

Practical SDR With OpenWebRx

Ben Matthews

**RMHAM-U
April 2022**

SDR?

- Wideband (typically) receiver (and/or transmitter) attached to a computer
- Do the bare minimum in hardware, everything else is software (math!)
- Math is hard though? I just want to listen to the radio
 - Cool, you've come to the right talk
 - If you want the details, see my Nerdfest talk from earlier this year

Legal Aside

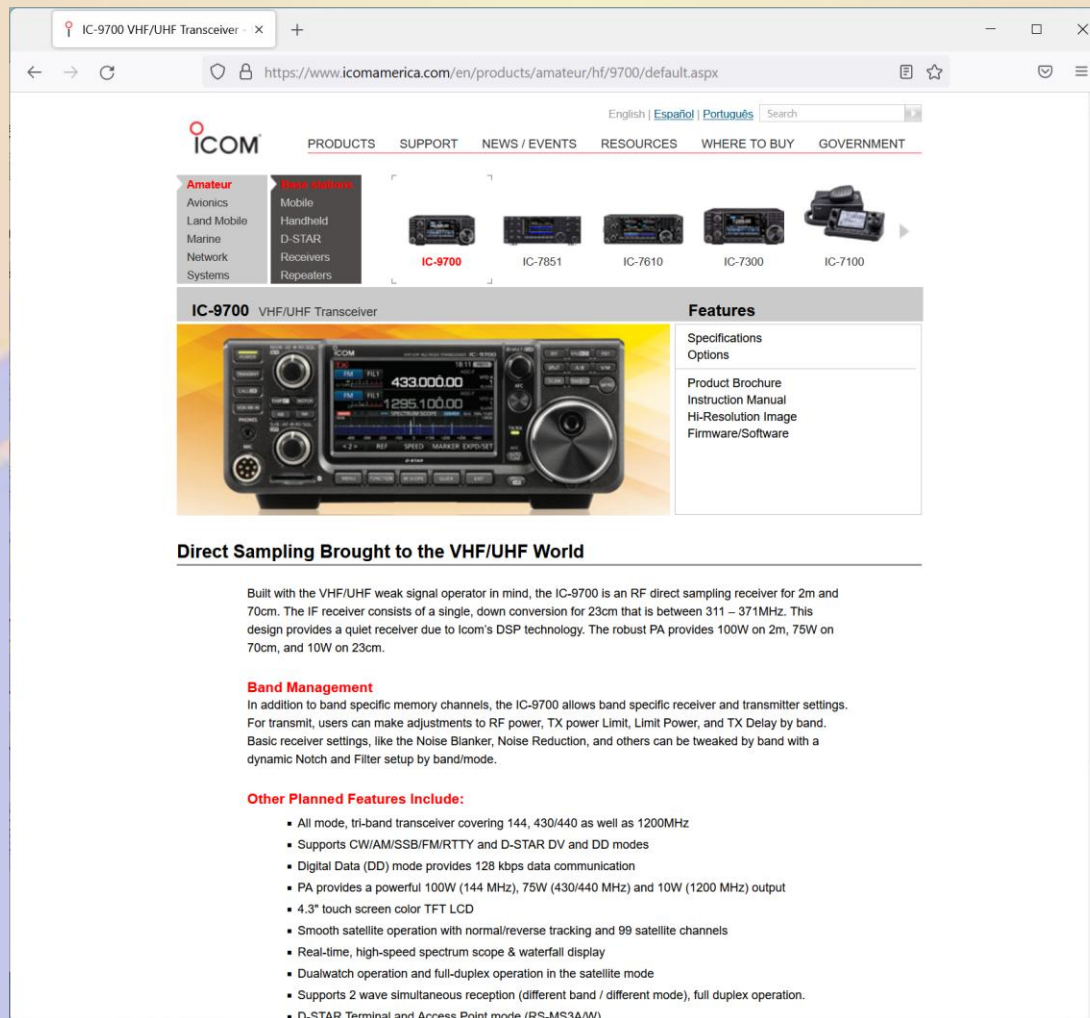
- SDRs are super flexible
- This is great, but it also provides the flexibility to do things the government might prefer you don't
- Be careful and stay inside the limitations of your license(s)
- Please keep Patent/IP law in mind. This talk may show software which could be encumbered in some jurisdictions. No legal advice is provided
 - For educational purposes, of course

So you bought an SDR, now what?

- Hardware?
- Software?
 - Windows?
 - Linux?
 - Web?
- Transmit?

Hardware

- Easy mode: SDR with the computer bundled



The screenshot shows the ICOM website's product page for the IC-9700 VHF/UHF Transceiver. The browser address bar shows the URL: <https://www.icomamerica.com/en/products/amateur/hf/9700/default.aspx>. The page features a navigation menu with links to PRODUCTS, SUPPORT, NEWS / EVENTS, RESOURCES, WHERE TO BUY, and GOVERNMENT. A sidebar on the left lists various product categories under the 'Amateur' heading. The main content area displays a row of product images for the IC-9700, IC-7851, IC-7610, IC-7300, and IC-7100. Below this, a large image of the IC-9700 transceiver is shown, along with a 'Features' section listing links to Specifications, Options, Product Brochure, Instruction Manual, Hi-Resolution Image, and Firmware/Software. The page also includes a section titled 'Direct Sampling Brought to the VHF/UHF World' and a 'Band Management' section. At the bottom, a list of 'Other Planned Features Include:' is provided.

ICOM PRODUCTS SUPPORT NEWS / EVENTS RESOURCES WHERE TO BUY GOVERNMENT

English | Español | Português Search

Amateur
Avionics
Land Mobile
Marine
Network
Systems

Next stations
Mobile
Handheld
D-STAR
Receivers
Repeaters

IC-9700 IC-7851 IC-7610 IC-7300 IC-7100

IC-9700 VHF/UHF Transceiver

Features

- Specifications
- Options
- Product Brochure
- Instruction Manual
- Hi-Resolution Image
- Firmware/Software

Direct Sampling Brought to the VHF/UHF World

Built with the VHF/UHF weak signal operator in mind, the IC-9700 is an RF direct sampling receiver for 2m and 70cm. The IF receiver consists of a single, down conversion for 23cm that is between 311 – 371MHz. This design provides a quiet receiver due to Icom's DSP technology. The robust PA provides 100W on 2m, 75W on 70cm, and 10W on 23cm.

Band Management

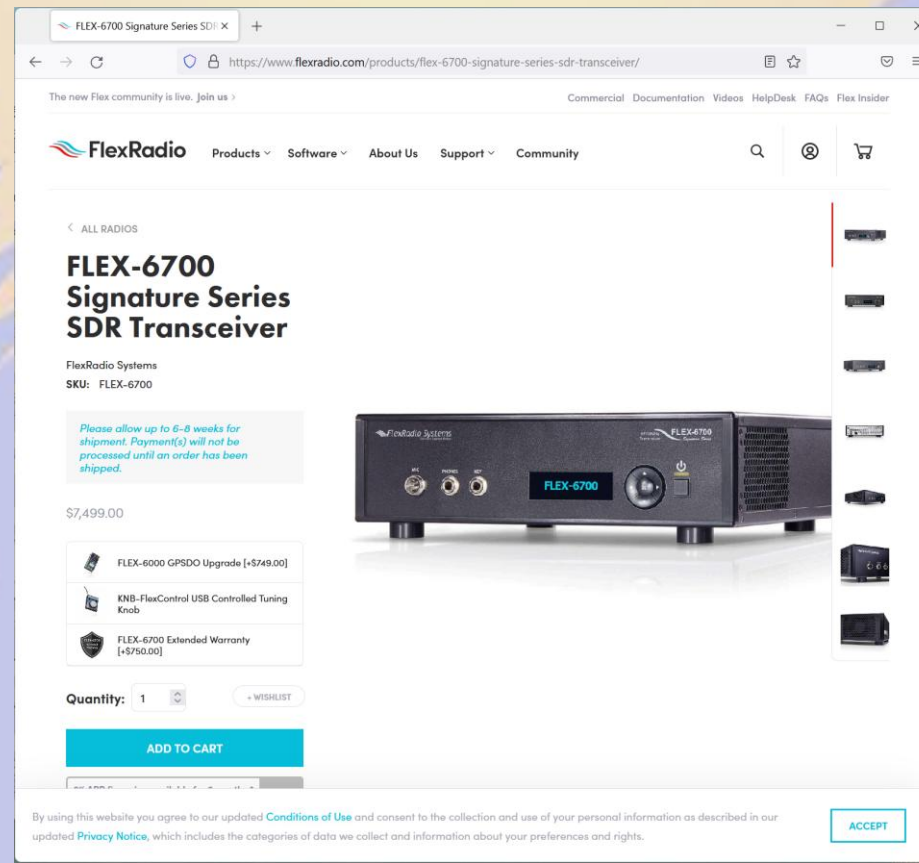
In addition to band specific memory channels, the IC-9700 allows band specific receiver and transmitter settings. For transmit, users can make adjustments to RF power, TX power Limit, Limit Power, and TX Delay by band. Basic receiver settings, like the Noise Blanker, Noise Reduction, and others can be tweaked by band with a dynamic Notch and Filter setup by band/mode.

Other Planned Features Include:

- All mode, tri-band transceiver covering 144, 430/440 as well as 1200MHz
- Supports CW/AM/SSB/FM/RTTY and D-STAR DV and DD modes
- Digital Data (DD) mode provides 128 kbps data communication
- PA provides a powerful 100W (144 MHz), 75W (430/440 MHz) and 10W (1200 MHz) output
- 4.3" touch screen color TFT LCD
- Smooth satellite operation with normal/reverse tracking and 99 satellite channels
- Real-time, high-speed spectrum scope & waterfall display
- Dualwatch operation and full-duplex operation in the satellite mode
- Supports 2 wave simultaneous reception (different band / different mode), full duplex operation.
- D-STAR Terminal and Access Point mode (RS-MS3A/W)

Hardware

- Slightly harder mode:
- Bundled software that looks like a real radio, filters, PA, etc (but you still provide your own computer)



Hardware

- Hard mode: Figure it out yourself
- RTLSDR
- HackRF
- BladeRF
- SDRPlay
- KrakenSDR
- ... Tons to choose from



Hardware

- Antenna system as you would any other radio
- External filters can be a good idea
- Good quality cables (Computers/USB can be RF noisy)
- Faster computers are better, but it really depends on how much bandwidth you need

So.. Where to start?

