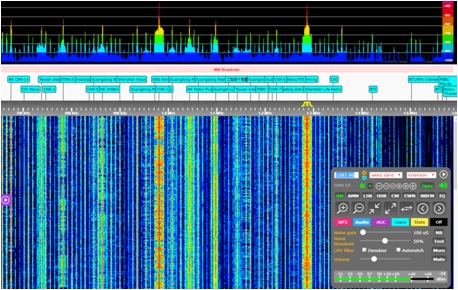
**Cover Page**

|  |
| --- |
| **Title:** Setting Up a WebSDR with OpenWebRX |
| **Subtitle:** A Comprehensive Report on Remote Radio Access and SDR Technology |
| **Prepared by:** Sanan Yagubov |
| **Date:** 07.07.2025 |
| **Organization:** Tubitak Bilgem |
| **Project Advisor:** |



**Introduction**

In recent years, the field of radio communications has undergone a significant transformation with the widespread adoption of **Software-Defined Radio (SDR)** technology. SDR represents a fundamental shift from traditional radio hardware by implementing signal processing and demodulation functions in software, thus offering greater flexibility, scalability, and accessibility. As a result, SDR has become increasingly popular not only among professional communication engineers but also within the amateur radio community.

Building upon SDR technology, **WebSDR** platforms introduce a new layer of convenience by enabling remote access to SDR receivers through standard web browsers. This means users can tune into a live radio signal from virtually anywhere in the world without needing specialized hardware or complex software installations. These platforms make it possible to share a single radio receiver among multiple users simultaneously, greatly enhancing the educational, experimental, and practical value of SDR systems.

One of the most powerful and actively developed open-source WebSDR platforms available today is **OpenWebRX**. Originally developed in 2015 by Hungarian student **András Retzler (HA7ILM)** as part of his bachelor’s thesis, OpenWebRX has since evolved into a community-driven project supported by radio enthusiasts and developers worldwide. It supports a wide range of digital and analog radio modes, making it suitable for diverse use cases including amateur radio operations, shortwave listening, weather monitoring, and signal analysis.

The primary goal of this report is to provide a comprehensive overview of OpenWebRX, including its history, key features, technical setup, and real-world applications. We will explore why one might choose to set up a WebSDR, what hardware and software components are required, and how OpenWebRX can be customized and deployed in different scenarios. The report also highlights the benefits of using OpenWebRX both individually and as a shared resource within the global radio community.

By studying the structure, configuration, and usage of OpenWebRX, readers will gain insight into how modern SDR technology can be integrated with web-based platforms to create accessible and powerful radio communication systems. Whether for educational purposes, research, personal exploration, or public service, OpenWebRX stands as a testament to the potential of open-source innovation in the world of wireless communication.

**What is a WebSDR?**

A **WebSDR** is a type of software-defined radio (SDR) system that can be accessed and controlled remotely through a web browser. The term combines “Web” for internet accessibility and “SDR” for software-defined radio, resulting in a platform that allows users to listen to radio signals from anywhere in the world via a simple web interface.

At its core, a WebSDR system consists of three main components:

1. **An SDR Receiver** – a hardware device capable of capturing radio signals across a wide frequency spectrum.
2. **A Host Computer** – typically running Linux, responsible for processing and demodulating the signals in real time.
3. **Web Server Software** – software like OpenWebRX that streams the processed signals to users via a web interface.

Unlike conventional radios that rely on analog components to tune and demodulate signals, SDR technology performs most of these functions in software. This enables WebSDR platforms to support multiple users simultaneously, each tuning to different frequencies and modes on the same device.

WebSDRs have gained popularity for several reasons:

* **Remote Accessibility**: Users can listen to shortwave, amateur, and other radio bands from any location.
* **Collaborative Use**: A single WebSDR installation can be shared with many users without requiring physical access.
* **Educational Value**: Students, researchers, and radio enthusiasts can explore the radio spectrum and analyze signals live.
* **Low Power Consumption**: Compared to full transceivers, WebSDRs often consume less electricity, making them efficient for continuous operation.

Furthermore, a WebSDR can be configured as either a **private** system, restricted to certain users, or a **public** system listed in directories such as [receiverbook.de](https://receiverbook.de" \t "_new), where thousands of global users can access it.

