Echoes user manual

Last updated on: Mar 06, 2021 - rel. 0.29

Dear reader,

this document derives from the manual ODT file released with version 0.26 with some additions related to 0.27. I've never written a wiki before, so please drop me a line if you see something wrong, unclear or have any other question about this software, thanks. GMB

Introduction

Echoes it's a radio spectral analysis software for RTL-SDR devices, designed for meteor scatter purposes. It's open source software currently hosted on SourceForge:

https://sourceforge.net/projects/echoes/

RTL-SDR are very cheap Software Defined Radios that uses DVB-T TV tuner dongles based on Realtek's RTL2832U chipset. With the combined efforts of Antti Palosaari, Eric Fry and Osmocom it was found that the signal I/Q data could be accessed directly, which allowed the DVB-T TV tuner to be converted into a wideband software defined radio by the means of a special software driver.

This driver must be installed in alternative to the official driver shipped generally in a CD-ROM along the dongle, if not already installed in your OS in the factory.

Echoes doesn't demodulate neither decode any human-made signal. Its main goal is to analyze and record the total power of natural signals and generate screenshots and tabular data (CSV, GNUplot) output in presence of particular peaks in a selected narrow range of frequencies. Since there is no demodulation, there is no provision for audio listening, except for a notify sound when an event has been recorded.

Echoes expresses the output power as dBfs – decibels at full scale – since the output signal from RTL-SDR is not calibrated. The zero dBfs value means the maximum amplitude of that signal. Lower values give negative dBfs.

A conversion to dBm requires to multiply the dBfs by a conversion factor depending of your hardware and the working frequency chosen. Here is an example of such calibration methods. Sincerely I didn't had the time to try it yet; I placed the link here just as a *memento* for the future.

Acronyms

Acronym	Meaning
AGC	Automatic Gain Control
CSV	Comma Separated Values
DSP	Digital Signal Processing
DVB-T	Digital video tuner
RTL	Realtek: producer of RTL-8139 receiver chip.
SDR	Software Defined Radio
UI	User Interface
GUI	Graphical User Interface
GRAVES	Grand Réseau Adapté à la Veille Spatiale, radar station located in Dijon (F) aimed to space debris monitoring.
RPM	Redhat Package Manager
OM	“Old Man” radio amateur
OSS	Open Source Software
RF	Radio Frequency
FFT	Fast Fourier Transform

Changes

v.0.18 : Jan 11, 2018

First official release

v.0.19 : Jan 17, 2018

1. fixed bug "echoes.dat - device not open"
2. added Italian manual.
3. Starting from this version, sources are available on SF as tarballs and srpms.

v.0.20 : Feb 21, 2018

1. Corrected manual, par.7.1 (“Stop after...” in place of “Shot duration”)
2. Progressive event count is zeroed at midnight and a new CSV statistic files is created
3. Improved readability in full report: the checkbox values are no more numeric 0/2 but strings NO/YES.

4. Full report files have now a naming convention.
5. Fixed .plt files to generate graph titles regularly formatted
6. The waterfall window can be shrunk enough to fit netbooks screens
7. The customizable logo set in Site Infos dialog is now showed on waterfall window too.
8. Fix: corrupted CSV files caused crash while stopping acquisition, during postprocessing. Fixed by skipping the malformed records.
9. HTML files are resources. No more need to install them.
10. IQ buffers have now tunable size, but default is still 100% that was the previous hardcoded value
11. Added notch filters, they can be added by right-clicking on ruler
12. Added some more checkboxes to report tab to allow generating reports with only plots included (no screenshots) and to restart acquisition automatically after report generation
13. Removed obsolete preference (autoclean)
14. Changed RTS file structure, that now is subdivided in sections. The sample RTS files have been updated to this new format.

---

v.0.21 : May 25, 2018

1. At 00.00 of each day, the csv, plots and shots are stored in dedicate subdirectories, one for each day (Daily Archive) .
2. The shots/plots are subdivided in further categories/subdirectories: iperdense/underdense/fakes on a event lasting basis.
3. The file names contains, besides the configuration name and date, also the acquisition type (automatic, periodic, continuous) in place of the time.
4. Only one statistic CSV will be produced for each day for the active configuration.
    - In automatic acquisition, it is opened in append mode
    - In continuous and periodic acquisition, it is created new for each session, eventually truncating an already present one.
5. Changed the "Erase files before acquisition start" checkboxes: now the first one deletes only statistic CSV files, while the second regards all the other files (shots, plots and reports).
6. Some installation-dependent parameters previously kept in RTS files, like the windows geometries, the gnuplot path and the events count, have been moved to a local configuration file (The registry under Windows, .config files under Linux) in order to allow the use of RTS files produced by other stations without impacts on those local settings.
7. The "Data lasting" and the "Erase files before..." controls have been moved from Output tab to the Preferences tab.

8. Added in Output tab the "Join time" control. It specifies the minimum time distance between consecutive echoes to be considered as a single event. The default is 1s. The count restarts at each echo, in order to make possible to recognize long dashed patterns as a single overdense event.
9. The area covered by the events - meaning the area covered by all pixels having power higher than lower threshold and contiguous to the maximum peak - is added as new columns and valorized only at Fall front (in other rows is zero)

---

v.0.22 : Aug 11, 2018

1. Statistic CSV: added 3 new columns:
    - Area echo interval: it's the area covered by the rectangle having height equal to event lasting and width equal to the detection interval. The ratio between the area covered by the echo and this area is useful to discriminate fake events.
    - Peaks count : counts all the peaks that crossed the higher threshold, including the maximum peak. It's another useful data for fake events discrimination.
    - LOS (Line Of Sight) speed of the event, calculated from its Doppler shift.
2. Added a control to reserve an amount of disk space to be kept free, in MB. If the available space falls below this limit, the acquisition stops spontaneously.
3. Thresholds controls now allow to set values with one decimal, for a finer thresholds tuning.
4. PPM error compensation now allows 2 decimals precision. The integer part is set in hardware, as in previous versions. The decimal part instead is used to apply a transparent offset to the waterfall.
5. The N filtering has been replaced by a sliding average of the last n scans, where n is the value set in spinbox.
6. Added an archive flattening button, that allows to specify a single directory where to copy the entire content of the hierarchical archive of the current configuration. Only the files together, without subdirectories. This is useful when building a video joining all the images.
7. Now it's possible to mix all the three kinds of plots in the same full report. The gnuplot data files generated in different days with different plot types can be mixed in the same full report. The previous versions instead plotted all the files using the format set when the report has been requested.
8. Automatic report generation: a checkbox has been added to set this feature. The acquisition stops at midnight UTC, the reports are generated, the files of the day are archived then the acquisition restarts automatically

---

v.0.23 : Oct 03 2018

Fixed the following bugs:

1. automatic report not working under linux

2. after an acquisition session longer than 1 day it is possible to generate a manual report (by pressing the button). This didn't work if Echoes were restarted before pressing the report button.

---

v.0.24 : Dec 24 2018

1. Fixed bug: the files in archive older than the expiration date weren't deleted.
2. Fixed bug: statistic CSV: when two consecutive peaks were joined, the area echo and area interval fields could assume negative values
3. Improved false positive filtering in automatic mode: besides event lasting, through some calculation now Echoes can discriminate most of false positives due to Graves carrier (still TODO: make editable the filter parameters) added two new preferences:
    - Enable event notification sound ("ping") and
    - Absolute thresholds (to apply the thresholds on S value instead of S-N)
4. The instantaneous S, N and S-N values are now showed at right-top corner of waterfall window. When taking a shot, the values showed are the peak ones, in order to match the instantaneous spectra graph at window's bottom.