- OpenWebRX and the ham's favorite computer, the Raspberry pi
- Linux is involved, but we'll present a web interface so you don't really need to know Linux
- Lots of decoders for various modes
- Let's try to replace a whole shack full of receivers
- Receive only for now. Some SDRs can transmit, but there's a bit more to that

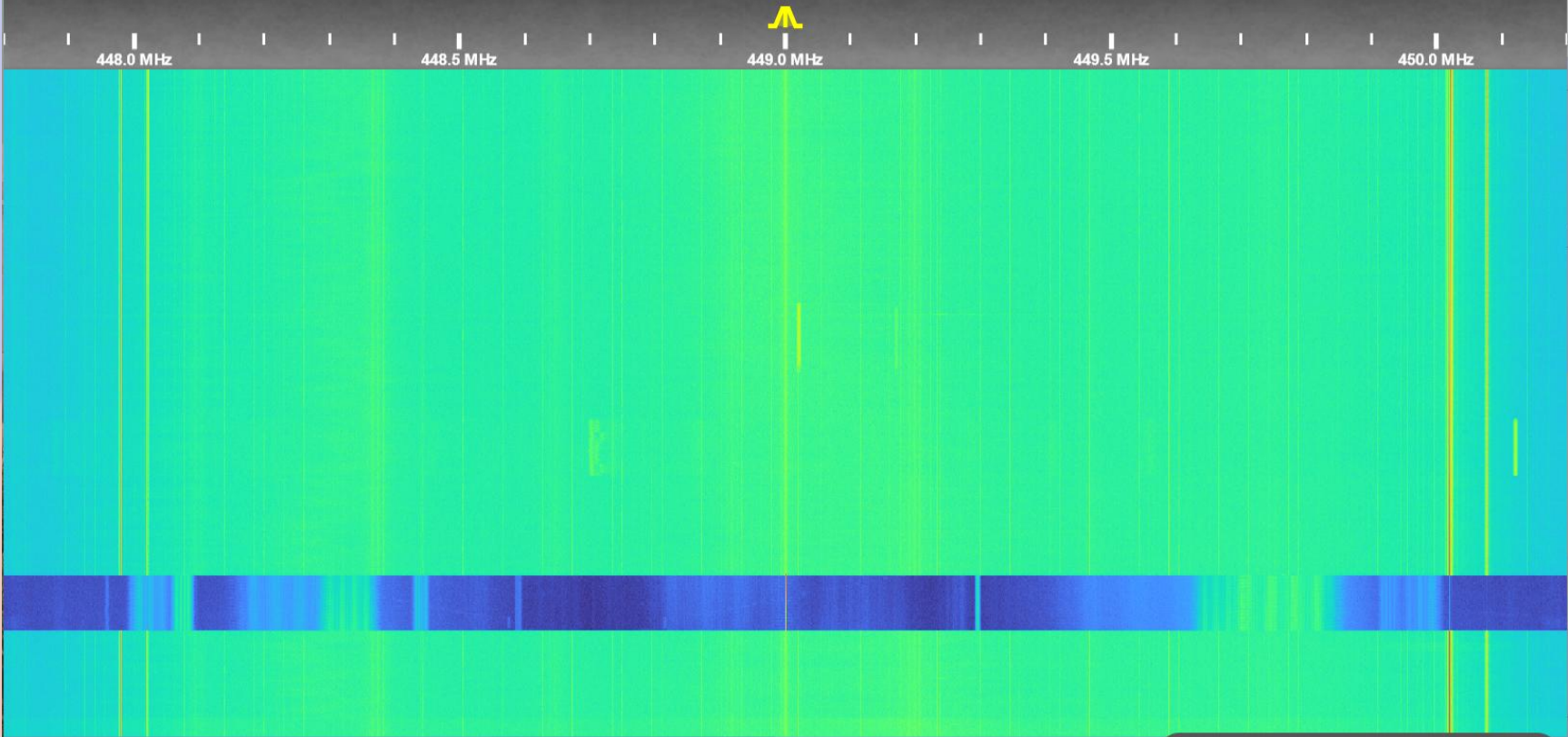
OpenWebRX



NOSZ

Westcreek, Colorado | Loc: DM79ma, ASL: 2700 m

- Status
- Log
- Receiver
- Map
- Settings



Under construction
We're working on the code right now, so the application might fail.

Audio buffer [0.2 s] Audio output [47.9 kbps] Audio stream [192 kbps]
Network usage [1355.1 kbps] Server CPU [27%] Clients [1]

449.0000 MHz

448.9294 MHz

RTL-SDR USB Stick 70cm Repeaters▼

FM WFM AM LSB USB

CW M17 FreeDV DRM

DIG ▼

SQ

-45.4 dB

Mode Selection

The screenshot displays the OpenWebRX web interface in a browser window. The address bar shows the URL `10.30.115.147/#freq=449000000,mod=nfm,sql=-150`. The interface includes a header with the OpenWebRX logo, a location indicator for "Westcreek, Colorado | Loc: DM79ma, ASL: 2700 m", and navigation links for Status, Log, Receiver, Map, and Settings. The main area features a spectrum plot with a frequency range from 448.0 MHz to 450.0 MHz. A red circle highlights the mode selection menu in the bottom right corner, which is currently set to "FM". Other modes visible include WFM, AM, LSB, USB, CW, M17, FreeDV, and DRM. A red banner at the bottom left states "Under construction" with the message "We're working on the code right now, so the application might fail." The bottom status bar shows system metrics: Audio buffer [0.2 s], Audio output [47.9 kbps], Audio stream [192 kbps], Network usage [1355.1 kbps], Server CPU [27%], and Clients [1].

