HAARP Scanner

Contents

Technic......1
Receiving examples......2
Long duration measurement......3
Mindcontrol......4
Installation......5
Complete Reference rtl_power......7

This project will give the possibility to search and prove the existance of HAARP.
It will describe how an automated station for measurements can be realized.

The weather control is a combination of this two factors:

- Nanoparticles in the atmosphere via Chemtrails
- Stimulation with radio waves in the frequency range of 1 to 10 MHz

Technic

You can use a cheap SDR receiver (Software Defined Radio).
Here you find a description for it:

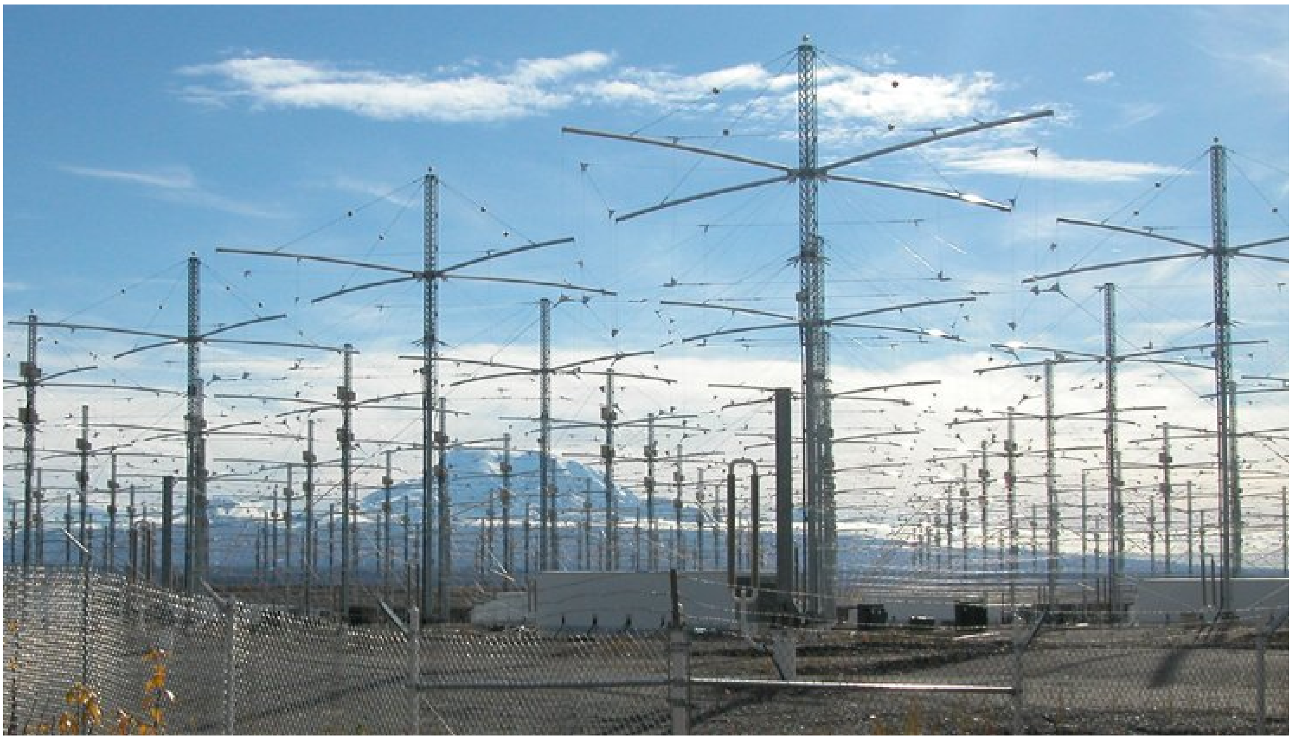
- http://www.rtl-sdr.com/about-rtl-sdr/
- https://de.wikipedia.org/wiki/Software_Defined_Radio
- https://wiki.ubuntuusers.de/Gqrx/
- http://www.amateurfunk-wiki.de/images/4/4e/Schulze07softwaredefinedradio.pdf

