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

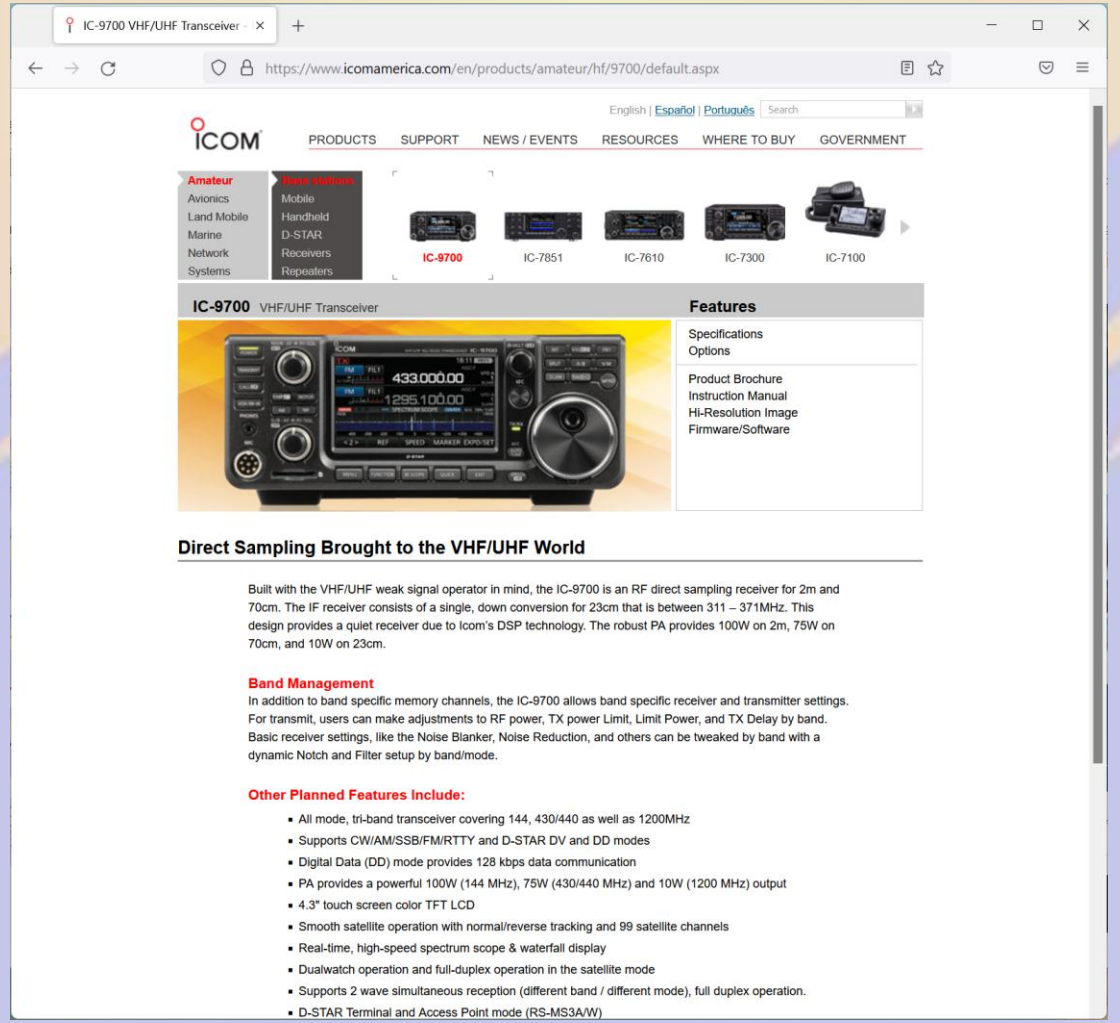
- SDRs are super flexible
- This is great, but it also provides the flexibility to do things the government might prefer you don't
- Be careful and stay inside the limitations of your license(s)
- Please keep Patent/IP law in mind. This talk may show software which could be encumbered in some jurisdictions.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

So you bought an SDR, now what?

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

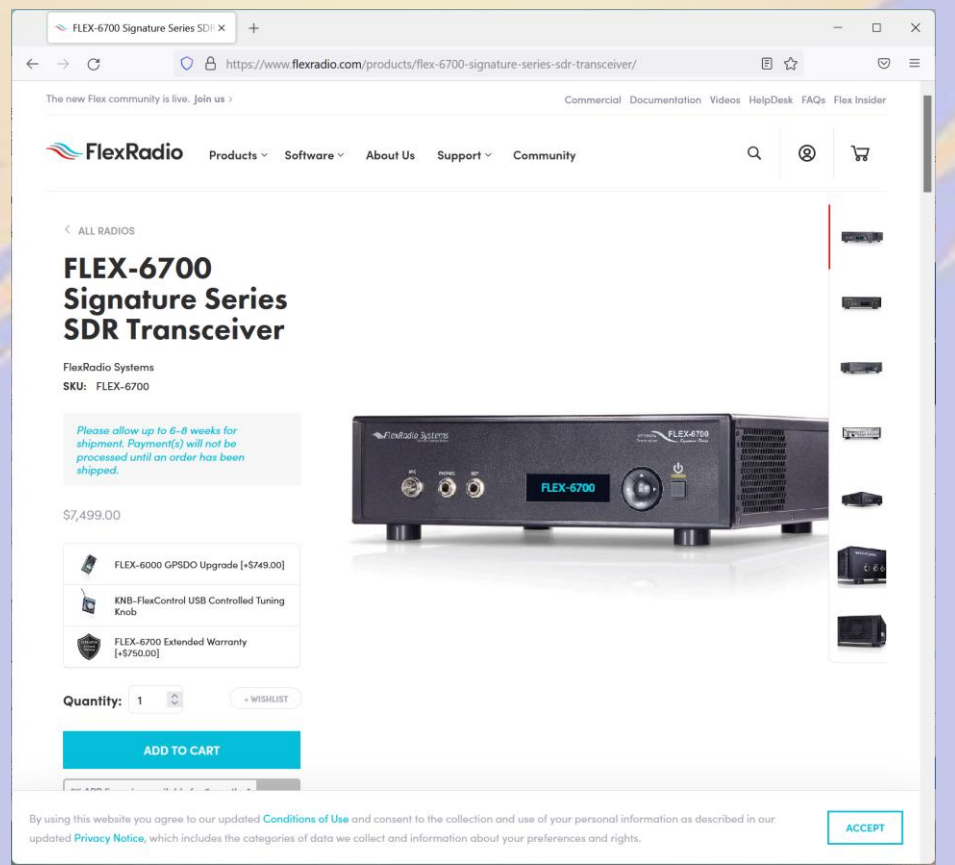
Hardware

- Easy mode: SDR with the computer bundled



Hardware

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



Hardware

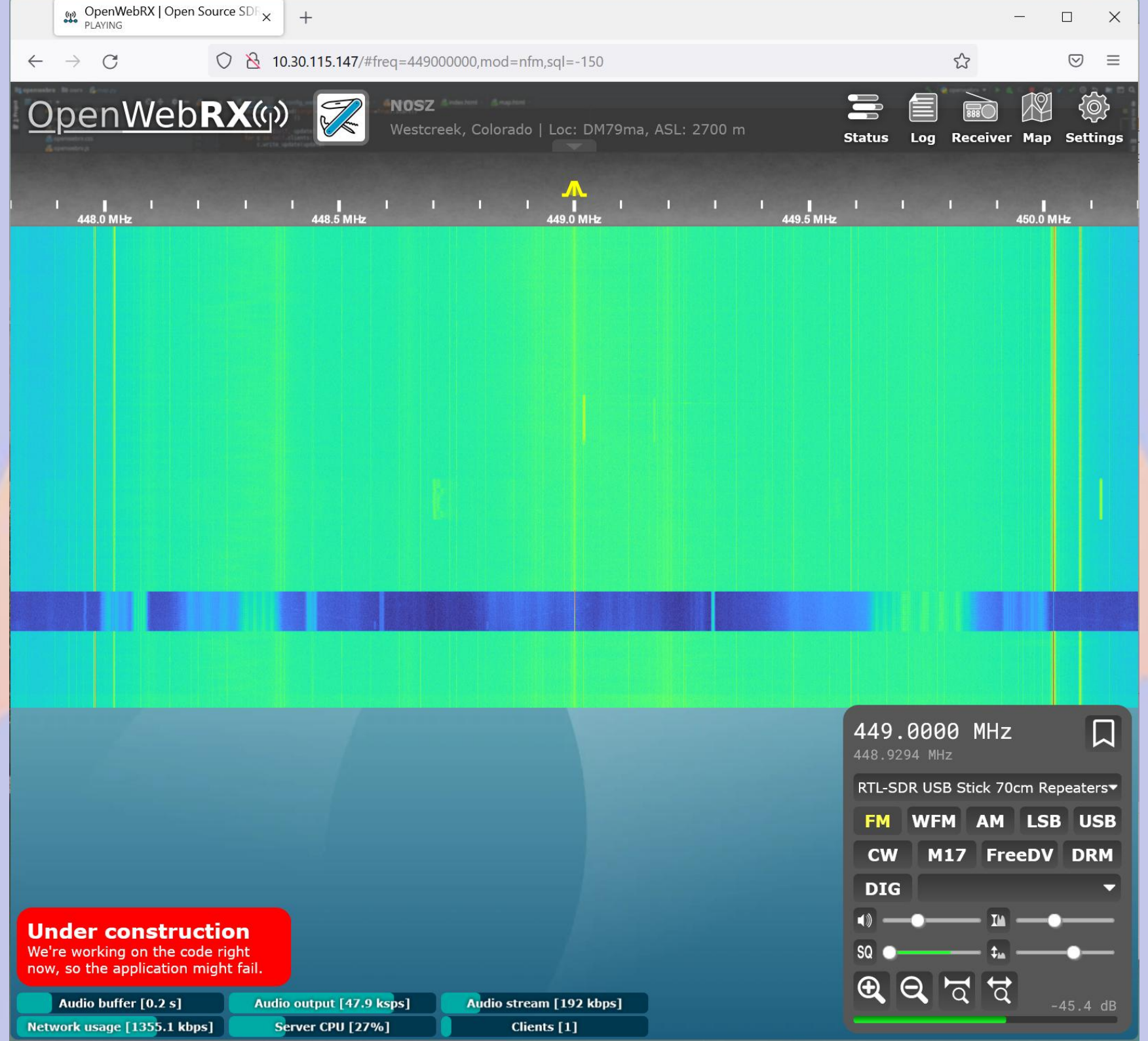
- Hard mode: Figure it out yourself
- RTLSDR
- HackRF
- BladeRF
- SDRPlay
- KrakenSDR
- … Tons to choose from

Hardware

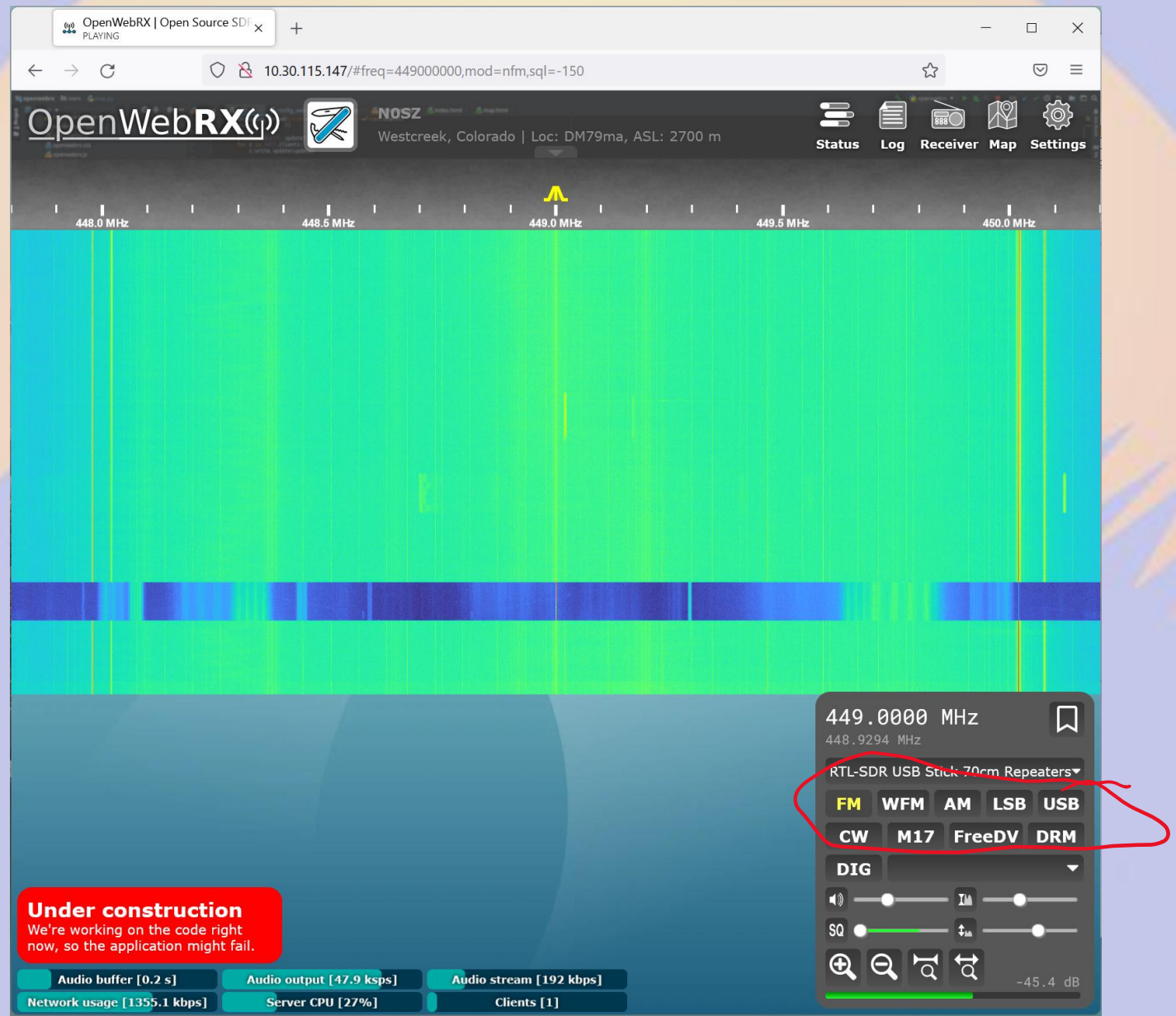
- Antenna system as you would any other radio
- External filters can be a good idea
- Good quality cables (Computers/USB can be RF noisy)
- Faster computers are better, but it really depends on how much bandwidth you need

So.. Where to start?