OpenWebRX | Open Source SDR
PLAYING

10.30.115.147/#freq=449000000,mod=nfm,sql=-150

OpenWebRX NOSZ
Westcreek, Colorado | Loc: DM79ma, ASL: 2700 m

Status Log Receiver Map Settings

448.0 MHz 448.5 MHz 449.0 MHz 449.5 MHz 450.0 MHz

449.0000 MHz
448.9294 MHz

RTL-SDR USB Stick 70cm Repeaters

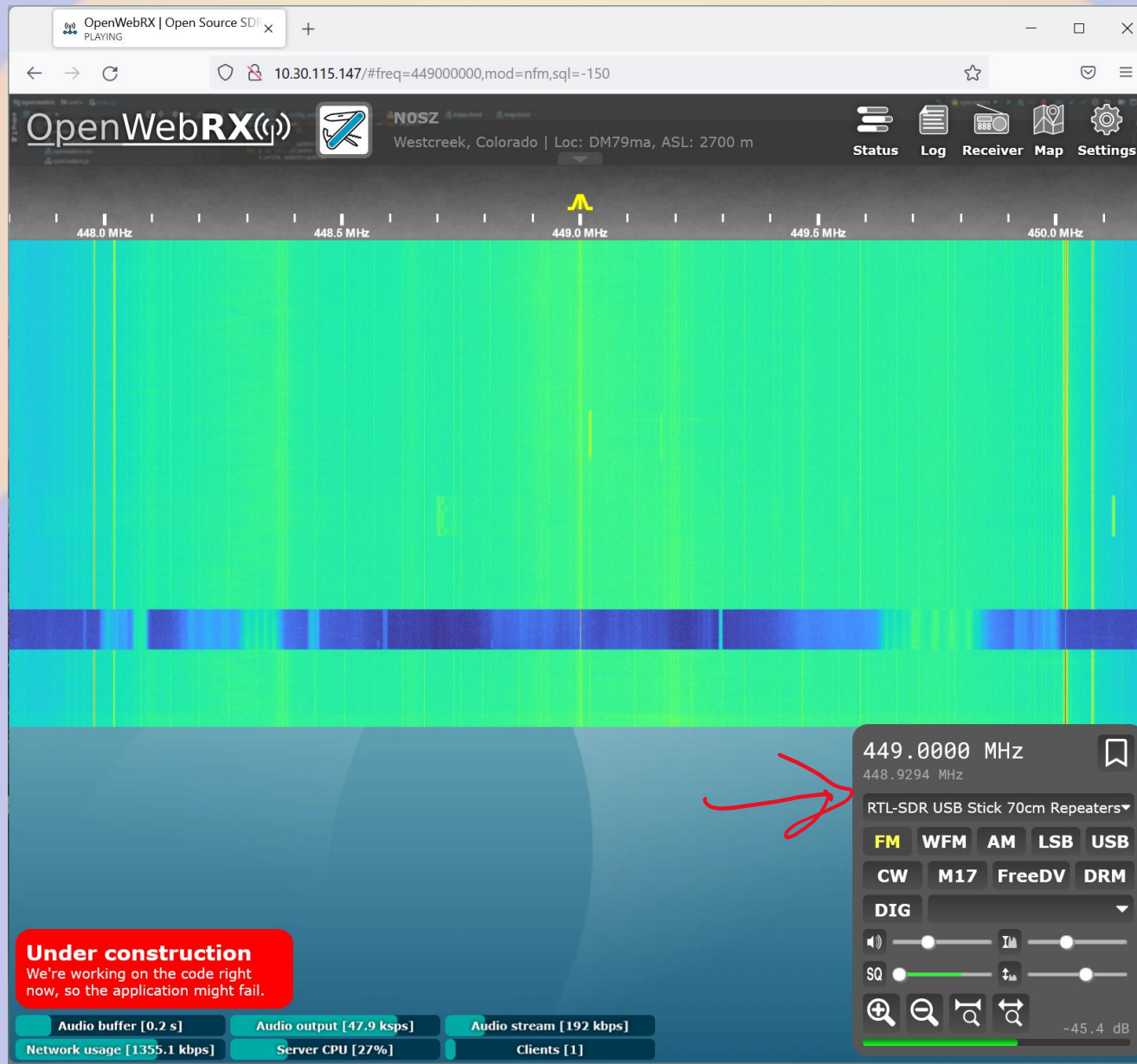
FM WFM AM LSB USB
CW M17 FreeDV DRM

DIG

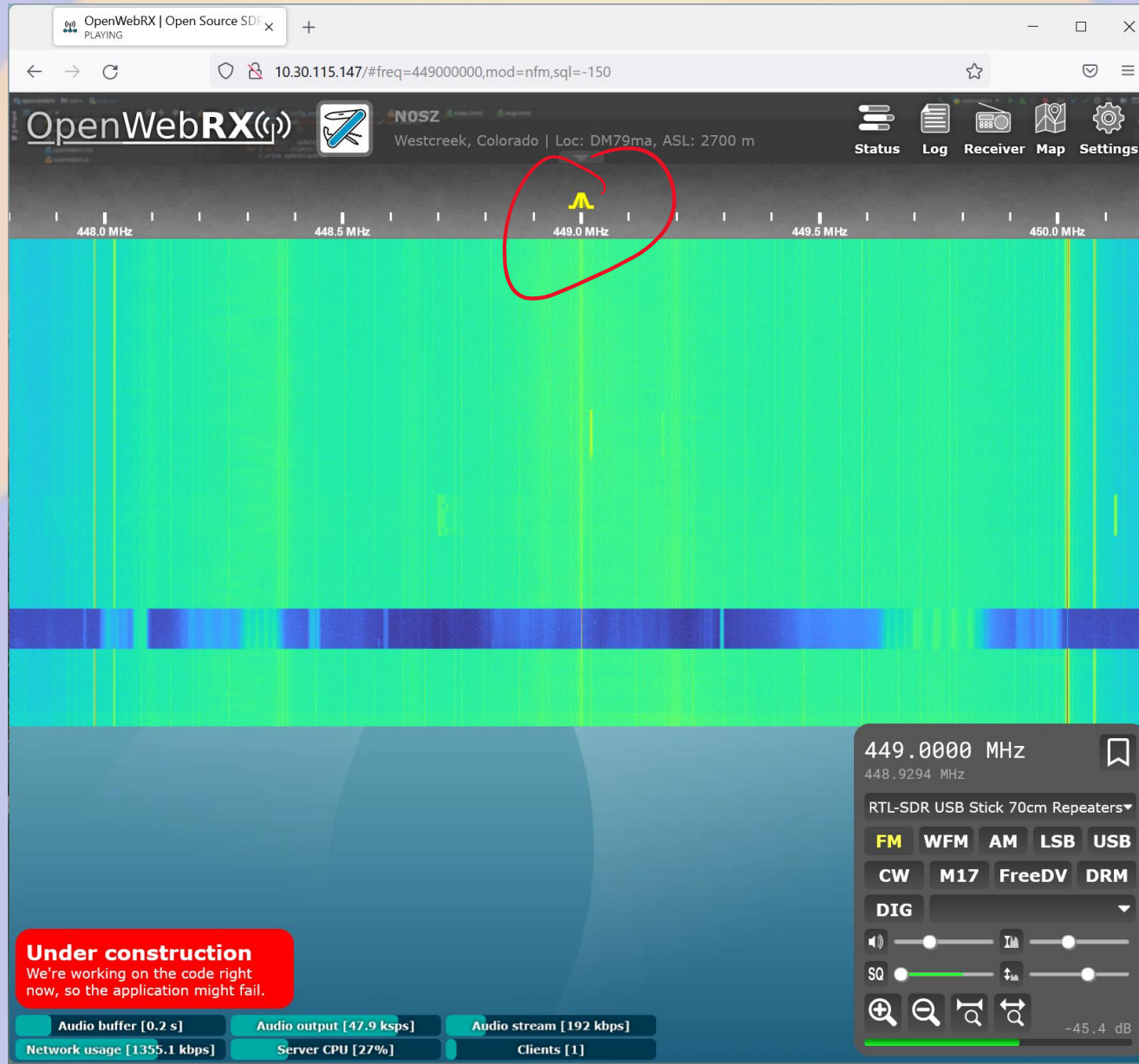
Audio buffer [0.2 s] Audio output [47.9 kbps] Audio stream [192 kbps]
Network usage [1355.1 kbps] Server CPU [27%] Clients [1]

Under construction
We're working on the code right now, so the application might fail.

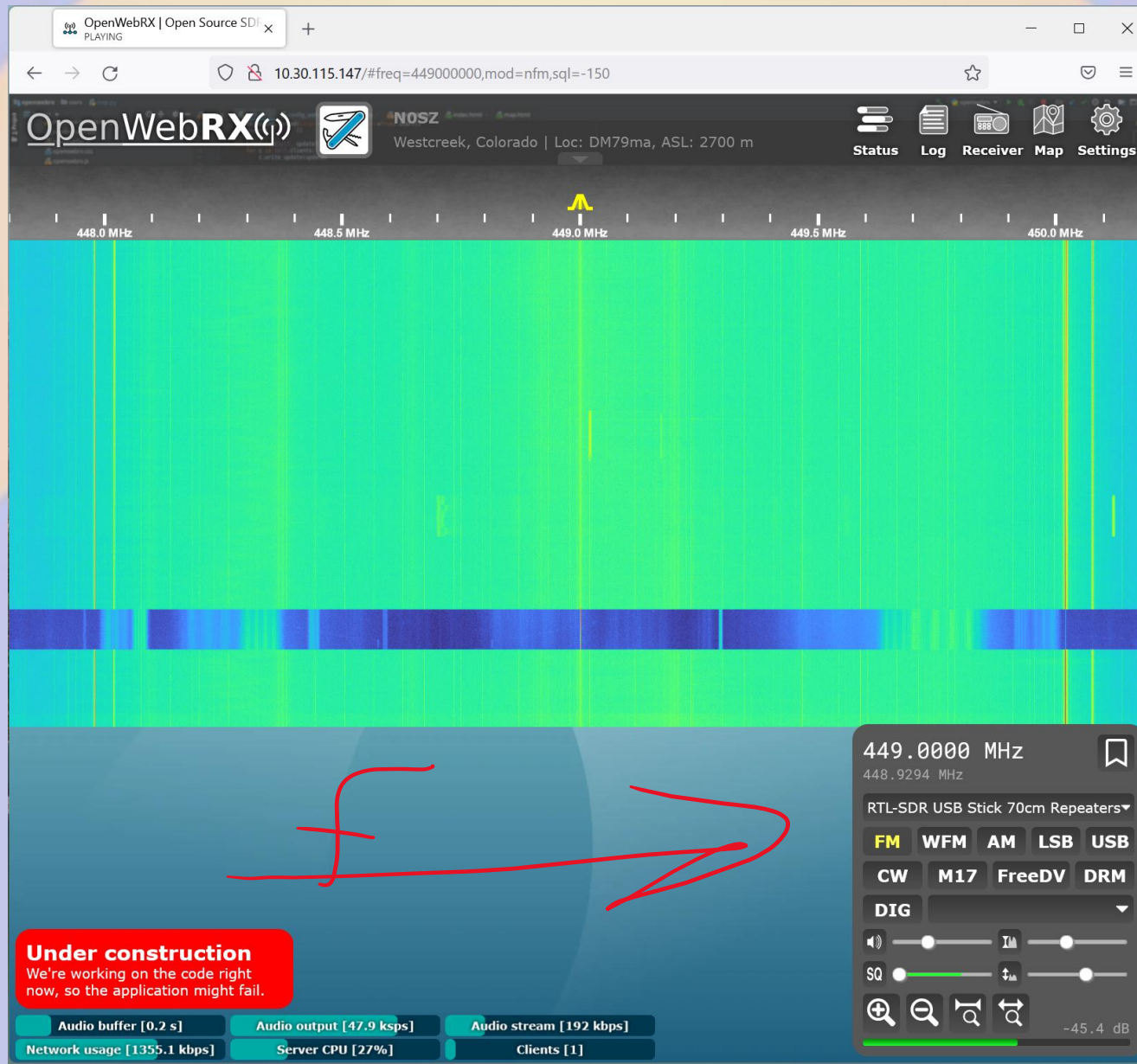
Band and Device Selection



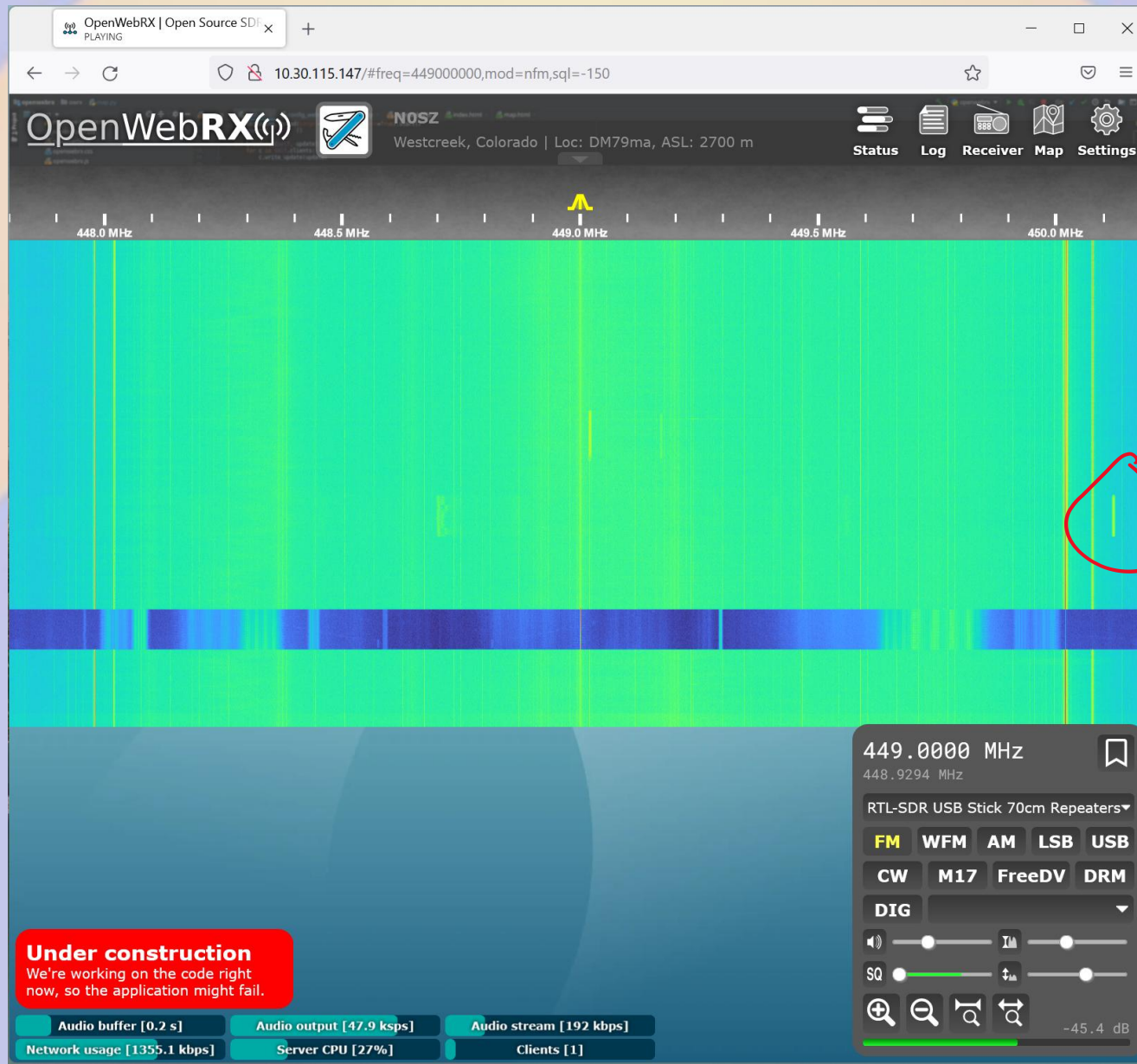
Dial Frequency



Time/Frequency



A Signal!



