An open-source digital radio protocol for amateur radio

The M17 protocol

Morgan Diepart

February 4, 2024

Table of contents

Who am I?

Figure 1: Morgan Diepart

- ▶ Research engineer at University of Liège (Belgium) in embedded systems and RF
- ▶ Licensed ham radio operator ON4MOD
- ▶ Joined M17Project in February 2023 (right after FOSDEM23)
- ▶ Doing mostly hardware design and firmware works

Amateur Radio

- It is a technical hobby
- Allows you to legally transmit on certain frequencies
- Extremely vast (antennas, transceivers, DX, ...)
- Most hardware produced by just a few brands...
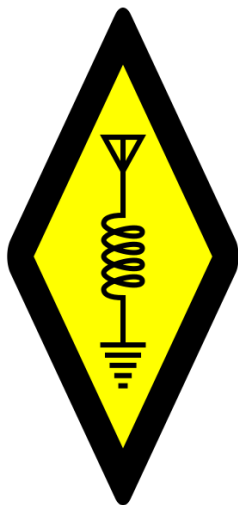
Open Source in Ham Radio

The *Ham Spirit* always promoted sharing designs, ideas, discoveries, ...

Open Source in Ham Radio

The *Ham Spirit* always promoted sharing designs, ideas, discoveries, ...

Which is not to be confused with...

Some countries disallow ciphered communications.
Forces manufacturers to publish their protocol's specifications.
That does not make it FOSS...

Some protocols freely available

Many protocols exist and you could use any of those...

Some protocols freely available

Many protocols exist and you could use any of those...

- ▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.

Some protocols freely available

Many protocols exist and you could use any of those...

▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.

▶ DSTAR: First protocol created for amateur radio. Designed for digital voice. VHF/UHF. Open specifications.

Some protocols freely available

Many protocols exist and you could use any of those...

- ▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.
- ▶ DSTAR: First protocol created for amateur radio. Designed for digital voice. VHF/UHF. Open specifications.
- ▶ YSF: Proprietary mode from Yaesu. Data and digital voice.

Some protocols freely available

Many protocols exist and you could use any of those...

- ▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.
- ▶ DSTAR: First protocol created for amateur radio. Designed for digital voice. VHF/UHF. Open specifications.
- ▶ YSF: Proprietary mode from Yaesu. Data and digital voice.
- ▶ FT-4 / FT-8 : Very long range (1000s km), very low power (10 W), very slow speed, HF

Some protocols freely available

Many protocols exist and you could use any of those...

- ▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.
- ▶ DSTAR: First protocol created for amateur radio. Designed for digital voice. VHF/UHF. Open specifications.
- ▶ YSF: Proprietary mode from Yaesu. Data and digital voice.
- ▶ FT-4 / FT-8 : Very long range (1000s km), very low power (10 W), very slow speed, HF
- ▶ DMR/TETRA/P.25: A whole range of industrial standards adapted for amateur radio use

Some protocols freely available

Many protocols exist and you could use any of those...

- ▶ AX.25 : Data, VHF/UHF. Adaptation of the X.25 protocol for amateur radio.
- ▶ DSTAR: First protocol created for amateur radio. Designed for digital voice. VHF/UHF. <u>Open specifications</u>.
- ▶ YSF: Proprietary mode from Yaesu. Data and digital voice.
- ▶ FT-4 / FT-8 : Very long range (1000s km), very low power (10 W), very slow speed, HF
- ▶ DMR/TETRA/P.25: A whole range of industrial standards adapted for amateur radio use

None is truly FOSS. Sometimes only one, closed source, implementation exists.

Taking DSTAR for an example

First versions of DSTAR specifications is released in 2001 by JARL (Japan).

Taking DSTAR for an example

First versions of DSTAR specifications is released in 2001 by JARL (Japan).

Uses the AMBE codec from DVSI.

Taking DSTAR for an example

First versions of DSTAR specifications is released in 2001 by JARL (Japan).

Uses the AMBE codec from DVSI.

Specifications are publicly available but no license specified.
A manufacturer does not have to follow the published specifications.

Taking DSTAR for an example

First versions of DSTAR specifications is released in 2001 by JARL (Japan).

Uses the AMBE codec from DVSI.

Specifications are publicly available but no license specified. A manufacturer does not have to follow the published specifications.

DSTAR de-facto became ICOM's proprietary mode.

Main obstacles

Licensing issues

Manufacturers exploit the lack of licenses to lock down their environments...

Main obstacles

Computing power at the time (2001) was limited.
The vocoder is implemented in an ASIC and the algorithms are
protected by patents, industrial secrets,...

Computing power at the time (2001) was limited.
The vocoder is implemented in an ASIC and the algorithms are
protected by patents, industrial secrets,...

The same algorithm (or variants of it) is used in all the mainstream
digital voices modes... (DSTAR, YSF, DMR, P25, NXDN,...).

Computing power at the time (2001) was limited.
The vocoder is implemented in an ASIC and the algorithms are protected by patents, industrial secrets,...

The same algorithm (or variants of it) is used in all the mainstream digital voices modes... (DSTAR, YSF, DMR, P25, NXDN,...).

There are historic reasons, but how does one tinker with it?

Computing power at the time (2001) was limited.
The vocoder is implemented in an ASIC and the algorithms are
protected by patents, industrial secrets,...

The same algorithm (or variants of it) is used in all the mainstream
digital voices modes... (DSTAR, YSF, DMR, P25, NXDN,...).

There are historic reasons, but how does one tinker with it?
Answer: you do not.