In this project a converter was used that has a frequency shifting for the low frequency range of 0 – 60 MHz to 125 MHz and up:

- http://www.darc-husum.de/sdr-konverter.html
- http://www.hb9f.ch/bastelecke/pdf/RTL-SDR_Stick/Up-Converter_fuer_den_RTL-SDR_Stick_Vers_B.pdf
- http://www.rudiswiki.de/wiki/AfuSDR-Rx
- https://www.rtl-sdr.com/rtl-sdr-direct-sampling-mode/

Receiving examples

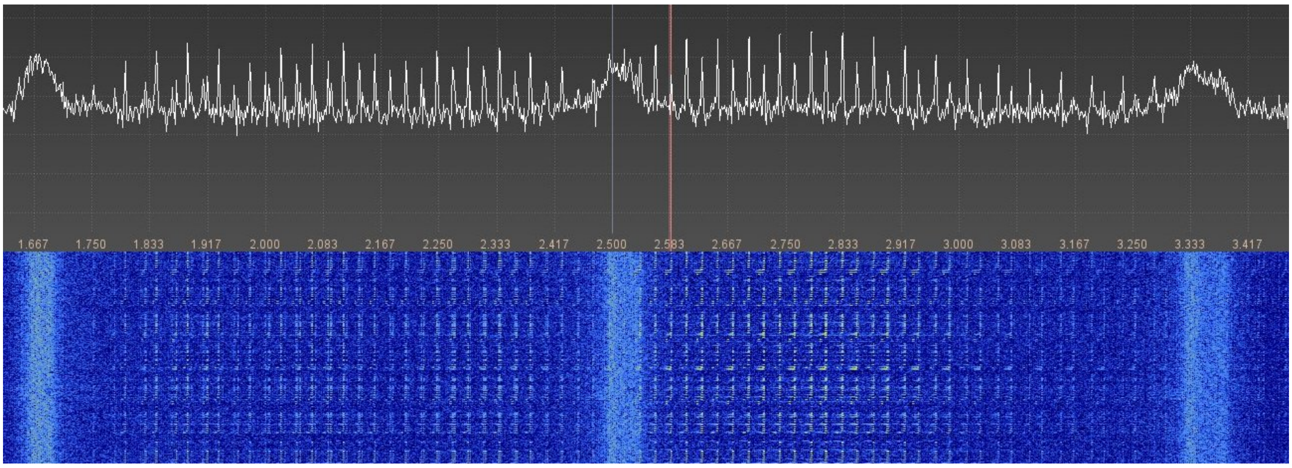
An HAARP sender is an array of antennas with an separate sender for each antenna. So it is possible to group this senders and send on multiple frequencies at the same time.



https://de.wikipedia.org/wiki/High_Frequency_Active_Auroral_Research_Program#/media/File:HAARP20l.jpg

You can see many carrier signals each side on side the next one.

It looks in an waterfall diagram like this:

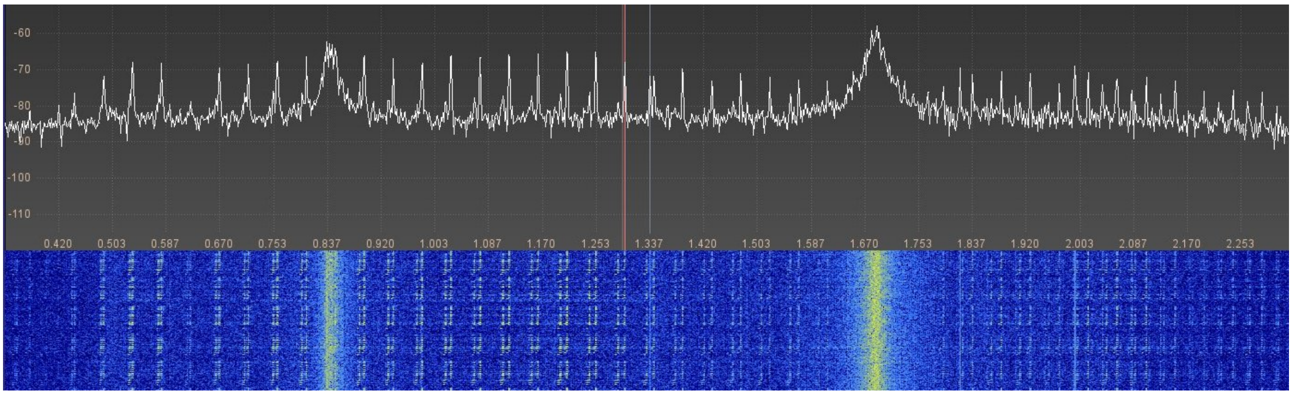


Here the frequency range 1.667 – 3.417 MHz.

