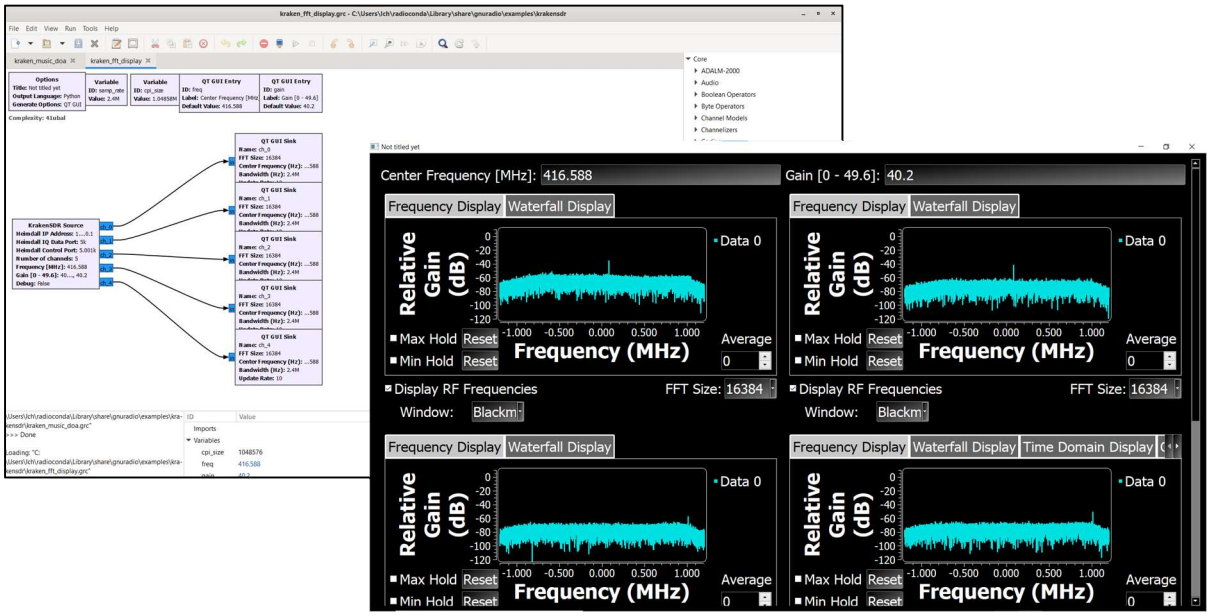
How to install and run gr-krakensdr natively on Windows

Version 1.0

7 April 2023

The gr-krakensdr software enables you to use the myriad features of GNU Radio and its graphical user interface GNU Radio Companion to process the signals received by the KrakenSDR hardware.