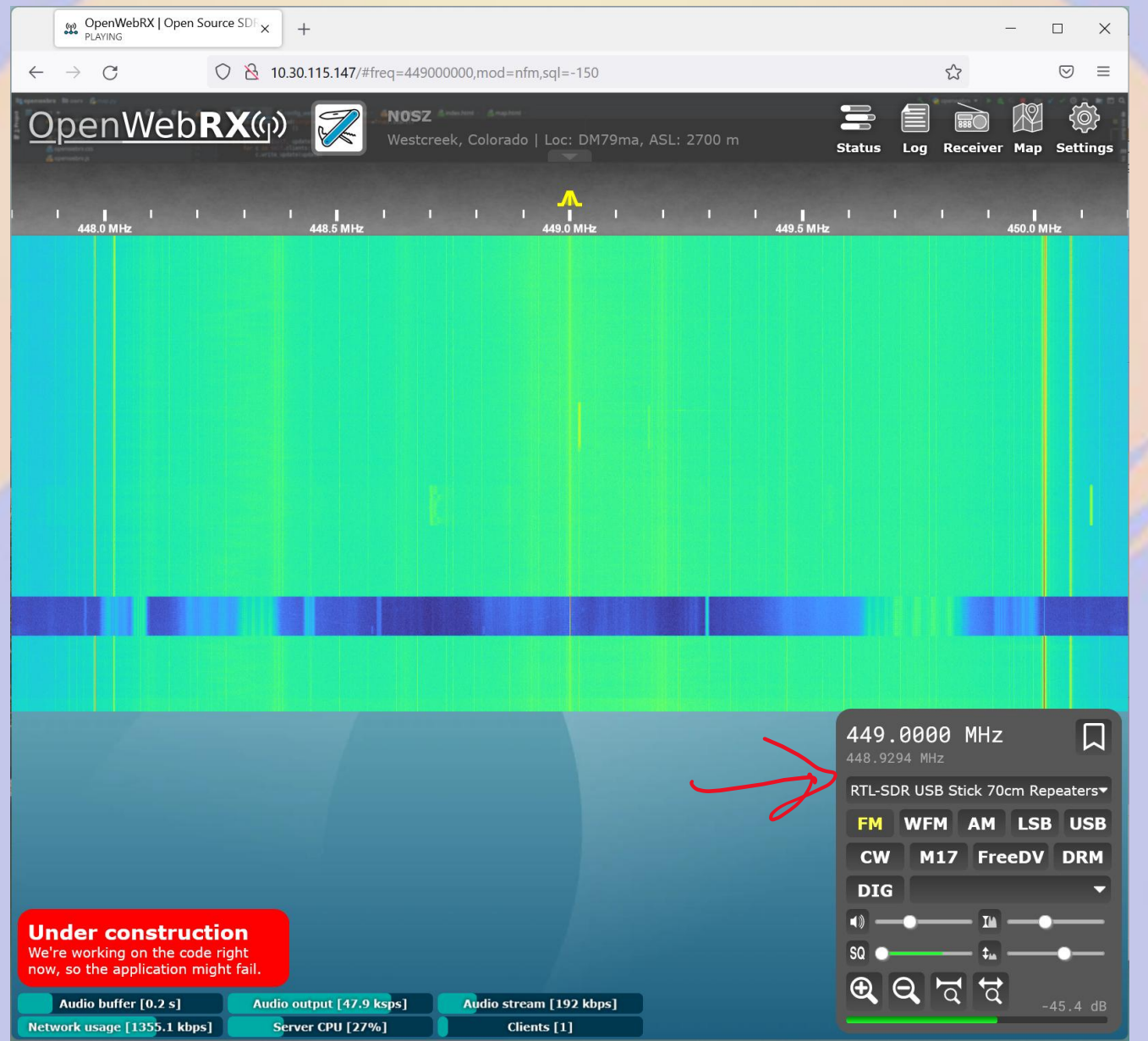
- OpenWebRX and the ham's favorite computer, the Raspberry pi
- Linux is involved, but we'll present a web interface so you don't really need to know Linux
- Lots of decoders for various modes
- Let's try to replace a whole shack full of receivers
- Receive only for now. Some SDRs can transmit, but there's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


OpenWebRX | Open Source SDR
PLAYING
10.30.115.147/#freq=449000000,mod=nfm,sql=-150
OpenWebRX
N0SZ
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
SQ
-45.4 dB
Under construction
We're working on the code right now, so the application might fail.
Audio buffer [0.2 s]
Audio output [47.9 ksps]
Audio stream [192 kbps]
Network usage [1355.1 kbps]
Server CPU [27%]
Clients [1]