Each carrier is switched on and off, so you can see a dashed line in the lower diagram.

It can be better seen in a video (see folder examples).

Here another frequency range of 0.4 – 1.67 MHz.



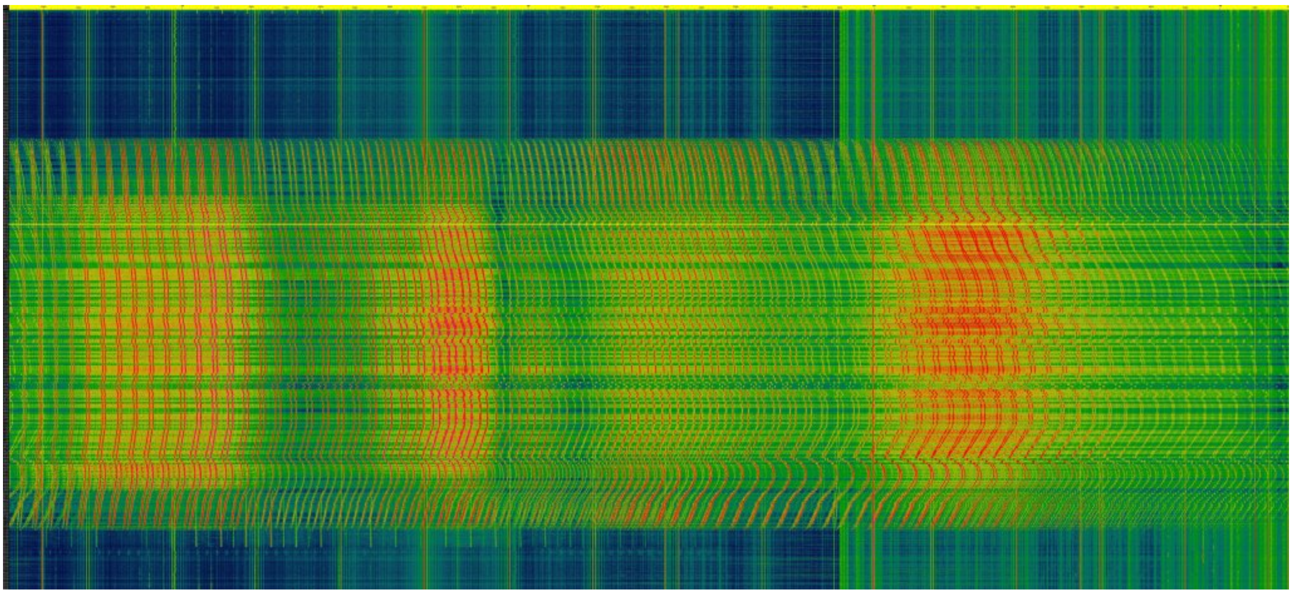
Long duration measurement

The examples where made with the software GQRX.
For long duration measurements there does not exist any software up to now.