Overall, a WebSDR is a powerful tool that combines the flexibility of SDR with the convenience of web-based interfaces, providing a practical solution for radio listening, experimentation, and education.

**Why Set Up a WebSDR?**

Setting up a **WebSDR** system offers numerous practical and strategic advantages for both amateur and professional users of radio technology. With the increasing availability of SDR hardware and open-source software platforms like OpenWebRX, it has become easier than ever to deploy a personal or shared WebSDR station. The motivations for setting up such a system can be grouped into several key areas:

**1. Remote Access to Your Station**

One of the primary reasons for setting up a WebSDR is to **access your radio equipment remotely**. If you are away from your physical radio station—whether traveling, at work, or living in an apartment without antenna options—a WebSDR allows you to continue monitoring the airwaves through a web browser from any device connected to the internet.

**2. Productive Use of Antennas**

When not actively transmitting, antennas can remain idle. A WebSDR puts them to productive use by turning your receiving setup into a **continuously available monitoring system**. This is especially useful for long-term propagation studies, scanning for emergency frequencies, or recording interesting transmissions.

**3. Lower Power Consumption**

Compared to traditional transceivers, SDR-based setups typically consume **significantly less power**. This makes them suitable for 24/7 operation, even when hosted on a Raspberry Pi or a low-power Linux PC. This efficiency is particularly advantageous in remote installations or off-grid setups powered by solar energy.

**4. Shared Access with the Community**

A WebSDR can be configured to allow public access, giving others the ability to use your receiver. This promotes **collaboration**, allows international users to explore your local RF environment, and helps foster the global amateur radio community. Sharing a receiver also makes rare or distant signals accessible to a broader audience.

**5. Educational and Research Opportunities**

Universities, schools, and research institutions use WebSDRs to **teach radio theory**, **signal processing**, and **wireless communication**. It allows real-time demonstrations of modulation types, propagation effects, digital modes, and more—all without needing dedicated hardware for each student.

**6. Monitoring and Surveillance**

WebSDRs can be employed for **monitoring specific frequency bands** for legal or experimental purposes. Emergency services, maritime operators, and aviation enthusiasts can all benefit from a WebSDR that’s focused on relevant communication bands.

Whether you are a radio hobbyist seeking remote access or a researcher needing a stable signal monitoring tool, setting up a WebSDR provides a flexible and scalable solution that fits a wide variety of use cases.

**History of OpenWebRX and Its Development?**

**OpenWebRX** is a prominent open-source WebSDR platform that emerged from an academic project and evolved into a widely adopted tool within the global radio community. Its origin story is a compelling example of how student-led innovation can grow into a sustainable, community-supported technology.

**The Beginning**

OpenWebRX was initially developed in **2015** by **András Retzler**, a Hungarian amateur radio operator with the callsign **HA7ILM**. At the time, András was a student at Budapest University of Technology and Economics, and he created OpenWebRX as part of his **Bachelor of Science thesis**. His aim was to build a WebSDR system that could work efficiently with low-cost SDR hardware, particularly the RTL-SDR USB dongle, and be accessible via any standard web browser.

The early version of OpenWebRX already included key features like real-time spectrum display, audio streaming, and basic control over tuning and mode selection. Unlike other WebSDR projects that were limited in accessibility or hardware support, OpenWebRX emphasized **user-friendliness**, **multi-user capability**, and **broad compatibility** with SDR devices.

**Community Involvement**

After its initial release, the project gained traction among radio amateurs, hobbyists, and developers. OpenWebRX was hosted on GitHub, allowing others to contribute improvements, report issues, and help with development. Over the years, various contributors expanded the platform’s capabilities, making it more powerful and flexible.

Today, OpenWebRX continues to be actively developed by volunteers and contributors across the globe. It has received significant updates, bug fixes, and performance enhancements, often driven by feedback from real-world users. Its open-source nature ensures that the software remains freely available, customizable, and transparent.