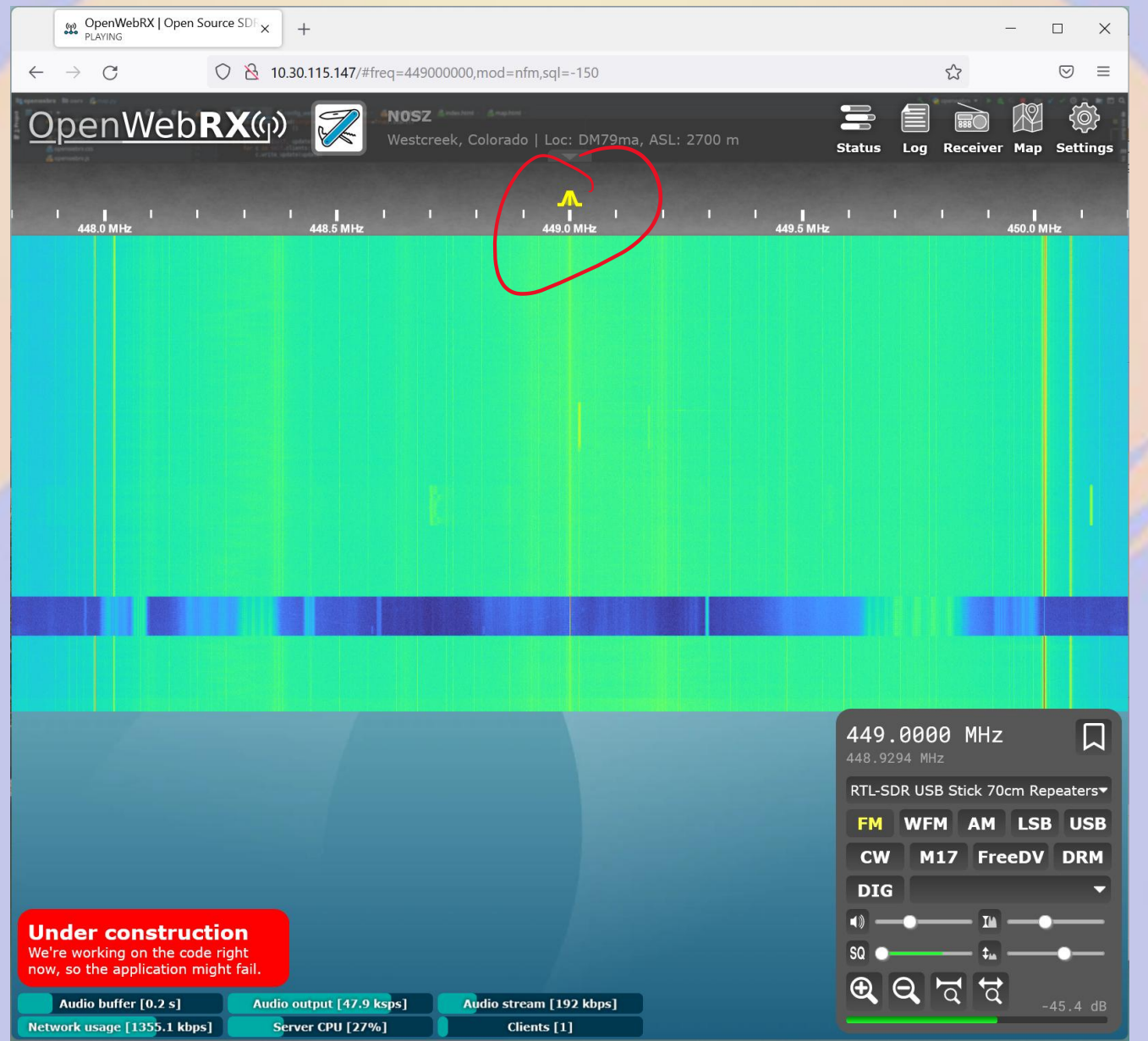
Mode Selection



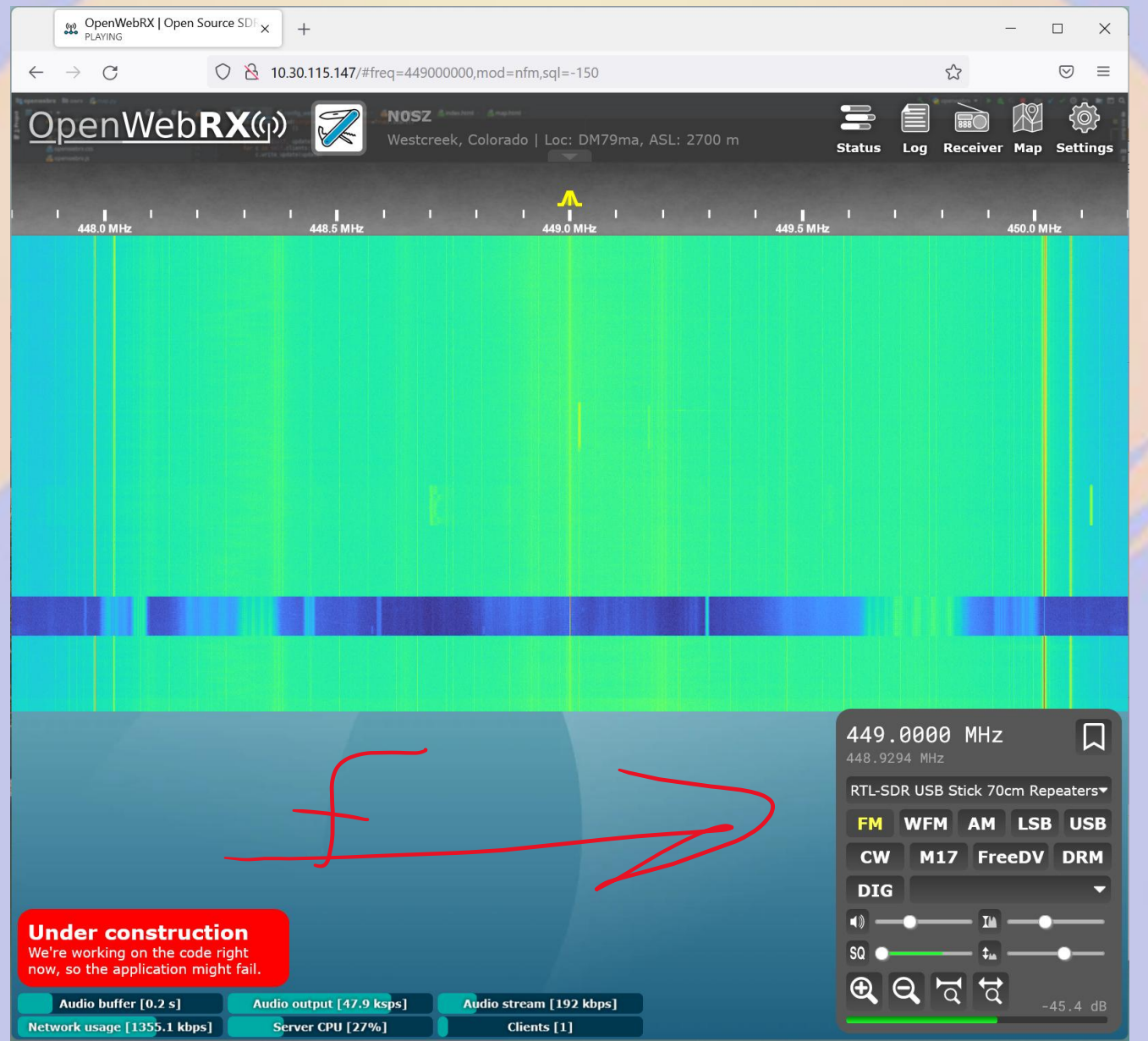
Band and Device Selection



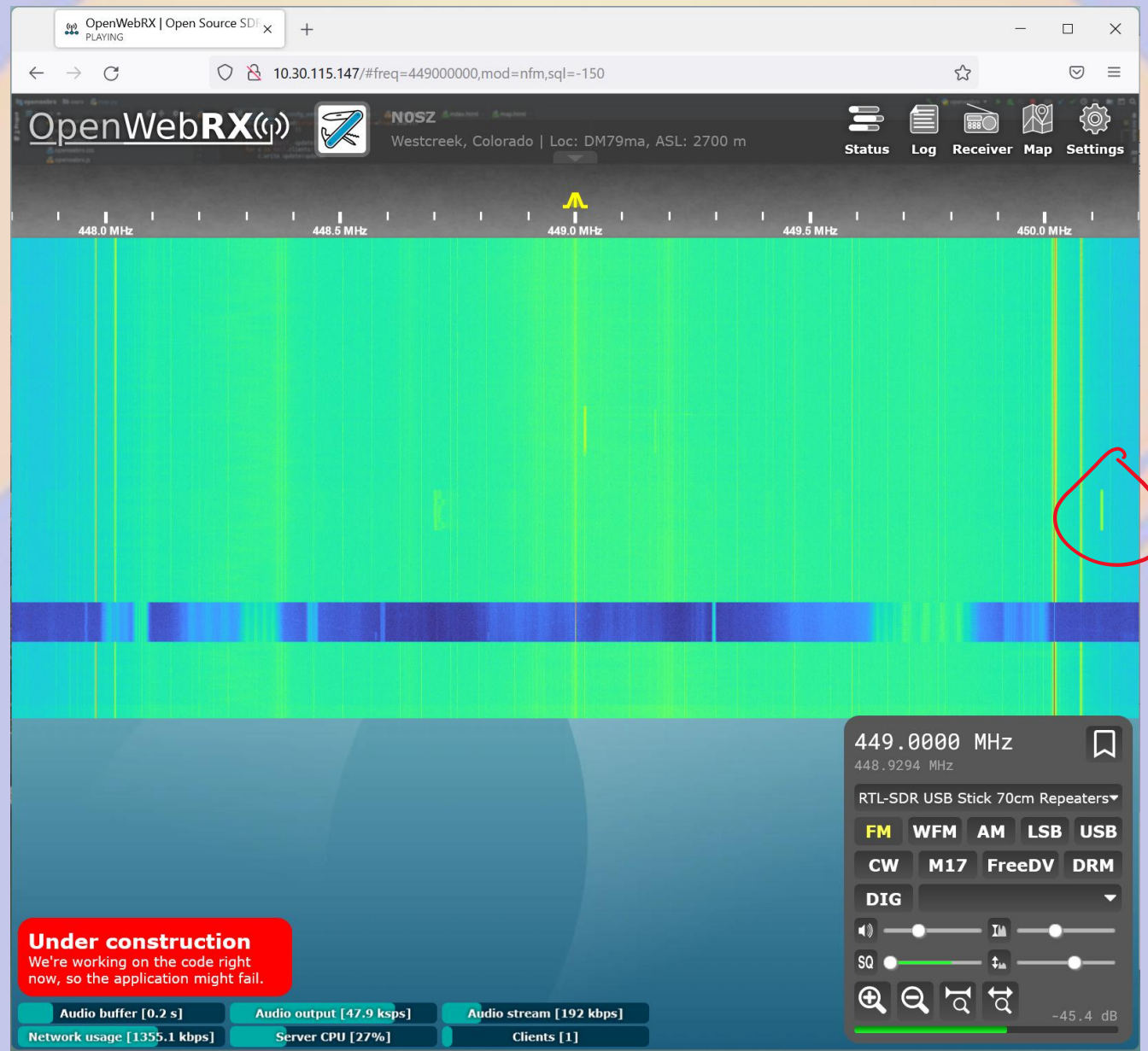
Dial Frequency



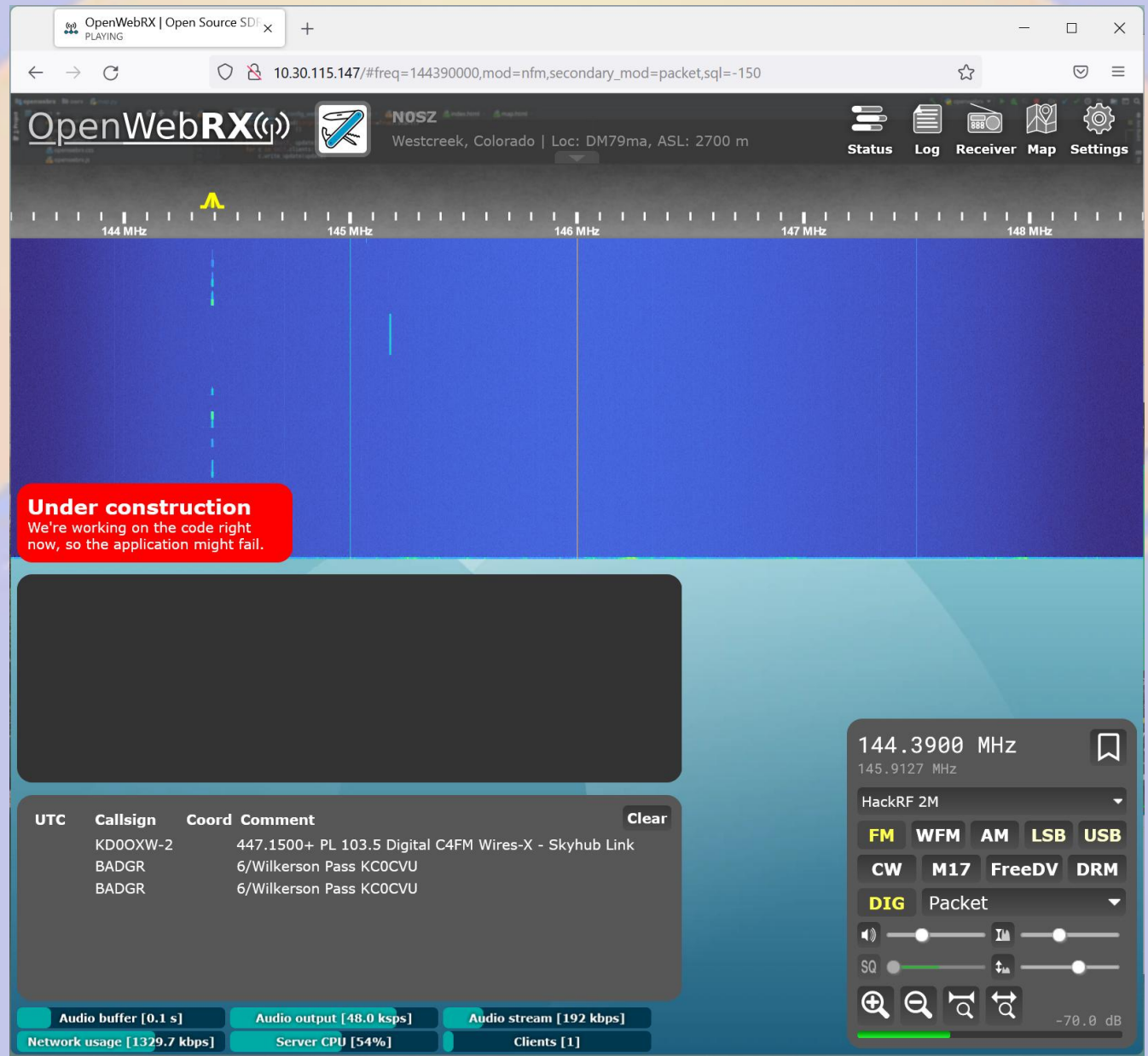
Time/Frequency



A Signal!


OpenWebRX | Open Source SDR
PLAYING
10.30.115.147/#freq=449000000,mod=nfm,sql=-150
OpenWebRX
N0SZ
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
SQ
-45.4 dB
Under construction
We're working on the code right now, so the application might fail.
Audio buffer [0.2 s]
Audio output [47.9 ksps]
Audio stream [192 kbps]
Network usage [1355.1 kbps]
Server CPU [27%]
Clients [1]

Packet


OpenWebRX | Open Source SDR
PLAYING
10.30.115.147/#freq=144390000,mod=nfm,secondary_mod=packet,sql=-150
OpenWebRX
N0SZ
Westcreek, Colorado | Loc: DM79ma, ASL: 2700 m
Status
Log
Receiver
Map
Settings
144 MHz
145 MHz
146 MHz
147 MHz
148 MHz
Under construction
We're working on the code right now, so the application might fail.
UTC
Callsign
Coord
Comment
Clear
KD0OXW-2
447.1500+ PL 103.5 Digital C4FM Wires-X - Skyhub Link
BADGR
6/Wilkerson Pass KC0CVU
BADGR
6/Wilkerson Pass KC0CVU
144.3900 MHz
145.9127 MHz
HackRF 2M
FM
WFM
AM
LSB
USB
CW
M17
FreeDV
DRM
DIG
Packet
SQ
-70.0 dB
Audio buffer [0.1 s]
Audio output [48.0 ksps]
Audio stream [192 kbps]
Network usage [1329.7 kbps]
Server CPU [54%]
Clients [1]

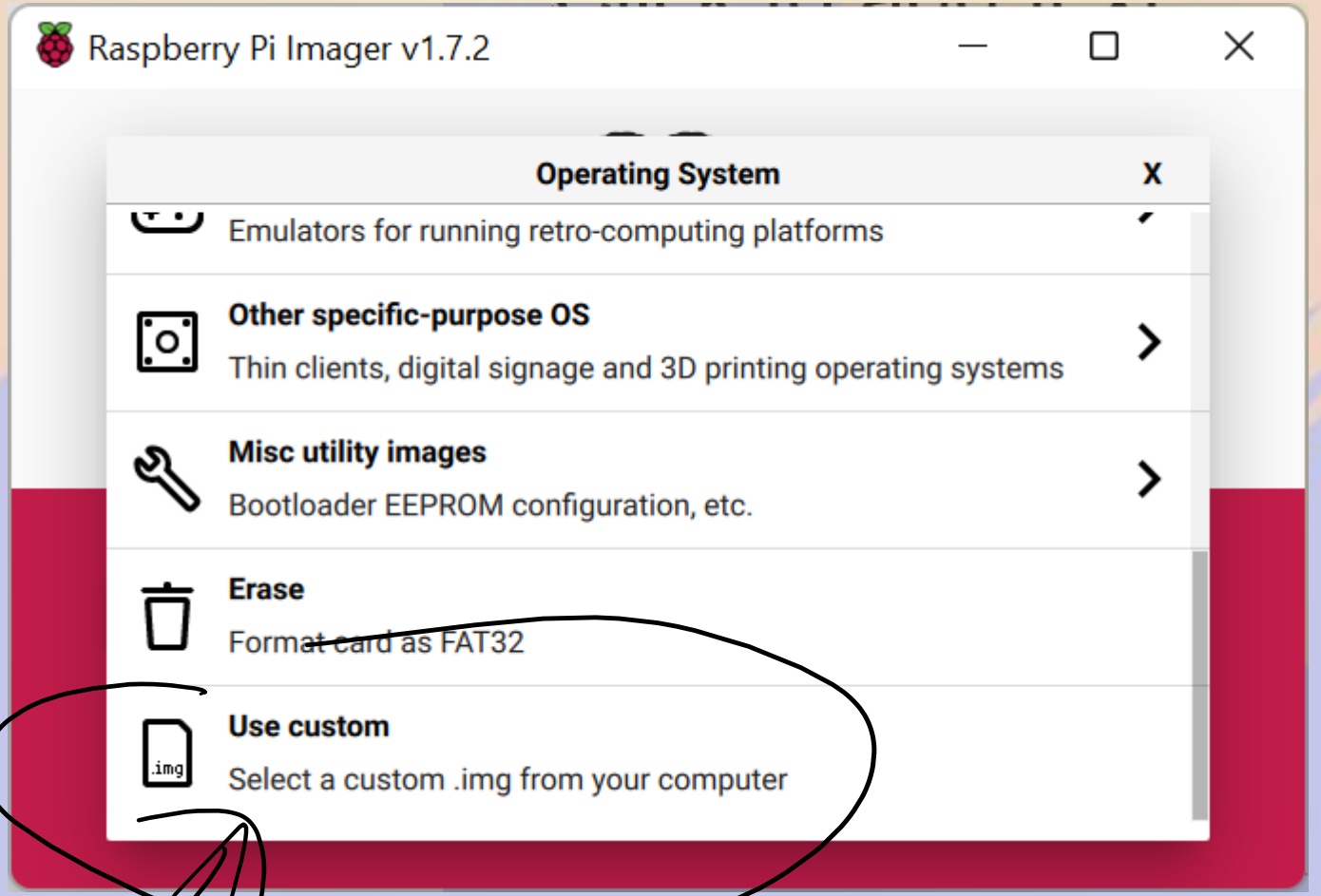
Neat, so how do I get this?

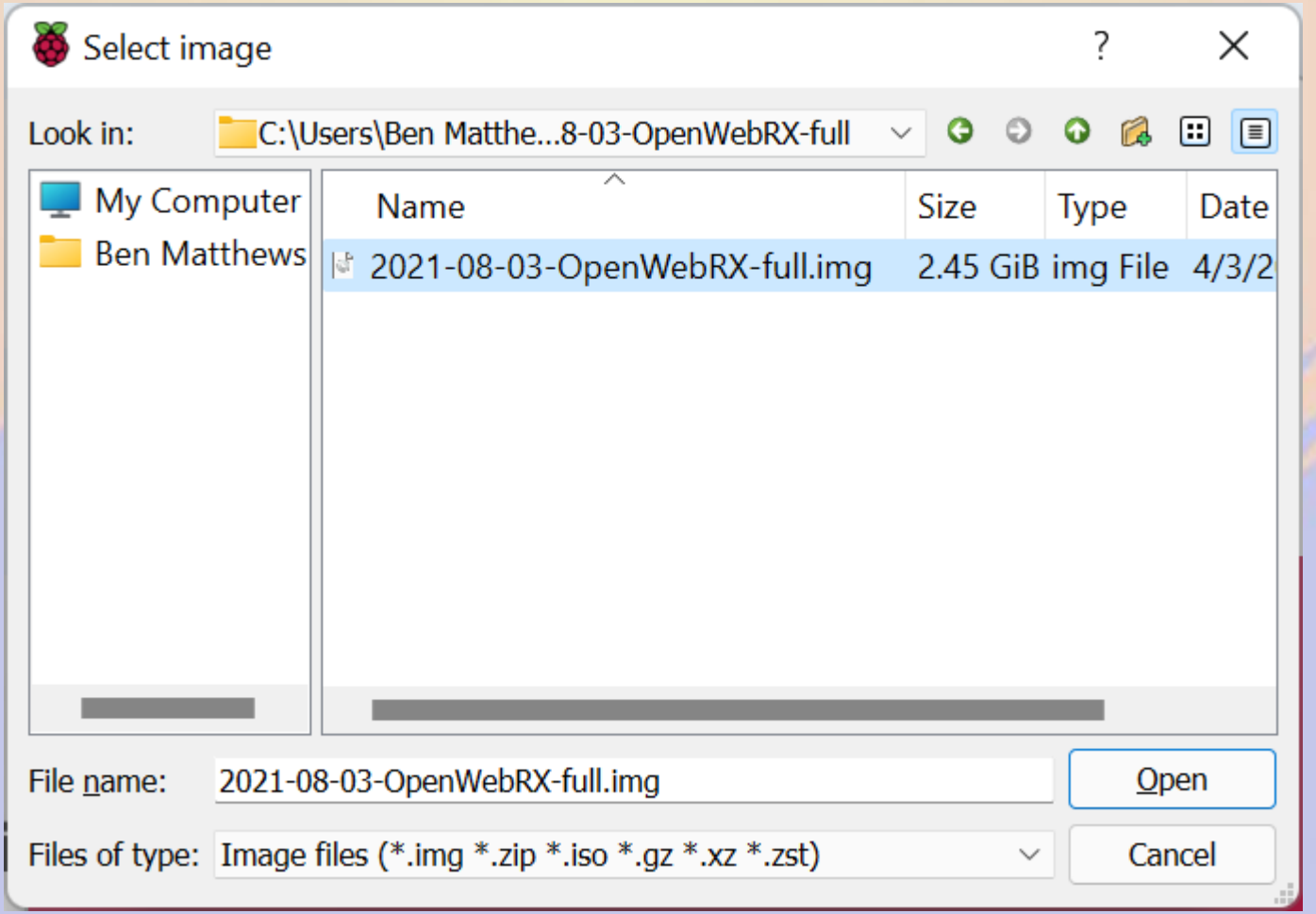
- Well, you could cheat. There are a bunch of OpenWebRX instances publicly available on the internet… but that's no fun
- Let's go Shopping:
  - Pi4
  - RTLsdr
    - or something better if you want, but I'm going to stick with RTLsdr because they're cheap
  - Something that'll work as an antenna for 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

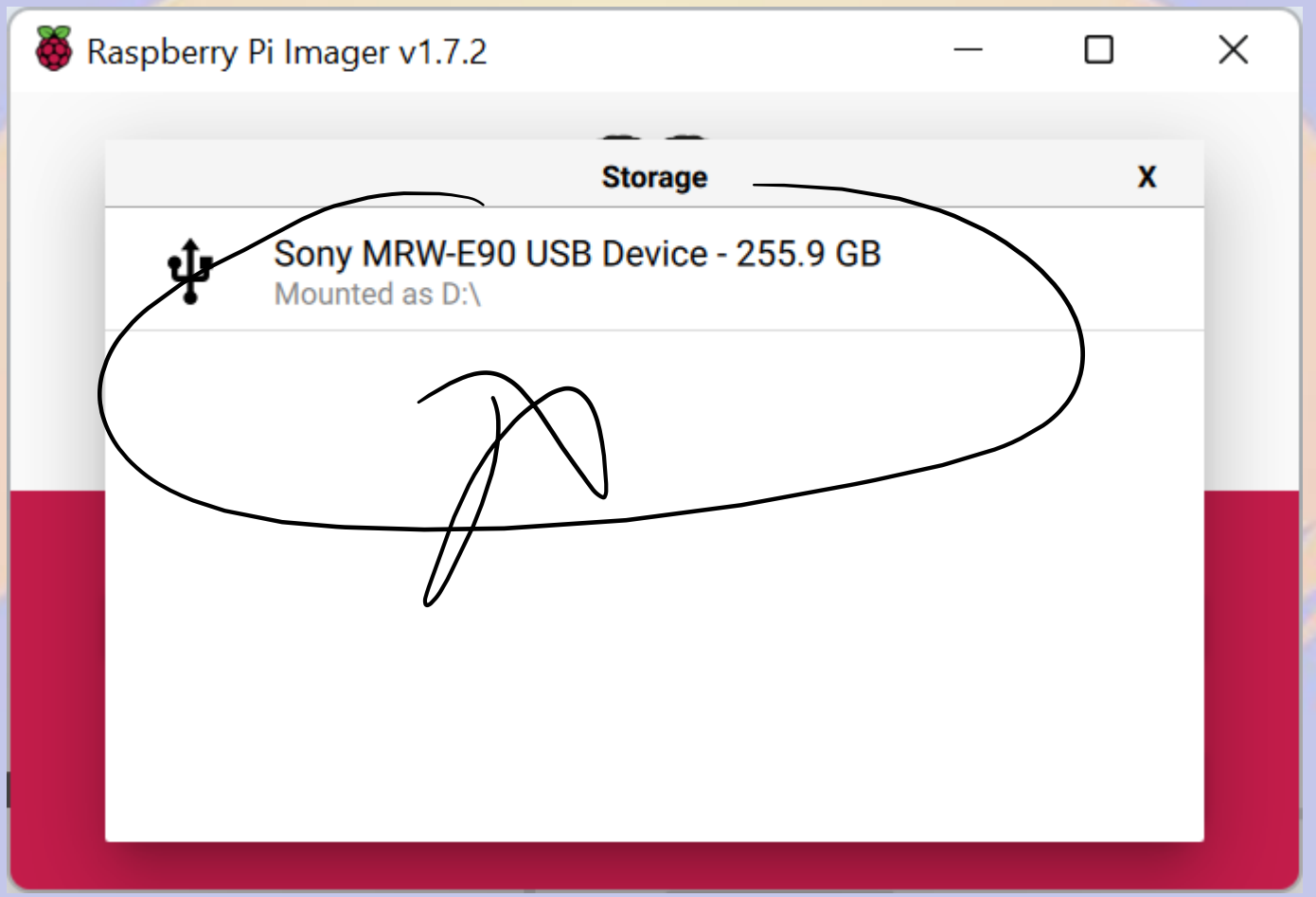
- Grab the latest image and unzip it:
- https://www.openwebrx.de/download/rpi.php
- (I hate prebuilt images, so I'll put some notes on how to install manually at the end of the deck for the Linux gurus)
- Install the Raspberry Pi imager tool from here:
- https://www.raspberrypi.com/software/
- (If you've 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
.img