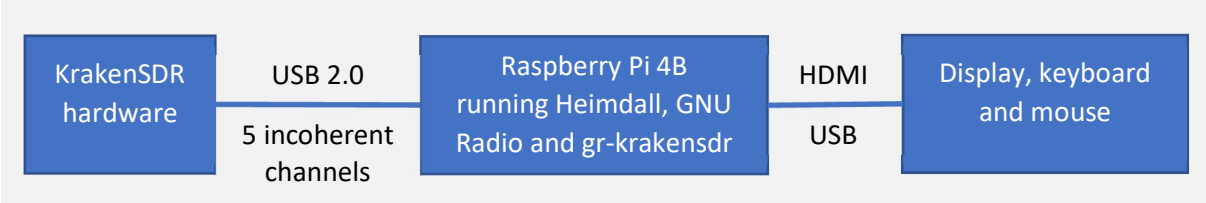
Contents

1. Intro

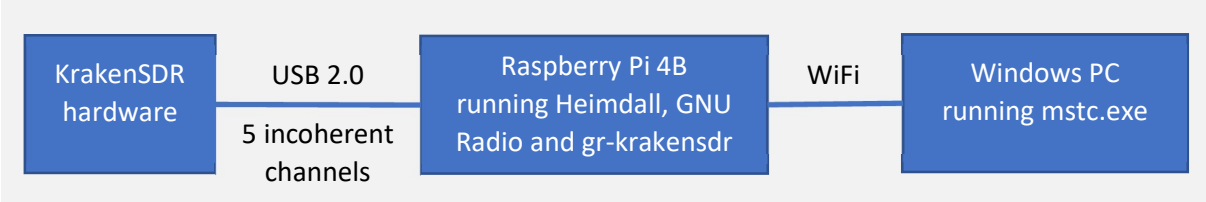
gr-krakensdr as supplied by KrakenRF in source form at https://github.com/krakenrf/gr-krakensdr or as included in the (non-desktop) KrakenSDR Pi image downloadable at https://github.com/krakenrf/krakensdr_doa/releases cannot be run as-is with the GNU Radio Companion. There are at least two variants to create a runnable gr-krakensdr/GNU Radio Companion system:

Variant 1: Running GNU Radio, gr-krakensdr and Heimdall on Raspberry Pi 4B

For this variant you will have to download from https://www.raspberrypi.com/software/ and install a desktop version of the Pi OS, install GNU Radio including GNU Radio Companion from a Linux package repository and integrate and configure the relevant software components extracted from the (non-desktop) KrakenSDR Pi image supplied by KrakenRF. This will lead to the following system architecture:



You would run the Pi's Linux GUI (and therein the GNU Radio Companion) on a display, keyboard and mouse connected to the Pi 4B via HDMI and USB. As an alternative you may use a PC as a remote GUI console, e.g. on Windows with mstsc.exe:

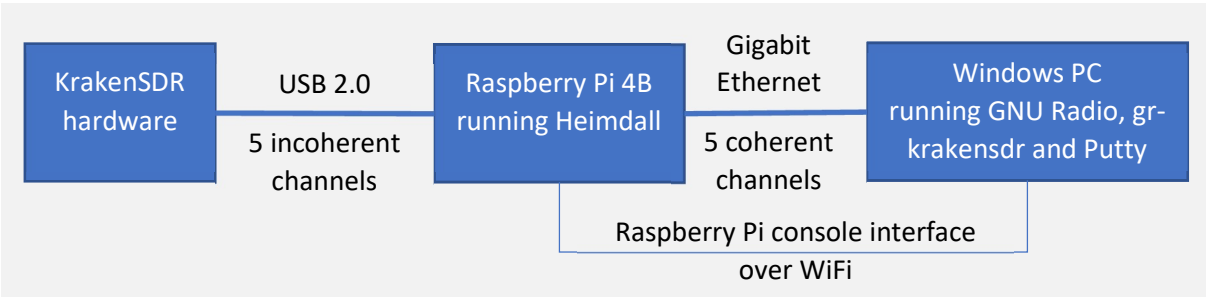


Heimdall is the KrakenRF-supplied software taking the clock-synchronized but not yet coherent samples from the KrakenSDR's five channels via USB 2.0 and sending it as five coherent channels further downstream the processing chain, e.g. to the direction-finding application krakensdr-doa or in our case to gr-krakensdr. Heimdall is quite a marvel, but that's another story.

The main disadvantage of this architecture is that GNU Radio digital signal processing and the GNU Radio Companion GUI can be very power-hungry and may easily ask too much even of a Pi 4B.

Variant 2: Running GNU Radio and gr-krakensdr on a PC

To improve signal processing performance GNU Radio and gr-krakensdr should be processed by a powerful PC connected to the Pi 4B via Gigabit Ethernet.



In this architecture the Pi 4B with installed (non-desktop) KrakenSDR Pi image will still be required, having a valuable role serving the KrakenSDR's USB 2.0 interface and running Heimdall to synchronize the KrakenSDR's 5 channels into 5 coherent data streams. Running Heimdall on the Pi 4B (and not on the PC) has the benefit of allowing a much longer physical distance between the user interface and the KrakenSDR, because USB 2.0 is limited to about 5 m whereas Ethernet will work across several ten meters (in theory up to 100 m).

WiFi via a router instead of Gigabit Ethernet may be overwhelmed by the KrakenSDR samples' data volume, but will still be required for the much less demanding Pi 4B textual console interface.