---

v.0.25 : Feb 14 2019

1. fixed bug reported by Kevin Palivec. A missed signal/slot connection could cause problems in program initialization

---

v.0.26 : May 31 2019

1. There is a new of thresholds mode: besides absolute and differential (the default) the "automatic" mode (be careful about confusion with the word "automatic", already used for acquisition) has been added. In this mode the thresholds must be specified as percentage of variation of instantaneous S-N against the average S-N, resulting in mobile thresholds that can self-increment or self-decrement depending of variations of the background noise level N along the day, preventing many fake captures.
2. The mode can be selected in the Preferences tab. Obviously, each time this preference is changed, the thresholds values in Output tab must be adjusted.
3. The postprocess carriers filter, introduced in 0.25, can now be excluded from GUI by unchecking the apposite checkbox in Reports tab.
4. A new postprocess filter has been added, suited for lightings, artificial electrostatic discharges and other kinds of disturbances producing an horizontal pattern covering the entire waterfall's bandwidth. This filter can be excluded by unchecking the apposite checkbox in Reports tab.
5. In HTML report the color scale used in counting tables now matches the colorgramme scale and the zero value is rendered in black. The table orientation remains the same.

6. Fixed the calculation of interval areas and echo areas in CSV reports; the lightings filter is based on these values and a wrong calculation caused the filter to fail sometimes.
7. On the waterfall, the average S-N difference value, used as reference for automatic thresholds (1), is now displayed at the right-top corner (brown color)
8. Still on waterfall, at bottom side, near the event counter, the geographical coordinates of the station have been added, since it can be an useful info to keep in the screenshots. The coordinates must be set in Reports/SiteInfo dialog.

---

v.0.27 : Jan 07 2021

1. Added sample rate downsampler to perform FFT on a narrower band than the entire sample rate. The old "bandwidth" control in waterfall window is now called "zoom" because the real bandwidth, after the downsampling, is tuned in main window "FFT" tab.
2. Added support for device "UDP server" that is another Echoes instance whose IP address has been specified in "preferences" tab. An Echoes instance connected to an UDP server becomes an UDP client. The UDP client clones the settings in "device" tab directly from the server, while all the other controls can be changed and saved independently.
3. Reporting: the colors scale used in hourly count tables (Colorgramme lookalike) are no more absolute but relative to the minimum/maximum value included in the report: the black value means zero, red is the maximum value.
4. New command line switch -d to activate the dumb server mode: the Echoes instance launched with -d works as an UDP server forwarding samples to other Echoes listening instances without recording any data for itself.
5. In statistic CSV, three new columns has been added:
    - "Unix timestamp" to match the time labels used in plots,
    - "Average S" (average of S values) used by saturation fake filter
    - "Std.dev." standard deviation of S for future purposes
6. Added new FFT windows "Hann", "Flat top" and "Blackman Harris 7°"
7. Nicer representation of dBms scale numbers in side graphs (when compiled with Qt >= 5.12)
8. Fixed HzShift calculation. Now it's the difference between the frequency when an event starts and the central frequency of the waterfall.
9. Added sample rate 226000 Hz for dongles.
10. The "Start" button has been split in two buttons "Start" and "Stop" to avoid unwanted acquisition restarts
11. Added a new "Generate Screenshots" checkbox to enable/disable the screenshots production

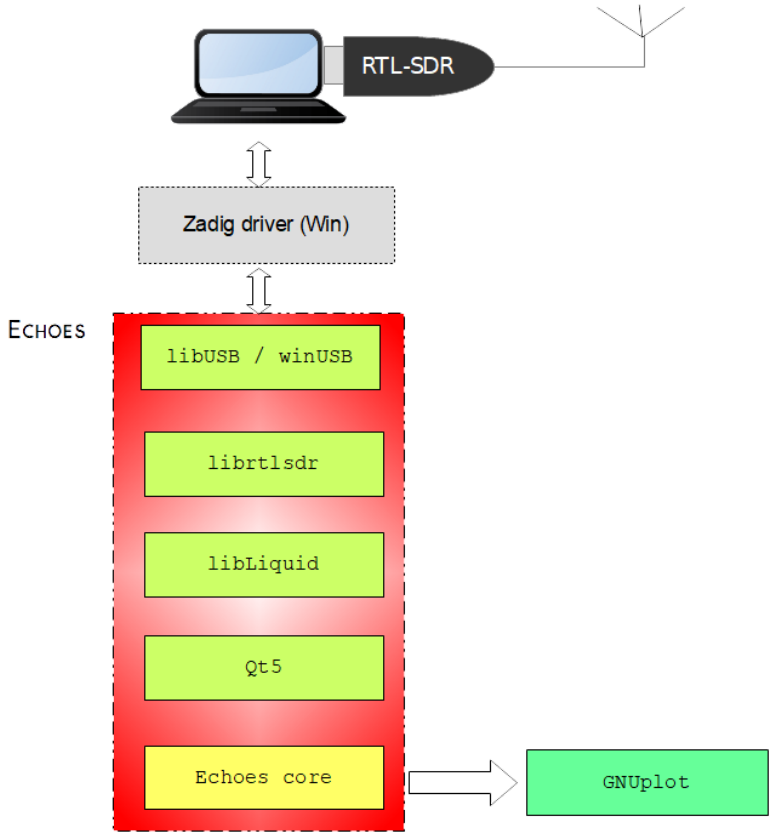
12. Added a new waterfall label "TOO NOISY" that appears when the N value exceeds a threshold set in "preferences" tab when the label appears, Echoes continues the acquisition but stops capturing events until the noise falls below that limit.
13. In "preferences" tab is possible to choose an alternate "Black&white" waterfall palette, besides the classical "Color"
14. The I/Q samples buffer size and FFTW flags controls in "fft" tab have been removed. The first because obsoleted by the downsampler feature, the second because the FFTW dependency has been removed from the project and replaced with liquidsdr library, that implements both the downsampler and the FFT.
15. Removed the main window's controls locking when acquisition is active. Some changes are immediately applicable, while others become active after acquisition restart.
16. The ODT manual has been replaced by a Wiki, published on SF. A snapshot of the wiki is deployed with the software.
17. The translations (currently Italian only) won't be maintained anymore, unless volunteers come out.
18. The rts files values are now expressed only with integers and strings in order to make them easier to edit manually. The previous float values have been converted to integers by multiplying them by a power of 10, while QDates are now expressed as strings.
19. Renewed program icons and logo

Overview

External dependencies

This manual is about what the yellow box at the bottom in the image below does. The remaining boxes are the hardware and software components that *Echoes* needs to do its job, components that are shortly described below. Curious people can follow the links for more information about these components.

ECHOES DEPENDENCIES



The Zadig tool is needed only once to install the patched WinUSB drivers. If you already used HDSDR or SDR# on the same machine, they are probably already present. These drivers are needed by user applications to access USB devices, that normally are managed by the operating system only. Please note that the original DVB-T driver supplied with the dongles cannot work if you use the dongle as SDR.

What wrote above is intended for Windows systems; but under the penguin the approach is similar: here the user applications access the USB ports by the means of libUSB library. Here too, the Linux distributions provide a driver for DVB-T dongles. These drivers will be blacklisted after the installation of the rtl-sdr package to avoid conflicts.