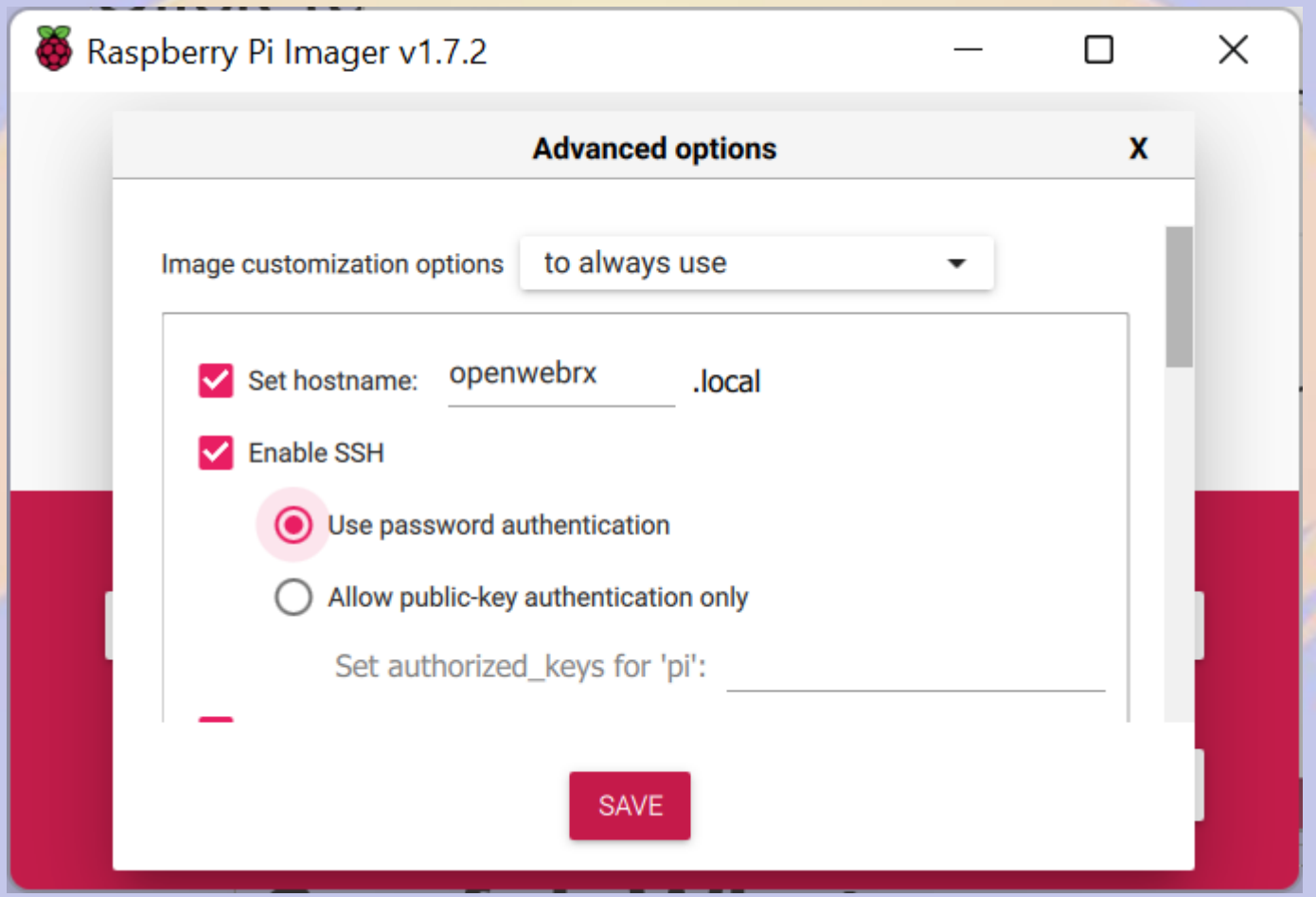

Select image
Look in:
C:\Users\Ben Matthe...8-03-OpenWebRX-full
My Computer
Ben Matthews
Name
Size
Type
Date
2021-08-03-OpenWebRX-full.img
2.45 GiB
img File
4/3/2
File name:
2021-08-03-OpenWebRX-full.img
Open
Files of type:
Image files (*.img *.zip *.iso *.gz *.xz *.zst)
Cancel

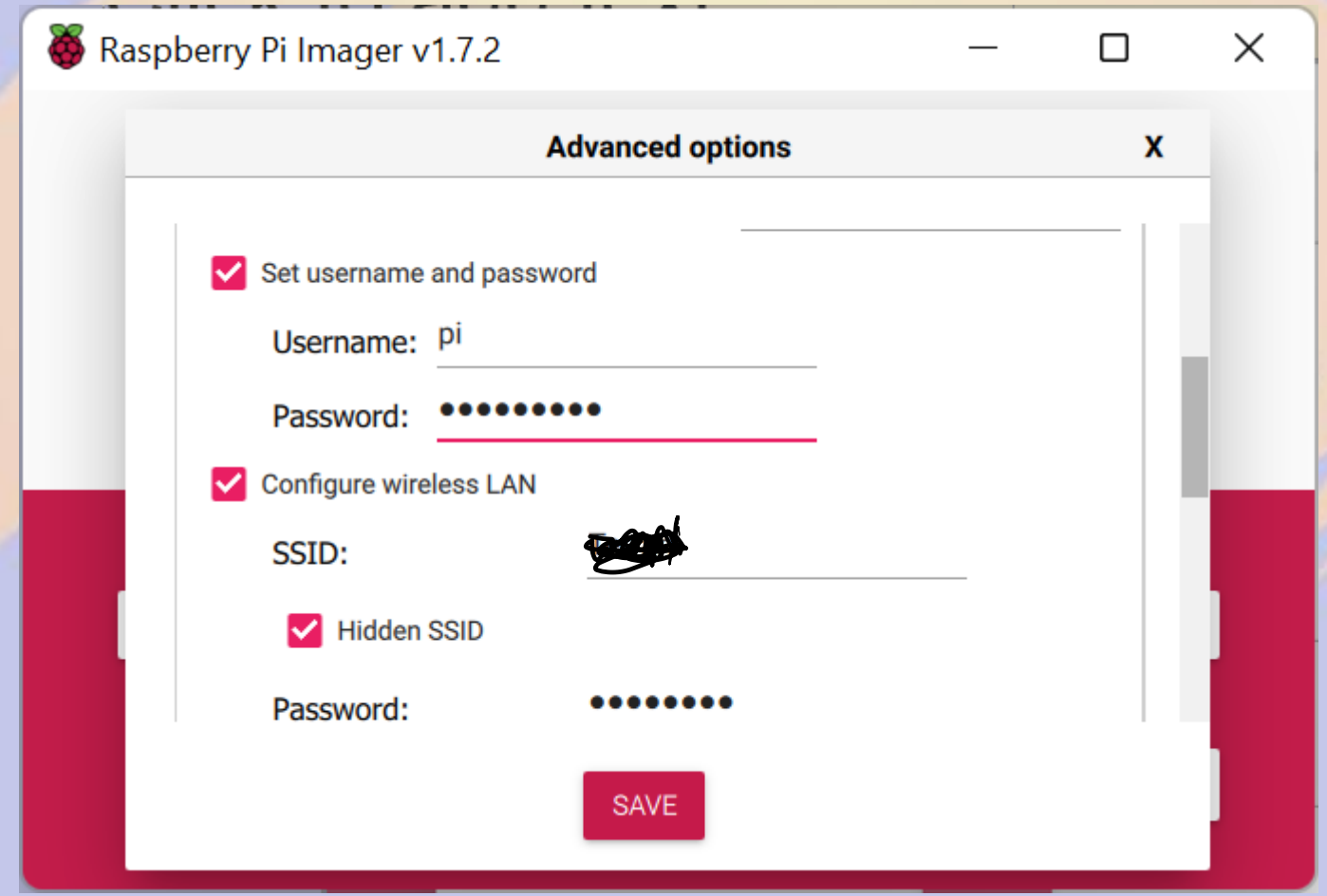

Raspberry Pi Imager v1.7.2
Raspberry Pi
Operating System
2021-08-03-OPENWEBRX-FULL.IMG
Storage
CHOOSE STOR...
WRITE

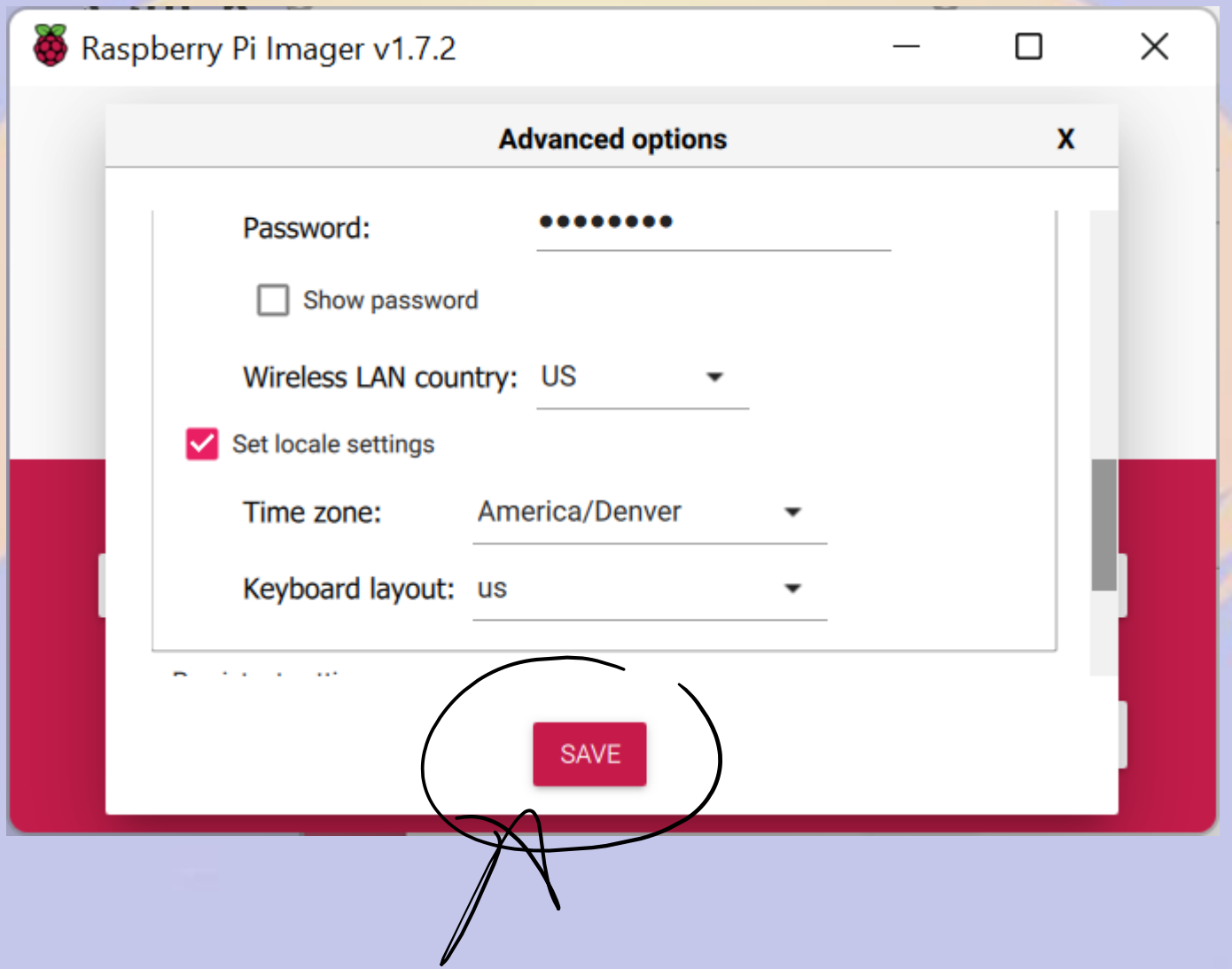


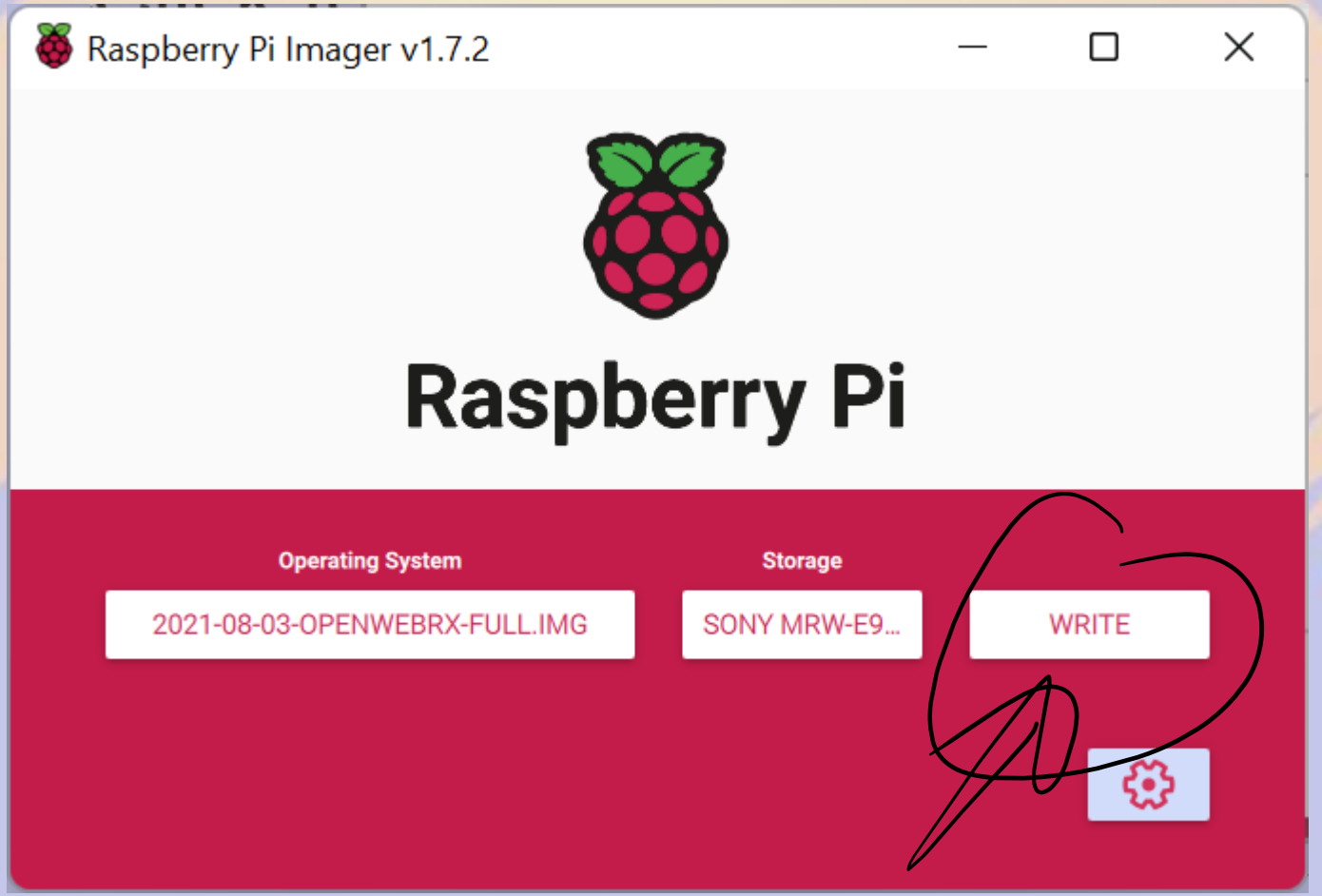
Careful: Whatever you choose here will be erased!