The PC operating system may be Linux, Windows or MacOS. However, the remainder of this document will focus solely on a Windows implementation.

2. gr-krakensdr

gr-krakensdr is supplied by KrakenRF at https://github.com/krakenrf/gr-krakensdr as a number of

- gr-krakensdr GNU Radio OOT (Out Of Tree) blocks[1] serving as interfaces between Heimdall and the myriad of GNU Radio signal processing blocks
  and
- gr-krakensdr GNU Radio application examples letting you run some simple GNU Radio Companion flowgraphs.

Both the gr-krakensdr OOT blocks and the gr-krakensdr application examples must be installed into your local the GNU Radio application.

3. Installation overview

To install the necessary software components on Windows you will have to

1. Install GNU Radio (including GNU Radio Companion)
2. Install the OOT blocks and application examples of gr-krakensdr into GNU Radio
3. Configure an Ethernet connection to the Pi 4B
4. Install Putty (putty.exe) as a remote console over WiFi and SSH to the Pi 4B

On the Pi 4B you will have to

1. Install the (non-desktop) KrakenSDR Pi image as supplied by KrakenRF at https://github.com/krakenrf/krakensdr_doa/releases/
2. Configure a WiFi connection to your PC
3. Amend some of the KrakenSDR Pi image's configuration files
4. Configure an Ethernet connection to the PC

The remainder of this document is a step-by-step instruction how to do this.

4. Installing GNURadio on Windows

You may access the GNU Radio tutorials at https://wiki.gnuradio.org/index.php/Tutorials and in YouTube to become familiar with GNU Radio. However, for the sole purpose of this installation this is not mandatory.

Compiling and installing GNURadio (https://gnuradio.org) directly from its repository at https://github.com/gnuradio/gnuradio on Windows is no easy task. Fortunately, there is a selection

[1] OOT blocks are third party GNU Radio-compatible software components which are not part of the set of blocks supplied with the standard GNU Radio distribution.

of compiled Windows ports listed at https://wiki.gnuradio.org/index.php/WindowsInstall from which I chose the well-maintained and regularly updated Radioconda distribution.

Radioconda is equivalent to GNU Radio and GNU Radio Companion embedded in the Conda package manager. Conda is the package manager component of the much larger Anaconda software suite at https://anaconda.org/. Installing Anaconda is not necessary for our purpose. If you want to become familiar with Conda I recommend the following introduction:
https://www.youtube.com/watch?v=23aQdrS58e0
and this Conda cheat sheet:
https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
However, deep Conda knowledge is not mandatory for the sole purpose of this installation. If ever you should be required to install a dependency package named e.g. “package_name” you may type

```
(base) C:\Users\your_account>conda install package_name
```

Download and install radioconda-Windows-x86_64.exe from https://github.com/ryanvolz/radioconda#download and take note of the instructions at https://github.com/ryanvolz/radioconda#windows.

I installed Radioconda only for use with my local Windows account C:\Users\my_account, so no admin privileges are required when making modifications.

After installation there will be a “radioconda” folder in C:\Users\your_account as well as in your Windows’ START menu of all installed apps, the latter containing an application called “Conda Prompt” which launches a special Windows console (cmd.exe) with activated Conda environment:

```
(base) C:\Users\your_account>
```

“(base)” indicates that you are now in the Conda base environment that has been installed together with and containing Radioconda. Always launch Radioconda via Conda Prompt and never via Windows standard cmd.exe or a direct link to an *.exe file in one of the C:\Users\your_account\radioconda directories.

Entering

```
(base) C:\Users\your_account>gnuradio-companion.exe
```

will start the GNU Radio

```
(base) C:\Users\your_account>gnuradio-companion.exe
>>> Warning: vocoder_codec2_decode_ps - option_attributes are for enums only, ignoring
>>> Warning: vocoder_codec2_encode_sp - option_attributes are for enums only, ignoring
<<< Welcome to GNU Radio Companion 3.10.5.1 >>>

Block paths:

        C:\Users\your_account\radioconda\Library\share\gnuradio\grc\blocks
```

and the GNU Radio Companion (hereinafter abbreviated as “GRC”) in a separate GUI window.