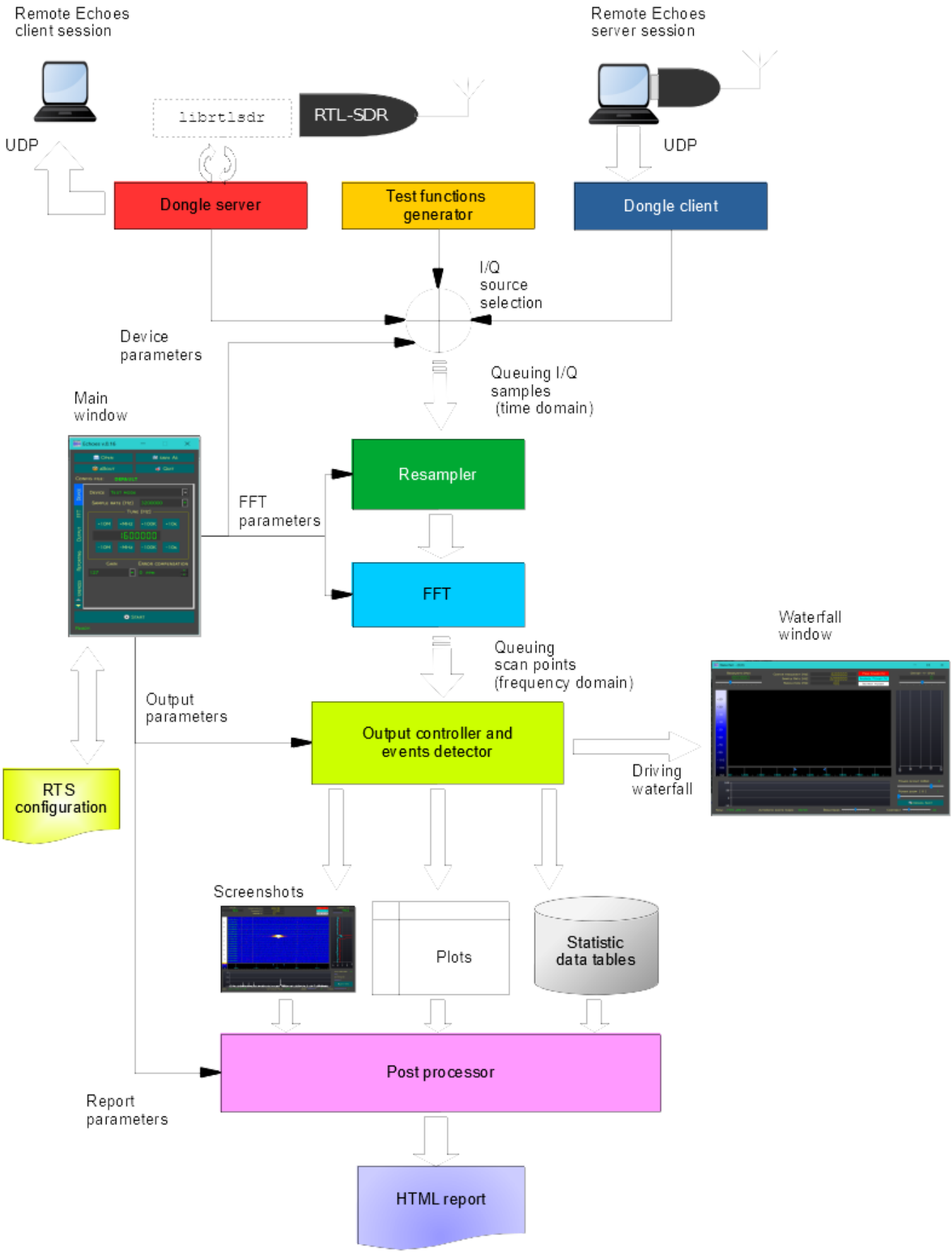
librtlsdr is an open source shared library available both for Windows and Linux. Under Linux, this library is part of the *rtlsdr* package (its name may vary depending of distros) that must be installed separately. Under Windows, this library is already linked in the executable.

libLiquid since rel.0.27 replaces libfftw because Liquid is a multipurpose DSP library that - besides FFT - includes other functions added in 0.27, like the rational resampler and more FFT windowing functions. Under Windows, this library is is already linked in the executable, while under Linux it can be found as a package that must be installed separately. Please be careful: **the version must be 1.3.2 or newer and released in July 2020 or later** .

Finally, GNUplot is a well known program for data plotting. It is not mandatory to install *gnuplot* to run *Echoes*, unless you want to run the program in console mode, when screenshots are not possible, and generate reports. In this case, *gnuplot* is required to plot the numerical dump files (.dat) generated by *Echoes* to get image files to be included in the report.

Core

Let's see now what is inside the yellow box seen above. The figure below resumes its functionalities in a block scheme. It's just a little more complicated than was in 0.26.



Starting from 0.27, *Echoes* can process IQ samples coming from remote Echoes sessions through UDP network messages. Any *Echoes* instance connected to an USB dongle becomes implicitly a *dongle server* listening the

UDP port specified in *preferences* tab (default is 12345 - if you have a better number let me know).

A secondary Echoes instance can receive the samples by specifying in *preferences* the same port number and the IP address of the server. On the server that IP address must be set as 0.0.0.0.

Once set the device and other desired parameters in main window (more detail in [Basics]), by pressing the *Start* button the selected device starts to produce IQ samples in time domain that are enqueued and sent to the radio acquisition thread.
This thread performs resampling (green block) and FFT (cyan block) and produces a frequency domain spectra, (shortly *scan*).

Each scan is then queued and forwarded to the output thread (pale green block) that draws a new row on the waterfall, detects events according to the given parameters and generates screenshots, plots and statistic data tables.

These data files are post-processed (pink block) and archived under a determinate directory structure. Post-processing is started on each new day at midnight UTC.

If required, once the given amount of daily data have been collected, a full HTML report (gray block) can be generated. The *Reporting* tab allows some report content's customization.

The settings changed in the windows are saved in a default configuration file (yellow/white block) that is silently reloaded each time the program is restarted. These settings can be saved with different names and retrieved later as needed.

Installation

Binary packages

Windows platform

Echoes installers are available for

- Windows 32 bit (*Install-echoes-0.27-Win32.exe*)
- Windows 64 bit (*Install-echoes-0.27-Win64.exe*)

A prerequisite for installation is the presence of the *WinUSB.dll* driver library. Such libraries are available at Zadig into a nice installer. So, when Zadig installer asks which DLL should be installed, please select WinUSB.dll.

When the *Echoes* installer finds another version of the program on your computer, it prompts the user for automatic removal. This prompt in fact does not take in count the program version, so a downgrade is still possible in case a newer version doesn't work as expected.

After old package removal, the installation of the new one proceeds. This process produces a log file echoes-install.log that is saved in the user's *documents* directory, to be analyzed in case of install / uninstall failures.

Linux platforms

Thanks to OBS, Echoes is available on SourceForge as binary packages for many distributions:

- Linux RPM packages for i586 and x86_64 (for *OpenSuse*, and other RPM-based distros)
- Linux DEB packages for i586 and x86_64 (*Debian 9*, *Ubuntu Stretch* and newer)
- Linux XZ packages for x86_64 (*ArchLinux*)
- Linux for *Raspberry PI* (*Raspbian 9 and 10*)

So, once downloaded, a package can be installed through the facilities provided by your distro to install binary packages (*rpm*, *dpkg*, *pacman* etc.).

A distribution is supported when it provides all the required dependencies, that must be installed before *Echoes*:

1. *rtl-sdr* library
2. *rtl-sdr-devel* (headers for building purposes only)
3. *liquidSDR* library (July 2020 or later – older versions miss the *rresamp* module)
4. *Qt5* (version 5.6 or later) including *QtCharts*. Newer Qt5 version
5. *pulseaudio* or *sox*
6. *GNUplot* (optional)

