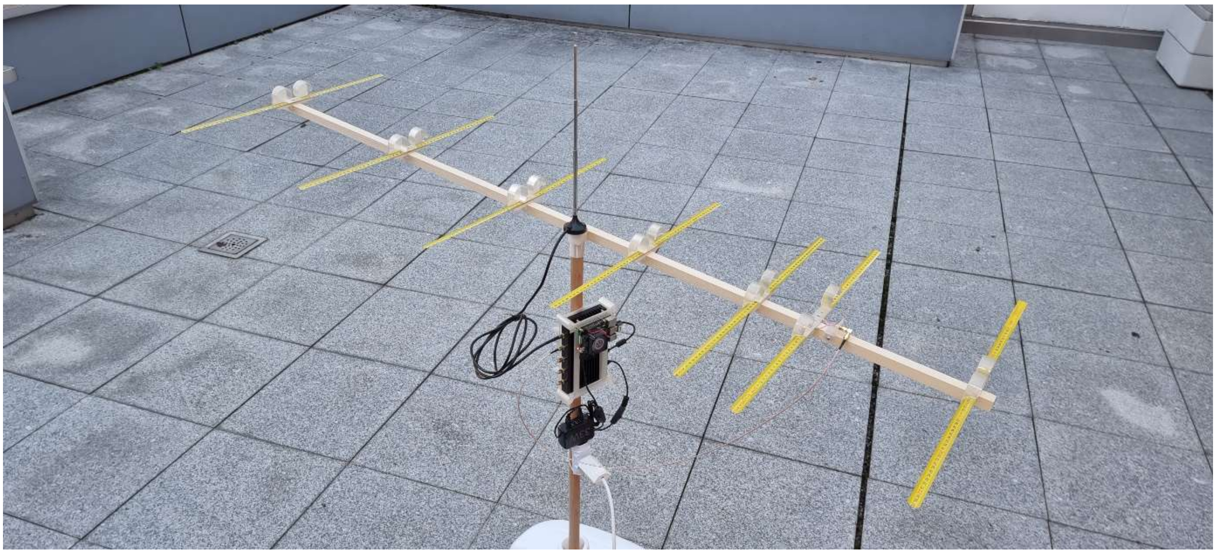
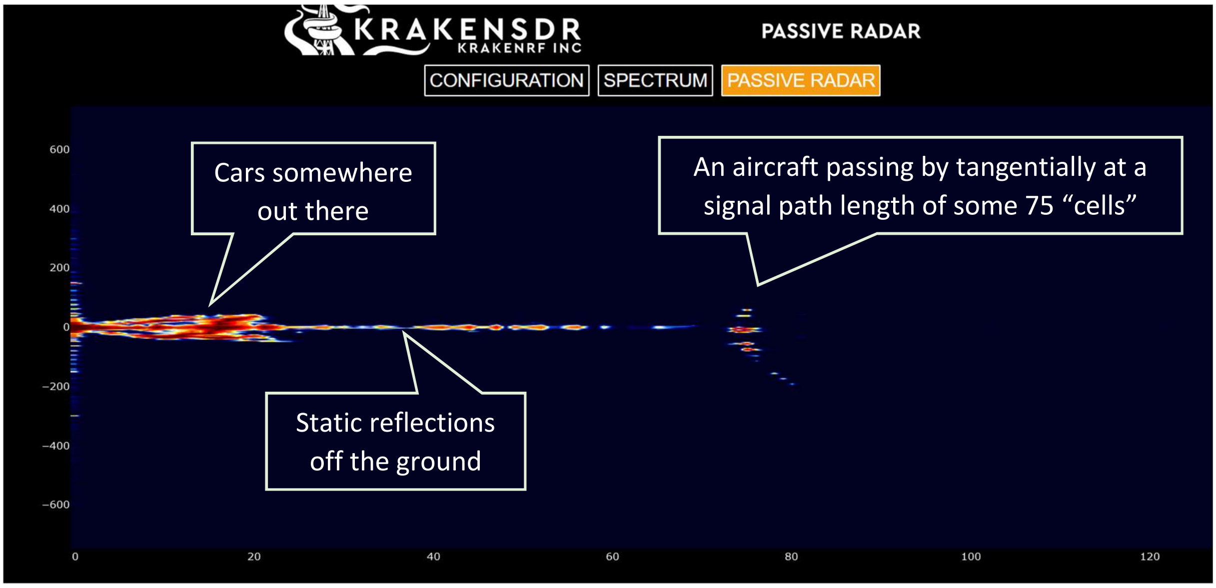
Installing and Operating the KrakenSDR with Raspberry Pi 4 as a Passive Radar Receiver

Document Version 1.9, 25 October 2022
Valid for KrakenSDR Pi 4 Image File V 1.3



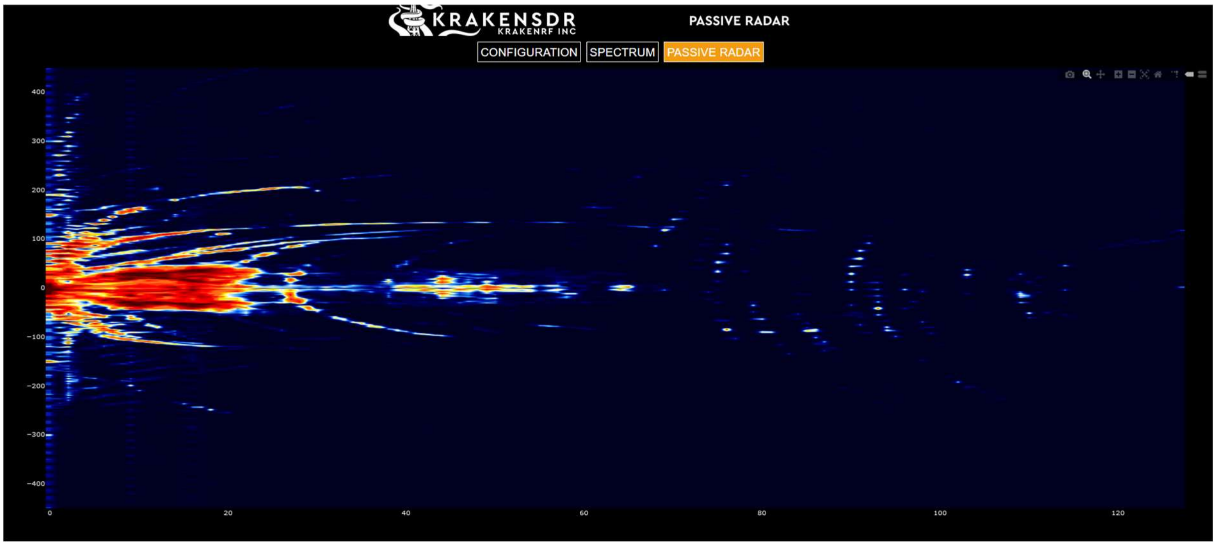
This is my current KrakenSDR passive radar setup: The KrakenSDR and a well-cooled Pi 4 mounted inside my 3D-printed frame and attached to a wooden 25 mm pole, together with a 33 W USB power supply. On top of the pole a $\lambda/4$ rod antenna connected to KrakenSDR's channel 0. Attached to the pole a self-made 227.4 MHz Yagi connected to KrakenSDR's channel 1.

This is what you might expect to observe if you operate such a setup at the frequency of a powerful (several kW) HDTV or DAB transmitter in the vicinity (a few km away):

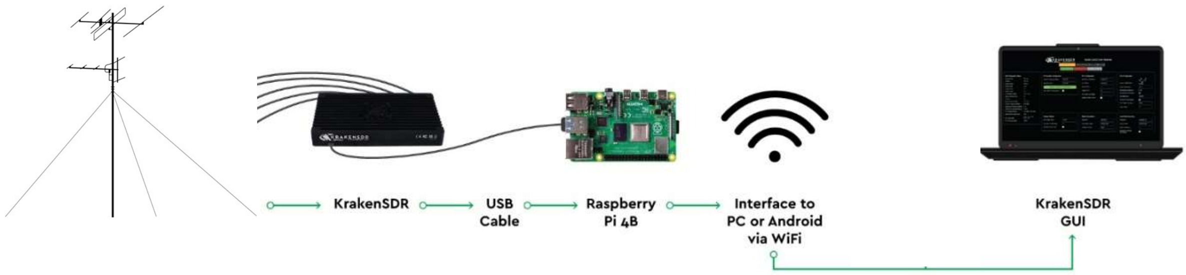


“My” transmitter is a 35 kW DAB radio station some 5 km away.

By activating the “PR Persist” option with a “Persist Decay” parameter value of 1.0 (= no decay) you may see – over time – many more aircraft:



The ellipse segments are traces of aircraft passing in a straight course at various radial speeds and bistatic (= signal path) distances. With my current setup of DAB radio transmitter position and power, its radiation characteristics, my antenna, KrakenSDR parameter settings etc. I am able to detect aircraft at up to about 100 km bistatic distance or 50 km effective range.



Intro

This is a step-by-step tutorial based on my own practical experience. It is addressed to KrakenSDR users with very limited pre-existing knowledge about the KrakenSDR, Raspberry Pi and its operating system. However, you should be familiar with your PC (Windows, Mac, Linux) and optionally your mobile, with basic WiFi setup procedures and basic radio frequency (RF) principles. You may have read about basic passive radar principles on the KrakenRF website including its continuative links or by self-study of other sources (searching on YouTube for “passive radar”).