5. Installing gr-krakensdr into GNU Radio

Download and unzip https://github.com/krakenrf/gr-krakensdr/archive/refs/heads/main.zip into a directory somewhere on your PC, say in a directory named “gr-krakensdr”.

gr-krakensdr is implemented in pure Python and YML and without any C or C++ code. As Python is an interpreted language without complier, gr-krakensdr is installed into GNU Radio without any compiling and you can ignore the compile instructions at https://github.com/krakenrf/gr-krakensdr#gnu-radio-oot-block-install-instructions.

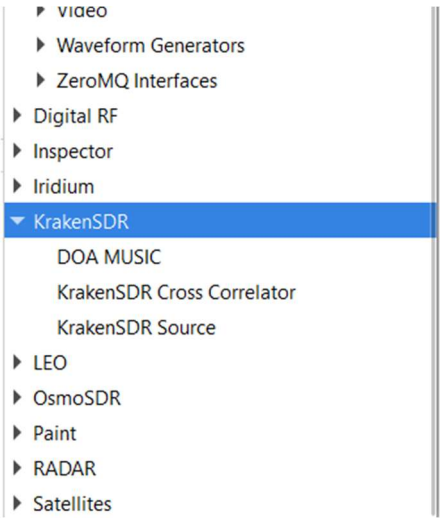
Copy the files (except “CMakeLists.txt” files) from the unzipped gr-krakensdr directory to your C:\Users\my_account\radioconda directory according to the following table:

Files	**Source directory gr-krakensdr\.....**	**Destination directory C:\Users\your_account\radioconda\.....**
*.yml	grc	Library\share\gnuradio\grc\blocks\krakensdr
*.py	python\krakensdr python\krakensdr\bindings	Lib\site-packages\gnuradio\krakensdr Lib\site-packages\gnuradio\krakensdr\bindings
*.py *.grc	examples	Library\share\gnuradio\examples\krakensdr

Launch the GRC from the Conda Prompt:

```
(base) C:\Users\my_account> gnuradio-companion.exe
```

In the GNU Radio blocks list at the lower right of your GRC GUI window you should now notice a directory “ ▸ KrakenSDR”. Opening it should reveal the three KrakenSDR OOT blocks:



You may drag e.g. “KrakenSDR Source” into the work area of the GRC, but do not yet run it as the GRC will crash with an incomplete installation and configuration, in particular if the Ethernet connection has not yet been configured and launched correctly.

6. Configuring a point-to-point Ethernet connection on Windows

For performance reasons I am suggesting you a direct point-to-point Gigabit Ethernet cable connection between PC and Pi 4B, i.e. not passing through a router. The data volume spit-out by Heimdall is quite heavy when all 5 KrakenSDR channels are active and the sample rate is high.

Activate your Windows' Ethernet interface and configure it with a valid, hard-coded IPV4 address. You may borrow some help from
https://docs.oracle.com/cd/E19657-01/html/E23956/z400096a1017631.html
or
https://kb.netgear.com/000037249/Setting-a-static-IP-address-on-your-network-adapter-in-Windows-for-direct-access-to-an-access-point
In a point-to-point Ethernet connection you may choose almost any IPV4 address, but it must be different from your WiFi IPV4 address. I am using **169.254.24.110** as the PC's IPV4 Ethernet address and **255.255.0.0** as a matching netmask. My IPV4 WiFi network address is 192.168.88.xx (= different).

7. Installing Putty on Windows

Later on you will have to configure some parameters and modify some file contents on the Pi 4B via the Pi's SSH text console, readily set-up in the KrakenSDR's Pi image. To do this from your PC via WiFi you will have to install Putty (putty.exe). Download Putty from https://putty.org/ to your Windows PC and install it there.

I am assuming here that you will access the Pi's console through your WiFi router and that your PC has a working WiFi connection to your router already.