**Key Milestones**

* **2015**: Initial version developed and released by András Retzler.
* **2016–2018**: Introduction of support for more SDR hardware such as AirSpy and SDRplay.
* **2019**: The community began integrating support for digital modes like FT8, JT65, and PSK31.
* **2020–2023**: Enhanced digital signal processing, decoding of additional modes (e.g., FreeDV, CW decoder, DMR), and improved web interfaces.
* **OpenWebRX+**: A fork or enhanced version of the original OpenWebRX, offering additional features like SSTV, FAX, D-Star, and more, while maintaining compatibility with most modern browsers and SDRs.

**Modern Use**

Today, OpenWebRX is not only used by individuals but also by educational institutions, DX monitoring networks, emergency communication groups, and hobbyist collectives. It represents a blend of simplicity and power, allowing both novices and experts to deploy high-functioning WebSDR solutions with minimal setup.

**Technical Requirements to Set Up OpenWebRX**

Setting up a fully functional WebSDR using **OpenWebRX** requires a combination of **hardware components**, **software tools**, and a **network connection**. Thanks to its flexible design, OpenWebRX can run on a variety of setups ranging from low-power single-board computers to full desktop systems.

Below is an overview of the key technical requirements needed for installation and operation:

**1. SDR Hardware (Software-Defined Radio Receiver)**

OpenWebRX is compatible with several SDR devices. Some of the most commonly used and supported receivers include:

* **RTL-SDR**: Affordable and popular USB dongle, ideal for beginners.
* **AirSpy**: Offers higher dynamic range and better sensitivity than RTL-SDR.
* **SDRplay**: Powerful SDRs capable of receiving large bandwidths (up to 10 MHz).
* **KiwiSDR**: Uses OpenWebRX-based software and supports wideband reception over Ethernet.

**2. Host Computer**

You need a computer to run OpenWebRX and interface with the SDR device. Supported systems include:

* **Raspberry Pi 3 or 4**: Excellent low-power choice for RTL-SDR-based setups.
* **PC or Laptop**: For higher performance and broader compatibility.
* **Operating System**:
  + Linux (Ubuntu/Debian recommended)
  + Docker (for containerized deployment)
  + VirtualBox (if using a virtual machine)

Note: Windows is not officially supported as a host OS, but Linux can be run inside a VM or Docker container on Windows systems.

**3. Internet or Network Access**

An internet connection is required for remote access. If you are only accessing the system locally, a home or office network will suffice. For public-facing WebSDRs:

* Ensure **port forwarding** is configured (default port: 8073 or 8074).
* Use a **static IP** or **dynamic DNS service** to maintain consistent access.

**4. Software and Dependencies**

OpenWebRX requires the following software and libraries:

* **Python 3**
* **GNU Radio or SoapySDR (optional, depending on hardware)**
* **ffmpeg** (for audio processing)
* **OpenWebRX source code** (available from [https://openwebrx.de](https://openwebrx.de" \t "_new) or GitHub)

The software is typically installed using terminal commands, and basic Linux command-line experience is recommended for the setup process.

**5. Web Browser (for Access)**

Clients (users) can access the OpenWebRX interface using any modern web browser:

* Chrome, Firefox, Edge, Safari (desktop or mobile)
* No plugins or extensions required

With these components in place, users can set up a powerful and flexible WebSDR system capable of streaming real-time radio content over the internet. The simplicity of the setup, combined with the low cost of SDR hardware, makes OpenWebRX accessible for both beginners and advanced radio enthusiasts.

**Supported Features and Radio Modes in OpenWebRX**

**1. Multi-User Support**

OpenWebRX is designed to handle **multiple users simultaneously**. Each user can independently tune to different frequencies and select different modes without interfering with one another (depending on the SDR hardware used). This is ideal for public WebSDRs shared with the global community.

**2. Broad Hardware Compatibility**

It supports many popular SDR devices, including:

* **RTL-SDR** (affordable and widely used)
* **AirSpy** (higher performance)
* **SDRplay** (supports up to 10 MHz bandwidth)
* **KiwiSDR** (WebSDR with integrated networking)

This broad compatibility makes OpenWebRX suitable for a wide range of users with varying performance needs and budgets.