This document refers to the KrakenSDR Pi 4 image file V1.3 provided by KrakenRF (see download link below). Later releases may differ in functionality. I may update this document along future releases.

Please note that with the current Pi 4 Image File V 1.3 the x- and y-axis units in the “Passive Radar” dashboard are “cells”, not km or Hz or km/h. See
https://github.com/krakenrf/krakensdr_docs/wiki/08.-Passive-Radar#range-doppler-units
for the formulas to convert “cells” to km and km/h.

KrakenSDR Key Weblinks

Main website: https://www. krakenrf.com
Main GitHub link: https://github.com/krakenrf
Passive radar wiki: https://github.com/krakenrf/krakensdr_docs/wiki/08.-Passive-Radar
WiFi connectivity: https://github.com/krakenrf/krakensdr_doa#choose-and-set-up-connectivity-option
Pi image download link: https://github.com/krakenrf/krakensdr_doa/releases

Crowd Supply: https://www.crowdsupply.com/krakenrf/krakensdr
https://www.crowdsupply.com/krakenrf/krakensdr/updates/

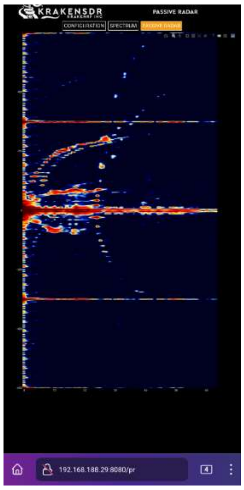
Abbreviations

CH0, CH1	KrakenSDR antenna channel 0, channel 1
DAQ	Digital AcQuisition
DOA	Direction Of Arrival
DSS	DAQ Subsystem Status
GUI	Graphical User Interface
IP	Internet Protocol
LNA	Low-Noise Amplifier
OS	Operating System, e.g. Linux, Debian, Windows
PD	Power Delivery
PR	Passive Radar
PRC	Passive Radar Configuration
PS	Power Supply
QC	Quick Charge
RF	Radio Frequency
RRC	RF Receiver Configuration
SDR	Software Defined Radio
SMA	SubMiniature version A
SSH	Secure SHell

Preparation

You will need

- The KrakenSDR with
  - A USB power supply (PS) of at least 2.4 A for the KrakenSDR. The KrakenSDR supports both native USB power as well as the USB-C PD (Power Delivery) standard. According to my experience the KrakenSDR can also be supplied through a USB QC 3.0 (QC for Quick Charge) outlet.



It is very important that you power the KrakenSDR with a low-noise PS: If you will later observe static equally spaced horizontal lines on the PR dashboard (see picture to the right), then this will be most probably due to PS noise inserted into the KrakenSDR via the USB-C power line.

I found this PS to be relatively low-noise: Hama 210537

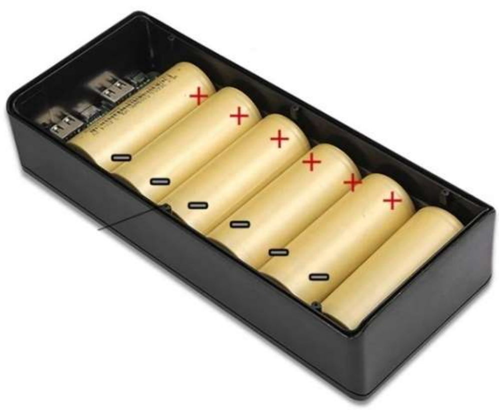
(it may be sold under various brand names). It has the additional advantage that you may supply both KrakenSDR and Pi from this single PS by connecting e.g. the Pi to the orange QC 3.0 connector and the KrakenSDR to one of the other connectors.

I found these PS to be noisy on the KrakenSDR, but they may still be suitable for the Pi:

ILEPO 20W QC 3.0

JDLIGHT HN-538

For mobile use I found this solution with the beefy 26650 Li-ion batteries:



https://www.aliexpress.com/item/1005004669688186.html

It is approximately as low-noise as the Hama 210537. As with the Hama 210537, the battery box seems to be able to supply both KrakenSDR and Pi at the same time (according to supplier's specs: 22.5 W max, 25 W peak). This box can also load the inserted batteries.

Strangely, after inserting the batteries, the pack seems to require being connected to an external USB PS at least for a short moment. Otherwise it won't work.

Inserting one or more batteries the wrong way may have an immediate devastating effect on the battery box's electronics and/or the other batteries. I did not dare to try.

Using a number of batteries in parallel has the effect that the batteries will slowly discharge each other over time by small differences in voltage: Say battery A has a slightly lower voltage than battery B, therefore consuming load current form B. Then B's voltage drops slightly below A's and the game runs the other way around, etc.

  - A USB cable between the PS and the KrakenSDR. It should be short to pick-up as little noise as possible from the environment.

- A Raspberry Pi 4B with
  - As much RAM as possible (I use 8 GB). However, the current KrakenSDR Pi image V1.3 uses less than 2 GB of RAM.
  - Massive cooling (e.g. an "Ice Tower"):

  - A fast USB 3.x memory stick of at least 8 GB to hold the Pi OS and KrakenSDR application. The current Pi 4 image file V 1.3 takes about 7.5 GB (uncompressed). You may use a Micro SD card instead, but the USB 3.x interface (the two blue USB-A connectors on the Pi) may be faster than the Micro SD interface, and an USB stick will be easier to insert and remove than a Micro SD card.
  - A USB power supply of at least 3 A for the Pi. In contrast to the KrakenSDR (see above) the Pi is more tolerant to PS noise.
  - A USB cable between the PS and the Pi.
  - A USB-A connector to USB-C connector type cable between the Pi and the KrakenSDR. It should be short to pick-up as little noise as possible from the environment.
  - The KrakenSDR Pi 4 image file V1.3 downloaded from https://github.com/krakenrf/krakensdr_doa/releases
  - An ISO imager software to load the KrakenSDR Pi 4 image file V1.3 onto the USB stick, e.g. for Windows https://www.balena.io/etcher/ or https://win32diskimager.download/ .

- For mobile usage you may power the KrakenSDR and the Pi from USB power packs as long as they are supplying sufficient amperage.

Using the KrakenSDR for PR will only require its CH0 and CH1. To save some energy you may disable CH2 to CH4 via the KrakenSDR-internal DIL-switches. You will have to open the

KrakenSDR's case for this. I have not done this so far, taking into consideration that the Pi exhausting its battery pack before the KrakenSDR exhausting his will not provide a practical advantage.

- A PC or a mobile to connect to the Pi, preferably through WiFi (or Ethernet), in order to display and operate
  - the Pi's console via a suitable SSH (Secure Shell) communication application, e.g. Putty for Windows https://www.putty.org/ or a Putty app for your mobile. You will need the Pi's console mainly for initial configuration and later for a clean Pi OS (Operating System) shutdown or reboot.
  - the KrakenSDR's GUI (Graphical User Interface) via a standard browser (e.g. Chrome, Edge, Firefox, Safari).
- A local WiFi network provided by (option 1:) a local hotspot provided by the KrakenSDR application on the Pi or (option 2:) via your mobile configured as a local hotspot or (option 3:) your local router or 3rd party hotspot
- An antenna connected to KrakenSDR's CH0 to receive the transmitter
  - If the transmitter is powerful and/or relatively close (e.g. a local TV or radio station a few km away) then you may do with a $\lambda/4$ (quarter wavelength) rod antenna.
  - If the transmitter is less powerful and/or more distant then you may require a directional antenna (e.g. a Yagi) tuned to the transmitter's frequency.
  - **If the transmitter you plan to use (or even a completely different transmitter) is extremely close (hundreds of meters, very few km) and powerful you may need a suitable SMA attenuator (e.g. 10 dB or much more) between your antenna and the CH0 connector of the KrakenSDR. Otherwise permanent damage may occur to your KrakenSDR.**
- An antenna connected to KrakenSDR's CH1 to receive the signals reflected from targets. Here you will require a directional antenna (e.g. a Yagi with reflector(s) and at 2 to 5 directors) to

  a) obtain sufficient gain and signal to noise ratio (S/N) for the weak reflected signals and
  b) weaken the direct-path signals from the transmitter.

  Caution: If the transmitter you plan to use (or even a completely different transmitter) is extremely close (hundreds of meters, very few km) and powerful you may need a suitable SMA attenuator (e.g. 10 dB or much more) between your antenna and the CH1 connector of the KrakenSDR. Otherwise permanent damage may occur to your KrakenSDR.
- You may obtain your Yagi(s) matching the transmitter frequency on the market ready to use or build one yourself matching the specific transmitter frequency to a hair. For the latter you will find Yagi calculators e.g. here:
  - Online: https://k7mem.com/Ant_Yagi_VHF_Quick.html
  - Windows application: https://www.vk5dj.com/yagi.html
  - (There are others)

  Here you will find 3D-printable parts to build a collapsible Yagi made of tape meter rods: https://www.thingiverse.com/thing:5472960
  It is my design and I am using it day and night.