Packet

The screenshot shows the OpenWebRX web interface in a browser. The address bar displays the URL: `10.30.115.147/#freq=144390000,mod=nfm,secondary_mod=packet,sql=-150`. The interface includes a top navigation bar with icons for Status, Log, Receiver, Map, and Settings. Below this is a spectrum plot showing a frequency range from 144 MHz to 148 MHz. A red banner in the lower-left corner of the plot area reads: "Under construction We're working on the code right now, so the application might fail." Below the plot is a table of callsigns and a settings panel on the right.

Under construction
We're working on the code right now, so the application might fail.

UTC	Callsign	Coord	Comment
	KD0OXW-2	447.1500+	PL 103.5 Digital C4FM Wires-X - Skyhub Link
	BADGR	6/Wilkerson Pass	KC0CVU
	BADGR	6/Wilkerson Pass	KC0CVU

Settings Panel:

- Frequency: 144.3900 MHz (145.9127 MHz)
- Mode: HackRF 2M
- Modulation: FM, WFM, AM, LSB, USB
- Modulation: CW, M17, FreeDV, DRM
- Modulation: DIG, Packet
- Audio buffer: [0.1 s]
- Audio output: [48.0 kbps]
- Audio stream: [192 kbps]
- Network usage: [1329.7 kbps]
- Server CPU: [54%]
- Clients: [1]

Neat, so how do I get this?

- Well, you could cheat. There are a bunch of OpenWebRX instances publicly available on the internet... but that's no fun
- Let's go Shopping:
 - Pi4
 - RTLsdr
 - or something better if you want, but I'm going to stick with RTLsdr because they're cheap
 - Something that'll work as an antenna for your band(s) of interest
 - Discones work well because they're very wide-band, but whatever you have
 - uSD for the Pi (16G should be plenty but get a name brand)
 - Maybe some peripherals for the Pi (if you want)
 - uSD reader

The easy way: Prebuilt Image

- Grab the latest image and unzip it:
- <https://www.openwebbrx.de/download/rpi.php>
- (I hate prebuilt images, so I'll put some notes on how to install manually at the end of the deck for the Linux gurus)
- Install the Raspberry Pi imager tool from here:
- <https://www.raspberrypi.com/software/>
- (If you've done this before, you can use whatever you want – just write the image to the card)



Raspberry Pi Imager v1.7.2



Raspberry Pi

Operating System

CHOOSE OS

Storage

CHOOSE STORAGE

WRITE



Raspberry Pi Imager v1.7.2



Operating System



Emulators for running retro-computing platforms



Other specific-purpose OS

Thin clients, digital signage and 3D printing operating systems



Misc utility images

Bootloader EEPROM configuration, etc.



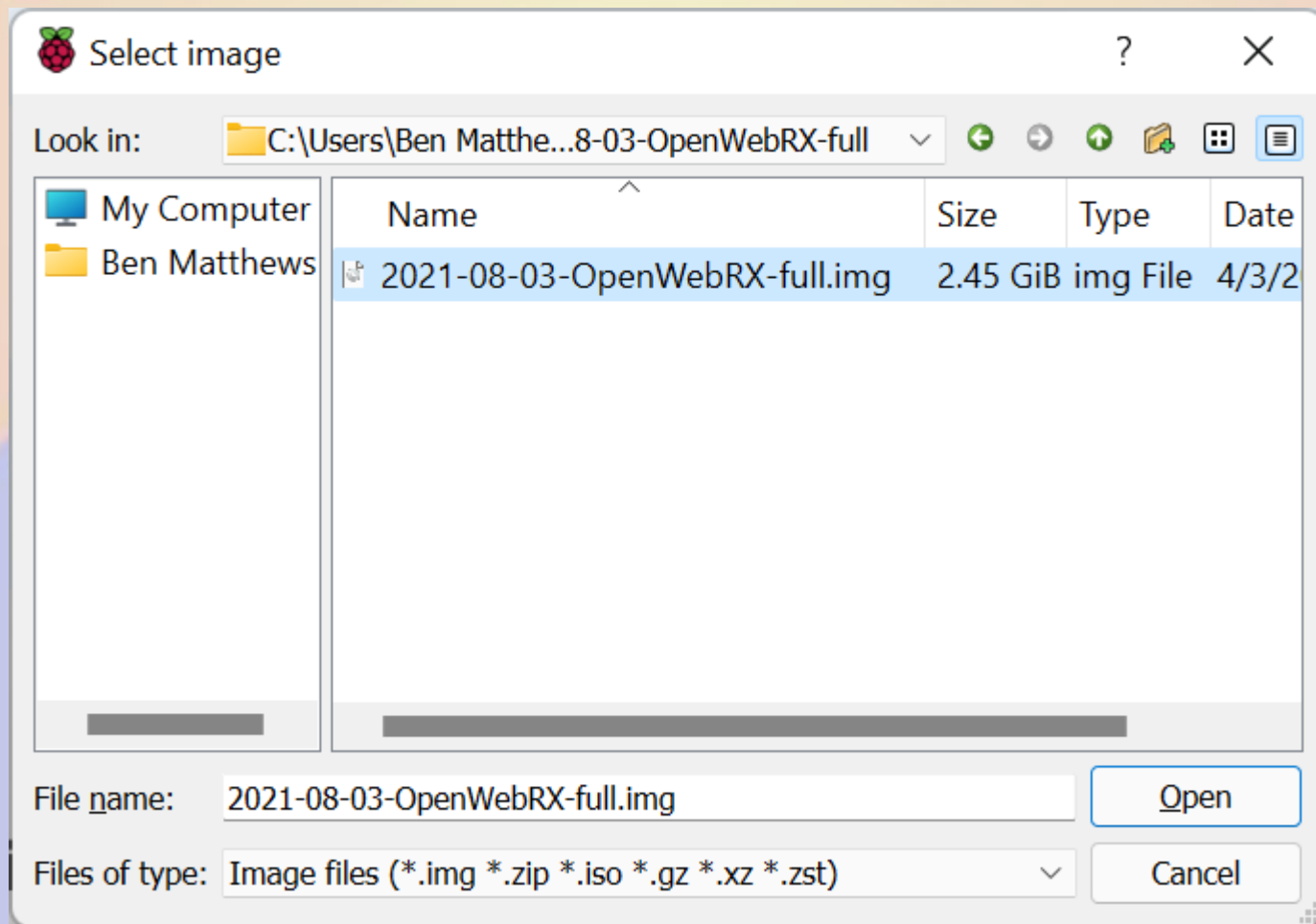
Erase

Format card as FAT32

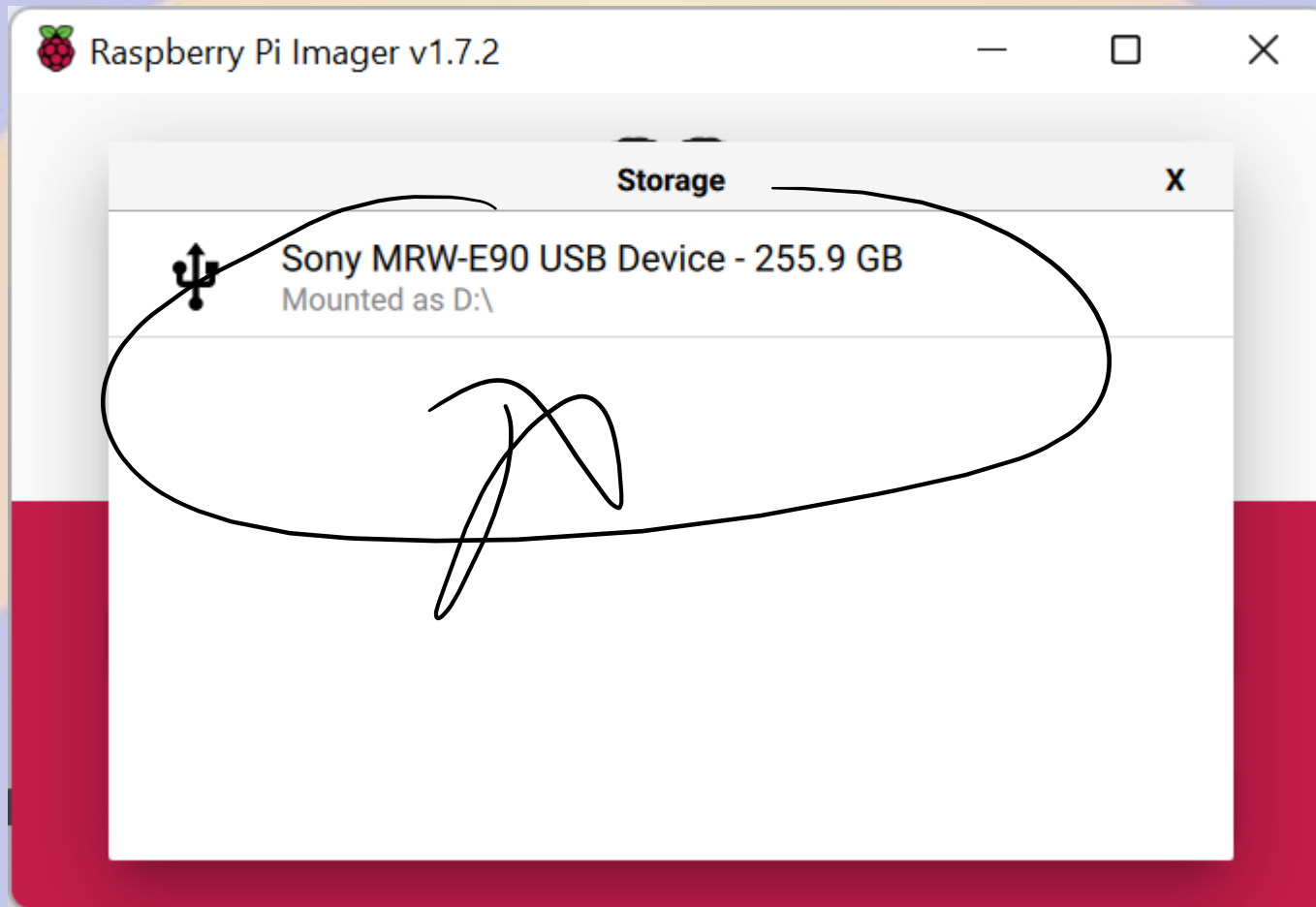


Use custom

Select a custom .img from your computer







**Careful: Whatever you
choose here will be erased!**



Raspberry Pi Imager v1.7.2



Raspberry Pi

Operating System

2021-08-03-OPENWEBRX-FULL.IMG

Storage

SONY MRW-E9...