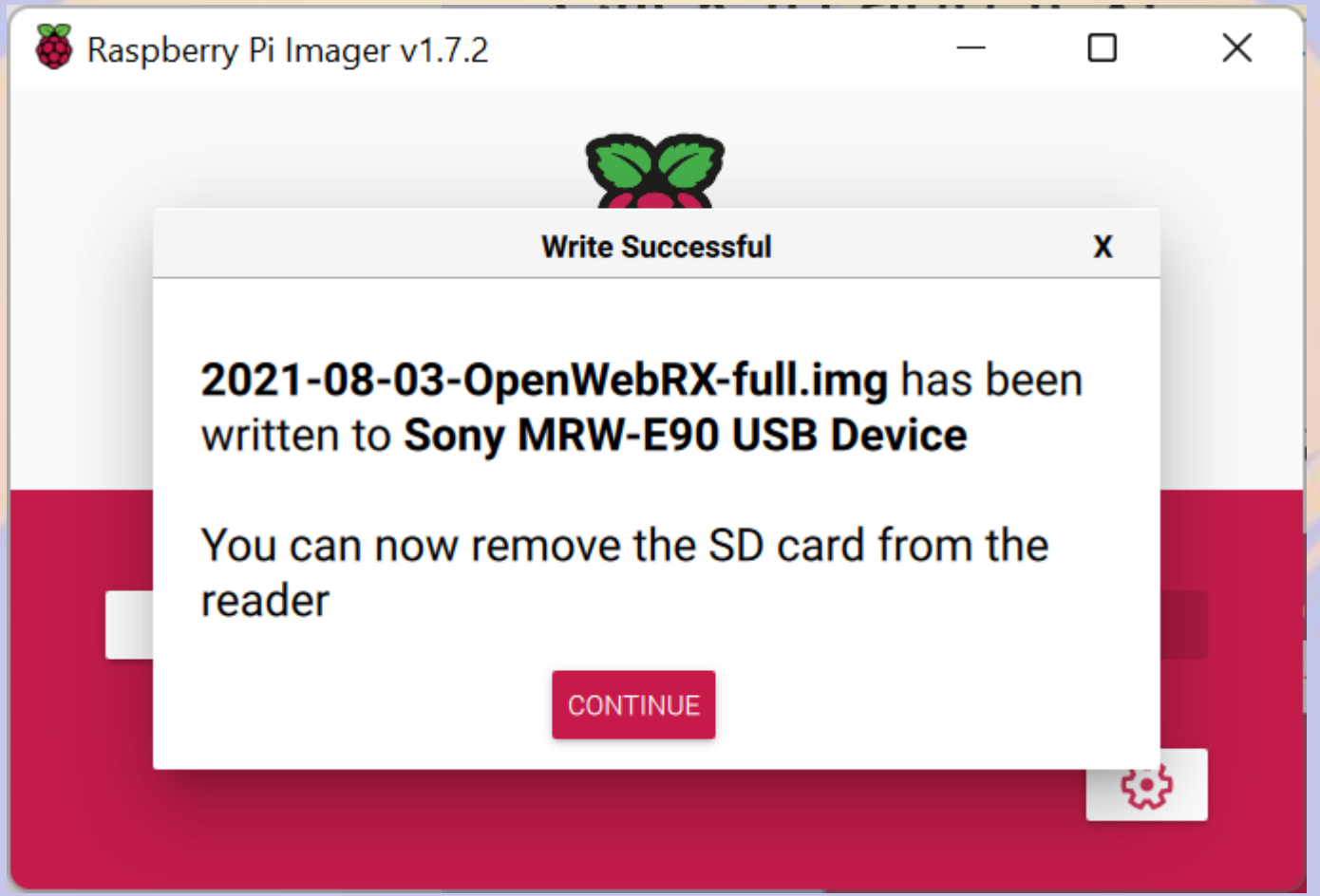

Raspberry Pi Imager v1.7.2
Raspberry Pi
Operating System
Storage
2021-08-03-OPENWEBRX-FULL.IMG
SONY MRW-E9...
WRITE


Raspberry Pi Imager v1.7.2
Advanced options
X
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
X
Set username and password
Username: pi
Password:
Configure wireless LAN
SSID:
Hidden SSID
Password:
SAVE


Raspberry Pi Imager v1.7.2
Advanced options
X
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
X
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

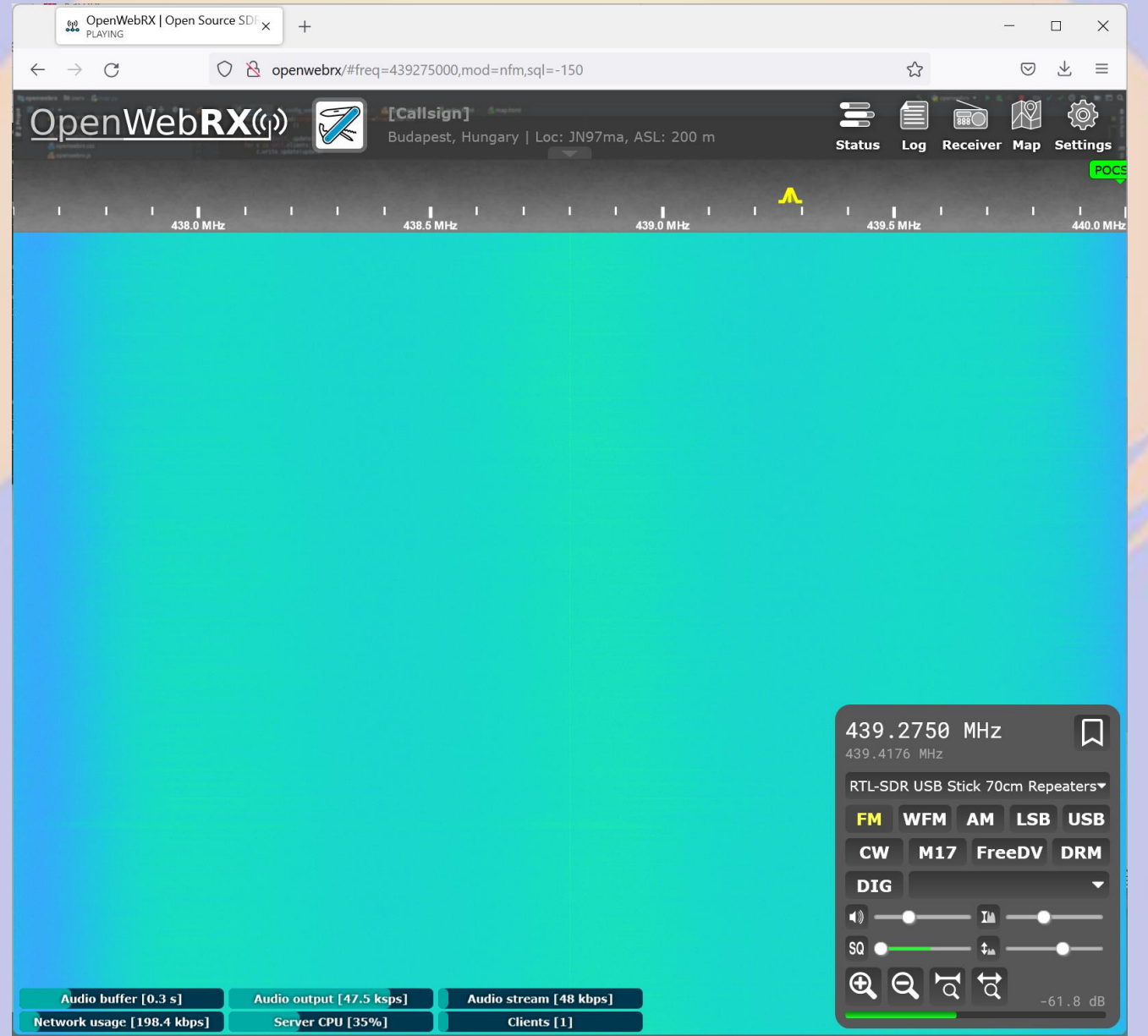
Success?

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

Connect to OpenWebRx

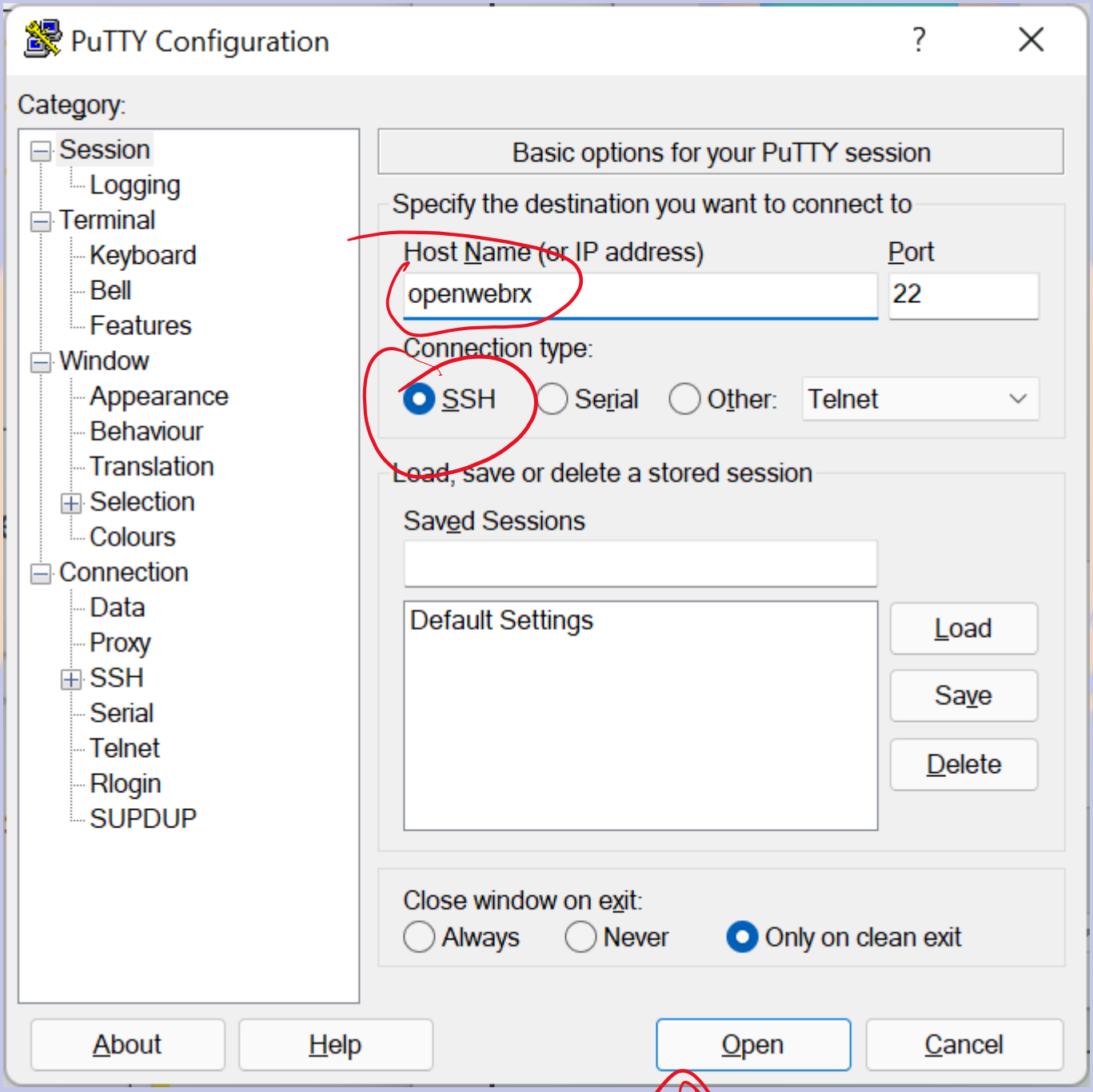
- If you're exceedingly lucky, you can just point a browser at http://openwebrx
- If not, you'll need to find the IP of your PI and point a browser at that
- Maybe you can ask your router
- If all else fails, you can connect an HDMI monitor to the pi and log in with the account you configured back a 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