Sources

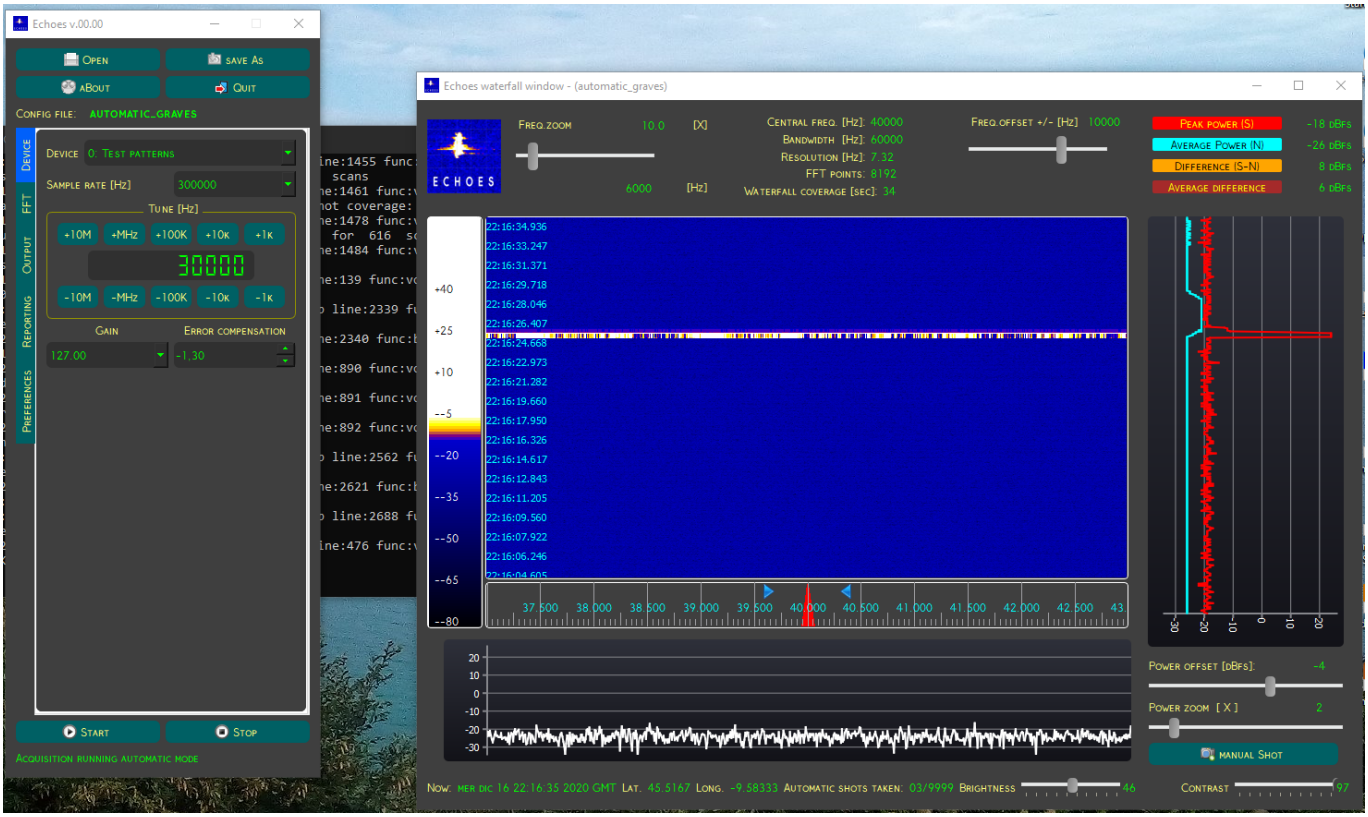
The program sources are available as tarball or packetized as source rpm.

Basics

Launching from GUI

> **Warning:** the RTL-SDR dongle must be plugged in before invoking the program because it won't be recognized if plugged in later.

When clicking on *Echoes* desktop icon, the program starts 'naked', without additional command line arguments, using the default parameters. The program will show two windows, the *main window* and the *waterfall window*. The picture below has been taken on a Windows machine so it shows a third window too, the *console window*.



1. The *Main window* (left side) contains the controls related to tuning, acquisition, data recording and report generation.

2. The *Waterfall window* (right side) contains the real-time graphical output of the program: a *waterfall diagram* with colors scale and tuning ruler, an *Instantaneous power* graph under a *Power history* graph at its right and zoom/offset sliders on frequency and power. The waterfall scrolls downward. Echoes works exclusively with UTC time; all the time references present in the logs and in the waterfall are in UTC.

3. The *Console window* (in background) opens automatically under Windows platforms to display some status messages and warnings generated by *rtlsdr* library. Moreover, this window can display also the program log (*echoes.log*) when Echoes is started with -v option. Under Linux, the same behavior is achieved by launching the program from a terminal shell session.

Now, before continuing the GUI description, let's talk a little about the command line options first.

Launching from a shell

Under Windows, open a command prompt *CMD.EXE* window, change the current directory to:

```
C:\program files\GABB\echoes
```

this because the installer doesn't insert that directory in user's path, so you must find the executable before starting it with the following command:

```
echoes --help  <Enter>
```

Under Linux the command is the same, but it's not required to change the directory first, since *Echoes* is installed in a directory included in $PATH (*/usr/bin*).

The above command recalls the *command line help*, listing all the available options when starting *Echoes* : *--help* is one of these.

```
pi@lamponel:~/echoes-git/trunk/echoes $ ./console_echoes --help
+++Console application+++
Usage: ./console_echoes [options]
Echoes is a RF spectrograph for RTL-SDR devices designed for meteor scatter v.XX.YY.
Architecture: arm  ABI: arm-little_endian-ilp32-eabi-hardfloat
(C)Giuseppe Massimo Bertani 2020.

Usage : echoes [options]


Options:
  -h, --help      Displays this help.
  -l <language>   loads the language qm file specified (defaults to local
                  language, otherwise english).
  -s <config>     loads the settings from user config file given.
  -w <wdName>     sets this directory as working directory instead of
                  $HOME/.echoes
  -n <level>      log level:
                  0: do not create a program log
                  1: only fatal messages (crashes) are logged
                  2: logs fatal and critical messages (alerts about possible
                  crashes)
                  3: logs warnings too (the program does not behave as expected)
                  4: logs status messages too (useful for console mode).
                  5: logs everything including debug messages (huge logs!).

  -c              console mode: acquisition and capture starts automatically, no
                  windows will be shown.
  -d              dumb console mode: acquisition starts automatically, no
                  windows will be shown and nothing is captured.  The IQ buffers
                  are forwarded to external clients only.
  -r              restores the hardcoded default settings (config_file if given
                  will be ignored).
  -v              verbose debug output on text console.
pi@lamponel:~/echoes-git/trunk/echoes $
```

Generally speaking, none of these options are mandatory to start *Echoes* with GUI, by clicking its icon on the desktop, but someday you could need to use them in a script or a batch file.

-c

The -c option starts *Echoes* without a GUI (console UI). The configuration used is the default one, (*default.rts*) unless another configuration file has been specified with the parameter -s<config_file>. The acquisition starts immediately. This operating mode is suited for headless, stand-alone stations. Being no GUI active, the snapshots can't be generated; but if *Generate GNUplot* has been enabled in the chosen configuration file, each event will generate a data file that can later be open with GNUplot. The configuration files can be easily edited with a text editor or can be created with *Echoes* itself by opening the program in GUI mode, set the desired parameters then save the new configuration with a name different than *default*.

Linux only: What said above, is valid for Linux too, when invoking *Echoes -c* from an X-terminal window. The *-c* option prevents *Echoes* to display windows, but a running X server is still mandatory to run it.

A Linux system could nevertheless be started in *multiuser mode* without graphics at all. On systems supporting *systemd* you can set that mode with the command:

```
systemctl set-default multi-user.target
```

By appending the command

```
/usr/bin/console_echoes -c ...etc...
```

at the end of the file */etc/rc.d/boot.local* , *Echoes* will be started automatically as a service at next reboot, without needing user login. Its textual output can be watched by the means of the command

```
journalctl -f | grep echoes .
```

console_echoes is simply an alias for *echoes* executable file. When called with that name, *Echoes* recognizes you want to exclude totally the graphical support and will run ever without an X server running.

-d

The -d option activates the *dumb mode*: the program reads the IQ samples from the local USB dongle and forwards them to a remote UDP client without process neither store data locally. This option includes *-c*

-l

The GUI language is chosen automatically depending of the operating system localization settings. Currently, the only alternative to the hardcoded language (English) is Italian, so the Italian language will be displayed automatically on Italian PCs. If you desire to override the default behavior and load a precise translation file (the extension *.qm* is implicit) it can be specified with the option *-l*.

-n < level>

Echoes can produce lot of debugging text output that can be read in the *echoes.log* file, under the default working directory. This file is re-created from scratch each time Echoes is launched. By setting the -n 0 option, the creation of this file will be inhibited, saving some CPU time and disk resources. Higher numbers (up to 5, with default 3) increase output verbosity.