WRITE





Raspberry Pi Imager v1.7.2



Advanced options



Image customization options

to always use



☒ Set hostname: openwebrx.local

☒ Enable SSH

☒ Use password authentication

☐ Allow public-key authentication only

Set authorized_keys for 'pi': _____

SAVE



Raspberry Pi Imager v1.7.2



Advanced options



Set username and password

Username: pi

Password: ●●●●●●●●



Configure wireless LAN

SSID:



Hidden SSID

Password:

●●●●●●●●

SAVE



Raspberry Pi Imager v1.7.2



Advanced options



Password:

••••••••



Show password

Wireless LAN country: US



Set locale settings

Time zone:

America/Denver



Keyboard layout: us



SAVE



Raspberry Pi Imager v1.7.2



Raspberry Pi

Operating System

2021-08-03-OPENWEBRX-FULL.IMG

Storage

SONY MRW-E9...

WRITE





Raspberry Pi Imager v1.7.2



Write Successful



2021-08-03-OpenWebRX-full.img has been
written to **Sony MRW-E90 USB Device**

You can now remove the SD card from the
reader

CONTINUE



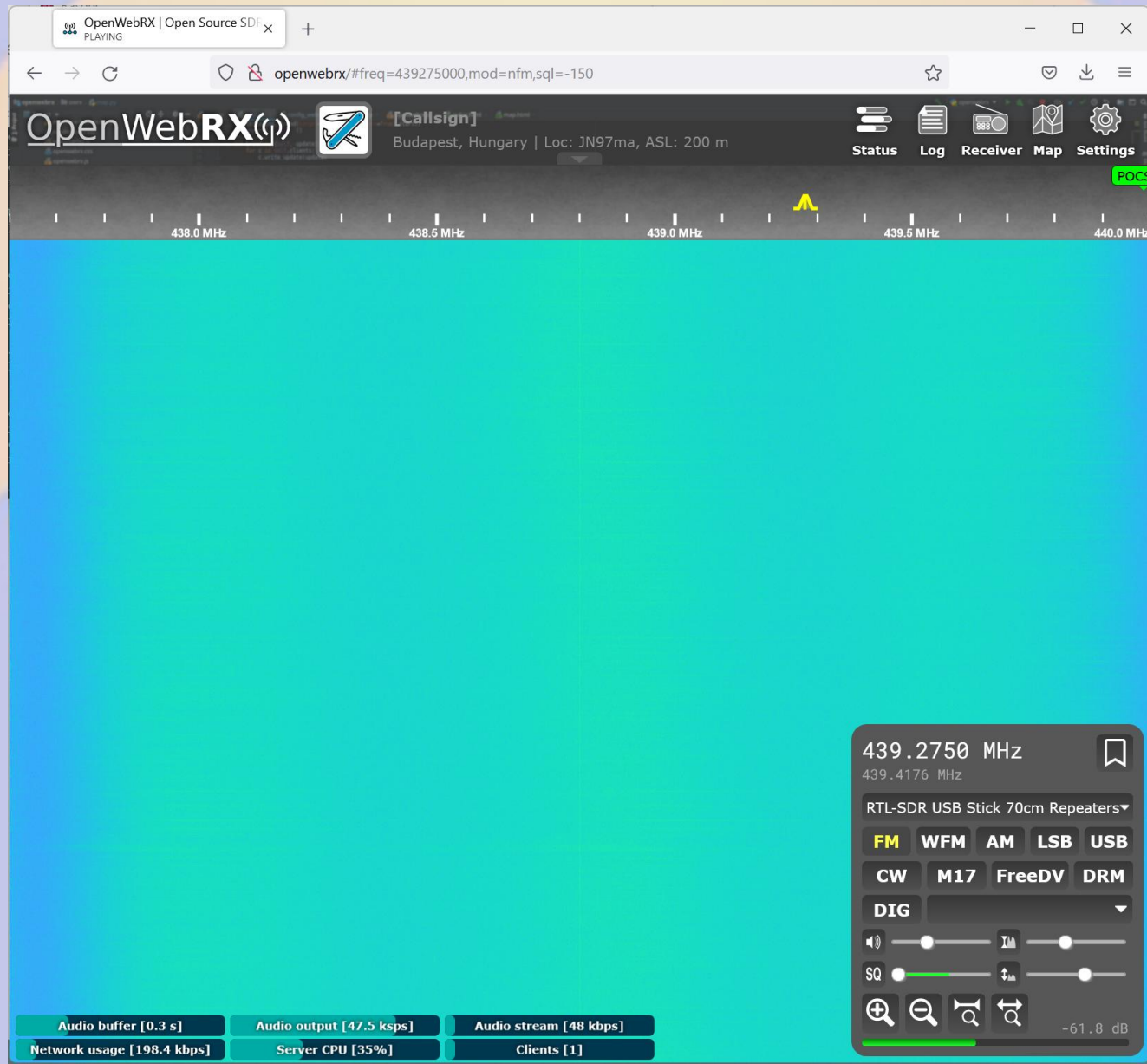
Success?

- Insert the uSD card into your pi
- Connect your RTLSDR
- Power up the pi
- Wait a bit

Connect to OpenWebRx

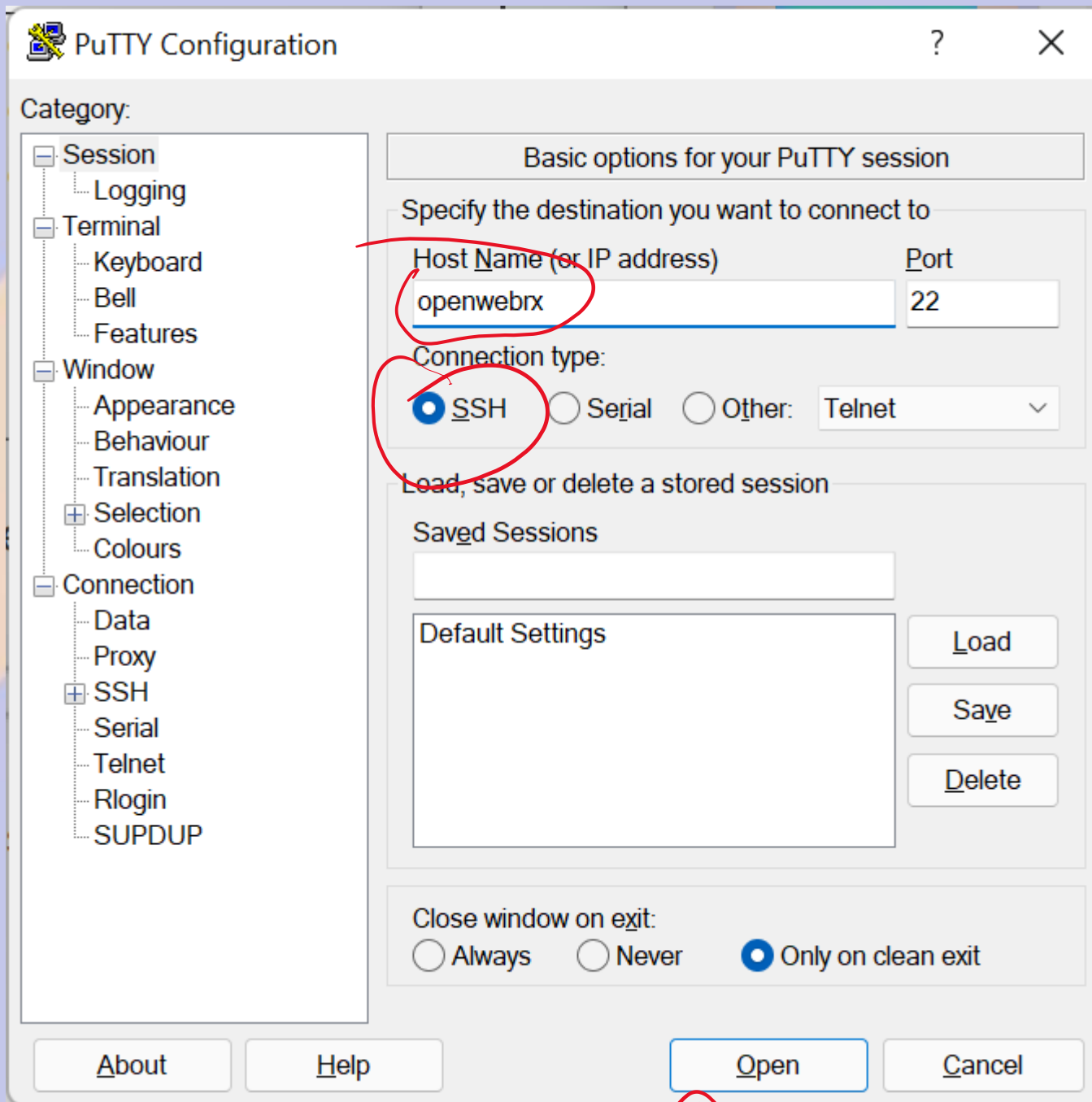
- If you're exceedingly lucky, you can just point a browser at <http://openwebrx>
- If not, you'll need to find the IP of your PI and point a browser at that
- Maybe you can ask your router
- If all else fails, you can connect an HDMI monitor to the pi and log in with the account you configured back a couple of slides and use "ifconfig wlan0" to get the IP ("ifconfig eth0" if you chose to connected a wired network)

You should see something like this



Let's add an account to configure with

- You'll need a ssh client (but you probably have one)
- Linux/Mac/Win10/Win11 people: `ssh pi@your_ip` or `ssh pi@openwebrx` in a terminal window (hopefully whatever address you used for the web interface should work)
- Other Windows people:
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