- Some form of antenna pole(s) or tripod(s) to hold the antenna(s) and possibly the KrakenSDR, the Pi and power supplies or power packs. Here you will find a 3D-printable frame to mount the Pi to the KrakenSDR and both of them to an antenna pole:
  https://www.thingiverse.com/thing:5480973
  It is my design and I am using it day and night.

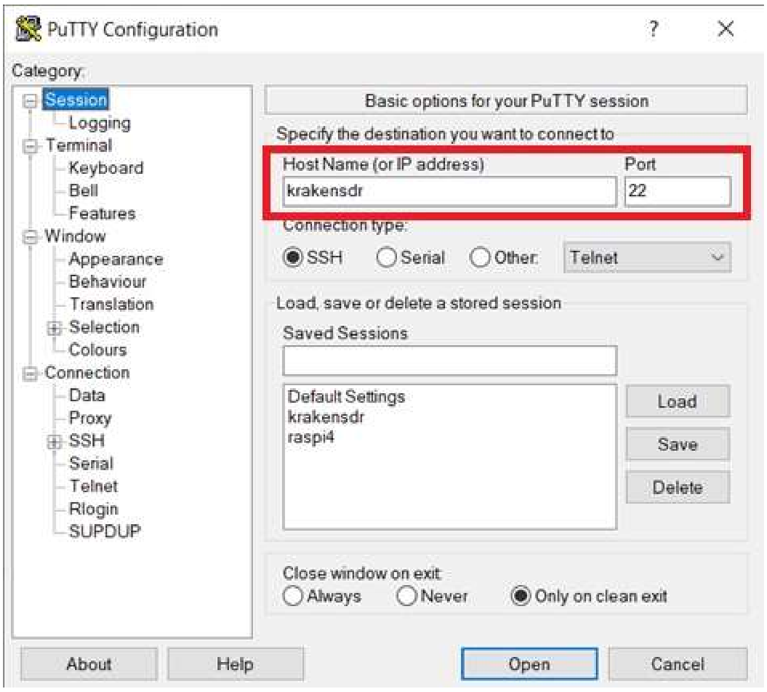
- Enthusiasm and patience to experiment with your KrakenSDR PR

Installation

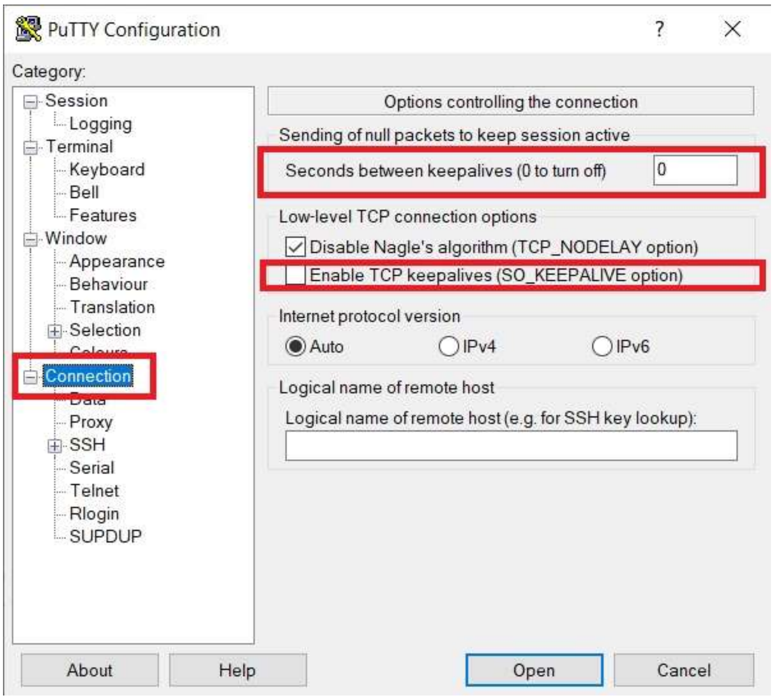
PC and optionally mobile:

- Download and install an ISO imager application on your PC (e.g. for Windows https://www.balena.io/etcher/ or https://win32diskimager.download/). There may be similar ISO imager applications in the Apple world.
- Download Putty from https://www.putty.org/ and install it on your Windows PC. There may be similar SSH applications to Putty in the Apple world. Optionally download and install a Putty app on your mobile.

  Putty under Windows:



The Pi's SSH server to which Putty on your PC or mobile connects terminates Putty sessions after a timeout. If you want to keep your Putty connection open for a long time without console input from your side, you will have to either cancel the timeout on the Pi's side or activate a "keepalive" function in Putty. You will find instructions for this here:
https://www.simplified.guide/ssh/disable-timeout



- Configure your mobile as local hotspot if you choose to use it as the local WiFi access point (option 2 or 3, see below). Otherwise use the Pi as local hotspot (option 1, see below) or use your router (it may be up and running already) or a local 3rd party hotspot (option 3, see below).

Pi:

- Connect the USB stick to your PC and
    - Download and install the KrakenSDR Pi 4 image V1.3[1] from https://github.com/krakenrf/krakensdr_doa/releases
      to your PC and install it on the USB stick by means of Etcher, Win32Imager or similar ISO imager utilities. **Take care to select the USB stick within these applications and not one of the other drives on your PC, as these imager applications will overwrite the target completely.** Upon completion there will be a "/boot" partition on the USB stick, visible under Linux and Windows, and another partition visible on Linux-based PCs, but not visible under Windows. If Windows asks you to format the so prepared USB stick, then please resist and say "No!".
    - Create an empty file named "ssh" (lower case, no extension) in the top-level directory (root directory) of the USB stick's "/boot" partition to enable WiFi access from the PC or mobile to the Pi's console via the SSH protocol.
    - For setup of the various local networking options see also https://github.com/krakenrf/krakensdr_doa#choose-and-set-up-connectivity-option

      You have the choice of

        - **Option 1**: Let the Pi create a WiFi hotspot named "krakensdr" to which you can connect with your PC or mobile. In its supplied implementation the KrakenSDR Pi image is not

[1] Later releases may differ in functionality.

configured as a gateway into the Internet and you will therefore not have Internet access with option 1.

The KrakenSDR Pi 4 image will create this WiFi hotspot by default whenever

- you either do not configure options 2 or 3 (i.e. if you do not supply a "wpa_supplicant.conf" file to the "/boot" partition of your USB stick as detailed below in options 2 and 3),
- or if the access point(s) configured in the "wpa_supplicant.conf" file is(are) not currently accessible by the Pi at boot time.

After having disconnected your PC or mobile from any other WiFi access point you will then be able

- to connect your PC or mobile to the Pi's hotspot named "krakensdr" with the WiFi access code "krakensdr" (i.e. same as the network name),
- access the Pi's console via Putty at IP address 192.168.50.5 / port 22 and
- access the KrakenSDR PR application with your browser at IP address 192.168.50.5 / port 8080, i.e. by typing http://192.168.50.5:8080 in the address field of your browser. Make sure that your browser allows "http" connections. "https" will not work.

Hotspot	Network name, access code	Host name, IP address	Putty port	Browser access
Pi	krakensdr krakensdr	krakensdr or 192.168.50.5	22	http://krakensdr:8080 or http://192.168.50.5:8080

- **Option 2**: Configure your PC, mobile or router as a local hotspot with the specific network name "KrakenAndroid" and the specific WiFi access code "KrakenAndroid. The Pi will look for a network of this name and with such WiFi access code when booting and try to connect. For option 2 to work as intended you will have to configure a "wpa_supplicant.conf" file in the Pi's "/boot" partition and directory with one of the entries containing the "KrakenAndroid" / "KrakenAndroid" credentials as supplied in "wpa_supplicant_example.conf". See the "wpa_supplicant.conf" file layout under option 3 below.

  A difficulty with option 2 may be to find the Pi's IP address as a prerequisite to connect your PC or mobile to Pi's console via Putty or the KrakenSDR PR application via your browser. Under Android 12 you may find this IP address at Settings / Connections / Mobile Hotspot and Tethering / Mobile Hotspot / Connected devices as "krakensdr". However, instead of using the IP address you may use the KrakenSDR's host name "krakensdr", i.e.

  - Type "krakensdr" instead of the IP address in Putty (and "22" in the port input field)
  - Type http://krakensdr:8080 instead of the IP address and port in your browser. Make sure that your browser allows "http" connections. "https" will not work.

Hotspot	Network name, access code	Host name, IP address	Putty port	Browser access
PC, mobile or router	KrakenAndroid KrakenAndroid (both to be configured in the hotspot)	krakensdr or IP address allocated by hotspot	22	http://krakensdr:8080 or http://192.168.xx.xx:8080

- **Option 3**: Use your existing local WiFi network (e.g. your PC or mobile hotspot, your local router or a local 3rd party hotspot) with their generic network names and WiFi access codes and supply the Pi with the necessary WiFi credentials of your existing local WiFi network in the Pi's "wpa_supplicant.conf" file, see below.

  With the USB stick still inserted into your PC: Configure the supplied "wpa_supplicant_example.conf" template file in the "/boot" partition of your USB stick with your mobile hotspot or existing local network WiFi details and save the edited file on the USB stick's top-level (root) directory as "wpa_supplicant.conf". Upon first boot the Pi will internalize this file to his /etc/wpa_supplicant directory and it will no longer be visible in the "/boot" partition of the USB stick. It may be a good idea to configure one entry in "wpa_supplicant.conf" with the generic WiFi details of your router and the other entry with the generic WiFi details of your mobile hotspot if you wish to use both of them (but only one at a time). Upon booting the Pi will run top-down through this list and connect to the first accessible WiFi network that matches the currently processed entry.