Under Linux, *echoes.log* is never created because the text output is sent to *syslog* to be collected in the system journal. *Echoes* messages can be showed with the command:

```
journalctl |grep echoes
```

-s <config_file>

The GUI settings can be saved in user defined *rts* files. This can be done by pressing the *Save As* pushbutton on the main window and specifying a path and a file name. The rts extension will be added by default. These files can be reloaded in future sessions by specifying the *-s* option followed by a rts file path.

-r

By default, each time it starts, Echoes looks for a default configuration file in the working directory called *default.rts*. This file is updated with the actual parameters set in the GUI when quitting the program. In this way, the program's settings remains persistent between consecutive invocations. The *-r* option can be set when you desire to reset the parameters to the hard coded settings (-s will be ignored in this case). The same result can be achieved by deleting *default.rts* .

-v

The debug messages dumped in *echoes.log* can be watched in real time on the console/terminal window by specifying the -v option. In this way, the messages will be printed on *stderr* too.

-w <working_dir_path>

The default working directory (*echoes/*) is normally created under the user's home directory, but another working directory can be specified with the *-w* option. In this directory will be stored the files generated by the program (logs, screenshots, data files, configurations, reports).

GUI description

The main window

This window controls the data acquisition and general program behavior. It's a small one, subdivided in tabs. The variable fields in window have green color, while fixed texts and pushbuttons are yellow.

After program startup (unless a configuration file has been chosen at command prompt via the *-s* option) the program configures itself following the settings present in *default.rts*.

The first time the program is invoked - when that file doesn't exist yet - it takes the hardcoded (*factory*) settings and creates *default.rts* in the working directory.

That file is updated with our latest GUI changes when exiting the program. The user can save the settings in a dedicated file by the means of “Save As” pushbutton. That file can be retrieved later from GUI by clicking the *Open* button or by specifying it as argument for *-s* command line option.

Some installation-dependent parameters, like *GNUplot path* and the daily events count, are recorded in a local configuration file (The *registry* under Windows, or *.config* files under Linux) in order to allow the portability of *.rts* files between different stations.

Pushbuttons

The two pushbuttons on the top *Open* and *SaveAs* manage the user's configuration files (*.rts*). Immediately below them, are the "About" and "Quit" pushbuttons. The first opens a dialog with program version and other general information, the second is self-explanatory.

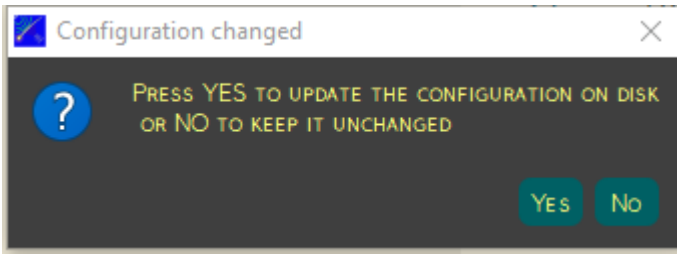
There is not a *Save* button; this operation can be done by pressing *SaveAs* then press *Ok*, since the proposed file name is the name of the last configuration file opened. The active configuration name is showed just below the buttons.

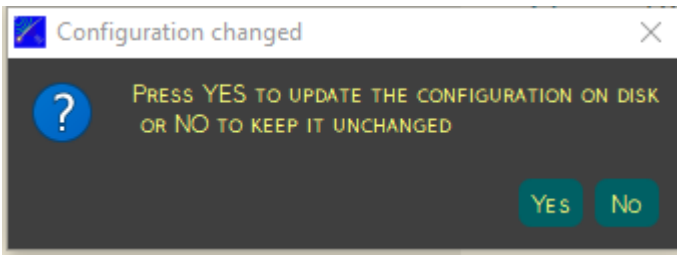
The last pushbuttons are *Start* and *Stop* at the bottom that start and stop the data acquisition thread. Below them, the status bar shows the acquisition status and error messages.

Even if *Start* has been pressed, the user can still operate on main window's controls; but some of the modifications won't be effective until the acquisition is stopped and restarted.

> **Note:** Depending of the *acquisition mode* and other parameters chosen, the acquisition could stop many seconds after pressing the *Stop* button, this because the programs avoids to truncate the data files while capturing events and postpones the stop after all data files are closed. Despite this, pressing *Quit* the program exits immediately regardless the acquisition status.

After pressing *Quit*, if the current configuration differs against the configuration file *rts* loaded, a small dialog box prompts to keep or not our changes in that file. The prompt doesn't appear if no *rts* file has been loaded first, the *default* configuration is always updated silently.





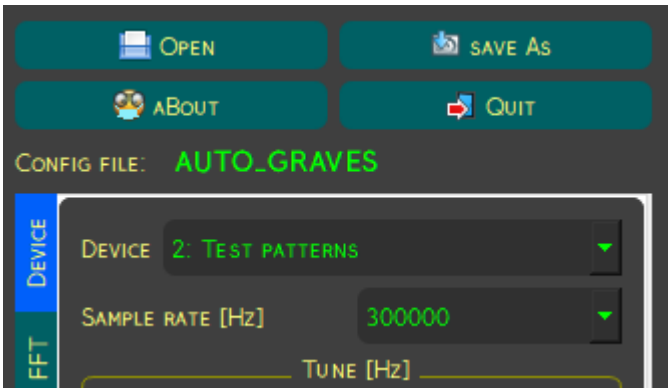
Device tab

The *Device* tab allows to control the samples acquisition parameters.

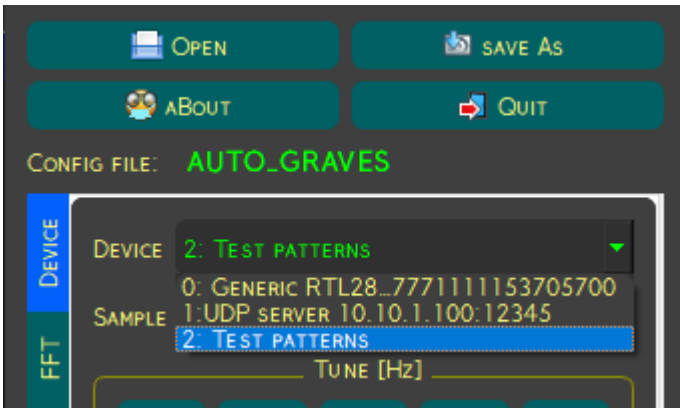
Once started, on main window all the available input sources are detected and made available through the device selector, on the *device* tab. This detection happens once when the program starts. Further devices plugged afterwards won't be recognized. There are three types of devices currently supported:

- RTL-SDR dongles, if plugged in
- Dongle servers, if configured in *preferences* tab. For more details see Advanced
- The *test patterns* device that's always showed. This is not a real device but a functions generator for program's development purposes.

By default, the device selected is the test device because it's always present.



Other devices appear in the drop-down list after clicking the listbox button at right side and can be selected.



When a configuration file is specified with *-s* option the selected device will be the one in use when that configuration file has been saved, or again *test patterns* if that device is not found. This happens - for instance - when the configuration file used comes from a different station with a different kind of dongle, having a different device identification string (*Generic RTL....* or similar) In this case, a new dongle will be listed in the drop-down, replacing the missing one.

The remaining controls in *device* tab operate on a plugged-in DVB-T dongle:

Sample rate

The drop-down list shows the possible values for sample rate (SR), from 250 kHz to 3.2 MHz. Default is 3.2 MHz.

Tune group

The tuning frequency can be selected with the buttons with resolution of 1kHz. To set a finer-grained tuning frequency, right-click on the digits and enter the frequency in the edit box.

Gain

The drop-down list shows all the provided manual gains for the dongle in dB. There is also the *AUTO* value to select automatic gain. Please note that *Echoes* always disables the *AGC*, at least it tries; unluckily it's not possible to exclude it completely by the means of software only.