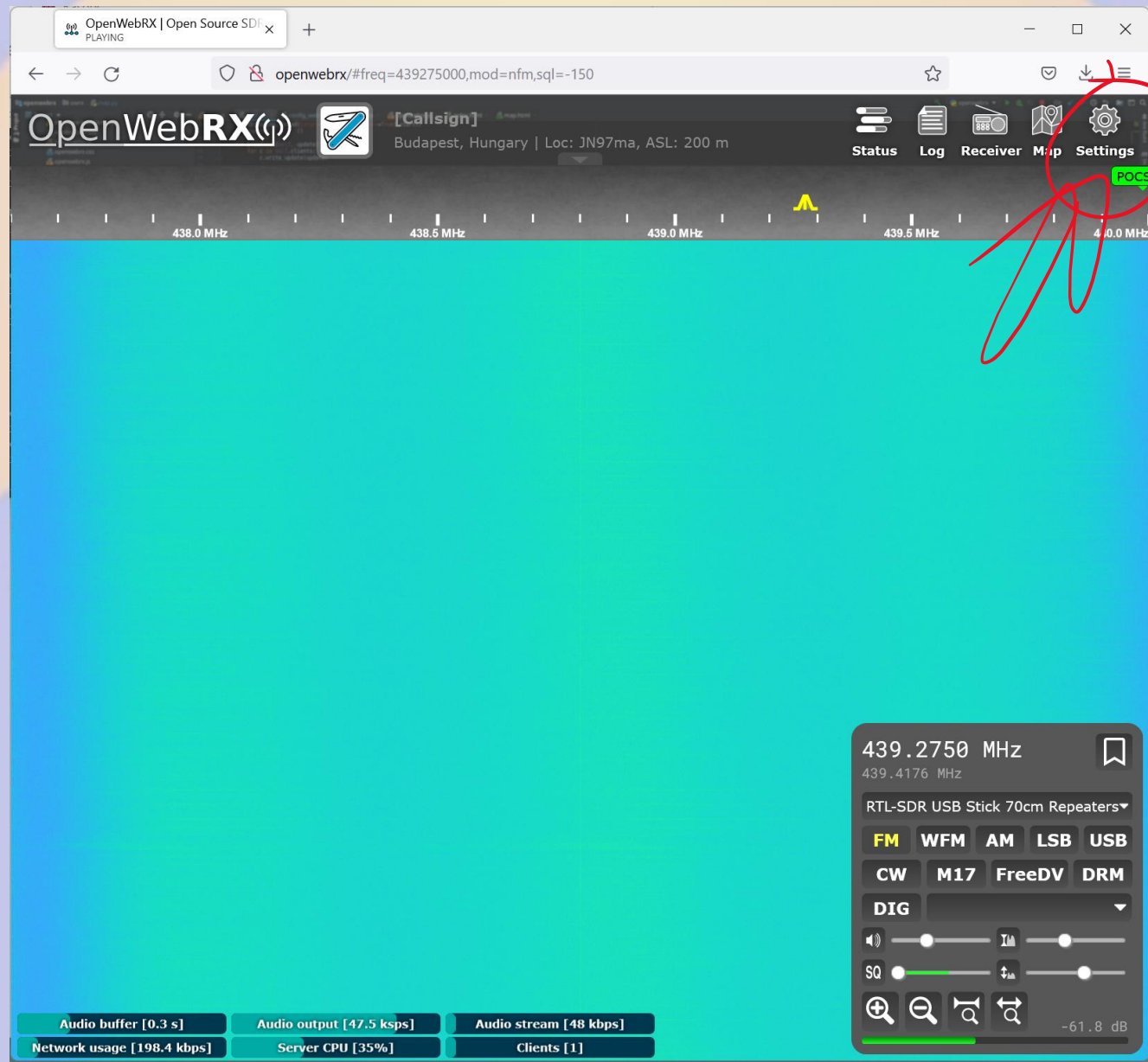
Adding a user

- Once logged-in, do:
- `sudo openwebrx admin adduser admin`
- You'll be prompted for a password, you can use whatever (but you'll need it for the next step)

pi@openwebrx: ~

```
File "/usr/lib/python3/dist-packages/owrx/admin/commands.py", line 48, in run
    if username in userList:
File "/usr/lib/python3/dist-packages/owrx/users.py", line 225, in __contains__
    self.refresh()
File "/usr/lib/python3/dist-packages/owrx/users.py", line 146, in refresh
    self.users = self._loadUsers()
File "/usr/lib/python3/dist-packages/owrx/users.py", line 161, in _loadUsers
    usersFile = self._getUsersFile()
File "/usr/lib/python3/dist-packages/owrx/users.py", line 149, in _getUsersFile
    config = CoreConfig()
File "/usr/lib/python3/dist-packages/owrx/config/core.py", line 34, in __init__
    CoreConfig.checkDirectory(self.data_directory, "data_directory")
File "/usr/lib/python3/dist-packages/owrx/config/core.py", line 47, in checkDirectory
    raise ConfigError(key, "{dir} is not writable".format(dir=dir))
owrx.config.error.ConfigError: Configuration Error (key: data_directory): /var/lib/openwebrx is not writable
pi@openwebrx:~ $ sudo openwebrx admin adduser admin
Please enter the new password for admin:
Please confirm the new password:
Creating user admin...
pi@openwebrx:~ $
```


Let's do some configuration



Configuration

OpenWebRX Login

openwebrx/login?ref=settings

OpenWebRX

[Callsign]

Budapest, Hungary | Loc: JN97ma, ASL: 200 m

Map Settings

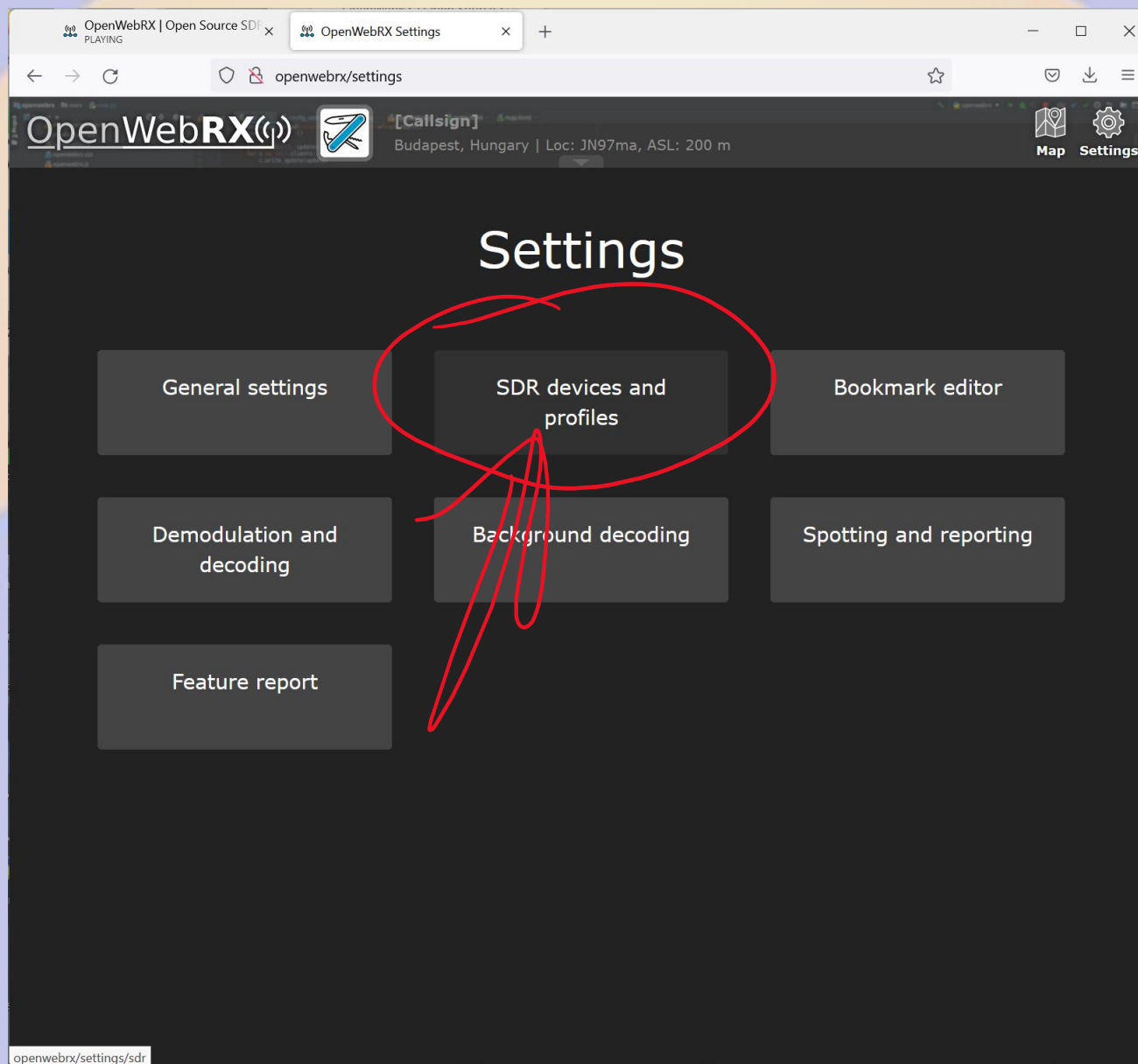
Username

admin

Password

.....

Login



Since we're using RTL-SDR, we can just recycle the example. If you have something else you can add on the bottom right

OpenWebRX | Open Source SDR PLAYING

OpenWebRX Settings

openwebrx/settings/sdr

OpenWebRX [Callsign] Budapest, Hungary | Loc: JN97ma, ASL: 200 m

Map Settings

Settings / SDR device settings

SDR device settings

RTL-SDR USB Stick State: Running	2 profile(s) Current profile: 70cm Repeaters Clients: INACTIVE: 2, USER: 3 Connections: 1
Airspy HF+ State: Stopped	5 profile(s) Current profile: 20m Clients: INACTIVE: 2 Connections: 0
SDRPlay RSP2 State: Stopped	5 profile(s) Current profile: 20m Clients: INACTIVE: 2 Connections: 0

Settings / SDR device settings

Add new device...