  Otherwise the Pi will activate option 1 and create a hotspot named "krakensdr".

```
country=XX # Your 2-digit country code
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
   ssid="KrakenAndroid"
   psk="KrakenAndroid"
   key_mgmt=WPA-PSK
   priority=4
}

network={
   ssid="Your generic mobile hotspot network name"
   psk="Your generic mobile hotspot WiFi access code"
   key_mgmt=WPA-PSK
   priority=3
}

network={
   ssid="Your generic router network name"
   psk="Your generic rooter network WiFi access code"
   key_mgmt=WPA-PSK
```

```
    priority=2
}
```

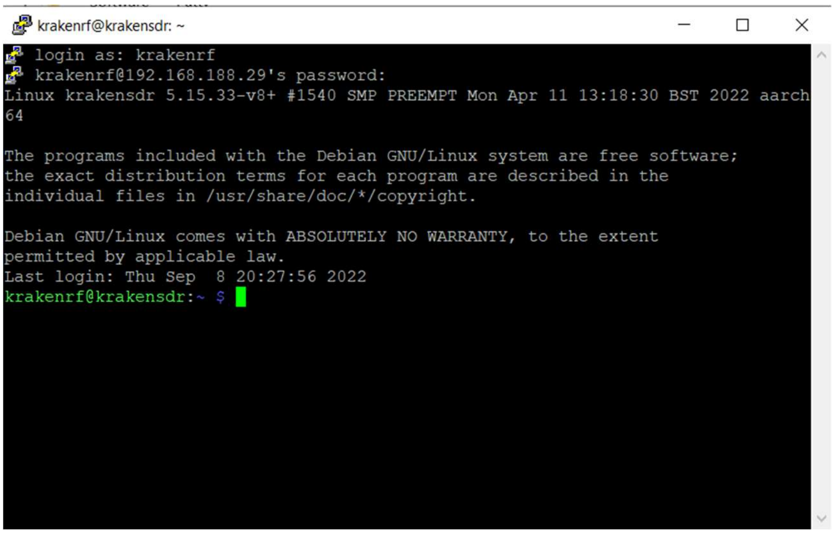
As with options 1 and 2 you may use the KrakenSDR's host name "krakensdr" instead of its IP address.

Hotspot	Network name, access code	Host name, IP address	Putty port	Browser access
PC, mobile, router, or local (e.g. public) hotspot	According to hotspot. Must be configured in Pi's wpa_supplicant.conf file.	krakensdr or IP address allocated by hotspot	22	http://krakensdr:8080 or http://192.168xx.xx:8080

- Plug the USB stick into one of the two blue USB 3.x connectors on the Pi.
- Connect the Pi to its power supply. It will now boot and (if supplied, see options 2 and 3) install the WiFi details of your "/boot/wpa_supplicant.conf" file in the "/etc/wpa_supplicant/wpa_supplicant.conf" file and delete "/boot/wpa_supplicant.conf". It will detect the "/boot/ssh" file, activate WiFi and wait for you to connect with your Putty application to the Pi's console. Further it will start the KrakenSDR application and set up a KrakenSDR GUI server. This will take between one and three minutes to complete, depending on the amount of your Pi's RAM and speed of the USB stick or Micro SD card.
- Launch Putty on your PC or mobile and connect Putty to the Pi. The Pi's hostname within the WiFi network under options 1 to 3 is "krakensdr" and the SSH port is 22.

  If you are unsuccessful connecting via the hostname you will have to connect via the Pi's IP address. In case the Pi is providing the hotspot (option 1), the Pi's IP address will be "192.168.50.5". Under the options 2 or 3 your access point will allocate an IP address to the Pi, e.g. something looking like "192.168.188.29" (but in your case almost certainly with different figures), visible on the access point's dashboard.
- Log into the preconfigured "krakenrf" account:

  Login name: "krakenrf"
  Password: "krakensdr"



- Edit the “start.sh” file by means of the Pi’s “nano” text editor (type “nano start.sh”) so that in the end of your editing (use arrow keys, not mouse) it looks like this:

```
#!/bin/bash

# This script is run on startup by a systemd service at /lib/systemd/system/krakensdr.service
sleep 2
#cd /home/krakenrf/krakensdr_doa                [insert the hash]
#./kraken_doa_start.sh                          [insert the hash]

# UNCOMMENT FOR PASSIVE RADAR STARTUP
# MAKE SURE TO COMMENT OUT DOA START ABOVE

cd /home/krakenrf/krakensdr_pr                  #[delete the leading hash]
./kraken_pr_start.sh#!/bin/bash                 #[delete the leading hash]
```

  The text in square brackets is explanatory and should not be part of the file. Save and close by typing “CRTL-O”, “Y” and “CTRL-X”.

- For overclocking of the Pi (provided that you can supply sufficient cooling) see https://github.com/krakenrf/krakensdr_doa#pi-4-overclock and e.g. https://beebom.com/how-overclock-raspberry-pi-4/
  It is not clear to me whether this really is useful for the KrakenSDR PR application.

- Connect the Pi to the USB-C “DATA” port of the KrakenSDR with the short USB-A to USB-C connector type cable mentioned in section “Preparation” above. **The USB-C “DATA” connection between Pi and KrakenSDR is very sensitive to interruptions. Any interruption (even very short ones, e.g. by incidentally wiggling with the connectors) will crash the connection. You will have to reboot the Pi or at least restart the KrakenSDR PR application on the Pi to recover. To reboot the Pi, go to the Pi’s console via Putty and enter “sudo reboot”. To restart the application see section “Advanced Topics for V1.3”.**

Antennas:

- Install the antennas on suitable mounts (e.g. tripods or poles).
  - The CH0 antenna (λ/4 rod or directional) should have a direct line of sight to the transmitter.
  - The CH1 antenna (directional as e.g. Yagi) should point in the direction where you are looking for targets. However, the reflected signals received by the CH1 antenna are at least thousands of times smaller than the signals received by the CH0 antenna. Pointing the CH1 antenna in the general direction of the transmitter will saturate CH1 with the signal and the noise of the transmitter, overwhelming the reflected signals. Ideally the CH1 directional antenna should point 180 degrees away from the transmitter, but angles between 90 and 270 degrees may also work.

- **Before connecting the antennas to the KrakenSDR:** Make sure that the input signal supplied by either antenna is below the +10 dBm (10 mW) limit of the KrakenSDR antenna input (even if this signal may originate from a completely different transmitter and frequency than you plan to work with), otherwise the respective KrakenSDR input channels may be permanently damaged. **In case of dangerously strong antenna signals install suitable attenuators at the end of both (!) channel’s antenna cables.**
  Normally this hazard is a concern only if the transmitter is powerful (tens of watts or more) and very close (hundreds of meters, very few km). A general formula is difficult to establish, as the

local signals strength is dependent on many factors and the transmitter power will probably be unknown[2]. In case of doubt take measurements at the end of antenna cable with a suitable RF power meter before attaching it to the KrakenSDR.

- Connect the antennas (potentially through attenuators, see above) to CH0 and CH1 of the KrakenSDR.

KrakenSDR:

- Connect the (low-noise) PS to the KrakenSDR via the short USB cable mentioned in section “Preparation” above. **Any power cycling of the KrakenSDR will cash the connection between the Pi and the KrakenSDR. You will have to reboot the Pi to recover or at least restart the KrakenSDR PR application on the Pi to recover. To reboot the Pi, go to the Pi’s console via Putty and enter “sudo reboot”. To restart the application see section “Advanced Topics for V1.3”.**

Operation (valid for Pi 4 Image File V1.3)

Antenna orientation:

- Position the CH0 antenna (λ/4 rod or directional as e.g. Yagi) within a direct line of sight to the transmitter, a directional antenna (e.g. Yagi) straight at the transmitter. Most public radio (e.g. DAB) transmitters are vertically polarized, so orient the antenna in the vertical plane (λ/4 rod straight up, Yagi with its elements in the vertical plane).
- Point the CH1 antenna (directional as e.g. Yagi) in the general direction of potential targets some 90 to 180 degrees azimuthally away from the transmitter. Orient the CH1 antenna 90 degrees tilted to the CH0 antenna (i.e. in most cases in the horizontal plane). Besides orienting the CH1 antenna at least 90 degrees azimuthal away from the transmitter, this polarization separation of 90 degrees will further attenuate the (unwanted) direct-path signal from the transmitter to the CH1 antenna while not impeding the (not specifically polarized) reflected signals into the CH1 antenna.

Power-up:

- Power both the KrakenSDR and the Pi, and your PC or mobile. It will take the Pi between one and three minutes to boot his OS, (option 1:) establish his “krakensdr” hotspot or (options 2 or 3:) connect to a local access point as configured in the “wpa_supplicat.conf” file and start the KrakenSDR application.

WiFi connection between PC or mobile and Pi:

- To verify whether the Pi is ready you may either connect via Putty to the Pi’s console or via your browser to the KrakenSDR’s dashboard (see below).

  The access to the Pi’s console via Putty may remain open in the background of your PR activities until shutdown or reboot. However, the Pi may shutdown the console connection after a period of inactivity. When required just reconnect through Putty and enter the login credentials again.