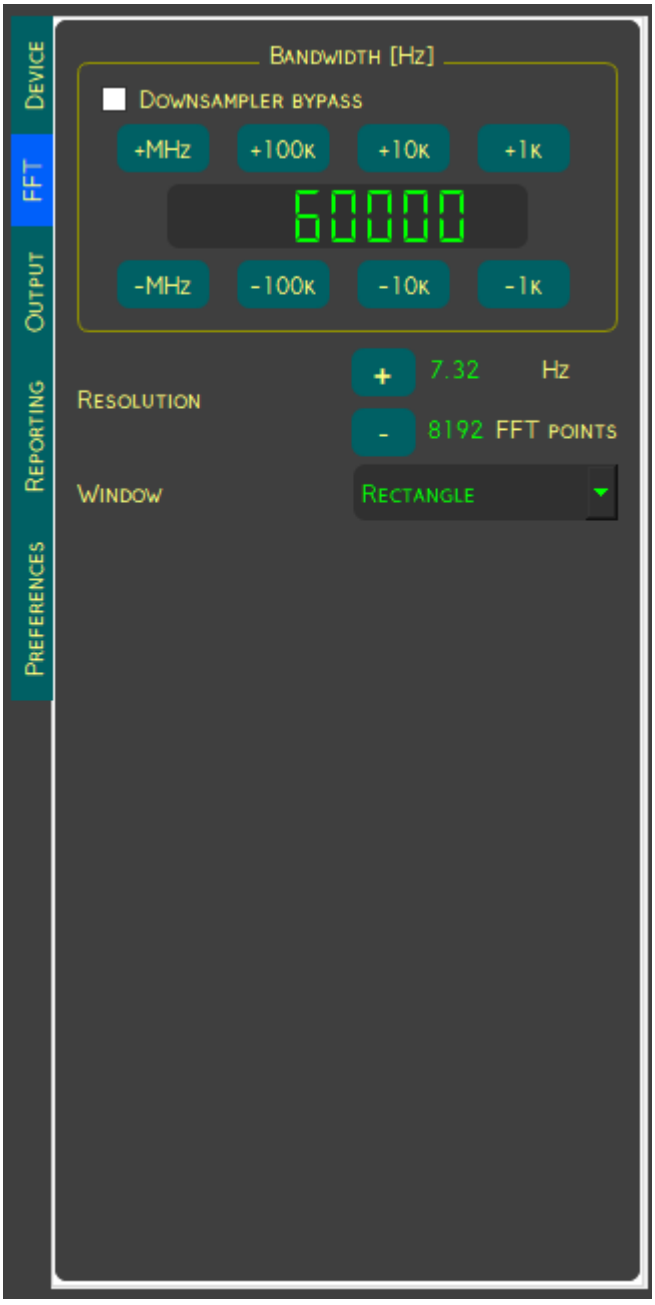
Error compensation

Enter a value only in case the frequency values on waterfall look shifted versus a reference signal. A precise well known fixed frequency carrier can help to verify the accuracy of the frequency scale. Shifts can be compensated by entering a value in ppm, that can be negative or positive and have 2 decimals precision. The integer part is managed by hardware, while the fractional part is used to apply a transparent offset correction to the waterfall for a finer centering.

> **Note:** even if the GUI allows to change the controls values at acquisition time, the changes in *Device* tab will be considered after restarting acquisition, with the exception of *Tune* that is immediately applied.

FFT tab

The second tab *FFT* encloses the controls that have effect on spectrogram quality.



Bandwidth group

Here is possible to set the downsampler output rate with 1kHz precision. For finer setting, right-click on the digits and enter the desired value. The FFT block will operate on data sampled at this frequency, and will produce spectra scans with equivalent bandwidth. The *Downsampler bypass* box, if checked, will set and lock the bandwidth control to the same frequency of sample rate, that is the same behavior of the older *Echoes* releases.

Window

This control applies an additional algorithm to the FFT output that aims to a better separation between the spectrogram output points and reduce the noise level (N). There are 10 kinds of windows available, the default is *Rectangle* meaning no window at all.

Resolution

The FFT resolution can be selected with the two **+/-** buttons. The FFT transforms the input signal in time domain to a number of *points* (or *bins*) in frequency domain that is a power of two; the buttons increment or decrement this power, and the resulting number of *FFT points* is showed at right, just below the value in Hz.

Since this value depends of the sample rate chosen

```
(Hz = SR/points)
```

that value will be mostly a non-integer frequency.

The smallest is the resolution step, the lower is the noise level (N) and the higher is the time to spend to get samples from the dongle. The default resolution is 4096 points (781.25 Hz with SR=3.2 MHz)

Output tab

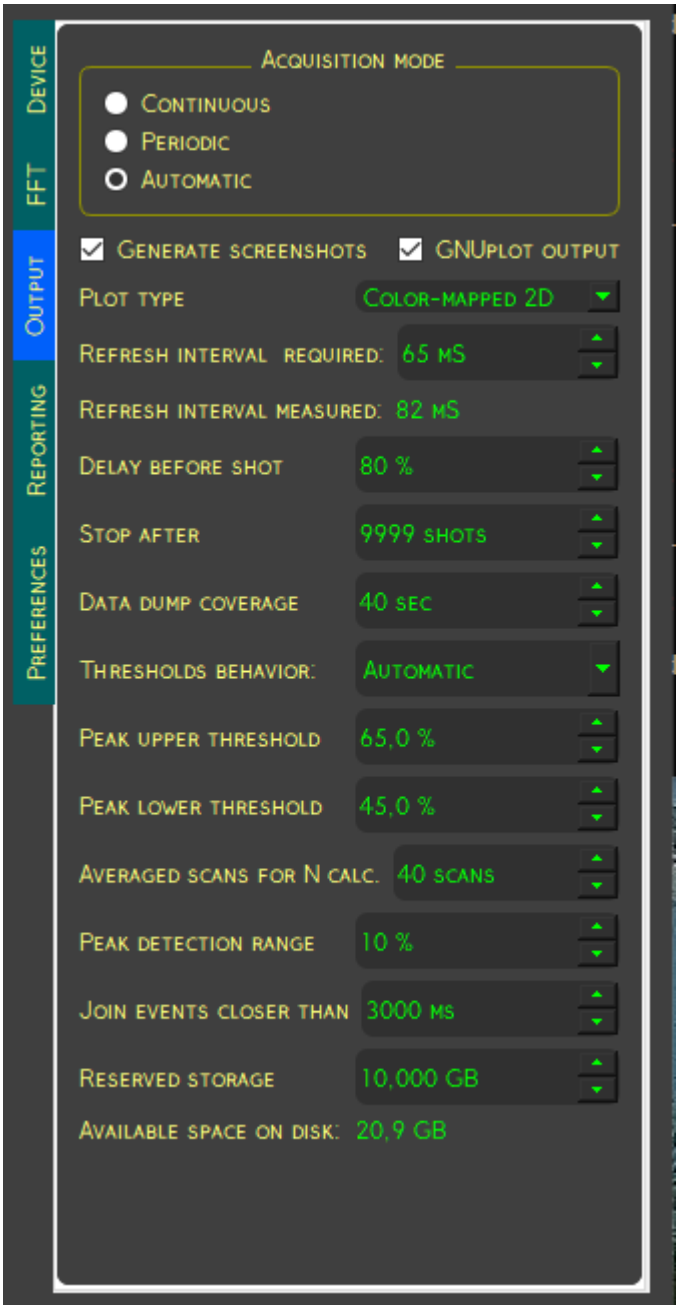
The third tab *Output* includes the controls that have effect on data files generation.

With the acquired data, *Echoes* produces four kinds of files:

- A **statistic CSV file**, later described in detail in Recording section. This files is always generated and its content depends of the *Acquisition mode* selected.
- **Waterfall screenshots in PNG format**. They are generated if the relative *Generate screenshot* box is checked (by default it is). The screenshot's time coverage varies with the *Refresh interval required* set and the program workload.
- **GNUplot's data dump files**. The scan data used to drive the waterfall are then saved in a circular buffer. The size of this buffer, expressed in seconds can be set with the control *Data dump coverage*. When the required number of seconds has been reached, all the data are dumped in a *DAT* file, in a format understandable by GNUplot. This means that, compared to screenshots that have length dependent on refresh time, the temporal coverage of dumps can be freely selected and last up to several minutes. Dumps are generated if the relative *GNUplot output* box is checked (by default it isn't).