8. Installing Heimdall on the Raspberry Pi 4B

Download the latest KrakenSDR Pi image supplied by KrakenRF at
https://github.com/krakenrf/krakensdr_doa/releases/
and install it according to the instructions at
https://github.com/krakenrf/krakensdr_doa
on your Raspberry Pi 4B. I am using https://win32diskimager.com/ to copy the image to the Pi's boot medium (microSD card or USB stick).

Before detaching this boot medium from your Windows PC, you will have to

- edit your WiFi credentials into the "wpa_supplicant.conf" file and
- make sure that there is an empty file named "ssh" (without extension)

on the boot medium's "boot" partition to enable both the Pi's WiFi access to your router and your Windows' Putty remote access to the Pi's console.

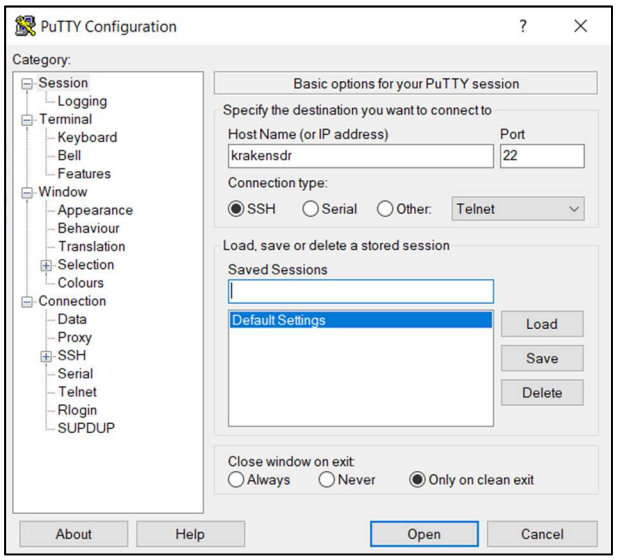
Your "wpa_supplicant.conf" may look like this:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=YOUR_COUNTRY (ISO 3166 two-character country code)

network={
   ssid="YOUR_WIFI_NETWORK_NAME"
   psk="YOUR_WIFI_PASSWORD"
   key_mgmt=WPA-PSK
   priority=2
}
```

Insert the boot medium into the Pi 4B and power it up.

The Pi's Linux console should now be reachable from your PC via Putty and WiFi. Enter the Pi's WiFi hostname "**krakensdr**" or its IPV4 address into Putty's hostname field. The IP port number for SSH is 22 and should be the default in Putty.



If the Pi's hostname does not work you will have to find the Pi's IPV4 address e.g. via the admin interface of your router.

Log into the Raspberry Pi 4B via Putty:

Preconfigured login name: **krakenrf**
Preconfigured password: **krakensdr**

Then on the Pi 4B edit the last line of the file "daq_chain_config.ini" in the directory "$home/krakensdr_doa/heimdall_daq_fw/Firmware/" with the "nano" text editor

```
krakenrf@krakensdr:~ $ nano
   krakensdr_doa/heimdall_daq_fw/Firmware/daq_chain_config.ini
```

from `out_data_iface_type = shmem`
to `out_data_iface_type = eth`

Make sure you accurately chose the above path. The nano does not accept mouse input for navigation but just arrow keys.

Write and close "nano" by CTRL-O + return + CTRL-X.

This will deviate Heimdall's output from the default "shmem" interprocess communication to the Pi 4B's Gigabit Ethernet interface.

Connect the Pi 4B via USB to the KrakenSDR and power the KrakenSDR. The Pi is running the "kraken_doa_start.sh" script by default, which is the KrakenSDR's direction-finding application. Change directory to $home/krakensdr_doa and launch the Heimdall-only mode of operation via the "./heimdall_only_start.sh" script:

```
krakenrf@krakensdr:~ $ cd krakensdr_doa

krakenrf@krakensdr:~/krakensdr_doa $ ./heimdall_only_start.sh
```

Whenever you disconnect the KrakenSDR from the Pi 4B or power-cycle either KrakenSDR or Pi 4B you will have to relaunch the Heimdall-only shell script. Starting it will automatically terminate the krakensdr_doa mode of operation. There is no explicit Heimdall-only stop script.

9. Configuring a point-to-point Ethernet connection on the Pi 4B

I am assuming the you configured the Pi's WiFi setup according to one of the options of https://github.com/krakenrf/krakensdr_doa#choose-and-set-up-connectivity-option.

The point-to-point Ethernet connection should be independent of the WiFi's DHCP server and provided with a hard-coded IPV4 address similar to the PC's Ethernet setup. Therefore, we3 have to supply the Pi with an IPV4 address manually.

Access the Pi's console through your PC's Putty. "ifconfig" is the Linux utility to manage network connections. "eth0" is the Pi's Ethernet interface.

