 100s of MB)
  - Performance issues
- Required Python to be installed
  - Auto-install on Windows sometimes failed

Continuous Integration Nightmares

Really, GitHub?

- FreeDV projects use CI to build and test releases/commits
  - freedv-gui in particular uses virtual audio devices and actually tests RX/TX
  - Tests on all three supported platforms (macOS/Linux/Windows)
- Consistent pass of all CI tests has been a problem since RADE's release
  - Same tests pass perfectly fine locally… 🤔

Continuous Integration Nightmares

Really, GitHub?

- Proper real-time audio hygiene
  - http://www.rossbencina.com/code/real-time-audio-programming-101-time-waits-for-nothing
- Performance optimization where applicable
  - Modifying libsamplerate to allow SIMD to happen
  - Reducing number of calls to libebur128 (AGC)
- Currently (2026): 1-2 tests out of 50+ on freedv-gui typically fail

Logging contacts

For the awards

- FreeDV had no easy support for logging contacts
  - Manual entry typically required
  - High demand feature, but don't want to implement a full logger
- Version 2.2.0 got logging support for the first time
  - WSJT-X network protocol over UDP

The AI Era arrives

Let the code slop begin

- ChatGPT first became accessible in 2022
  - Claude, Grok, etc. followed
- Finally made it possible to automate significant parts of the coding gruntwork
  - Example: porting the Python parts of RADE to C

Porting RADE to C

Voice go brrrrr

- Version 2.3.0: Python went away from RADE entirely
  - Basically gave Claude the RADE codebase and told it to port to C
  - Got a lot correct from the start
- Still needed to review/test the results prior to full release
- Found and fixed several issues prior to 2.3.0's release and afterward

What about Docker?

FlexRadio support for FreeDV

- Previously, we weren’t able to run FreeDV on Flex radios themselves
  - Not enough CPU available even for the legacy modes :(
- This changed with the Flex 8000 and Aurora series
  - Docker support + additional processing power (Raspberry Pi CM4)
  - Simply upload a file and FreeDV gets added as a mode (FDVU/FDVL)
- Would still not have been possible without getting rid of Python first

Older Flex radios can join in too

No Docker? No problem!

- Flex radios allow sending waveform data over the network
  - This means that the actual FreeDV code can run elsewhere
- Flex 6000 series support this as well as 8000/Aurora
  - Install the Flex AppImage on a Raspberry Pi 4 or newer
- Auto detection via UDP discovery
  - TCP control channel + VITA-49 over UDP for audio

Additional third party support

Use FreeDV however you want

- Users organically integrated FreeDV into their own software
  - Examples: AetherSDR, Thetis-RADE
  - Many more listed at https://freedv.org/alternative-freedv-rade-v1-software/
- Made possible by having a full C implementation of RADE

Making integration easier

Just drop FreeDV in and go

- Began splitting off the backend code into a new repository
  - https://github.com/tmiw/freedv-backend
  - (Note: This will move to the freedv organization by EOY.)
- Designed as enough to implement your own FreeDV client
  - Just add UI (or lack thereof) and add this as a Git submodule
  - BSD licensed (along with RADE itself) to encourage commercial adoption

What’s next?

Can future be predicted?

- Version 2 of RADE
  - More information at Pacificon :)
- Further (commercial?) radio integration?
- It's up to you!

How can I get involved?

Be the change you want

- Tell your preferred radio manufacturer that you want FreeDV
  - Enough people doing so will get their attention
- Integrate FreeDV into your own projects
  - SDR clients, web-based receivers, etc.

How can I get involved?

Be the change you want

- Contribute to the FreeDV codebase
  - freedv-gui: Reference client for FreeDV (Windows, macOS, Linux)
  - freedv-backend: Backend audio/network/etc. logic.
  - freedv-integrations: Official FreeDV radio integrations (FlexRadio/web SDR)
  - rade_c / radae: RADE encoder/decoder.
- C++20 for all but rade_c/radae (straight C/Python for the latter)

How can I get involved?

Be the change you want

- Create a PR and/or issue for bug fixes
  - Needs GitHub account
- New features: Submitting a Feature Request is highly encouraged
  - Create GitHub issue -> Choose “Feature Request”
  - PRs may be delayed in review/approval until FR submitted/approved

Thanks for Listening!

Any questions?

- Contact me anytime with questions: mooneer@gmail.com
  - Will also be at Pacificon with a booth and a different FreeDV talk
- FreeDV Discord (scan QR code or visit freedv.org)