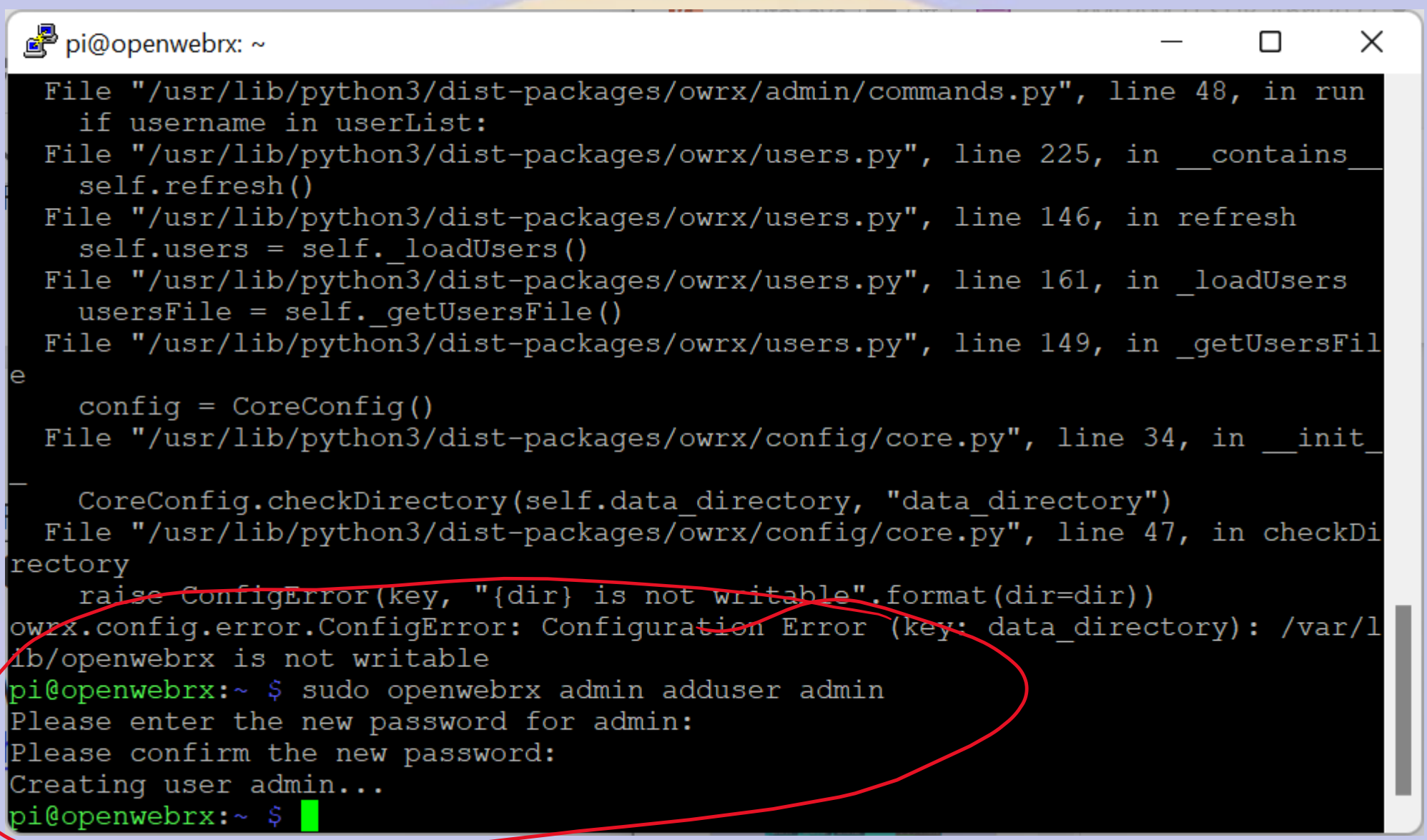
Let’s add an account to configure with

- 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

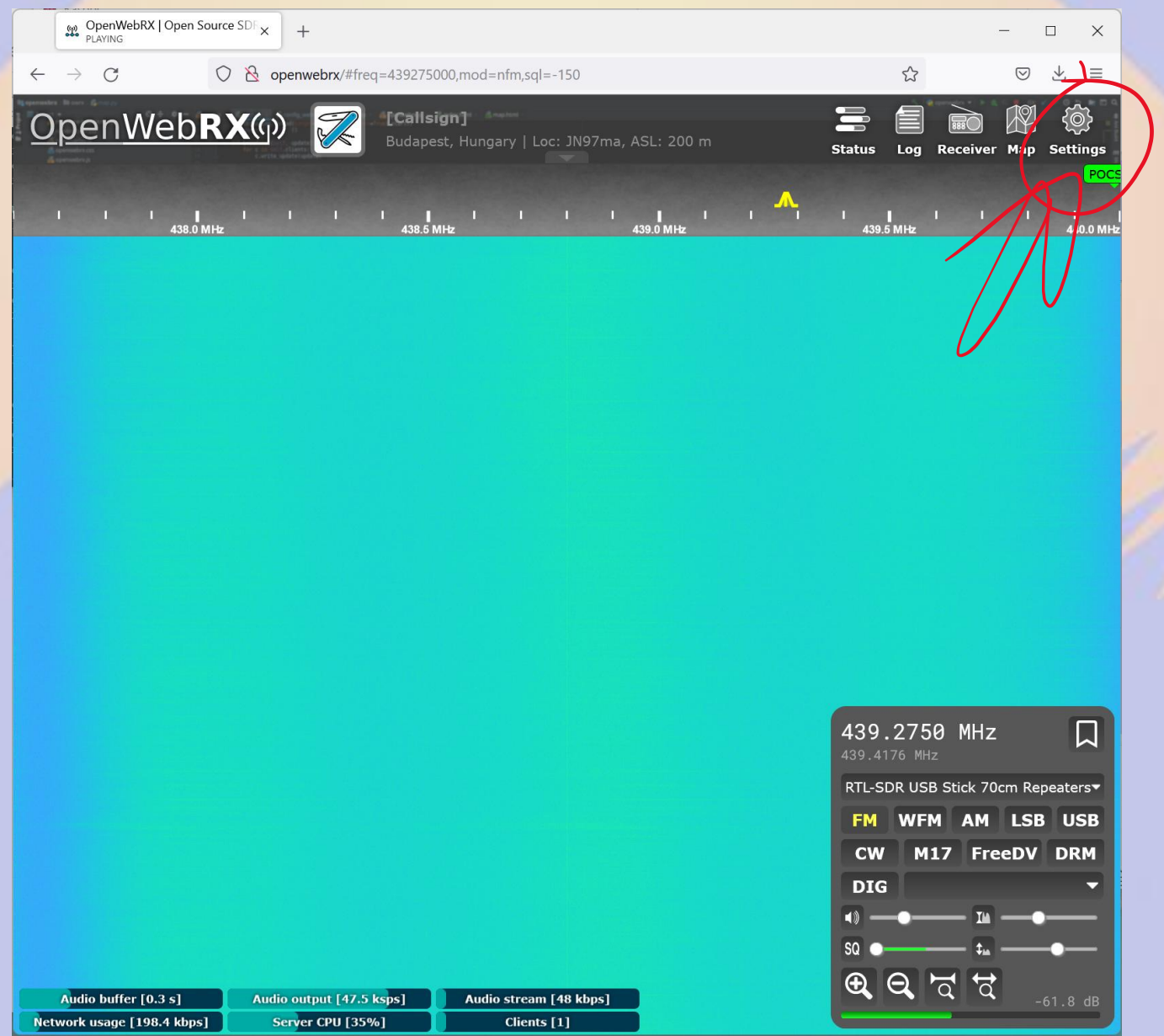

PuTTY Configuration
?
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Serial
Telnet
Rlogin
SUPDUP
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
openwebrx
22
Connection type:
SSH
Serial
Other:
Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel

Adding a user

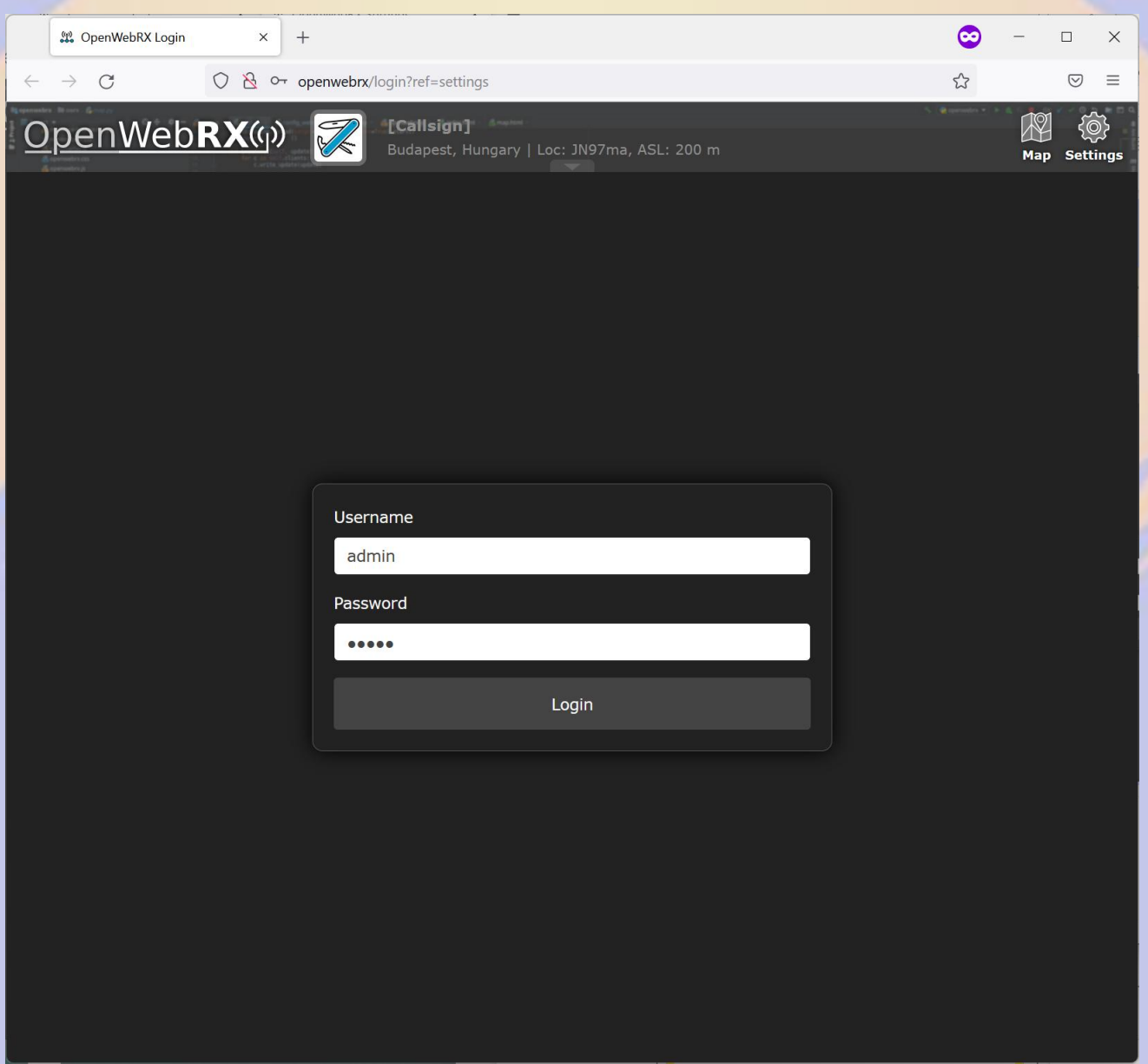
- Once logged-in, do:
- `sudo openwebrx admin adduser admin`
- You'll be prompted for a password, you can use whatever (but you'll need it for the next step)

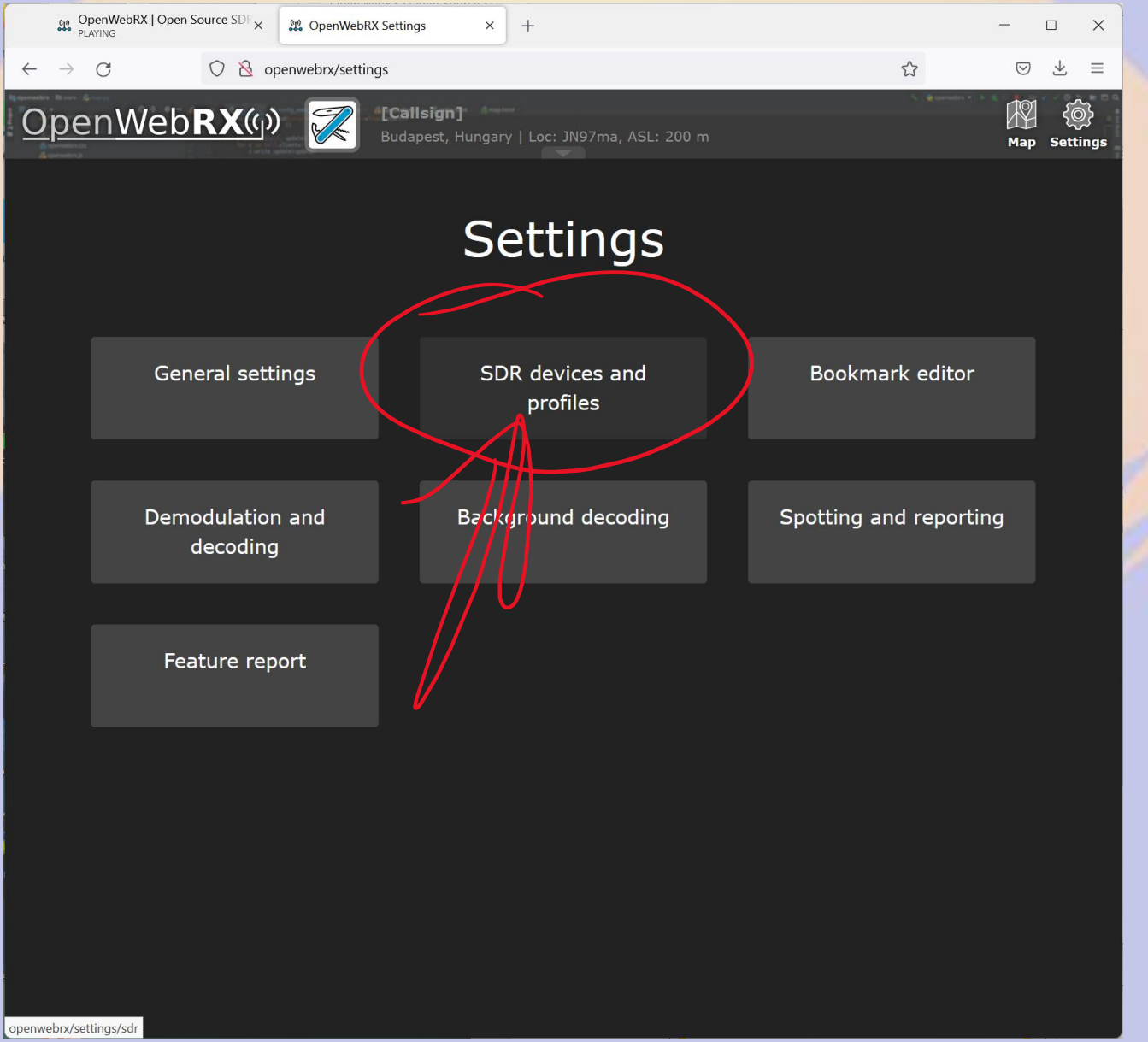

pi@openwebrx: ~
File "/usr/lib/python3/dist-packages/owrx/admin/commands.py", line 48, in run
if username in userList:
File "/usr/lib/python3/dist-packages/owrx/users.py", line 225, in __contains__
self.refresh()
File "/usr/lib/python3/dist-packages/owrx/users.py", line 146, in refresh
self.users = self._loadUsers()
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