```
krakenrf@krakensdr:~ $ ifconfig eth0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::e65f:1ff:fe27:8a47  prefixlen 64  scopeid 0x20<link>
        ether e4:5f:01:27:8a:47  txqueuelen 1000  (Ethernet)
        RX packets 41  bytes 4086 (3.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 72  bytes 18908 (18.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

As you can see, no IPV4 address is configured in the eth0 interface yet. You must now configure an IPV4 address matching the PC's IPV4 network address (169.254.24.110), but differing in the rightmost byte:

```
krakenrf@krakensdr:~ $ sudo ifconfig eth0 169.254.24.100 netmask 255.255.0.0

krakenrf@krakensdr:~ $ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.24.100  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::e65f:1ff:fe27:8a47  prefixlen 64  scopeid 0x20<link>
        ether e4:5f:01:27:8a:47  txqueuelen 1000  (Ethernet)
        RX packets 44  bytes 4362 (4.2 KiB)
        …
```

Now the Pi and the PC should be able to exchange data via Ethernet. Unfortunately, such configuration will not be persistent across boot and you may have to manually repeat its configuration after each Pi 4B boot. But as you will have to start Heimdall after each boot anyway you may include the IPV4 configuration into heimdall_only_start.sh:

```
krakenrf@krakensdr:~ $ nano krakensdr_doa/heimdall_only_start.sh

#!/bin/bash
sudo ifconfig eth0 169.254.24.100 netmask 255.255.0.0
#source /home/krakenrf/miniforge3/etc/profile.d/conda.sh <- required for systemd
auto startup (comment out eval and use source instead)
eval "$(conda shell.bash hook)"
conda activate kraken

./kraken_doa_stop.sh
sleep 2

cd heimdall_daq_fw/Firmware
#sudo ./daq_synthetic_start.sh
sudo env "PATH=$PATH" ./daq_start_sm.sh
```

Write and close "nano" by CTRL-O + return + CTRL-X and run heimdall_only_start.sh:

```
krakenrf@krakensdr:~ $ cd krakensdr_doa
krakenrf@krakensdr:~ $ cd ./heimdall_only_start.sh
```

Now connect an Ethernet cable between PC and Pi 4B.

10. Checklist

You are now ready for the first GRC/gr-krakensdr run on Windows! Make sure the following prerequisites are fulfilled:

KrakenSDR:

- The KrakenSDR is powered and connected to the Pi 4B via a short USB cable.

Pi 4B:

- The Pi4B is powered.
- The configuration changes in
  wpa_supplicant.conf
  ```
  krakenrf@krakensdr:~ $ sudo cat /etc/wpa_supplicant/wpa_supplicant.conf
  ```
  and
  $home/krakensdr_doa/heimdall_daq_fw/Firmware/daq_chain_config.ini
  and
  krakensdr_doa/heimdall_only_start.sh
  have been performed as detailed above.
- The Pi 4B is connected to your PC via an Ethernet cable.

Windows PC:

- Your Windows PC is running and has a WiFi connection to your router.
- Putty, Radioconda and gr-krakensdr are installed on your PC as detailed above.
- A suitable point-to-point Ethernet IPV4 address is configured on your Windows.

