

# M17 & OpenRTX

Silvano Seva - IU2KWO

10th February 2024

*whoami*



- Also known as Redman
- Born and living in Milan, Italy
- Ham radio operator since 2017 as IU2KWO
- Firmware developer by profession (and by passion)
- Co-founder and developer of OpenRTX
- Member of the M17 team since 2021

# What is M17

M17 is a community of open source developers and radio enthusiasts. We're building understandable systems in support of the hackers and experimenters' history of ham radio.

Main goals:

- an open source protocol for digital radio
- open source software
- open source hardware

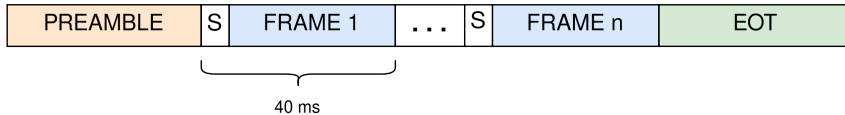
## Brief history

- 2019:
  - first experiments with Codec2, STM32 and RFM26W module
  - first prototype of TR-9, an M17-compatible handheld
- 2021:
  - first ARDC grant
  - ARRL's Technical Innovation Award
  - MMDVM and OpenWebRX start to support M17
- 2022:
  - Module17 modem board by Mathis DB9MAT
  - OpenRTX supports M17 on MD-(UV)380
- 2023:
  - First OpenHT prototype

## Technical specs

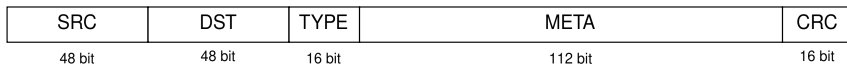
- 9600 bps 4FSK modulation (4800 symbols/second)
- 12.5kHz channel spacing
- packet and stream mode
- voice transport over stream, Codec2 at 3200 bps (voice only) or 1600 bps (voice + data)
- callsign-based ID
- 16 Channel Access Numbers
- optional AES encryption or XOR scrambling

# Physical layer



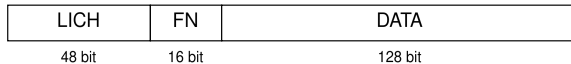
- Complete transmission:
  - preamble, at least 40ms of alternating +3/-3 symbols
  - one of more frames, of 40ms each
  - End of Transmission marker
- Frames:
  - 8 symbols of sync burst (frame specific)
  - 368 symbols of actual data

# Link Setup Frame



- First frame following the preamble for both packet and stream
- TYPE field:
  - stream/packet indicator
  - data type indicator
  - encryption type and subtype
  - Channel Access Number
- META field:
  - application specific content (ext. callsign, GNSS data, ...)
  - AES encryption nonce

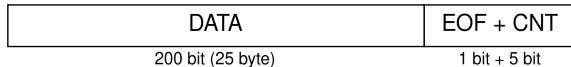
# Stream



- No limit to the number of bytes sent
- Link Information Channel (LICH):
  - 40-bit chunk of the LSF + 3-bit counter
  - allows to re-build the LSF in case of late entry
- Frame number:
  - free-running counter from 0 to 0x7FFF
  - most significant bit used as EoS marker



# Packet



- Up to 823 bytes of data can be sent in a single packet
- A transmission consists of:
  - LSF as first frame
  - 1 to 33 packet frames
- Net throughput between 3kbps and 4.7kbps

# IP networking

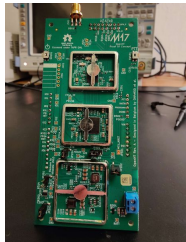
- Primarily designed to allow repeater/hotspot interconnection
- IP-capable devices can connect directly to the network
- Network nodes are called “reflectors”:
  - up to 26 “modules” per reflector
  - all the clients in the same module communicate together
  - is possible also to connect modules of different reflectors

# M17 hardware



## Module17:

- an M17 modem to be used with already existing radios
- currently only for voice, in the future also for data
- revision 1.0 almost ready



## OpenHT:

- SDR handheld transceiver: FPGA + STM32
- currently a prototype, < 25mW RF output