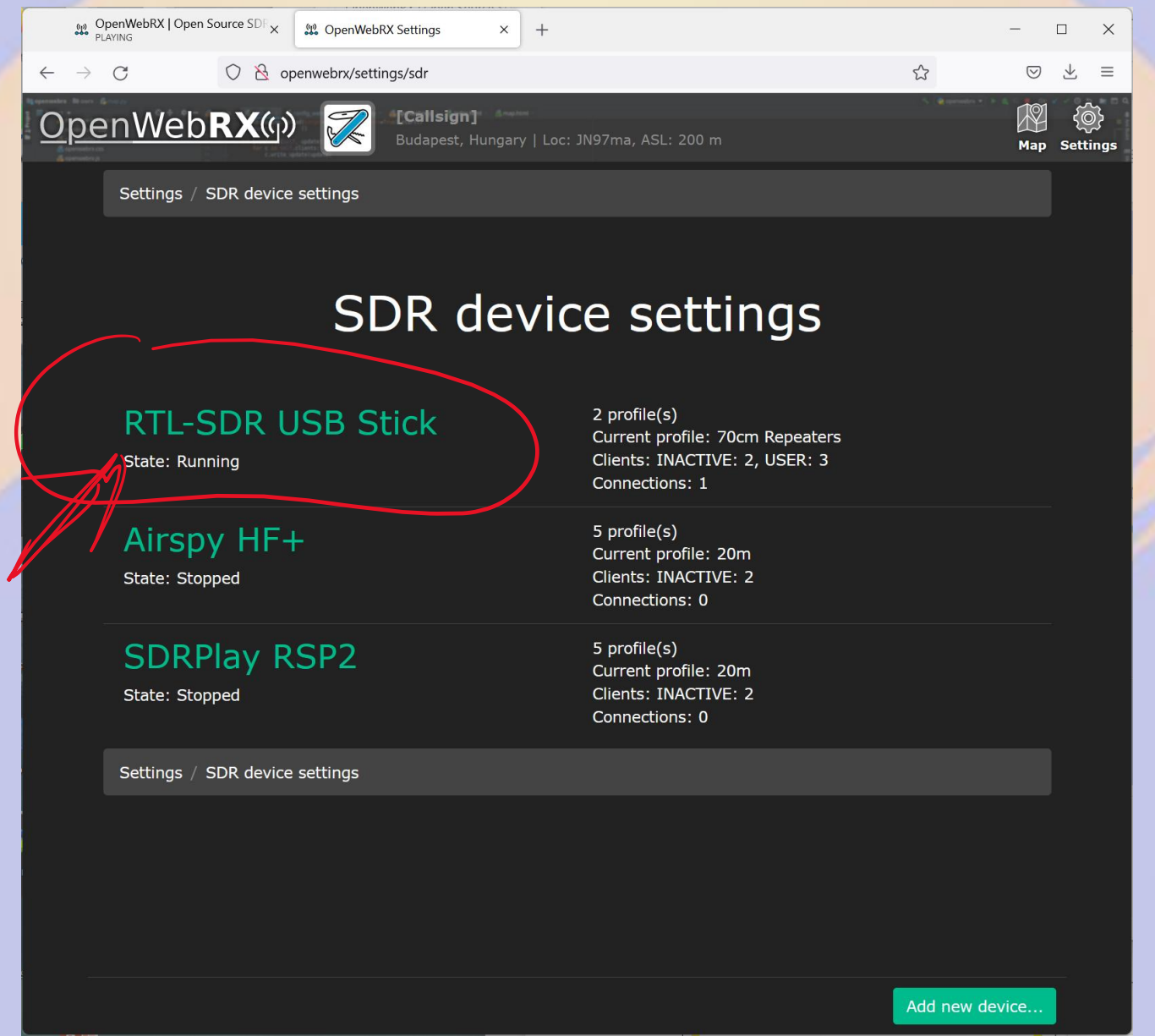


Configuration

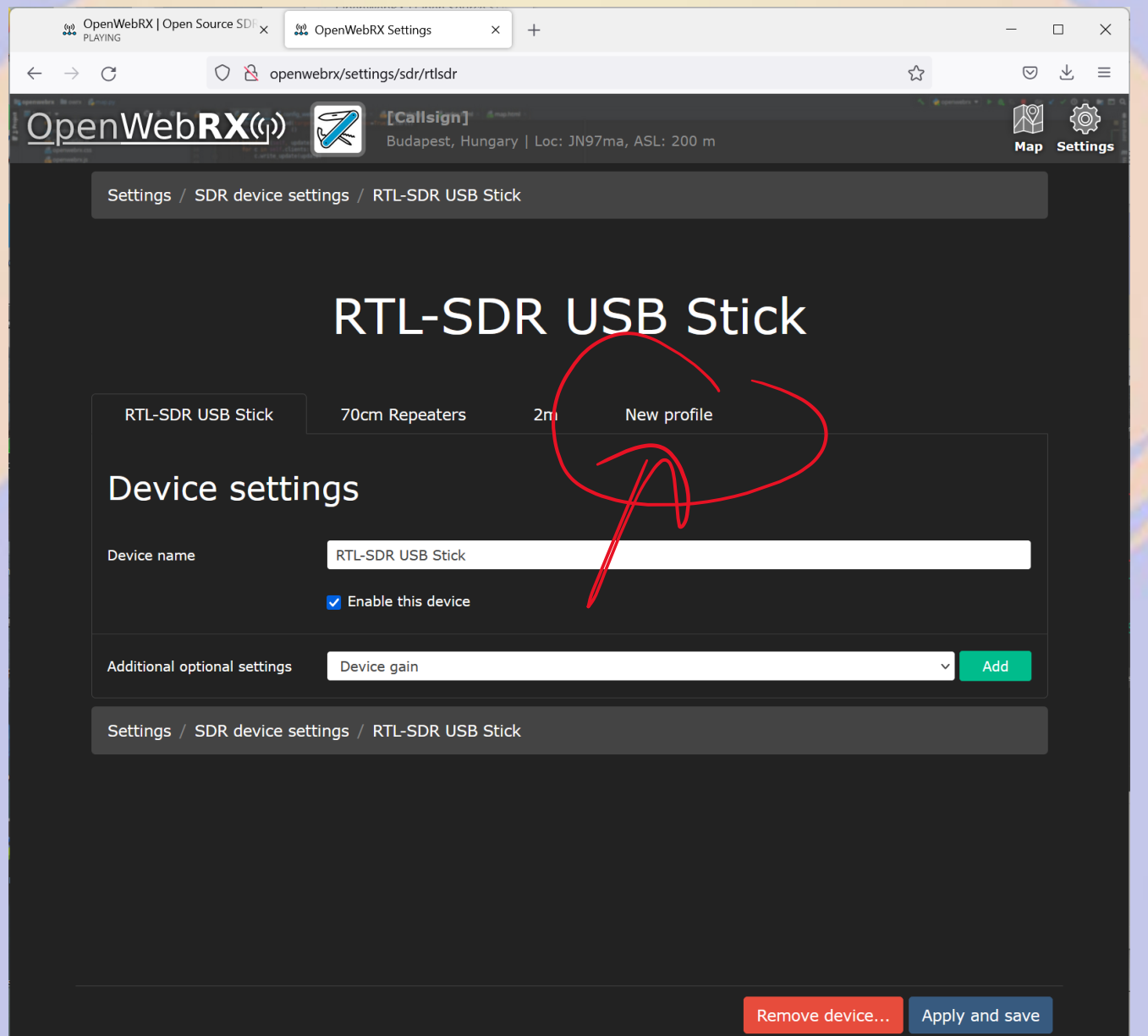


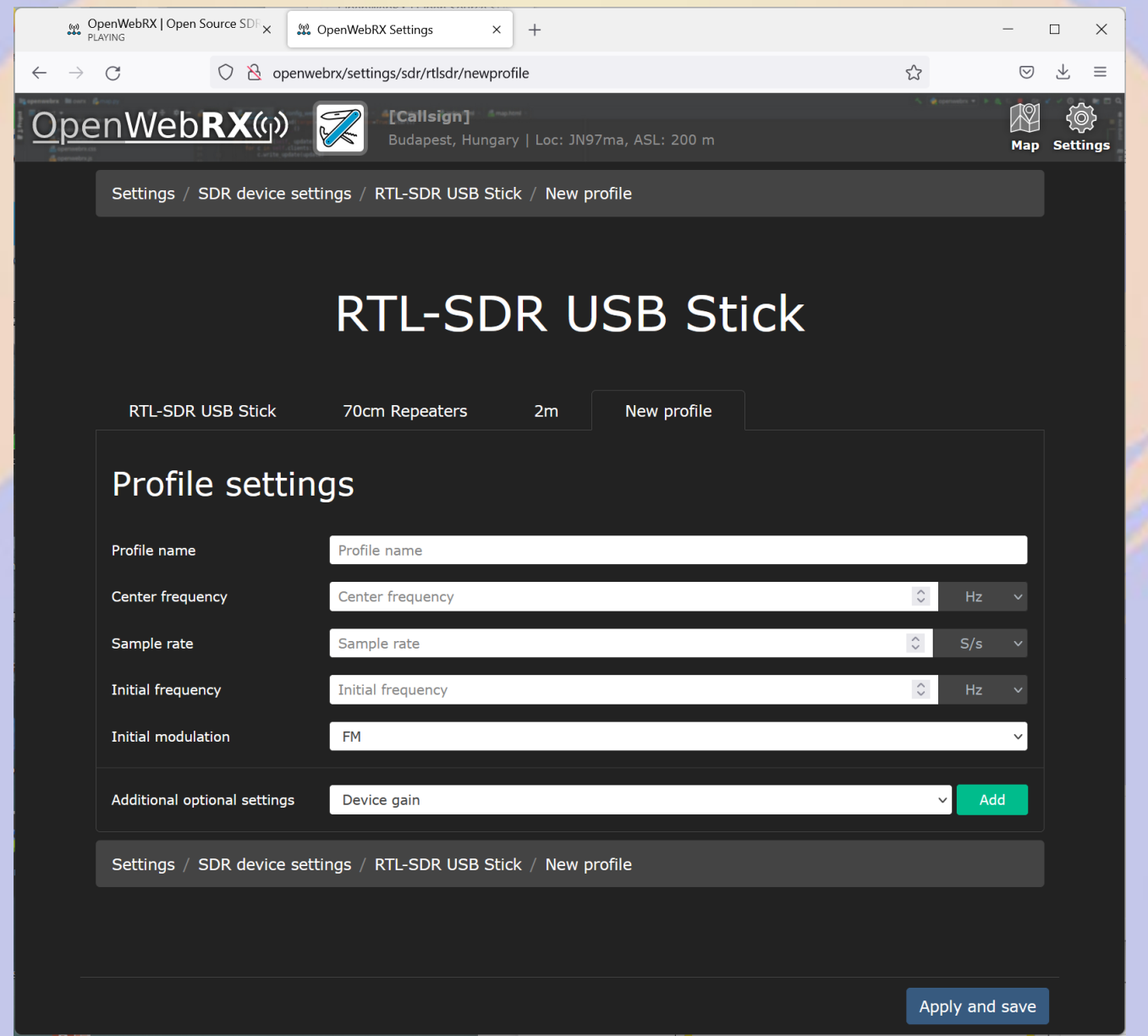

OpenWebRX | Open Source SDR
PLAYING
OpenWebRX Settings
openwebrx/settings
OpenWebRX
[Callsign]
Budapest, Hungary | Loc: JN97ma, ASL: 200 m
Map
Settings
Settings
General settings
SDR devices and profiles
Bookmark editor
Demodulation and decoding
Background decoding
Spotting and reporting
Feature report
openwebrx/settings/sdr

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



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

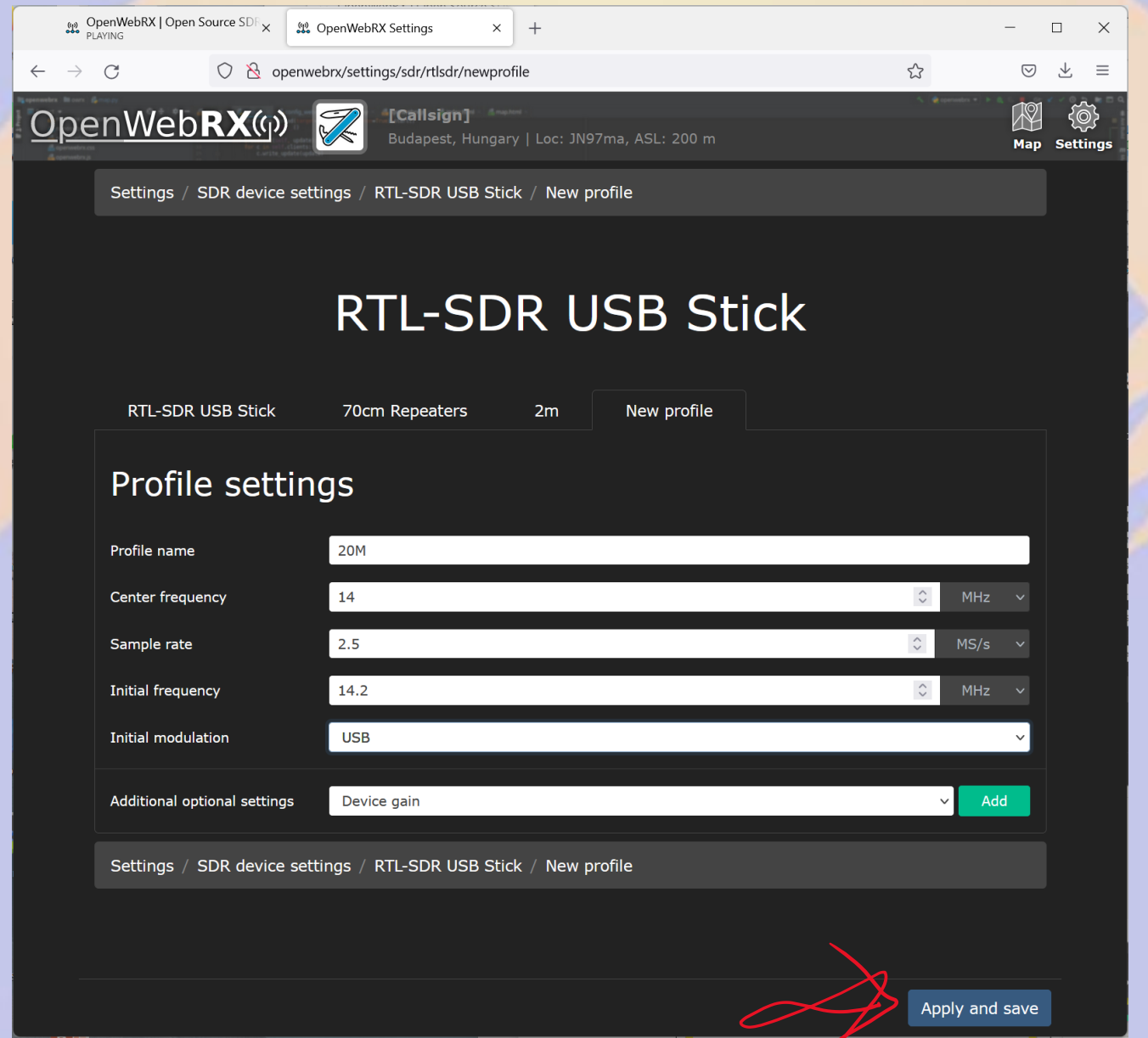


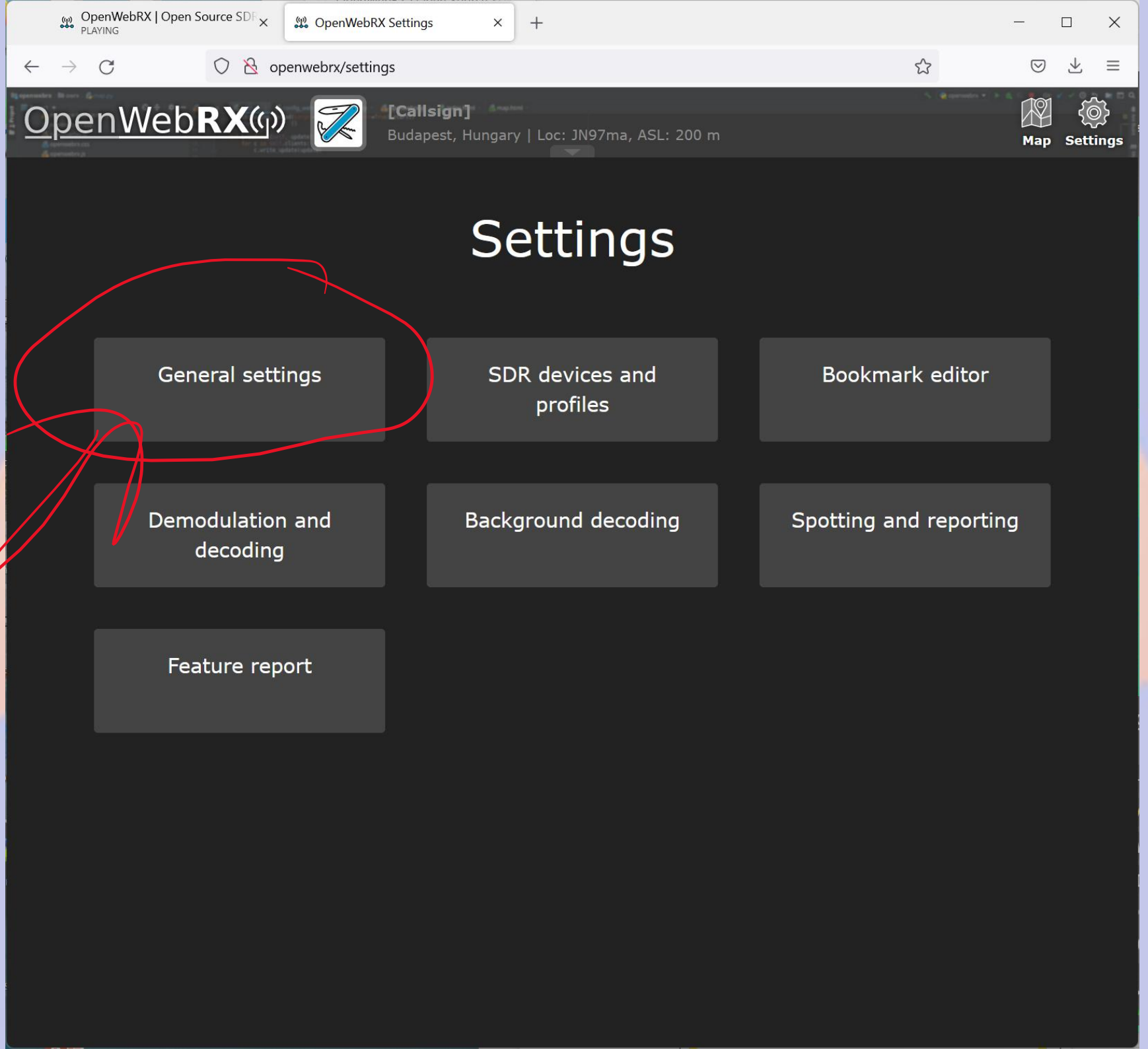

OpenWebRX | Open Source SDR
PLAYING
OpenWebRX Settings
openwebrx/settings/sdr/rtlsdr/newprofile
OpenWebRX
[Callsign]
Budapest, Hungary | Loc: JN97ma, ASL: 200 m
Map
Settings
Settings / SDR device settings / RTL-SDR USB Stick / New profile
RTL-SDR USB Stick
RTL-SDR USB Stick
70cm Repeaters
2m
New profile
Profile settings
Profile name
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.