Next:

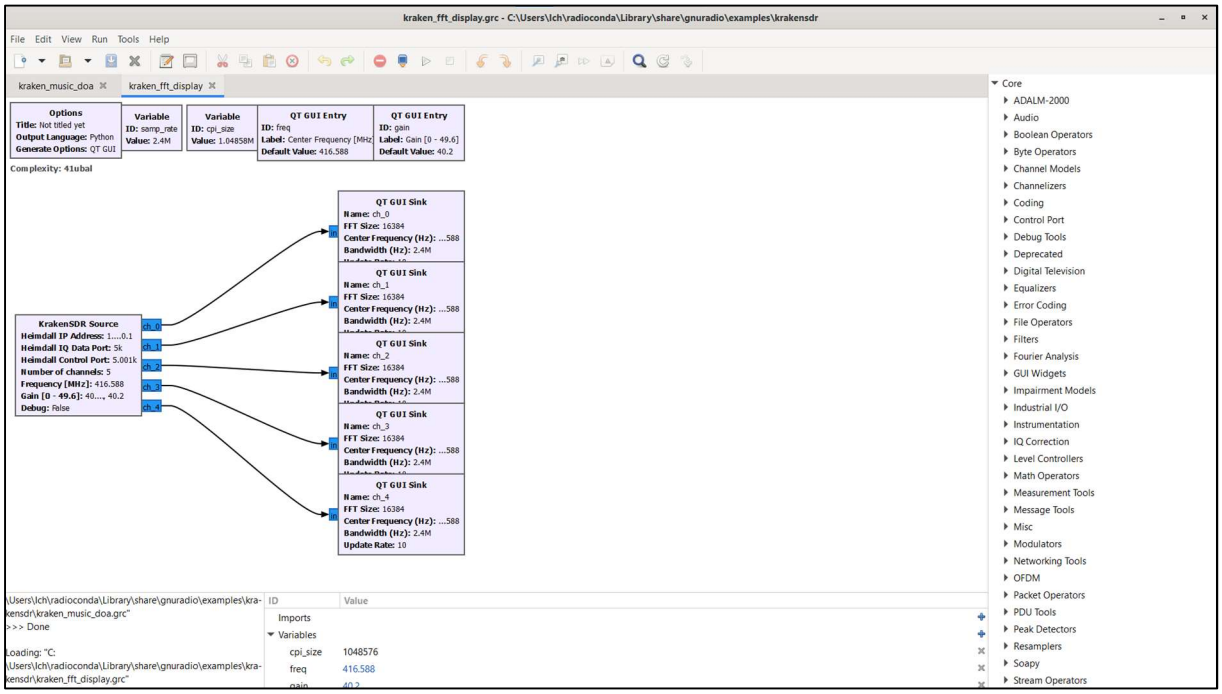
- Start Putty and GRC (GNU Radio Companion) on the PC.
- Log into the Pi's console via Putty and WiFi and start Heimdall on the Pi 4B.

We are almost there. A small configuration in GRC (Heimdall's IP address) is still missing.

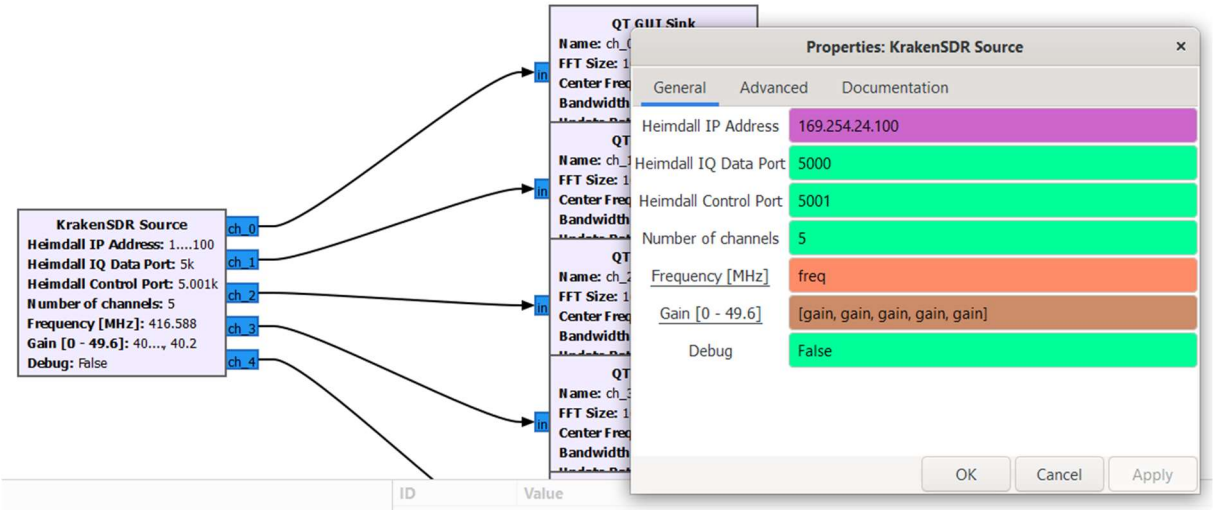
11. Running an example on GNURadio Companion