Need to add configuration for each band we're interested in

OpenWebRX | Open Source SDR PLAYING

OpenWebRX Settings

openwebrx/settings/sdr/rtlsdr

OpenWebRX [Callsign] Budapest, Hungary | Loc: JN97ma, ASL: 200 m

Map Settings

Settings / SDR device settings / RTL-SDR USB Stick

RTL-SDR USB Stick

RTL-SDR USB Stick 70cm Repeaters 2m New profile

Device settings

Device name

☒ Enable this device

Additional optional settings

Settings / SDR device settings / RTL-SDR USB Stick


OpenWebRX | Open Source SDRPLAYING

OpenWebRX Settings

openwebrx/settings/sdr/rtlsdr/newprofile

☆📍📄☰

OpenWebRX



[Callsign]

Budapest, Hungary | Loc: JN97ma, ASL: 200 m

Map

Settings

Settings / SDR device settings / RTL-SDR USB Stick / New profile

RTL-SDR USB Stick70cm Repeaters2mNew profile

Profile settings

Profile name

Profile name

Center frequency

Center frequency

Hz

Sample rate

Sample rate

S/s

Initial frequency

Initial frequency

Hz

Initial modulation

FM

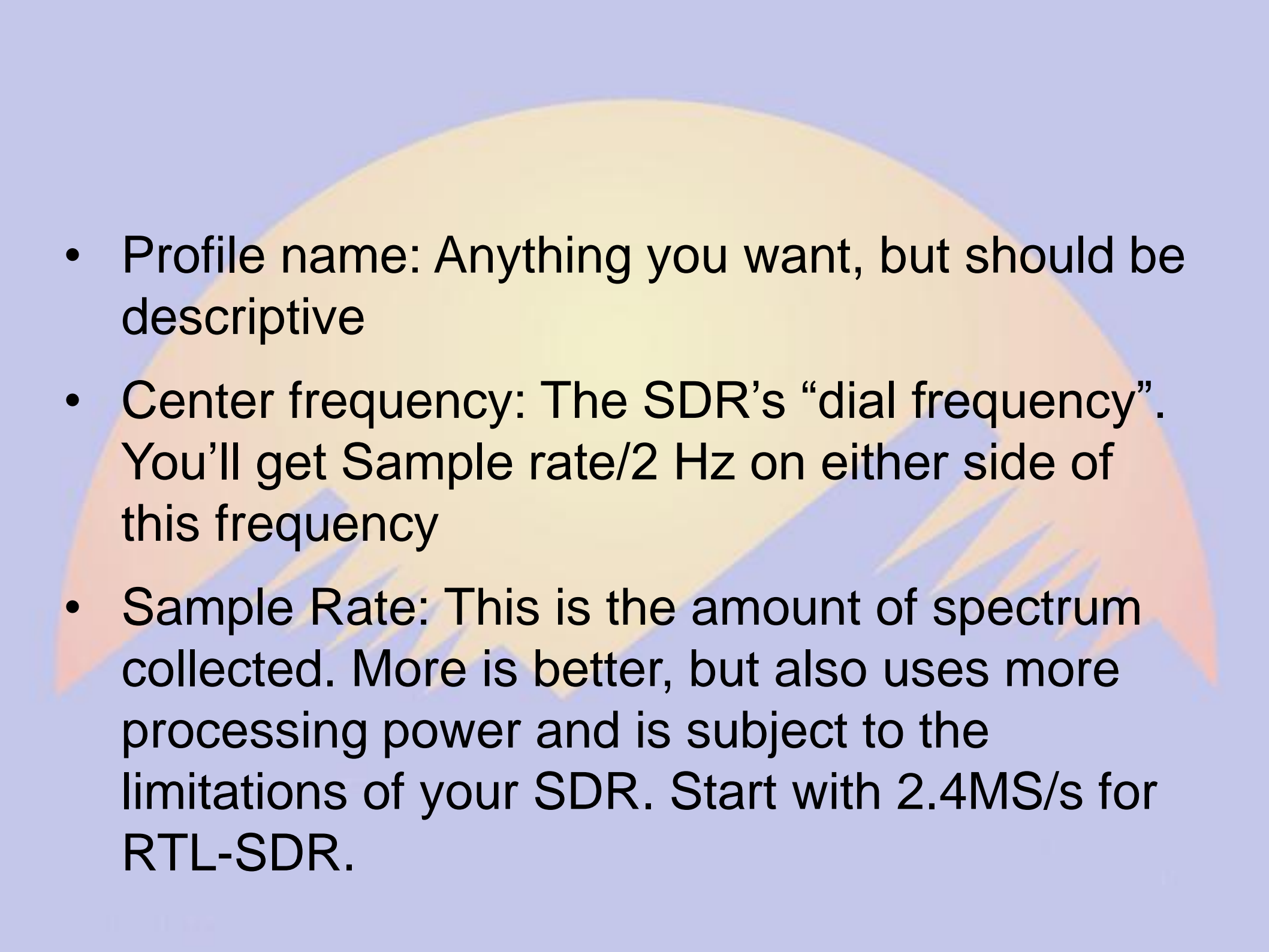
Additional optional settings

Device gain

Add

Settings / SDR device settings / RTL-SDR USB Stick / New profile

Apply and save

- 
- Profile name: Anything you want, but should be descriptive
 - Center frequency: The SDR's "dial frequency". You'll get Sample rate/2 Hz on either side of this frequency
 - Sample Rate: This is the amount of spectrum collected. More is better, but also uses more processing power and is subject to the limitations of your SDR. Start with 2.4MS/s for RTL-SDR.

- Initial Frequency: The interface will display a chunk of spectrum based on your sample rate, but you probably only want to listen to a few KHz. This is the default dial frequency where you're listening
- Initial Modulation: Set this based on the band segment you're configuring. FM for repeaters, USB or LSB for HF, etc.
 - You can always use different modulation if you want – this just sets the default

Example:

OpenWebRX | Open Source SDR PLAYING

OpenWebRX Settings

openwebrx/settings/sdr/rtlsdr/newprofile

OpenWebRX [Callsign] Budapest, Hungary | Loc: JN97ma, ASL: 200 m

Map Settings

Settings / SDR device settings / RTL-SDR USB Stick / New profile

RTL-SDR USB Stick

RTL-SDR USB Stick 70cm Repeaters 2m New profile

Profile settings


Profile name	<input type="text" value="20M"/>		
Center frequency	<input type="text" value="14"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Sample rate	<input type="text" value="2.5"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Initial frequency	<input type="text" value="14.2"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Initial modulation	<input type="text" value="USB"/>		
Additional optional settings	<input type="text" value="Device gain"/>		<input type="button" value="Add"/>

Settings / SDR device settings / RTL-SDR USB Stick / New profile

Apply and save

OpenWebRX | Open Source SDR PLAYING x OpenWebRX Settings x +

← → ↻ openwebrx/settings ☆

OpenWebRX  [Callsign] Budapest, Hungary | Loc: JN97ma, ASL: 200 m Map Settings

Settings

General settings

SDR devices and profiles

Bookmark editor

Demodulation and decoding

Background decoding

Spotting and reporting

Feature report

General Settings: Customize Metadata

OpenWebRX | Open Source SDR PLAYING OpenWebRX Settings

openwebrx/settings/general

General Settings

Receiver information

Receiver name

Receiver location

Receiver elevation meters above mean sea level

Receiver admin

Receiver coordinates

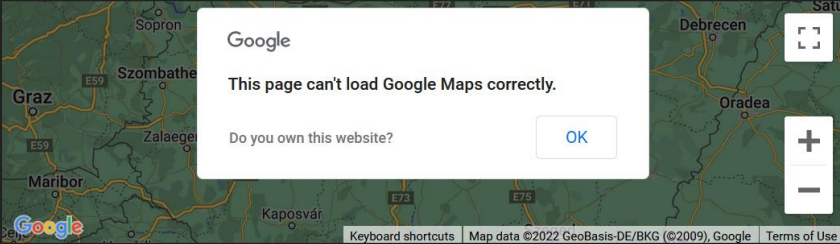


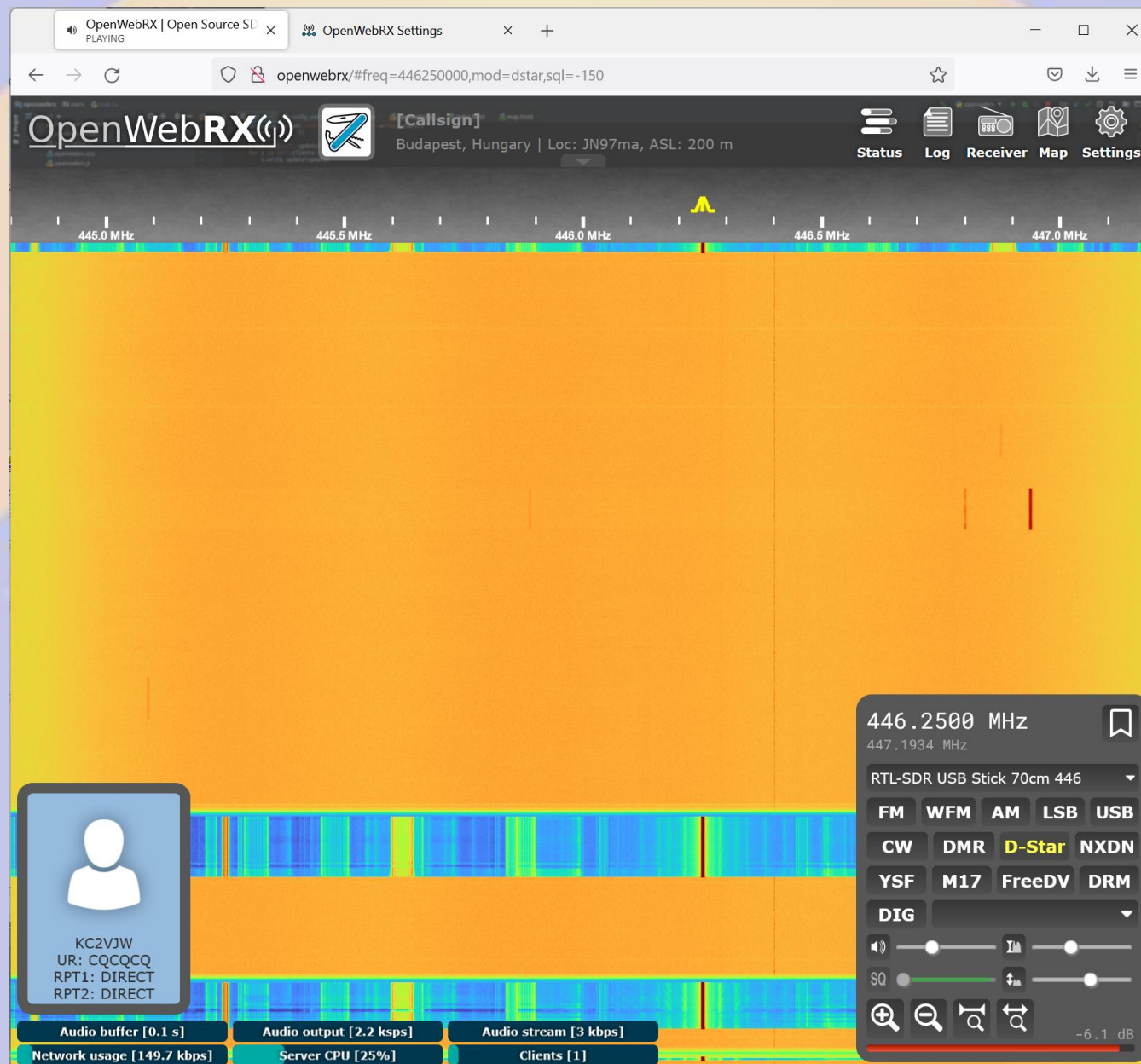
Photo title

Photo description

Apply and save

Google
This page can't load Google Maps correctly.
Do you own this website? [OK](#)

Digital Voice?