**3. Web-Based User Interface**

OpenWebRX features a responsive and intuitive **web interface** with the following capabilities:

* Real-time **waterfall** and **spectrum displays**
* Frequency tuning via mouse, touchscreen, or manual input
* Selection of modulation modes
* Audio playback in the browser (no plugins required)
* Signal strength display and waterfall color customization

**4. Supported Modulation Modes**

OpenWebRX supports a variety of **analog** and **digital** radio modes, including:

**Analog Modes:**

* **AM (Amplitude Modulation)**
* **FM (Frequency Modulation)**
* **SSB (Single Side Band – USB/LSB)**
* **CW (Continuous Wave, i.e., Morse Code)**

**Digital Modes (Standard):**

* **FreeDV** – Low bitrate digital voice over HF
* **FT8, JT65** – Weak signal digital communication modes
* **PSK31** – Phase Shift Keying used in keyboard-to-keyboard chat
* **Packet** – AX.25 packet radio

**5. OpenWebRX+ Features (Extended Functionality)**

Some deployments use **OpenWebRX+**, a community fork or enhanced version of the software that includes additional decoding capabilities:

* **CW Decoder** – Automatic Morse code decoding
* **RTTY** – Radio teletype
* **DMR, D-Star, YSF** – Digital voice modes for VHF/UHF
* **SSTV** – Slow-scan television for image transmission
* **FAX** – Weather fax decoding

These advanced features make OpenWebRX+ especially appealing to experienced operators and experimenters.

**6. Frequency Range and Band Segments**

The frequency range depends on the SDR hardware. For example, SDRplay devices can capture up to **10 MHz of bandwidth**, enabling coverage of:

* Amateur radio bands (80m to 33cm)
* Shortwave broadcasting
* VHF/UHF public services
* Weather and maritime bands

Overall, the feature set of OpenWebRX makes it a **versatile and powerful platform** for monitoring, decoding, and analyzing radio signals across the RF spectrum.

**Real-World Example: KP4MD WebSDR Station**

To better understand the practical application of OpenWebRX, we can examine a real-world example: the **KP4MD WebSDR Station**, operated by radio amateur **Carol Milazzo (callsign: KP4MD)**. This station demonstrates how OpenWebRX can be implemented in a functional, public-facing environment to provide wide frequency coverage and remote accessibility.

**Station Overview**

* **Location**: Citrus Heights, California, United States
* **Operator**: Carol Milazzo – KP4MD
* **Web Access URL**: <http://sdr.n6na.org:8074>

This WebSDR is publicly accessible and provides live streaming of radio signals through a user-friendly web interface. It allows multiple users to listen to various frequencies without needing any additional software or equipment.

**Antenna Setup**

The KP4MD station utilizes a combination of antennas optimized for different frequency ranges:

1. **40m Full Wave Horizontal Loop**
   * Positioned approximately **6 meters above ground**
   * Covers **0.5 MHz to 54 MHz** effectively
   * Ideal for HF bands (shortwave and amateur radio)
2. **VHF-UHF Bowtie Antennas**
   * Designed to cover frequencies from **88 MHz to 910 MHz**
   * Used for monitoring FM broadcast, weather stations, airband, and other VHF/UHF services

These antennas ensure broad coverage from the lower HF spectrum up to high UHF frequencies, making the station suitable for a wide variety of monitoring purposes.

**Hardware and Capabilities**

* **SDR Device**: SDRplay receiver
  + Can receive **up to 10 MHz bandwidth** at a time
  + Allows dynamic tuning within that range
* **Web Interface**: OpenWebRX
  + Multiple users can connect and tune independently (within the selected band)
  + Intuitive controls for frequency, mode, and audio
* **Software Features**:
  + Real-time waterfall and spectrum display
  + Support for various modes including AM, FM, SSB, CW, FT8, and more

**Band Selection and Tuning Etiquette**

The KP4MD WebSDR provides access to:

* **15 amateur radio bands** from **80 meters to 33 centimeters**
* **General coverage segments** from **0.5 MHz to 910 MHz**

However, due to hardware limitations (specifically SDRplay), the receiver can only tune to **one band at a time**. Therefore, users are expected to follow basic **WebSDR etiquette**:

* **Do not change the band** while someone else is listening, as it interrupts their session.
* Use the **chat box or status display** (if available) to coordinate with other listeners.