In the GRC open the file (File/Open)
C:\Users\your_account\radioconda\Library\share\gnuradio\examples\krakensdr\kraken_fft_display.grc

You should now see this screen:



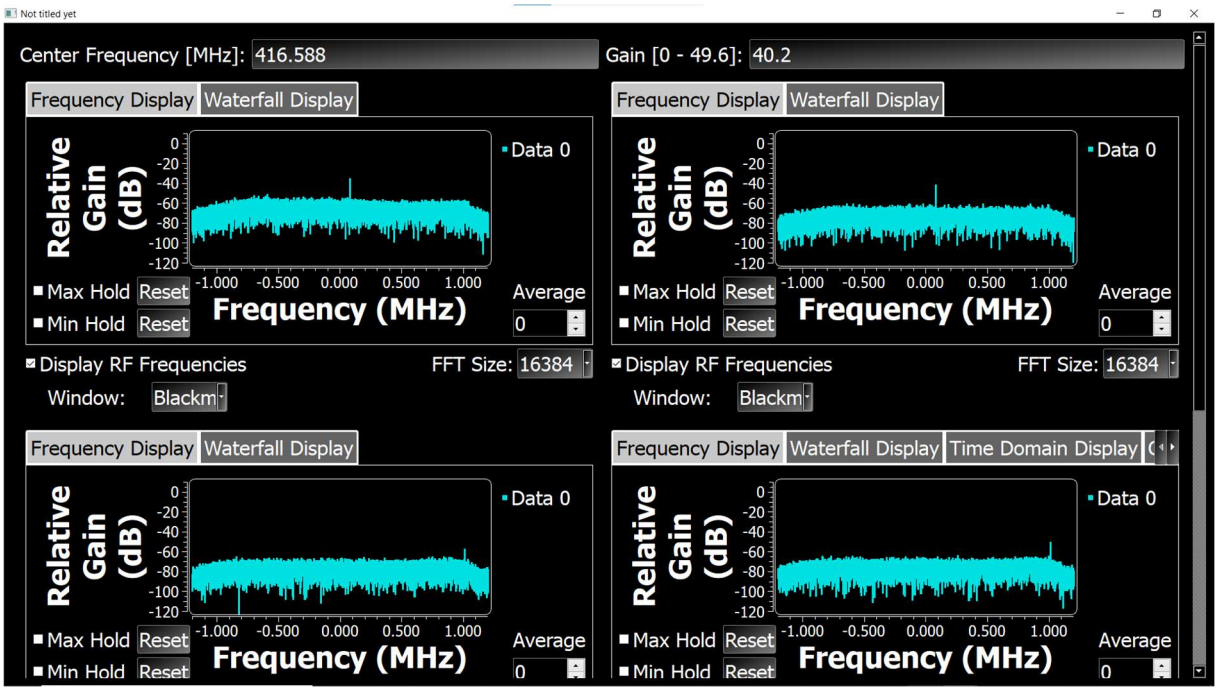
Double-click on the KrakenSDR Source block to the left and replace the default Heimdall IP address 127.0.0.1 with the Pi's Ethernet IPV4 address 169.254.24.100.



Click "Apply" and "OK".

Run the Example (Run/Execute).

After a few seconds you should now see something like this:



Congratulations!

Have fun with processing KrakenSDR’s coherent signals in the GRC on Windows!