DJ0MZ: QPMRpanCAT Web App V0.1

QuickStart

Contents

Description

This web-based CAT control and Panadapter interface connects directly to the Q900 or PMR-171 radio via USB or Bluetooth using the Web Serial and Web Audio APIs. It provides CAT (Computer Aided Transceiver) for frequency, mode, split, transmit and receive audio settings, together with live spectrum and spectrogram (waterfall) displays (spectrumscope, bandscope).

Access: dj0mz@darc.de

Operation controls are on the leftmost panel, general TX and RX settings are on the middle panel and Information is provided on the rightmost panel.

Usage

1. Open the application in Chrome or Chromium-based browser.
2. Connect your radio via USB cable for CAT and Panadapter, or Bluetooth serial port for CAT only.
3. Click ‘Open CAT’ to initialize the serial connection.
4. Click ‘Open Panadapter’ to initialise the I/Q connection.
5. Sometimes the sync operation mutes the RX audio; if this happens, click on NB, NR or change the volume.
6. There are several tuning methods:

7. Set the band and send the frequency.
8. Scan the band with the left and right arrowheads, with adjustable frequency steps.
9. Retrieve the contents of a saved memory.
10. With mouse over the spectrum (upper canvas of the Panadapter) press the SPACE bar and use the mouse wheel.
11. Click on a frequency in the spectrum.
12. Adjust volume, AGC, and noise reduction as needed.
13. Click ‘Write All’ to synchronize the rig with the current app setting
14. Use the A-B dropdown, ‘A=B’, Split, and the memory dropdown to manage VFO A and VFO B pairs.

Installation

Compatibility

This web app runs only on browsers which support WebAudio and WebSerial. Currently this means Chromium-based browsers such as Chromium, Chrome and Edge, but not Safari or Firefox.
The web app will not run as a standalone file but must be located in the HTML file space of a web server such as localhost or other web server.

Linux

1. Ensure you are running a recent version of Chrome or Chromium (version 112 or newer).
2. Connect your transceiver via USB; identify the device node (e.g. /dev/ttyUSB0).
3. Give your user access to the serial device:

```
sudo usermod -a -G dialout $USER
```

4. Log out and back in for permissions to take effect.
5. Launch the web app by opening the index.html file in Chrome.
6. Grant serial and microphone permissions when prompted.

Windows

1. The web app has not been tested on Windows, the following instructions are therefore only a general guideline.
2. Install Chrome or Edge (Chromium-based) if not already available.
3. Connect your transceiver via USB; confirm the COM port in Device Manager.
4. Launch the web app via index.html.
5. When the app requests serial access, select the correct COM port.
6. The app should now display 'Connected' and populate device settings.
7. Grant serial and microphone permissions when prompted.

Structure and Dataflow

The app consists of an HTML UI frame with embedded JavaScript. The system is divided into two cooperating JS scripts plus static HTML and CSS. The first script (CAT control) manages serial communication, CAT command packing, frequency caching, and synchronization with the rig. It also manages VFO A/B logic, memory storage, and all button events.

The second script (Panadapter) manages the audio input, FFT processing, waterfall and spectrum display, and the link between audio and CAT state. It can source data from the rig's IQ stream or from an attached audio device.

The dataflow is as follows:

- The user interacts with the toolbar and panels to change rig settings.
- These controls call central functions like setTargetFrequencyHz(), which updates the UI, caches values in memory, and sends the corresponding packed CAT frame through the Web Serial interface.
- Audio data enters via Web Audio, is transformed through FFT, and displayed in two canvases (spectrum and waterfall).
- The two panels remain synchronized so that changing the frequency in the CAT layer updates the spectral center frequency and vice versa.
- The VFO memory bank stores pairs of A/B frequencies persistently using localStorage.

The app uses asynchronous serial reads with CRC checking to ensure integrity, and schedules writes sequentially with small delays (50–100 ms) to prevent race conditions. Audio processing uses throttled FFT updates (typically 30 frames per second) to limit CPU load.

Dafydd Gibbon, DJ0MZ, 2025-10-19

- *Manual drafted by ChatGPT5 and postedited.*
- *Specification and UI: DJ0MZ.*
- *HTML and JS coding: DJ0MZ with ChatGPT5 for WebAudio and WebSerial and debugging.*