# OpenRTX

- An open-source firmware for ham radio devices
- Designed to be:
  - modular
  - easily portable to new devices
  - easily extendable to new protocols
- Currently supporting FM and M17 modes

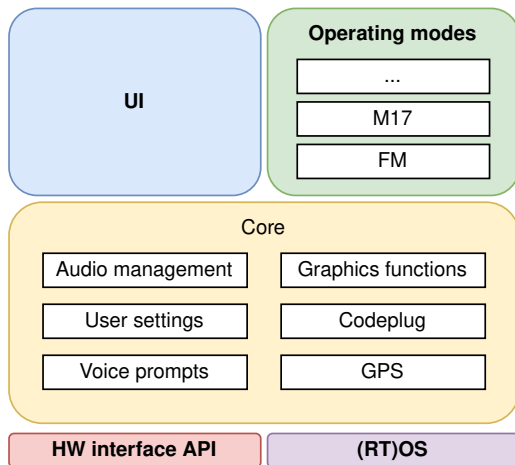
# Timeline

- *March 2020*: project starts as a port of OpenGD77 to the TYT MD-380
- *September 2020*: original idea abandoned, “official” beginning of OpenRTX
- *January 2021*: first alpha release with working FM on the TYT MD-380
- *February 2021*: first TX tests of M17 mode on the MD-380
- *April 2021*: support for GD-77, DM-1801 and MD-UV380
- *May 2022*: release v0.3.3 brings full support for M17 voice transmission
- *November 2022*: implemented voice prompts for vision impaired operators
- *October 2023*: support for Lilygo T-TWR Plus (and various technical improvements)
- *More to come ...*

## Supported devices

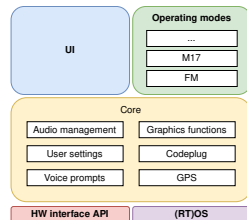
- TYT MD-380/Retevis RT3 (FM, M17)
- TYT MD-UV380/Retevis RT3s (FM, M17)
- Radioditty GD-77 (FM)
- Baofeng DM-1801 (FM)
- Module17 (M17)
- Lilygo T-TWR Plus (FM)

# Internals



# Internals

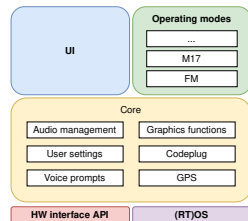
- Interface with the operating system:
  - thread management done using the Posix API
  - all the remaining parts use the standard C library
  - an RTOS is preferred on embedded devices
- Interface with the hardware:
  - APIs for display, keyboard, audio, radio and nonvolatile memory
  - “platform” API for device initialization and other stuff (LEDs, ...)
  - more devices can share a single API implementation (e.g. the display driver for MDx)





# Internals

- User interface:
  - currently a “standard” GUI + an ad-hoc GUI for Module17
  - you can write your own from scratch, if you want
  - future plans to make the standard GUI scriptable/expandable
- Operating modes/protocols:
  - C++ here, but simple
  - all the operating modes are subclasses of a generic “OpMode” class
  - pre-defined functions: enable, disable, periodic update (33Hz), squelch status
  - still some work to do: functions to get/set mode-specific data (e.g. configuration)



## M17 support

- First work done on the TYT MD-380, then extended to the MD-UV380
- Everything is handled in the MCU
- Hardware must have the following connections:
  - mic to MCU
  - RF stage to MCU, DC to  $\sim 3\text{kHz}$
  - MCU to speaker
  - MCU to RF stage, DC to  $\sim 3\text{kHz}$
- Current limitations:
  - you need to mod the radio hardware
  - the MCU has to be powerful enough
  - Codec2 uses floating point math

# Codeplug

- Trying to make something which:
  - is open and free
  - supports common ham radio needs (direct comm, repeaters, hotspots)
  - is portable across devices, both for end users and developers
- Currently WIP, an RFC open at <https://github.com/OpenRTX/openrtx.github.io/pull/32>
- Technical details:
  - binary format
  - up to 65'535 channels, contacts and banks (like DMR zones)
  - currently supporting FM, DMR and M17 operating modes
- May become a separate entity from the firmware



<https://openrtx.org>

<https://m17project.org>

[s.seva@openrtx.org](mailto:s.seva@openrtx.org)