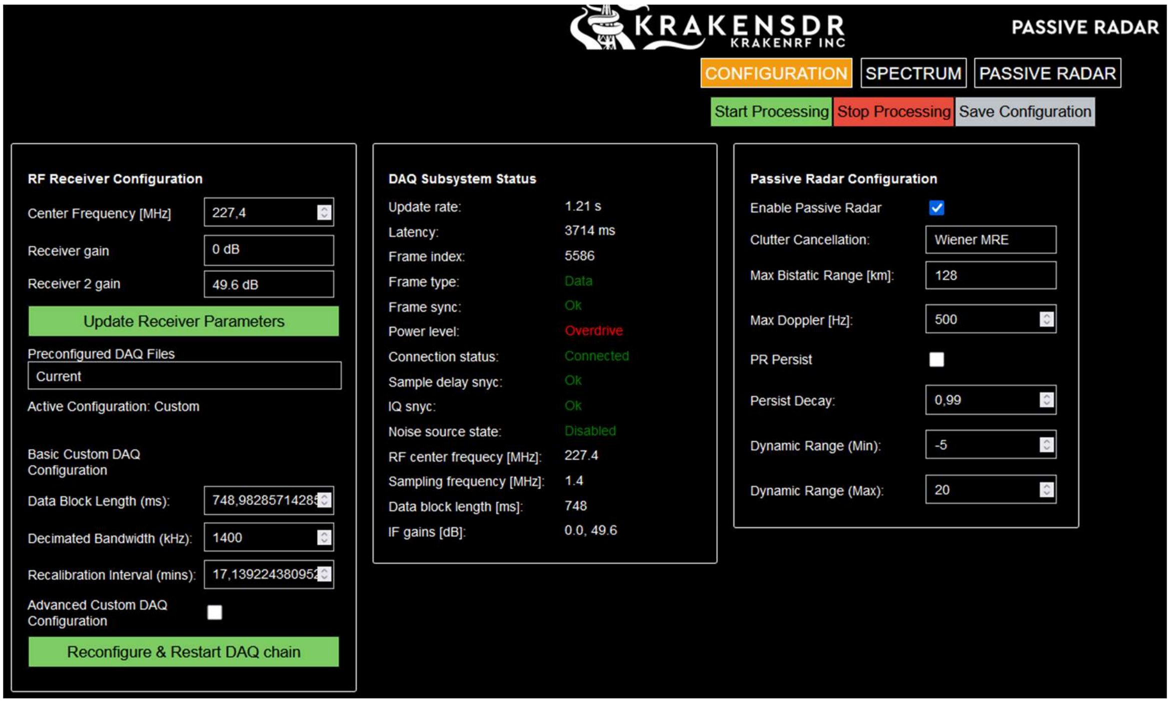
[2] See e.g. https://www.electronicdesign.com/technologies/communications/article/21796484/understanding-wireless-range-calculations

Access the KrakenSDR’s GUI:

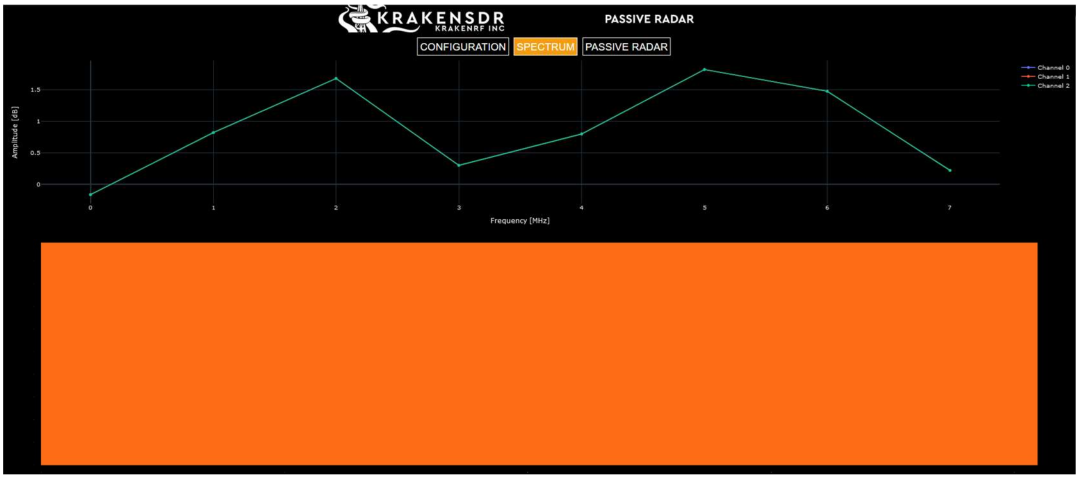
- Open the browser on your PC or mobile and connect to “krakensdr:8080”.

  On some browsers (e.g. Firefox, depending on its setup parameters) it may be necessary to tell the browser that this is not a “https://...” but a “http://...” connection: In this case type http://krakensdr:8080. If the browser still refuses to connect, enter the Pi’s IP address followed by “:” and the KrakenSDR’s dashboard port 8080 (i.e. type e.g. “192.168.188.29:8080”, or in case the Pi is providing the hotspot (option 1) http://192.168.50.5:8080).

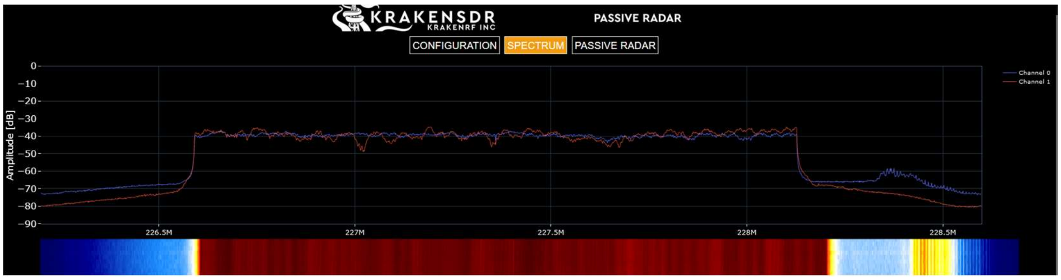
  You should now see the KrakenSDR’s “Configuration” dashboard (http://krakensdr:8080/config). **Try to abstain from randomly changing values and clicking buttons. You may end up with a completely botched setup.**



- In a first step you may want to solely obtain confirmation that your antennas are receiving signals from the supposed transmitter and defer optimizing visualization parameters to a second step.
  - At the top left you will note the “RF Receiver Configuration” (RRC). Enter your transmitter’s centre frequency in MHz (with decimals) and the KrakenSDR’s amplifier gains for CH0 (“Receiver gain”) and CH1 (“Receiver 2 gain”). Assuming a nearby (few km) strong (several kW) transmitter you may choose 0 dB for CH0. CH1 receiving the weak reflected signals will require a much stronger gain. Choose 25.4 dB as a start for CH1.
  - Click “Update Receiver Parameters”, then click the grey “Save Configuration” button.
  - Click “Start Processing”. After a few seconds the red fields in “DAQ Subsystem Status” (DSS) will turn green, with the possible exception of “Power level” which may remain with a red “Overdrive”.
  - Click the “Spectrum” button at the top. After initially noticing an orange square under a green zig-zag line



for a few seconds you should then see the CH0 and CH1 spectrums at around -40 dB and left/right centred above a waterfall diagram:

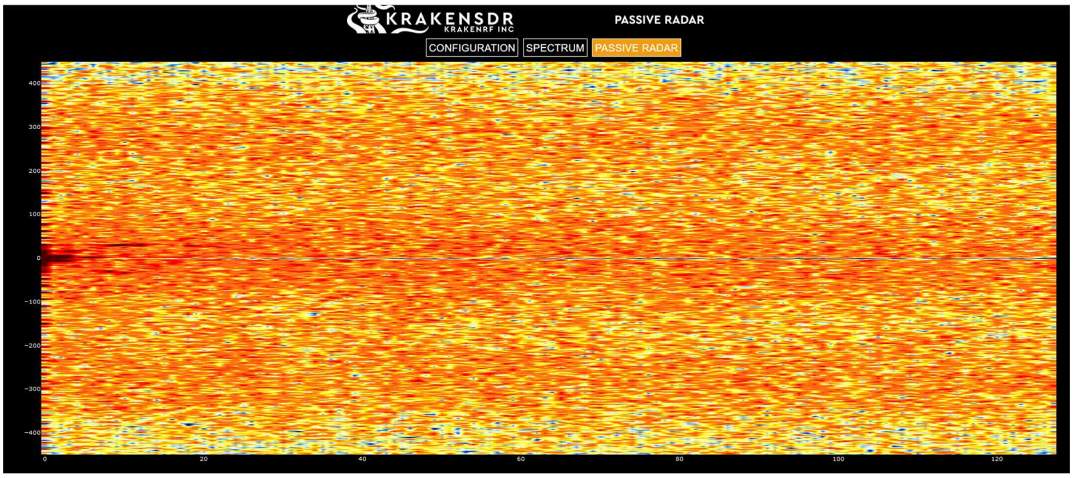


If you observe the big orange square for an extended period of time (tens of seconds), then you either have not started processing or the KrakenSDR may be receiving an insufficient signal level on at least one of CH0 or CH1, i.e. at least one of the antennas may not be connected or the gain of at least one of CH0 or CH 1 may be grossly insufficient, or the transmitter may be off.