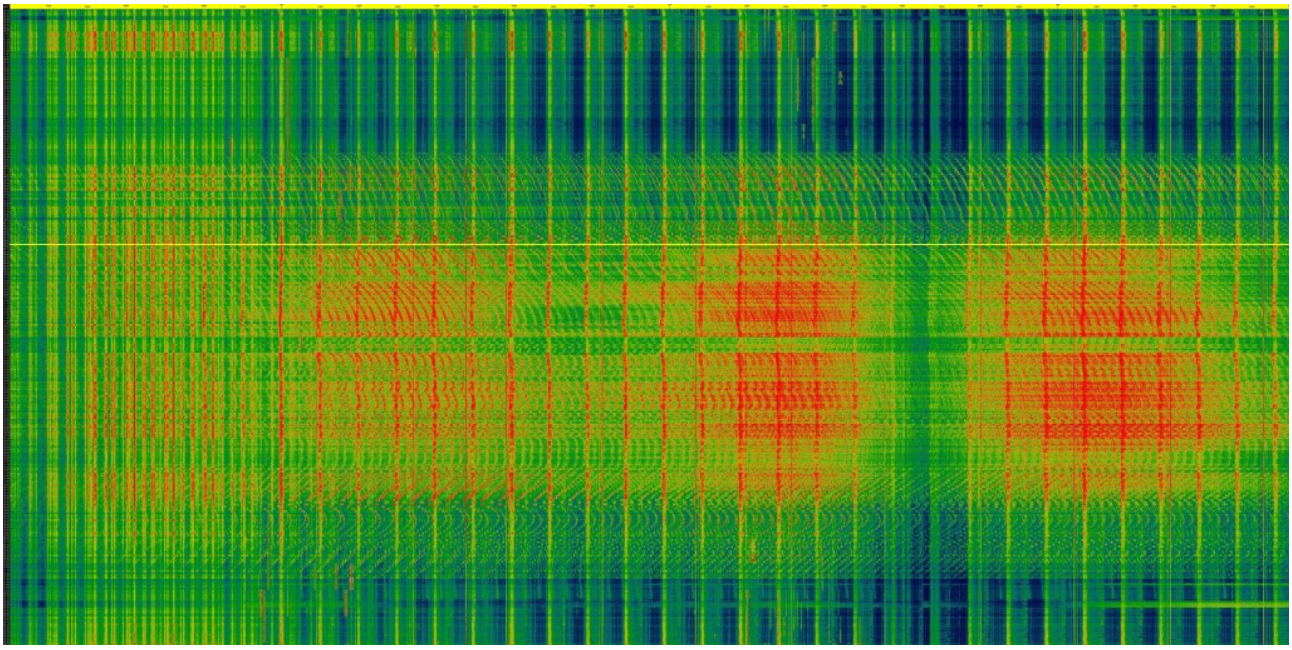
Base of the software are programs of Kyle Keen: http://kmkeen.com/rtl-power/
This includes **rtl_power** and the script **heatmap.py**, to visualize the measured data.

With rtl_power every 5 or 10 minutes a measurement of 10 seconds is done - the results are saved and visualized as heatmap.
Once a day all single measurements will be pieced together to a graphic of the day.
The raw data is packed and archived.
This needs about 61 MB a day for the archive and 46 MB for one PNG graphic.

Here is an example for the frequency range 0.5 - 4 MHz on 2017.06.17



Here the corresponding frequency range 5 - 8 MHz on 2017.06.17
9:00 is marked where the main sending starts.



Since the beginning of the measurements this "sending program" of HAARP could be protocolized continuous since 5 month.

Mindcontrol

With this software different frequency ranges can be observed simultaneously.
But of course only in the receiving range of the SDR receiver.

You will find additional frequency ranges from 60 MHz up to 960 MHz, showing abnormality in the local living region.
Here is the assumption that this is used for mindcontrol.

But the observation of the signal power like in HAARP is not sufficient for the analyze here. You need to make a spectral analysis of the modulation of the signal, to find the information you are searching for. This is not implemented so far.

Installation

In Debian you need this linux packages:

- rtl_sdr
- perl
- python
- python-imaging
- imagemagick

As standard it is defined to run the software in **/srv/SDR** .
So the zip file should be unpacked in **/srv** and everything should be fine.

The main program is **`haarpscan.pl`**.
Here only the path of the called programs is configured.

All other configurations will be done in **`haarpconfig.pm`**.
Please have a look at the next chapter or the homepage of rtl_power for understanding the parameter: http://kmkeen.com/rtl-power/

The parameter **`gain`** is for the RTL2832 chip.

With the parameter **`freq_offset`** the frequency shift of the receiver is defined to correct it in the heatmap.