Digital Voice?

- Sure, but the IP lawyers are making things difficult
- FreeDV works fine out of the box
- ... but you wanted your favorite AMBE based protocol didn't you?
 - DMR
 - DSTAR
 - YSF
 - NXDN
 - (No P25 support yet)

AMBE?

- How do we stuff ~8kHz of audio into 1.2kbits/sec?
- Compress it to within an inch of its life
- AMBE is a proprietary way of doing this
 - I mentioned FreeDV earlier – it uses something called Codec2 to do the same, for free! (but you can't really buy a radio to do that)
- There are a couple of variants but legally, you need to buy a chip to do this compression for you

How does the AMBE chip work?


- Take raw PCM samples, and feed them to the AMBE chip over a high-speed UART, get compressed symbols in return
- Or vice-versa
- Some of the chips can support several streams simultaneously (maybe you want to transcode from the variant P25 uses to the variant DMR uses?)
- The easiest way for us to get this is to buy a dongle with the AMBE chip and a USB<>Serial chip

AMBE Dongles?

NW Digital Radio ThumbDV™ x DVstick 30 x +

← → ↻ https://nwdigitalradio.com/products/thumbdv™ ☆ ⌵ ⌵ ⌵

NW DIGITAL RADIO Home Catalog 3rd Party Support 🔍 🛒



Amateur Radio ThumbDV™

\$99.95 ~~\$119.95~~ **SALE**

Shipping calculated at checkout.

ADD TO CART

Buy with Pay

[More payment options](#)

DVSI AMBE3000 Digital Voice Vocoder on USB

HW only, SW Provided and supported by others:

BlueDV by David, PA7LJM
DSTAR, DMR, & Yaesu Fusion. Runs on Android, iOS, Linux & Windows

Buster by Anna, NH6Z
DSTAR on MacOS. Available on the App Store

[AMBEServer](#)
Allows network access to your ThumbDV™

SHARE TWEET PIN IT

YOU MAY ALSO LIKE

NW Digital Radio ThumbDV™ x DVstick 30 x +

← → ↻ https://www.combitronics.nl/index.php?route=product/product&path=61&product_id=103 ☆ ⌵ ⌵ ⌵

COMBITRONICS
THE TECHNOLOGY COMPANY

Shopping Cart 0 Item(s) - 0.00€ 🔍 Search


Welcome visitor you can [login](#) or [create an account](#).

[Home](#) | [Wish List \(0\)](#) | [My Account](#) | [Shopping Cart](#) | [Checkout](#)

DVMEGA Raspberry pi (i)BlueStack Arduino Complete Solutions USB-Wifi Displays

Home • DVMEGA • DVstick 30

DVstick 30



Product Code: DVstick30
Availability: In Stock

Price: **95.00€**

Qty: 1 **Add to Cart** - OR - [Add to Wish List](#) [Add to Compare](#)

12 reviews | [Write a review](#)

Share

Description **Reviews (12)** **Documentation**

This USB-stick facilitates the use of a PC to communicate on Dstar, DMR and C4FM! Just install BlueDV (latest update supports C4FM as well now), and use the PC-microphone and PC-speakers to communicate to reflectors/talkgroups. Ideal solution to use on a laptop whilst travelling.

[Technical datasheet here](#)

How to setup: check AMBE in BlueDV (setup screen), select comport just below that, and **set the baudrate to 460800**. That will make sure BlueDV will recognise the device. Check the extra image above to see the settings!

[Driver for windows 10 in case the comport is not recognised:](#) [Download](#)

Our webshop uses cookies... [More...](#) [I understand](#)

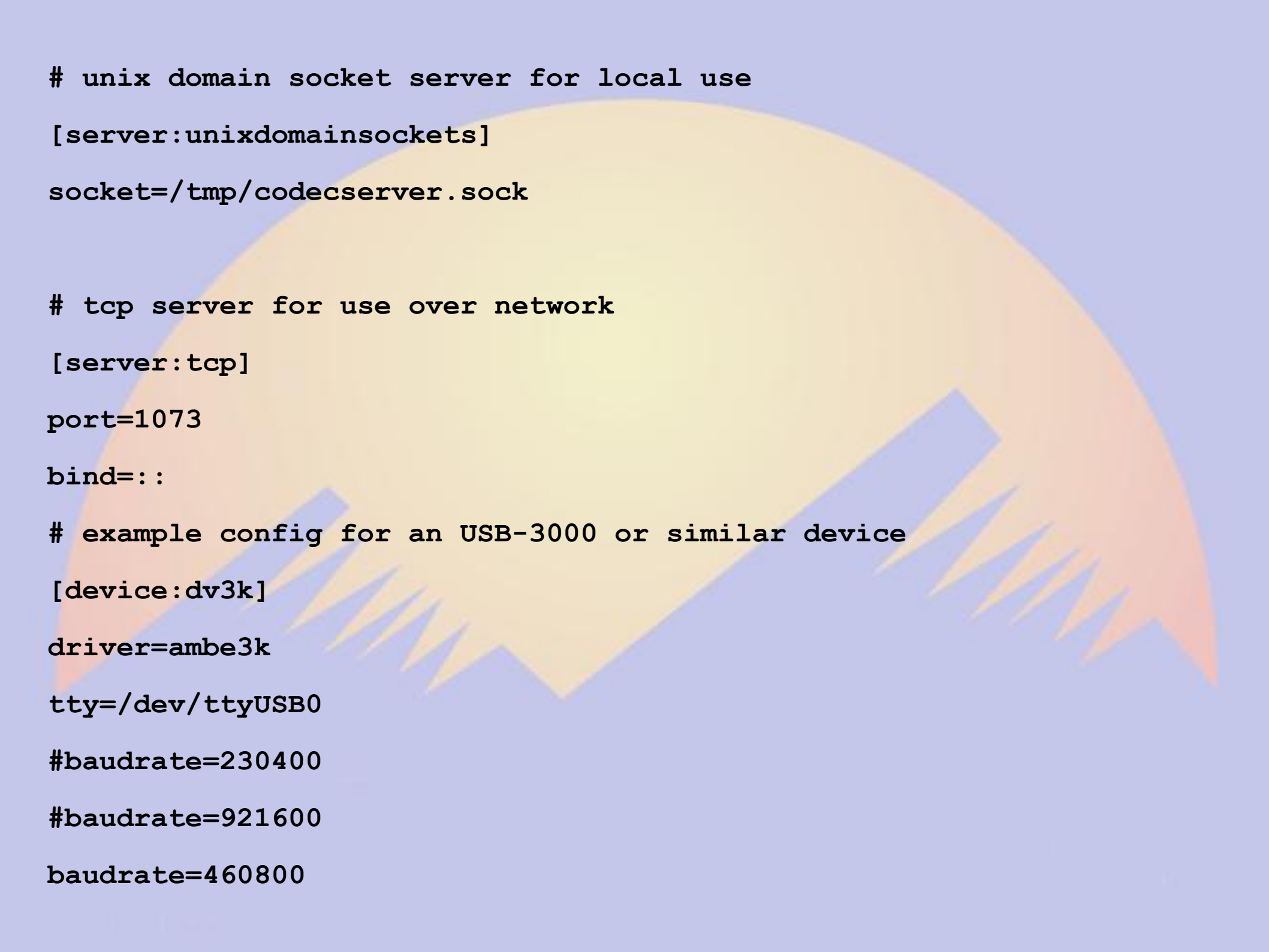
for DVstick33 please contact [Chris at info@combitronics.nl](#) directly

How do we give OpenWebRX access to our ABME dongle?

- Codecserver
 - Not the same as AMBEServer from the DSTAR folks, but it does something similar
 - The digital voice world is pretty fragmented
- Preinstalled in the OpenWebRX Image or available from <https://github.com/jketterl/codecserver>

Configuring Codecserver

- SSH into your OpenWebRx system as before
- Open `/etc/codecserver/codecserver.conf` in an editor
- `sudo nano /etc/codecserver/codecserver.conf`
- Uncomment all the configuration lines
- As long as you don't have any other USB<>Serial devices the defaults should work fine (except, maybe, baud rate)
- Save and exit



```
# unix domain socket server for local use
```

```
[server:unixdomainsockets]
```

```
socket=/tmp/codecservice.sock
```

```
# tcp server for use over network
```

```
[server:tcp]
```

```
port=1073
```

```
bind=::
```

```
# example config for an USB-3000 or similar device
```

```
[device:dv3k]
```

```
driver=ambe3k
```

```
tty=/dev/ttyUSB0
```

```
#baudrate=230400
```

```
#baudrate=921600
```

```
baudrate=460800
```

AMBE and baud rate

- Some AMBE dongles have their baud rate set in hardware
- Most AMBE dongles don't seem to be well documented
- Luckily, there are only a few common values
- Unplug, wait a few seconds and then replug your dongle between attempts, then restart codecserver

AMBE and baud rate

- Try these:

#baudrate=230400 #older ThumbDV

#baudrate=921600 #Some other devices

baudrate=460800 #newer ThumbDV

Codecserver

- Restart codecserver with “systemctl restart codecserver”
- Check status with “systemctl status codecserver”
- If it's happy, you should see something like:

```
Apr 08 15:06:41 openwebrx codecserver[377]: Product  
id: AMBE3000R
```

```
Apr 08 15:06:41 openwebrx codecserver[377]: ;  
Version:  
V120.E100.XXXX.C106.G514.R009.B0010411.C0020208
```

```
Apr 08 15:06:41 openwebrx codecserver[377]: detected  
AMBE3000, creating one channel
```


Codecserver

- Once you have codecserver happy, it's probably easiest to just reboot the pi
 - Though “systemctl restart openwebrrx” is probably good enough
- The digital voice modes should appear in the OpenWebRx UI

But... isn't this “software defined radio” What's with the hardware vocoder?

- Yeah... Patent law
- I didn't tell you this, but there are a couple of software implementations.
- <https://github.com/fventuri/codecservice-mbelib-module>

Can I play with this for free?

- Sure, it's a web app – people have made instances publicly available
- The biggest indexer of websdrs recently shutdown, but the KiwiSDR people have a list of their nodes (based on OpenWebRx):
- <http://rx.linkfanel.net/snr.html>

A stylized background featuring a large, semi-circular sun in shades of yellow and orange, with a blue sky and blue mountain peaks at the bottom.

Questions?
Live Demo?

ben@kc2vjw.com

Building From Source

- See <https://github.com/matthb2/openwebrxsdrplaycontainer/blob/master/Dockerfile>