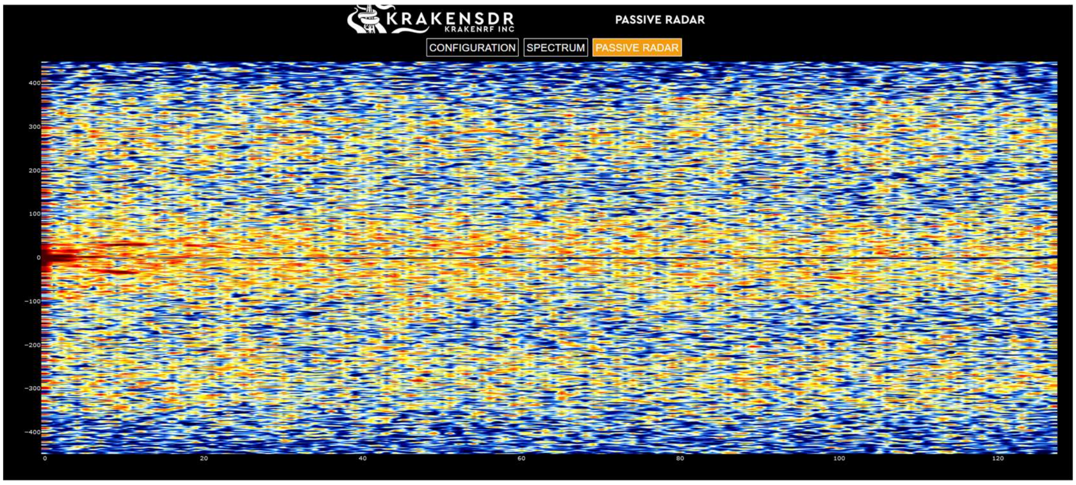
- Click on “Configuration”. Make sure the “PR Persist” clickbox in the PRC section is empty.
- Click on “Passive Radar”. After a few seconds you should see the PR dashboard:

Wait a few seconds for the display to stabilize before taking conclusions.

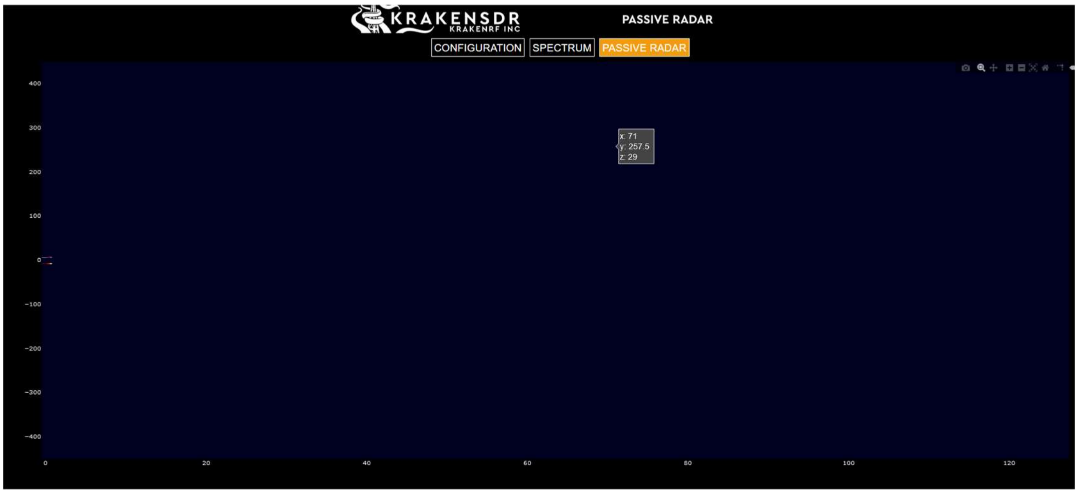
The dashboard may not look as clean as in the above example. In extreme cases it might look like this (totally cluttered):



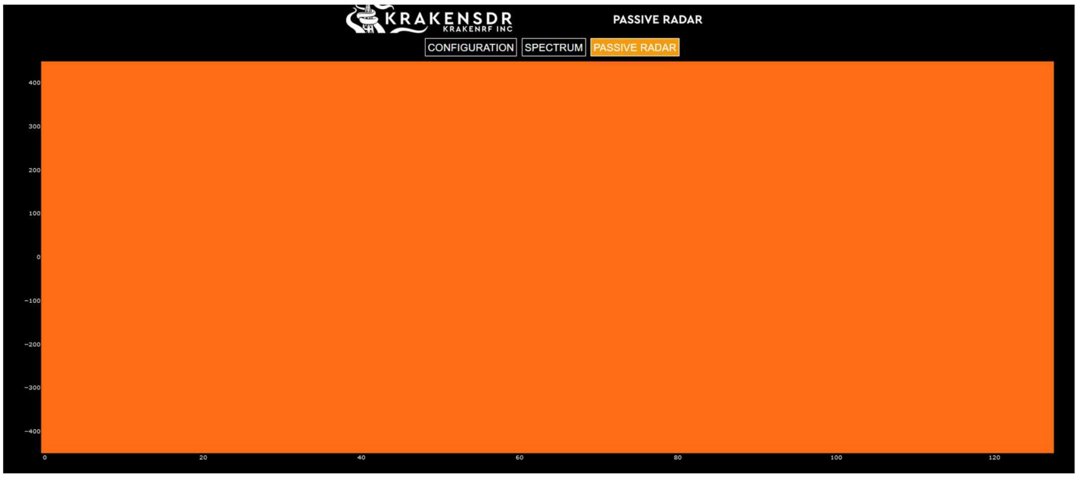
or like this (heavily cluttered):



or like this (just about empty):



Do not let this bother you for the moment. It will improve in step 2. However, if you observe a big orange square for an extended period (tens of seconds):



then

- At least one of CH0 or CH1, i.e. at least one of the antennas may not be connected or
- At least one of the CH0 or CH 1 gains in the “Configuration” dashboard may be grossly insufficient or
- The “Dynamic Range (Min)” parameter in the “Configuration” dashboard may be way too high or
- The “Dynamic Range (Min)” parameter in the “Configuration” dashboard may be equal or higher than the “Dynamic Range (Max)” parameter.

Check the antenna connections. If this des not help then one or more of the parameters in the “Configuration” dashboard may be grossly inadequate. We will deal with this in step 2.

In the KrakenSDR application based on the Pi image file V1.3 the x-axis shows the “bistatic” distance in “cells” (see my Intro for further information)

- from the transmitter to the target
- plus from there back to your position
- minus the distance between the transmitter and the KrakenSDR (because CH0 will receive the transmitter signal delayed by the distance between the transmitter and the KrakenSDR).

To compute the scientifically correct “bistatic distance” (i.e. from the transmitter to the target plus from there to the KrakenSDR) you will have to add the distance between the transmitter and the KrakenSDR. V1.4 of the KrakenSDR application may include a field where you can enter this distance to be included in the then correct bistatic distance computation.

The y-axis shows the doppler shift in “cells” (see my Intro for further information), i.e. the speed at which the target is moving radially towards you (positive values) or radially away from you (negative values). It is not the speed at which the target is moving along its trajectory in 3D space.

If you see a screen filled of a lot of noise, then either the gain of at least one of CH0 or CH1 is way too high or the “Dynamic Range (Min)” parameter in the “Passive Radar Configuration”

(PRC) column of the "Configuration" dashboard is too low. If CH0 is the cause and you have set the CH0 gain in the "Spectrum" dashboard to 0 dB (i.e. the transmitter is close and powerful), then you may have to insert an attenuator of 10 dB or more between the antenna and CH0 input.

Otherwise you have now a working, but possibly suboptimal, KrakenSDR PR setup. Congratulations! You are close to total success!

- In a second step you may now want to optimize your setup for a clean "Passive Radar" dashboard display.
  - Return to the "Configuration" dashboard and click "Stop Processing".
  - Under RRC you may select one of several "Preconfigured DAQ Files".
    Caution: Exclusively use the "pr_ch2_..." configuration files in the FRC section! By selecting and activating one of the other files of V1.3 you may end up with a completely botched setup.
    The "pr_ch2_..." configuration files mainly differ in the preconfigured "Data Block Length (ms)". Do not modify this parameter directly. Longer data block lengths lead to slower updating of the "Passive Radar" dashboard.

    After having selected a different "pr_ch2_..." preconfigured DAQ File you will have to click "Reconfigure & Restart DAQ chain" to activate it. You will notice that in the DSS section the parameter "Connection status" will indicate an orange "Reconfiguring" text for a few seconds while the activation is taking place.

    If this text is shown for a prolonged time (several ten seconds) then the KrakenSDR application may have crashed, sometimes because you forgot to "Stop Processing". You will have to reboot the Pi by typing "sudo reboot" via Putty into the Pi's console.

    But normally, after a few seconds, "Connection status" shows a green "Connected" and you may click "Start Processing" and "Save Configuration" and switch to the "Passive Radar" dashboard to observe the results.
  - **Caution: In the RRC section do not manually modify any fields**
    - in the "Basic Custom DAQ Configuration" section or
    - in the "Advance Custom DAQ Configuration" section
    except as detailed below. You may end up with a completely botched setup. The values of the fields do have complex boundary conditions for their values and relationships.