With **`haarpdaemon.sh`** the haarpscan will be started as daemon.
The direct standard call is "`haarpscan.pl -d -l`"

It is not needed to run the main program as root if all rights are configured correct.
You can copy **`rtl-sdr.rules`** to **`/etc/udev/rules.d/`** for this.

With "`haarpscan.pl -r`" all frequency ranges will be listed and the needed pathes for the data are created.

For debugging and showing the executed calls use "`haarpscan.pl -t -s -b 2`".

The script **`piecing.sh`** can piece the already measured data of the day together without deleting it. This is the difference between option -p and -q.

All options can be shown with call of the main program without parameter:

```
haarpscan.pl [option]
             -d daemon mode
             -t single test run
             -l with logfile
             -r range list and directory creation
             -p piece together heatmaps
             -q piece together heatmaps and delete (daily job)
             -u cleanup files (daily job)
             -s simulate without execution for debugging
             -c [file] analyze csv file
             -a [path] reprocess csv data in path
             -b [level] debug
```

And now much success with your analyze …

Complete Reference rtl_power

If you want to stream directly to a file, provide the file name as the final argument. On some platforms this may hit a 2GB file limit. Use output redirection (`rtl_power ... > log.csv`) for unlimited file size.

`-f lower:upper:bin_size [Hz]`

Set a frequency range. Values can be specified as an integer (89100000), a float (89.1e6) or as a metric suffix (89.1M). The bin size may be adjusted to make the math easier. Valid bin sizes are between 0.1Hz and 2.8MHz. Ranges may be any size.

`-i <integration_interval>`

Collect data for this amount of time, report it and repeat. Supports 's/m/h' as a units suffix. Default is 10 seconds. Minimum time is 1 second, but for extremely large ranges it may take more than 1 second to perform the entire sweep. Undefined behavior there.

`-e <exit_timer>`

Run for at least this length of time and exit. Default is forever. Like the other times, this supports 's/m/h' units.

`-1`

Enable single-shot mode, default disabled. Perform a single integration interval, report and exit. It is not necessary to use `-e` with this option.

`-d <index>`

When using multiple dongles, this indicated which. You can also identify dongles by the text in the serial number field of the EEPROM.

`-g <gain>`

A floating point gain value. The dongle will use the closest gain setting available.

`-p <error>`

Correct for the parts-per-million error in the crystal. This will override a ppm value retrieved from eeprom.

-w <window>

The window is a shaping function applied to the data before the FFT. Each will emphasize or deemphasize certain aspects. The default is none (aka boxcar, rectangular). Options include: hamming, blackman, blackman-harris, hann-poisson, bartlett, and youssef.

-c <crop_percent>

The crop sets how much of the bandwidth should be discarded. 0% discards nothing, 100% discards everything. The edges of the spectrum are lower quality than the middle. There is less sensitivity, gain roll-off and out-of-band aliasing. Higher values of crop will produce a better spectrum, but do so more slowly. Values may be a decimal (-c 0.1) or a percent (-c 10%). Default crop is 0%, suggested crop is between 20% and 50%.

This setting has no effect on bins larger than 1MHz.

-F 0 | 9

Not exactly the best named option, this configures the downsampler and the downsample filters. Downsampling is only used when the total bandwidth range is under 1MHz. (Like in the radar example above.) Omitting the `-F` option uses the default downsampler, rectangular. This downsampler is very fast but has bad spectral leakage.

Filters with minimal leakage are `-F 0` and `-F 9`. 0 is a plain filter, but has bad droop at the edges of the spectrum. 9 uses the same filter as 0, but has a 9-point FIR filter to correct the droop. Rectangular needs the least cpu, 0 needs more, and 9 most of all. It is suggested to use 0 with `-c 50%`.

-P

Enables peak hold. The default behavior is to average across time. Peak hold uses the maximum value across time. Note that averaging improves the SNR, and peak hold will tend to make a spectrum look much worse.

-D

Enable direct sampling. Requires that you have first modified the dongle for direct sampling.

-O

Enable offset tuning. Only applies to E4000 tuners