**Educational and Public Value**

Stations like KP4MD serve not only the radio hobbyist community but also provide educational value for students and researchers. By listening to real-time signals, users can observe propagation effects, learn about modulation, and decode digital transmissions from around the world.

This example illustrates how OpenWebRX can be deployed effectively using relatively simple hardware to create a robust, shared radio resource that benefits users globally.

**Etiquette and Best Practices for WebSDR Users**

While **OpenWebRX** is designed to support multiple users simultaneously, certain hardware limitations—particularly in SDR devices like SDRplay—mean that not all operations can be done independently by every user. To ensure a smooth and respectful experience for everyone using a shared WebSDR, it’s important to follow a set of established **etiquette guidelines and best practices**.

**1. Respect Other Users**

If the SDR hardware can only monitor **one band at a time** (as is the case with SDRplay), any user who changes the band setting will disrupt the listening experience for everyone else. Therefore:

* **Do not switch frequency bands** if other users are connected.
* If unsure whether others are online, check the interface for indicators or chat messages.
* Some setups display a user count or activity log—use this to avoid interfering.

**2. Coordinate Using Chat (if available)**

Many public WebSDR interfaces include a **chat box** or messaging window. This can be used to:

* Announce your intent to change a band
* Request permission before making changes
* Discuss signal conditions or interesting finds with other users

Using the chat promotes a sense of **community and cooperation**.

**3. Use the Interface Responsibly**

OpenWebRX provides powerful tools for tuning, filtering, and decoding signals. To maintain system performance:

* Avoid excessive or rapid changes in frequency and mode
* Do not overload the system by opening too many browser tabs or sessions
* Refrain from playing loud audio or disturbing signals if the platform allows sharing

**4. Share the Resource**

If you’re using a **public WebSDR**, remember that it is a shared resource maintained voluntarily. Be mindful of others who may also wish to:

* Listen to rare DX signals
* Analyze digital modes
* Conduct propagation experiments

You can also **limit your session time** during peak activity periods to give others a chance.

**5. Acknowledge the Host**

Public WebSDRs are typically set up and maintained by individuals or institutions. It's a good practice to:

* Read and follow any posted rules or usage notes
* Avoid behavior that could lead to IP blocking or restrictions
* Consider donating or supporting the project if you find it valuable

**6. Educational Courtesy**

If you’re using a WebSDR for educational purposes, encourage **students or new users** to observe etiquette as well. Teach them how to:

* Tune carefully
* Respect others' listening
* Report technical issues kindly (if contact info is available)

**Summary**

Using WebSDRs like OpenWebRX is a privilege made possible by dedicated operators and open-source developers. Observing proper etiquette ensures that the system remains **functional, respectful, and enjoyable** for all users.

**Conclusion**

The rapid advancement of software-defined radio (SDR) technology has revolutionized the way individuals and institutions interact with the radio frequency spectrum. Among the most accessible and innovative developments in this space is **OpenWebRX**, an open-source platform that allows SDR receivers to be operated remotely via a web browser.

Throughout this report, we have explored the **foundations, advantages, technical requirements, features**, and **real-world applications** of OpenWebRX. We have seen how it provides a **flexible, low-cost, and user-friendly solution** for monitoring radio signals, supporting multiple users simultaneously, and operating on a wide range of SDR hardware. From casual listeners and amateur radio operators to educators and researchers, OpenWebRX serves a broad and diverse community.

Setting up a WebSDR using OpenWebRX offers many benefits:

* It enables **remote access** to radio equipment.
* It encourages **collaboration and knowledge sharing**.
* It supports **real-time experimentation and learning**.
* It operates with **minimal power consumption and hardware requirements**.

Additionally, public deployments such as the **KP4MD WebSDR station** highlight how OpenWebRX can be used to create global, always-available listening posts for various radio bands. These stations not only serve the amateur radio community but also provide educational and scientific value.

During the course of this study, it became evident how important proper **user etiquette** and **best practices** are in maintaining a stable and cooperative WebSDR environment. Respectful usage ensures that such community resources remain sustainable and beneficial for everyone.

In conclusion, OpenWebRX is more than just a technical tool—it is a gateway to exploration, learning, and connection within the radio spectrum. Whether deployed privately or shared publicly, it represents the future of how we listen to and engage with the invisible world of radio waves.