    The only input field that you may need to adapt in the DAQ configuration is the "Sample Rate [MHz]" in the "Advanced Custom DAQ Configuration" section, set at a default 2.4 MHz for HDTV. In the "Spectrum" example screenshot above you will have noted that the effective spectrum of this DAB transmitter is less than 2.4 MHz. In order to avoid inclusion of noise from the frequencies to the left and right of the effective bandwidth, the bandwidth should be limited to the effective bandwidth of the transmitter or just below. Ideally the width of the "Spectrum" dashboard should match the transmitter's bandwidth. This can be approximated by the "Sample Rate [MHz]" parameter:

    In the RRC section activate the "Advance Custom DAQ Configuration" click box. Scroll down to the "Sample Rate [MHz]" field and select an appropriate sample rate depending on your transmitter type. The sample rate should be equal to or just below the transmitter bandwidth. E.g. for DAB's 1.5 MHz bandwidth the sample rate should be 1.4 MHz. Check with the "Spectrum" dashboard.

Then deactivate the "Advanced Custom DAQ Configuration" click box and click "Reconfigure and Restart DAQ chain" button. You will notice that in the DSS section the parameter "Connection status" will indicate an orange "Reconfiguring" text for a few seconds while the activation is taking place.

If this text is shown for a prolonged time (several ten seconds) then the KrakenSDR application may have crashed, sometimes because you forgot to "Stop Processing". You will have to reboot the Pi by typing "sudo reboot" into the console via Putty.

But normally, after a few seconds, "Connection status" shows a green "Connected" and you may click "Start Processing" and "Save Configuration" and switch to the "Passive Radar" dashboard to observe the results.

- Return to the "Configuration" dashboard. The "Dynamic Range" parameters in the PRC section limit the signals displayed in the "Passive Radar" dashboard according to their strength:
    - "Dynamic Range (Min)" cuts off signals below the selected dB value. A reasonable range may be between -20 and -5.
    - "Dynamic Range (Max)" cuts- off signals above the selected dB value. A reasonable range may be 10 to 30.

  You do not need to cycle "Stop Processing" / "Start processing" to activate parameter changes. Check with the "Passive Radar" dashboard.

  There is a relationship between the receiver gains in the RRC section and the dynamic range in the PRC section: Increasing the gain of e.g. CH1 may lead to noise in the "Passive Radar" dashboard, which you could reduce by increasing "Dynamic Range (Min)". To obtain a low-noise dashboard while at the same time detecting distant reflectors you will have to experiment with CH1 gain ("Receiver 2 gain") and "Dynamic Range (Min)". **Optimize CH1 gain ("Receiver 2 gain") and "Dynamic Range (Min)" such that the "Passive Radar" dashboard is just free of noise.** The noise level may depend on weather conditions (clouds, rain, snowfall) and (if you are standing nearby) the reflections of the transmitter's direct-path via your body into the antenna of CH1.

- Select a suitable "Max Bistatic range (km)" setting. Remember that in V1.3 this range is the path length in "cells" (not km, see my Intro for further information) between your position and transmitter and from there to the target and back to your position. You do not need to cycle "Stop Processing" / "Start processing" to activate the change, but the change will reset persistence (see below). Check with the "Passive Radar" dashboard.

- Enter a suitable "Max Doppler (Hz)" value. Remember that in V1.3 this range is the doppler speed in "cells" (not Hz, see my Intro for further information). You do not need to cycle "Stop Processing" / "Start processing" to activate the change, but the change will reset persistence (see below). Check with the "Passive Radar" dashboard.

- Optimize the CH0 and CH1 gains in RRC depending on your range settings (more range set = more gain required, either by increasing RRC (Click “Update Receiver Parameters”) or by adding directors to your CH1 Yagi). Click “Update Receiver Parameters”, but you do not need to cycle “Stop Processing” / “Start processing”. Check with the “Passive Radar” dashboard.
- Activate / deactivate the “PR Persist” click box in the PRC section dependent on whether you prefer persistence in the “Passive Radar” dashboard. You may alter the level of persistence with the “Persist Decay” parameter (0.0 … 1.0). A value of zero equals no persistence and a value of one means no decay, i.e. for each pixel its highest value will be preserved across updates. You do not need to cycle “Stop Processing” / “Start processing” for this. Check with the “Passive Radar” dashboard.
- Click “Save Configuration”. Otherwise the changes will not be persistent across reboot.

- **Do not shut down the Pi by simply powering it off. To shut down the Pi’s OS go to the Pi’s console via Putty and enter “sudo shutdown now”. Then power off after a few seconds (i.e. when the Pi’s green LED stops flashing and remains off).**

Advanced Topics for V1.3

- It may make sense to save an empty “ssh” file and your “wpa_supplicant.conf” file locally on your PC. If you ever have to rewrite the USB stick you can simply copy these files back onto the USB stick.
- In V1.3 there are some preconfigured DAQ files (i.e. the “non-pr_2ch…” files) whose selection will crash the KrakenSDR application. You may delete these unnecessary directories (whose names do not begin with “pr_2ch…”) via Putty:
  “rm -r /home/krakensdr/krakensdr_pr/heimdall_daq_fw/config_files/kerberos_default”
  “rm -r /home/krakensdr/krakensdr_pr/heimdall_daq_fw/config_files/kraken_default”
  “rm -r /home/krakensdr/krakensdr_pr/heimdall_daq_fw/config_files/kraken_development”
  “rm -r /home/krakensdr/krakensdr_pr/heimdall_daq_fw/config_files/unit_test_k4”

- To stop the KrakenSDR PR application via Putty and the Pi’s console without shutting down the Pi type
  “cd /home/krakenrf/krakensdr_pr”
  “./kraken_pr_stop.sh”
  The Pi will continue to run and you may restart the KrakenSDR PR application as detailed below.

- To restart the KrakenSDR PR application via Putty and the Pi’s console type
  “ ./start.sh >/dev/null &”

- You may have configured a different “Decimated Bandwidth [kHz]” in the “Configuration” dashboard, e.g. replacing the standard 2’400 kHz by 1’400 kHz for the DAB transmitter you are using. After switching to a different “Preconfigured DAQ File” (e.g. from “pr_2ch_2pow22” to “pr_2ch_2pow20”), the KrakenSDR Pi image V1.3 release will reset the bandwidth to the preconfigured value of 2’400 kHz. To permanently change the preconfigured value to your specific setting, use Putty to log into the Pi’s console and type:
  “cd krakensdr_pr/heimdall_daq_fw/config_files”
  “nano pr_2ch_2pow**20**/*”   [example for the “pr_2ch_2pow**20**” config file]
  Edit the sample rate [in Hz] as follows:

"sample_rate = 1400000" [example for DAB's standard 1.5 MHz bandwidth)]
Leave "nano" by typing CTRL-O,Y,CTRL-X.

Pushing the Clutter Limit

- To detect very distant objects, you should optimize CH1 gain and "Dynamic Range (Min)" such that the "Passive Radar" dashboard is just free of "clutter" (the fine blue "snow" slowly filling the screen while in "PR Persist" mode). I define the "clutter limit" as the setting of both CH1 gain and "Dynamic Range (Min)" that just does not lead to clutter while in "PR Persist" mode. At the clutter limit, increasing CH1 gain to the next higher value in the list or lowering "Dynamic Range (Min)" would lead to clutter.

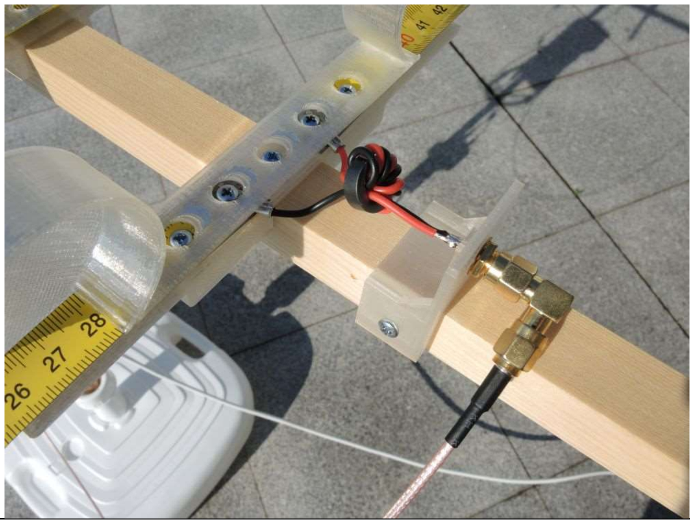
  At first sight, increasing/decreasing CH1 gain or decreasing/increasing "Dynamic Range (Min)" seems to have an identical effect on the clutter limit. A systematic test reveals that this is not true (all figures in dB):

PS Type	**CH1 gain**	**Dynamic Range (Min)**	**Clutter limit at**
Hama 210537	44.5	5	39.5
	44.5	0	44.5
	43.4	-10	53.4
	40.2	**-20**	**60.2**
	Clutter at any gain	-30	N/A
Battery Box	49.6	5	44.6
	44.5	0	44.5
	42.1	-10	52.1
	38.6	**-20**	**58.6**
	Clutter at any gain	-30	N/A
Preconfigured DAQ file "pr_2c_2pow20" / Sample Rate [MHz]: 1.8 / transmitter: DAB @ 5 km / CH0: λ/4 rod antenna / CH 1: Yagi with one reflector and 5 directors			

  According to my tests it seems advisable to set "Dynamic Range (Min)" to -20 and to feel for the clutter limit exclusively with CH1 gain.