The vocoder being an integral part of the protocol. It it possible to have a fully FOSS protocol?

And how?

The solution
Codec 2

The solution came in the form of Codec 2 released in 2010 by David Rowe (VK5DGR).

The solution
Codec 2

The solution came in the form of Codec 2 released in 2010 by David Rowe (VK5DGR).

Fully open-source, no patents, no industrial secrets.

The solution
Codec 2

The solution came in the form of Codec 2 released in 2010 by David Rowe (VK5DGR).

Fully open-source, no patents, no industrial secrets.

Since then the computing power increased significantly, so much that a 32-bits ARM MCU is enough.

This last brick allowed the emergence of fully open-source
protocols such as:

Then came...

This last brick allowed the emergence of fully open-source protocols such as:

▶ FreeDV (GPL v2.1)
  Using Codec 2 at a low bit-rate, narrower bandwidth in HF.
  Published in 2012.
  It is the reference Codec 2 implementation.

Then came...

This last brick allowed the emergence of fully open-source protocols such as:

- ▶ FreeDV (GPL v2.1)
  Using Codec 2 at a low bit-rate, narrower bandwidth in HF.
  Published in 2012.
  It is the reference Codec 2 implementation.

- ▶ M17 (GPL v2)
  Using Codec2 at highest bit-rate, standard FM bandwidth, for VHF and up.
  Published in 2019.

The M17 protocol
Finally!

Has all the features one can expect

- ▶ Packet mode
- ▶ Stream mode
- ▶ AES encryption
- ▶ Specifications for traffic over IP
- ▶ ...

The M17 protocol

Finally!

Has all the features one can expect

- ▶ Packet mode
- ▶ Stream mode
- ▶ AES encryption
- ▶ Specifications for traffic over IP
- ▶ ...

Most importantly, the specifications are open-source and licensed under GPL v2.

The M17 Project

> Because M17 is now more than just a protocol!
> Getting rid of the proprietary vocoder allows implementations...

The M17 Project

Because M17 is now more than just a protocol!
Getting rid of the proprietary vocoder allows implementations...

- for your computer (mvoice, GNURadio OOT module, libm17, ...),

The M17 Project
Beyond the protocol

Because M17 is now more than just a protocol!
Getting rid of the proprietary vocoder allows implementations...

- ▶ for your computer (mvoice, GNURadio OOT module, libm17, ...),
- ▶ for your phone (Droidstar),

The M17 Project

Because M17 is now more than just a protocol!

Getting rid of the proprietary vocoder allows implementations...

▶ for your computer (mvoice, GNURadio OOT module, libm17, ...),

▶ for your phone (Droidstar),

▶ for your radios,

The M17 Project

Because M17 is now more than just a protocol!
Getting rid of the proprietary vocoder allows implementations...

- ▶ for your computer (mvoice, GNURadio OOT module, libm17, ...),
- ▶ for your phone (Droidstar),
- ▶ for your radios,
- ▶ on servers (reflectors),

The M17 Project

Because M17 is now more than just a protocol!
Getting rid of the proprietary vocoder allows implementations...

- ▶ for your computer (mvoice, GNURadio OOT module, libm17, ...),
- ▶ for your phone (Droidstar),
- ▶ for your radios,
- ▶ on servers (reflectors),
- ▶ ... .

A whole ecosystem.

Module17

Modem that allows you to TX M17 using FM radio

- ▶ Open-source protocol...
- ▶ Open-source hardware
- ▶ Open-source software
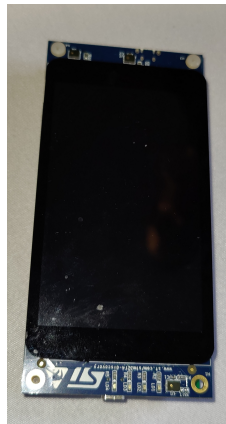- ▶ Affordable
- ▶ Build it yourself!

OpenHT
Open source portable transceiver

If you can modulate it we can send it...
For now...

▶ Open-source hardware*

▶ Open-source firmware*

▶ 430 MHz / 2.4 GHz @25 mW

FPGA toolchain and IPs are not

open-source

OpenHT
Open source portable transceiver
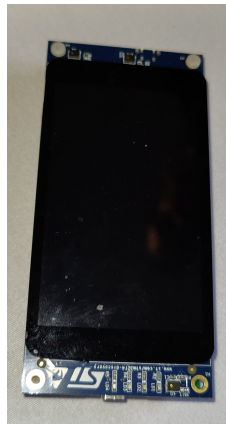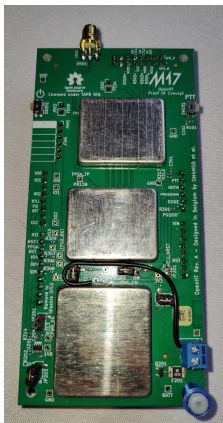
If you can modulate it we can send it...

For now...

▶ Open-source hardware*

▶ Open-source firmware*

▶ 430 MHz / 2.4 GHz @25 mW

FPGA toolchain and IPs are not open-source

For the future

▶ OpenRTX

▶ USB-C charging!!

▶ Open-source FPGA

▶ 5W output

The larger ecosystem

A shootout to very interesting projects close to M17

- ▶ OpenRTX, *the* open-source firmware,
- ▶ WPSD, *the* hotspot software,
- ▶ MMDVM, *the* hotspot hardware,
- ▶ ...

Thank you for your attention!

m17project.org

Checkout the Ham Radio infobooth in building AW.