Open source, multi-user SDR receiver software with a web interface   http://openwebrx.org

sdr # rtl-sdr # waterfall # html5 # software-defined-radio

ⓘ **207** commits	⎇ **8** branches	◌ **6** releases	👥 **3** contributors	⚖ AGPL-3.0

master ▾	New pull request			Find file	Clone or download ▾

ha7ilm [2] Fixed bug related to disabled audio autoplay starting from Chrome 66		💬 1 Latest commit 4e30fd5 May 7, 2018

📁 htdocs	[2] Fixed bug related to disabled audio autoplay starting from Chrome 66	May 7, 2018
📄 .gitignore	Added nanoscroller, retabbed index.wrx and did some work on the digid…	May 3, 2017
📄 CONTRIBUTING.md	Update CONTRIBUTING.md	Feb 14, 2016
📄 CONTRIBUTORS	Added CONTRIBUTORS file.	Feb 14, 2016
📄 ICLA.txt	Update ICLA.txt	Feb 14, 2016
📄 LICENSE.txt	Create LICENSE.txt	Nov 29, 2015
📄 README.md	Changed website URL	May 6, 2018
📄 config_webrx.py	Merged feature/digitalmods	Jul 12, 2017
📄 csdr.py	Few fixes	Jun 28, 2017
📄 openwebrx.py	This version of merged gl3 and digitalmods actually works okay	May 30, 2017
📄 rxws.py	Retabbed anything else in python	May 2, 2017
📄 sdrhu.py	sdr.hu is now HTTPS	May 6, 2018

📖 **README.md**

OpenWebRX

Setup guide for Ubuntu | Knowledge base on the Wiki | Receivers on SDR.hu

OpenWebRX is a multi-user SDR receiver software with a web interface.

It has the following features:

- csdr based demodulators (AM/FM/SSB/CW/BPSK31),
- filter passband can be set from GUI,
- waterfall display can be shifted back in time,
- it extensively uses HTML5 features like WebSocket, Web Audio API, and <canvas>,
- it works in Google Chrome, Chromium (above version 37) and Mozilla Firefox (above version 28),
- currently supports RTL-SDR, HackRF, SDRplay, AirSpy and many other devices, see the OpenWebRX Wiki,
- it has a 3D waterfall display:

News (2015-08-18)

- My BSc. thesis written on OpenWebRX is available here.
- Several bugs were fixed to improve reliability and stability.
- OpenWebRX now supports compression of audio and waterfall stream, so the required network uplink bandwidth has been decreased from 2 Mbit/s to about 200 kbit/s per client! (Measured with the default settings. It is also dependent on `fft_size` .)
- OpenWebRX now uses sdr.js (*libcsdr* compiled to JavaScript) for some client-side DSP tasks.
- Receivers can now be listed on SDR.hu.
- License for OpenWebRX is now Affero GPL v3.

News (2016-02-14)

- The DDC in *csdr* has been manually optimized for ARM NEON, so it runs around 3 times faster on the Raspberry Pi 2 than before.
- Also we use *ncat* instead of *rtl_mus*, and it is 3 times faster in some cases.
- OpenWebRX now supports URLs like: `http://localhost:8073/#freq=145555000,mod=usb`
- UI improvements were made, thanks to John Seamons and Gnoxter.

News (2017-04-04)

- *ncat* has been replaced with a custom implementation called *nmux* due to a bug that caused regular crashes on some machines. The *nmux* tool is part of the *csdr* package.
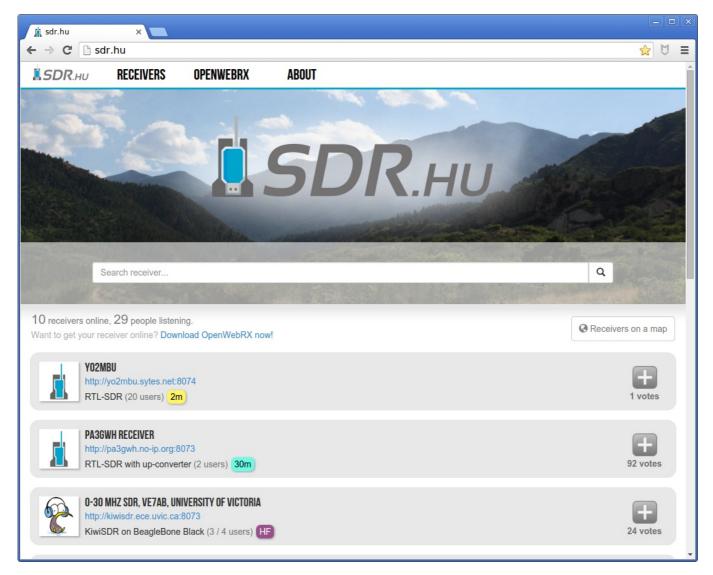- Most consumer SDR devices are supported via rx_tools, see the OpenWebRX Wiki on that.

News (2017-07-12)

- OpenWebRX now has a BPSK31 demodulator and a 3D waterfall display.

> When upgrading OpenWebRX, please make sure that you also upgrade *csdr*!

OpenWebRX servers on SDR.hu

SDR.hu is a site which lists the active, public OpenWebRX servers. Your receiver can also be part of it, if you want.

Setup

OpenWebRX currently requires Linux and python 2.7 to run.

First you will need to install the dependencies:

- libcsdr
- rtl-sdr

After cloning this repository and connecting an RTL-SDR dongle to your computer, you can run the server:

```
python openwebrx.py
```

You can now open the GUI at http://localhost:8073.

Please note that the server is also listening on the following ports (on localhost only):

- port 4951 for the multi-user I/Q server.

Now the next step is to customize the parameters of your server in `config_webrx.py`.

Actually, if you do something cool with OpenWebRX, please drop me a mail:
Andras Retzler, HA7ILM <randras@sdr.hu>

Usage tips

You can zoom the waterfall display by the mouse wheel. You can also drag the waterfall to pan across it.

The filter envelope can be dragged at its ends and moved around to set the passband.

However, if you hold down the shift key, you can drag the center line (BFO) or the whole passband (PBS).

Setup tips

If you have any problems installing OpenWebRX, you should check out the Wiki about it, which has a page on the common problems and their solutions.

Sometimes the actual error message is not at the end of the terminal output, you may have to look at the whole output to find it.

If you want to run OpenWebRX on a remote server instead of *localhost*, do not forget to set *server_hostname* in `config_webrx.py`.

Licensing

OpenWebRX is available under Affero GPL v3 license (summary).

OpenWebRX is also available under a commercial license on request. Please contact me at the address *<randras@sdr.hu>* for licensing options.