  However, this is not the end of the line:

- To connect the Yagi dipole to the CH1 coaxial antenna cable via a simple two-wire line is a hack, because it may allow common mode noise to enter the coaxial antenna cable. Use a balun (common mode choke) to filter this noise. Remember that the reflected signal received by the CH1 Yagi is extremely weak.



  There are two types of baluns:

  - Common mode rejection baluns
  - Impedance matching baluns

I am talking about the first variant here. Take a small ferrite toroid and wind the thin two-wire line between the dipole and the coaxial cable a few turns (I use four) in parallel (!) onto the ferrite.

The effect is this:

PS Type	**CH1 gain**	**Dynamic Range (Min)**	**Clutter limit at**
Hama 210537	**44.5**	**-20**	**64.5**

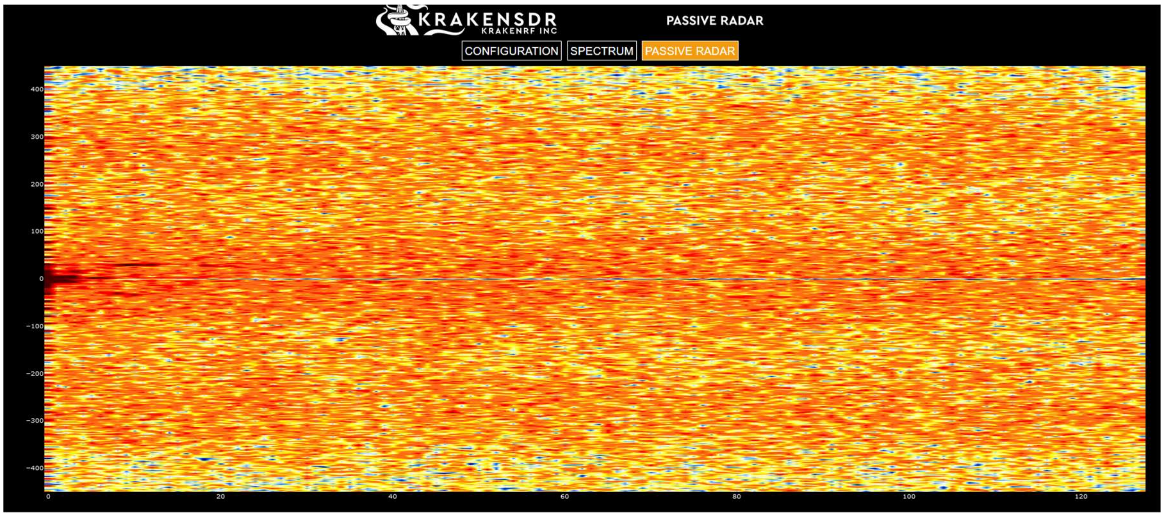
This simple provision pushes the clutter limit another 4 dB.

Pushing the Range Limit

- I tried to increase range with an LNA
  https://www.tindie.com/products/gpio/ultra-low-noise-amplifier-10-mhz-to-4000-mhz/
  preceded by a DAB band-pass filter (with my setup working at 227.4 MHz)
  https://www.aliexpress.com/item/1005004098440841.html
  without success: It amplified the signal as well as the noise. I had to reduce CH1 gain by the amount of the LNA's amplification to get a clutter-free "Passive Radar" dashboard. My conclusion was that a LNA will make sense after I will have obtained a clutter-free "Passive Radar" dashboard with maximum CH1 gain (49,6 dB) and a "Dynamic Range (Min)" of -20 dB. I am almost there (44.5 dB and -20 dB), but will have to lower noise even further before having another LNA try.
- Deeply believing that the earth is a sphere I made some altitude calculations: At my practically achievable bistatic distances of about 100 km (approx. 50 km effective range) the curvature of the earth requires the airplane to be at a height of 200 m above ground or more, so the curvature of the earth is not a limiting factor.
- Transmitters above medium to short wave lengths tend to direct their transmission power to potential listeners in the closer and possibly farther vicinity and not into outer space. Their antenna radiation characteristics normally supply a flat cone around the antenna position. If an airplane is passing far away and high up it may not be illuminated by the transmitter.

Troubleshooting

- If your Passive Radar dashboard looks like this (or similarly cluttered or even fully orange)



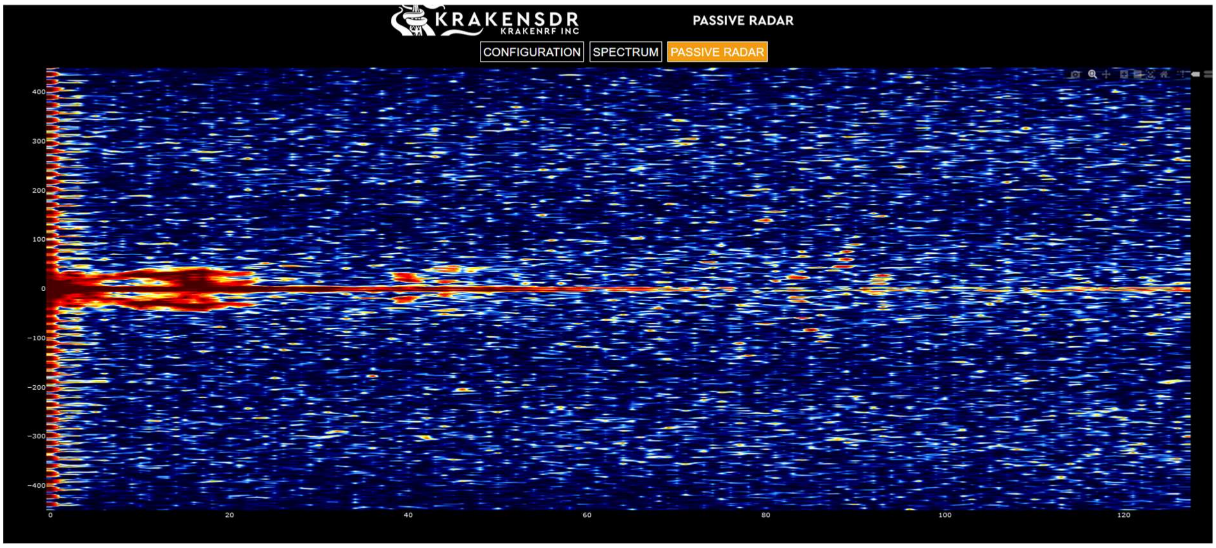
then

  - The CH0 and/or CH1 gain(s) in the RRC section of the "Configuration" dashboard may be too high: Reduce the gains (one at a time) and click "Update Receiver Parameters".
  - The "Dynamic Range (Min)" in the PRC section of the "Configuration" dashboard may be too low: Increase it.
  - The "Dynamic Range (Min)" parameter in the "Configuration" dashboard may be equal or higher than the "Dynamic Range (Max)" parameter: Increase the "Dynamic Range (Max)" parameter.

  If this does not help:

  - The CH0 antenna is a directional antenna (e.g. Yagi) and points to a strong nearby transmitter and therefore provides the KrakenSDR with a saturating signal even at a CH0 gain of 0 dB: Use a $\lambda/4$ rod antenna for CH0 instead.
  - The CH1 antenna may point towards the transmitter. Turn the antenna away from the transmitter.
  - The directivity of your CH1 antenna may be insufficient relative to the power and closeness of the transmitter: Use a more directive antenna, e.g. a Yagi with more directors.

- If your Passive Radar dashboard looked normal, but then suddenly looked cluttered like this,



then the reason may be that you or somebody else approached the antennas. In particular this may happen if you set the gains and “Dynamic Range (Min)” close to the clutter limit (which elsewise is a good idea).

- If you had a clutter-free Passive Radar dashboard before shutting down, but a Passive Radar dashboard full of clutter when restarting next time (without having changed anything in the Configuration dashboard), then you probably either forgot to save the latest configuration changes or you started in “PR Persist” mode. The KrakenSDR may clutter the Passive Radar dashboard upon processing the first sample before settling to a clutter-free display as before. The “PR Persist” takes this first cluttered sample undifferentiated. To clear the Passive Radar dashboard:
    - Go to the Configuration dashboard and deactivate the “PR Persist” clickbox.
    - Go to the Passive Radar dashboard and then back to the Configuration dashboard.
    - Activate the “PR Persist” clickbox.

- If, after having run the KrakenSDR application for some time without observing it in the browser, the browser GUI appears to have become unresponsive (you have to wait up to several 10 seconds for updates), the output queue of the GUI may have become full and has to be emptied before the GUI will become responsive again. Use patience to overcome this situation.

- If the GUI has become totally blocked, then the reason may be that the USB-C connection between Pi and KrakenSDR or one of the PS connections was interrupted (even very shortly as e.g. during fiddling with the respective cables). If the Pi’s console through Putty is still responsive, then you may restart the KrakenSDR PR application by entering
“cd /home/krakenrf/krakensdr_pr”
“./kraken_pr_start.sh”
However, if the console is unresponsive you will have to power-cycle both Pi and KrakenSDR.

To be continued …