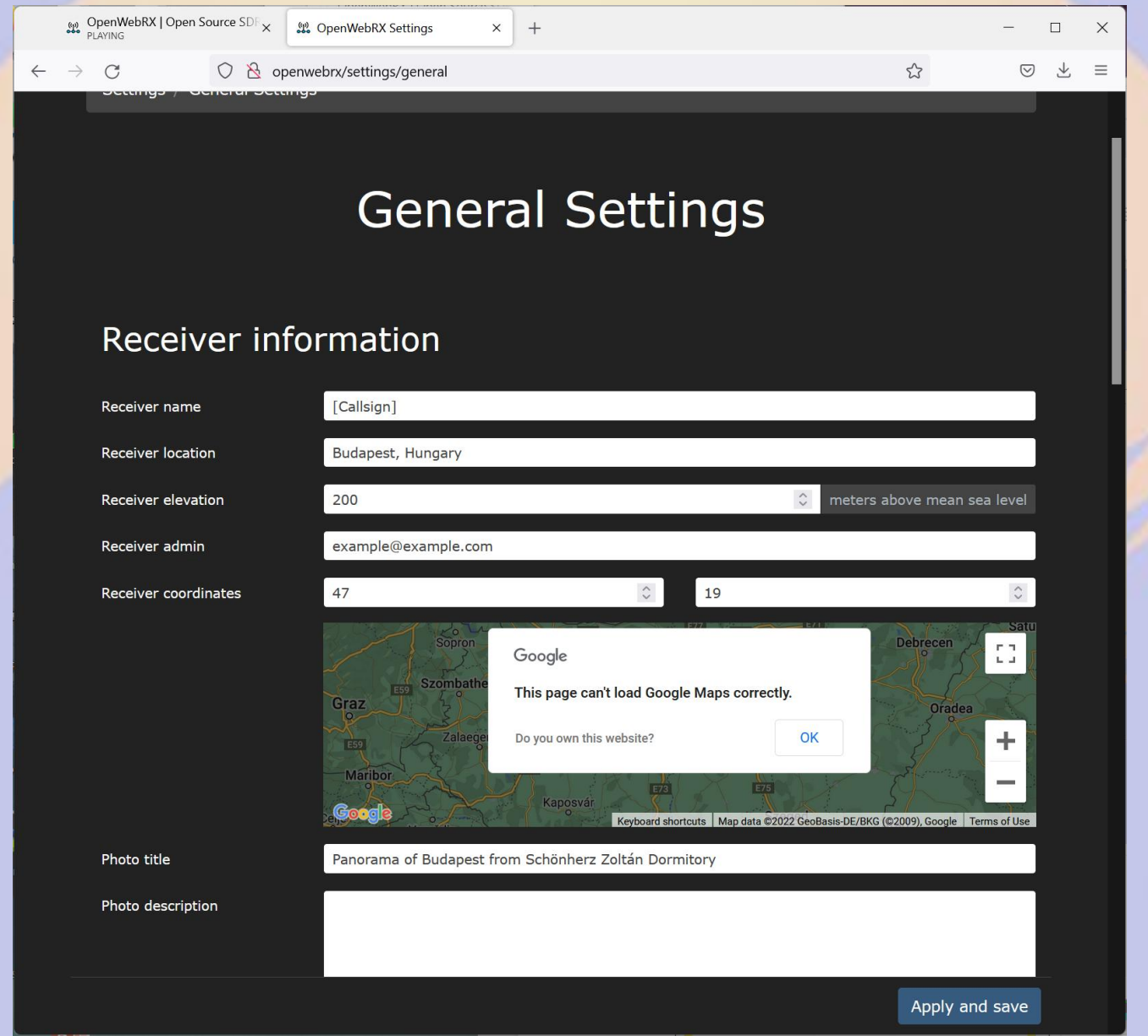
- Initial Frequency: The interface will display a chunk of spectrum based on your sample rate, but you probably only want to listen to a few KHz. This is the default dial frequency where you're listening
- Initial Modulation: Set this based on the band segment you're configuring. FM for repeaters, USB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

Example:

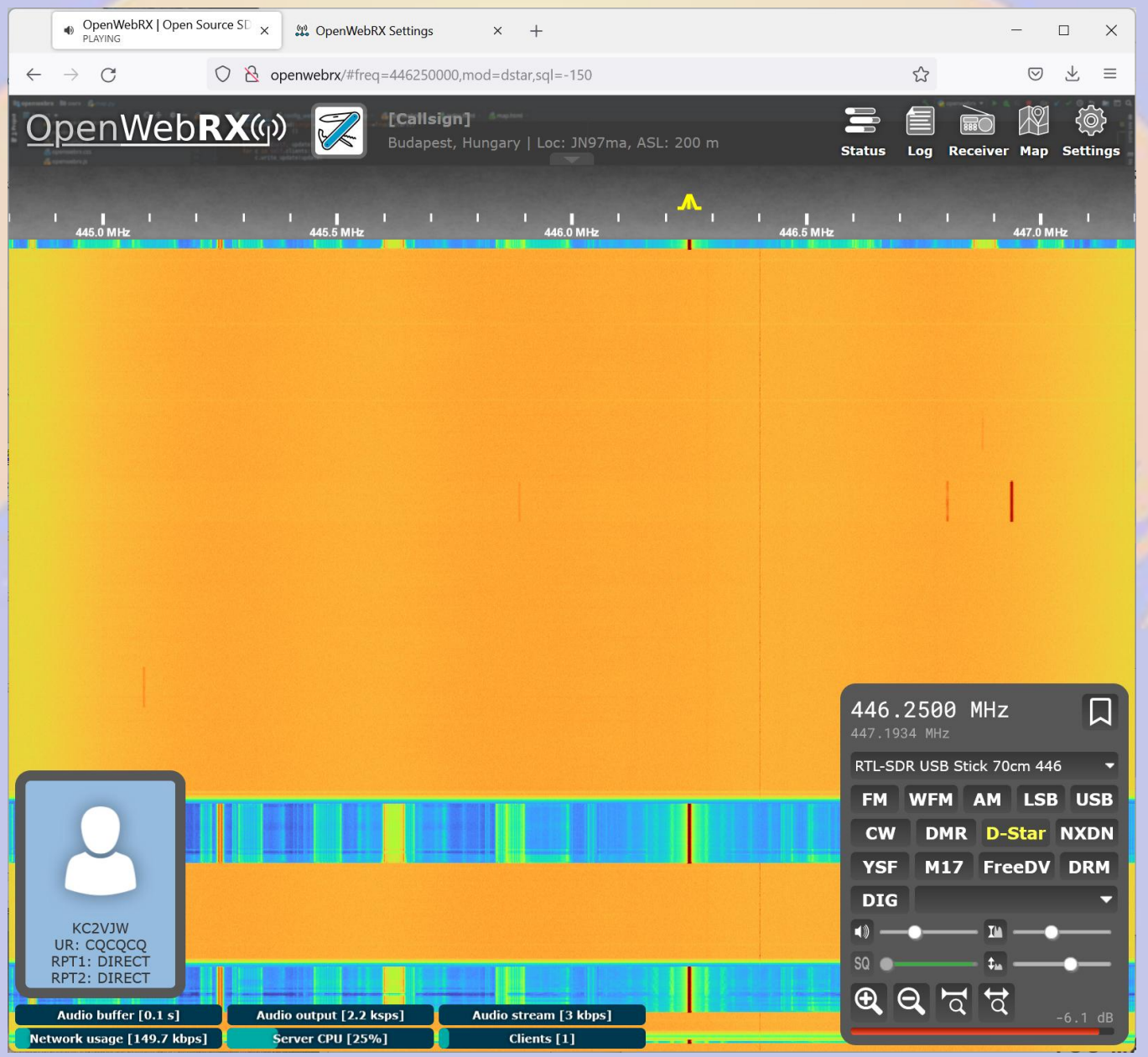

OpenWebRX | Open Source SDR
PLAYING
OpenWebRX Settings
openwebrx/settings/sdr/rtlsdr/newprofile
OpenWebRX
[Callsign]
Budapest, Hungary | Loc: JN97ma, ASL: 200 m
Map
Settings
Settings / SDR device settings / RTL-SDR USB Stick / New profile
RTL-SDR USB Stick
RTL-SDR USB Stick
70cm Repeaters
2m
New profile
Profile settings
Profile name
20M
Center frequency
14
MHz
Sample rate
2.5
MS/s
Initial frequency
14.2
MHz
Initial modulation
USB
Additional optional settings
Device gain
Add
Settings / SDR device settings / RTL-SDR USB Stick / New profile
Apply and save


OpenWebRX | Open Source SDR
PLAYING
OpenWebRX Settings
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



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)

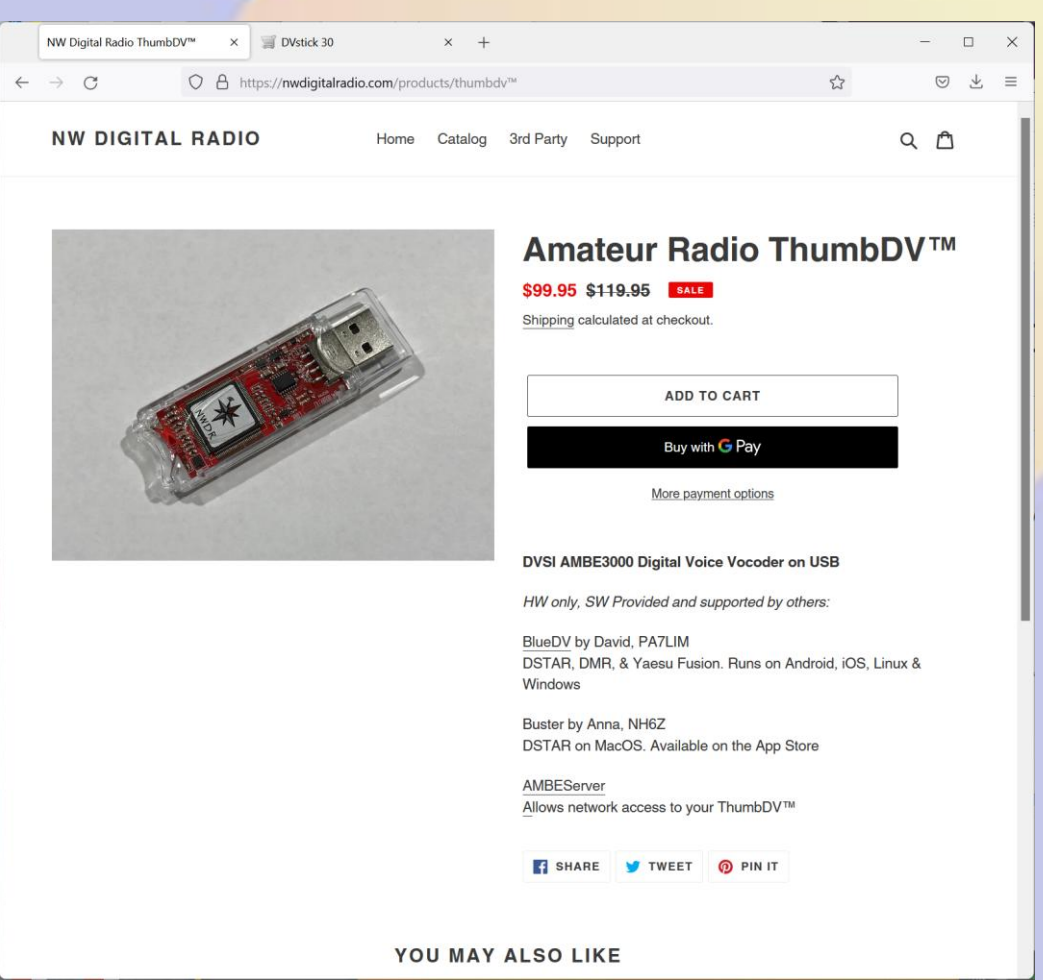
AMBE?

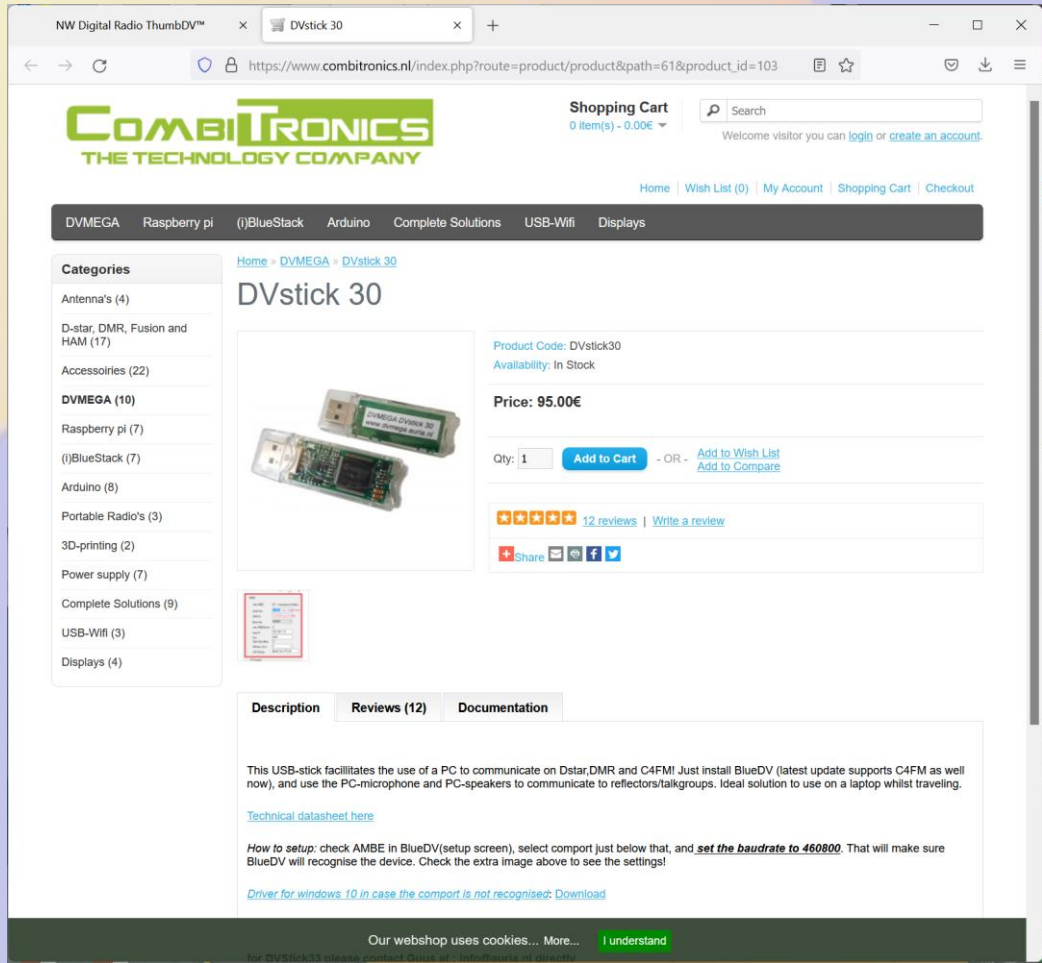
- How do we stuff ~8kHz of audio into 1.2kbits/sec?
- Compress it to within an inch of its life
- AMBE is a proprietary way of doing this
  - I mentioned FreeDV earlier – it uses something called Codec2 to do the same, for free! (but you can't really buy a radio to do that)
- There are a couple of variants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





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/codecserver.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 openwebrx" 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/codecserver-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

Questions?

Live Demo?

ben@kc2vjw.com

Building From Source

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