The *Plot type* control selects the way in which the data will be represented by GNUplot. More detail on this aspect is in Recording section.

- **GNUplot's commands file PLT**. When opened with GNUplot, this file starts a carrousel of all the dump files found in working directory, with the same color scale used in screenshots. The file can be easily edited by the user to customize the color scale, size and other plotting parameters. For GNUplot commands reference, the GNUplot manual is for you.



After this premise, let's continue describing the *Output* tab controls:

Refresh interval required

This control sets the waterfall update cadence timer (aka *tick*). The default value is 100 mS, this means that when *Start* is pressed, an internal timer every 100 mS awakes the output controller to display all the scans the radio produced in meantime and updates the output files. If this operation takes more than 100mS, the following tick will be lost, and the waterfall will be updated 200 mS later instead of 100. This is the reason why I called it *required* interval. The effective *Measured interval* is available some seconds after acquisition start and

its value appears immediately under the *Refresh interval required* control. My advice is to keep the interval request few mS under the measured one, in order to optimize the program's CPU load.

Acquisition mode

The most important parameter to set is *Acquisition mode* because it has effect on the whole program behavior and the contents of generated files. Moreover, depending of the mode chosen, some controls can become ineffective. This is not the case of the *Refresh interval required* seen above.

- **Continuous mode**: when the acquisition starts, the waterfall runs continuously without automatisms. The statistic CSV file records data related to every scan line showed on the waterfall. When acquisition is stopped, the file is closed, and a new empty one is created when restarting the acquisition. There is no provision for events detecting, every scan is treated and recorded as a single event. No automatic screenshots neither dumps will be produced, but is still possible to capture screenshots manually by pressing the button *Manual shot* on waterfall window. This is the default mode the first time *Echoes* is started.

- **Periodic mode**: if this mode has been chosen,while acquisition is running, *Echoes* will capture screenshots and/or dumps on a fixed period basis. This period is set by the *Data dump coverage* control. By default, this control is set to zero seconds. By setting, for instance, 60 seconds, the program will generate a dump and a screenshot every 60 seconds. Please note that if the waterfall coverage is below 60 seconds, the screenshots and the dumps will remain time-aligned but screenshots won't cover the entire time. The CSV content is similar to *Continuous mode* with one column added that holds the name of the screenshot file where that scan has been plotted. This operating mode should be used to practice with the program and get to understand how to set the thresholds and other parameters that allow you to run the program in automatic mode.

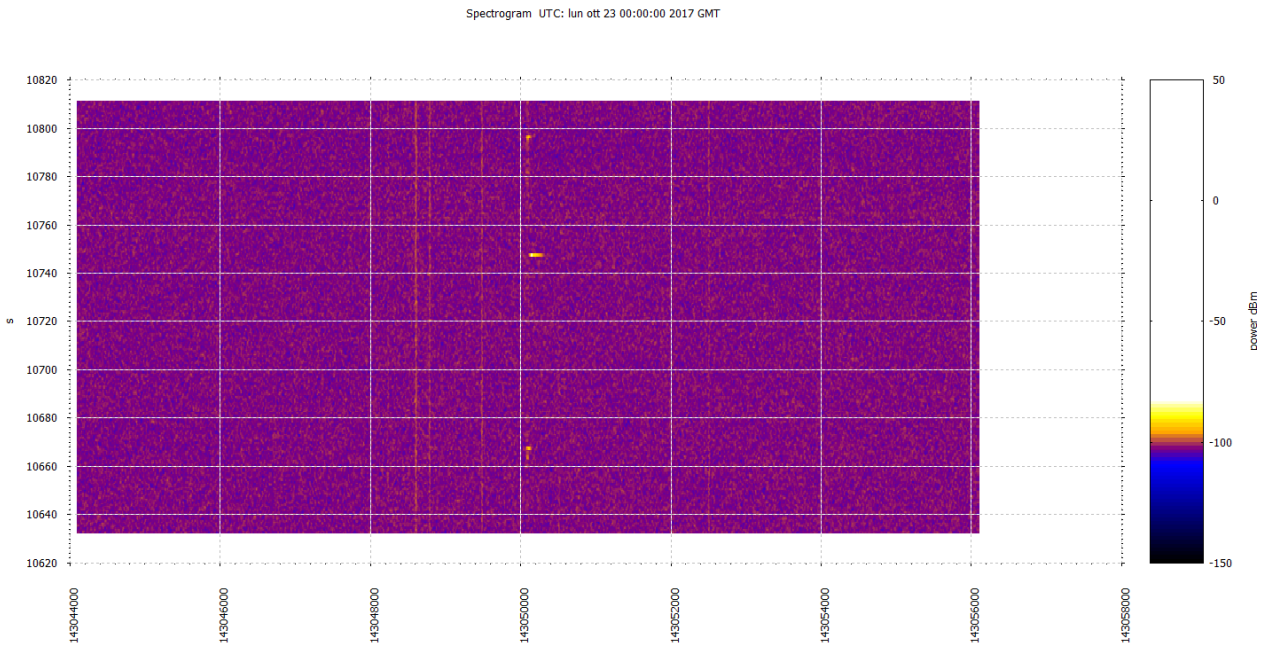
> **Note:** In *Continuous* and *Periodic* modes the statistic CSV file tends to grow a lot; in a single day it can reach several tenths of MB.

- **Automatic mode**: when this mode is selected, the program tries to recognize meteor echoes in the spectrum. This try can be more or less successful depending of how some controls - those related to *thresholds* - are set to match your specific setup. When a peak is detected a screenshot and/or dump can be generated immediately, but in most of cases is more useful to add some amount of delay to let the peak reach the lower part of the waterfall, scrolling downward. In this way, longer lasting echoes (*overdense*) can be fully captured in the screenshot. While taking the screenshot, the *Instantaneous power graph* replays the power data recorded when the peak was detected, so the shot will simultaneously record the power peak in frequency domain, in time domain (*Power history* graph) and both (waterfall). For each event detected, in the statistic CSV only 3 rows are recorded: one at event starts, one at peak and one at the end, but there are several more data columns, explained in detail in Recording section.

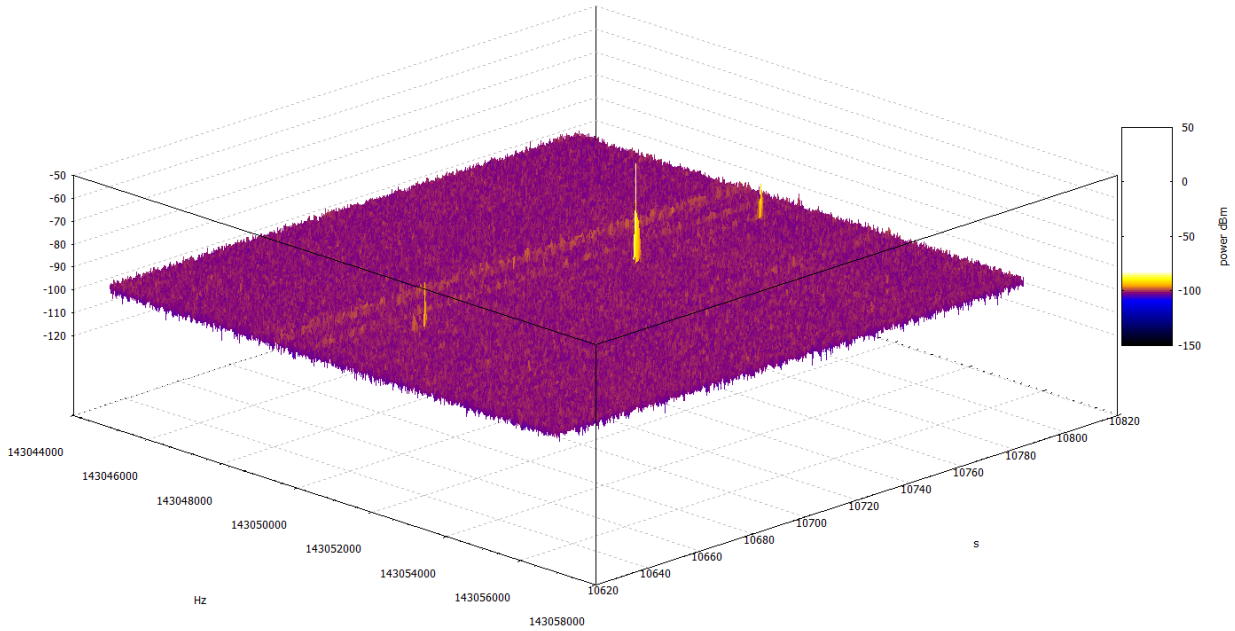
Plot type

This control specifies in which way we want to represent the dump files:

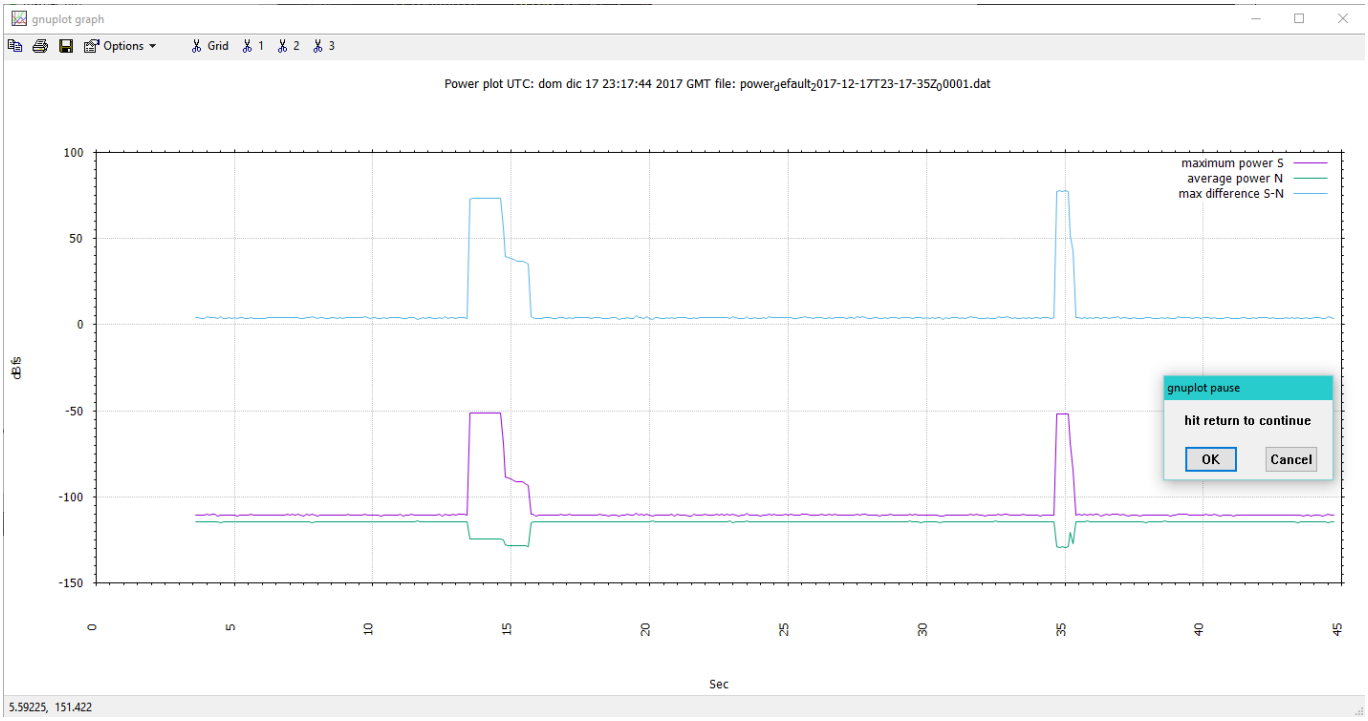
- **Color-mapped 2D** : it represents the waterfall, similarly to a screenshot, with de difference that time coverage can be longer.



- **Perspective 3D** : a fancy 3D view of the waterfall, with power on Z-axis.



- **Power 2D** : the peak power S, the average power N and their difference (S-N) in dBfs are plotted continuously in a ECG-like graph, with no frequency information, like the *Power history* graph with the difference that the graph covers the entire acquisition session's time.



Delay before shot

This control delays the shot triggering by some amount of seconds after the event detection. The delay is specified as percentage of the screenshot / dump coverage. In this way, even if dumps coverage is different than screenshot coverage, the echo will appear in the same position on both.

For instance, if screenshot coverage is 30 sec and the dump is 60 sec long, setting a delay of 50% the screenshot will be taken 15 seconds after the peak, while the dump will occur after 30 seconds.

Stop after

Here you can set the number of shots to be captured before stopping acquisition automatically. The default number is zero, because the default mode is Continuous and it takes no shots; in Periodic or Automatic modes this number can be set high, for instance 9999.

Since in Automatic mode the event counter is zeroed at every midnight UTC, it is very unlikely that 9999 will be reached, so the acquisition won't never stop. In Periodic mode instead it will.

In Continuous mode, if this number is set higher than zero, the acquisition won't start, and the status bar at main window's bottom will show the warning

```
Please set shots=0 and upThreshold=0
```

Data dump coverage

The scan data used to drive the waterfall are saved in a circular buffer. The size of this buffer, expressed in seconds can be set with this control.

In Periodic mode, when the required number of seconds has been reached, all the data are dumped in a *DAT* file.

In Automatic mode, the dump file is created only if there is a detected pending event, otherwise the scans remains in the buffer until they reach the maximum age to be silently deleted. Even if a dump has been created, the scans dumped in the file aren't deleted before being aged, because there could be two or more events closed in time that trigger further dumps.So, the same scan can appear in more than one dump file.

In Continuous mode this control is ignored.

Upper and lower peak thresholds and their behavior

The event capturing feature available in Automatic mode is based on two thresholds, *Peak lower threshold* and *Peak upper threshold*.

An event starts when the reference data value exceeds the lower threshold; at this moment the event counter is incremented and the first of the three rows is added to the statistic CSV buffer (*Raise stage*).

When the reference data value exceeds the upper threshold, the second row is added to the buffer (*Peak stage*). This row is continuously updated until the monitored data reaches the highest value.

When the reference data value falls under the lower threshold, the third row is added to the buffer (*Fall stage*), the event terminates and the three rows are appended to the statistic CSV file.

Ok boss but... what the heck is a *reference data* ?

A reference data is the statistic CSV columns that triggers an event. The statistic CSV columns are described in Recording section, but only three of them can be selected with the *Thresholds behavior* control.

- **Absolute** : the reference data is the peak **S** value expressed in dBfs. The S value is the highest power value detected in a scan *within the detection frequency range* (explained later in this section). Depending of the daily variations of average power **N**, this reference could determine the collection of an higher number of false positives during the day than during the night.
- **Differential** : the reference data is the difference between the *S* value and the average power detected *in the entire scan* **N** that is considered noise. The **(S-N)** value is expressed in dBfs. This reference is less impacted by daily variations of *N* than absolute *S*, but is given as stated that if the *N* level raises, weaker echoes will be lost in any case.
- **Automatic** : the program calculates continuously a mobile *(average S-N)* value and the reference data is the difference between the instantaneous (S-N) value and the average (S-N). This difference is expressed as percentage of the average.
  For instance, by setting the lower threshold to 40%, an event is started if the *(S-N)* value exceeds the value:
  instantaneous lower threshold = *(average S-N)* + ((40 x *(average S-N)*) / 100)
  If *(average S-N)* is 7 dBfs, the instantaneous lower threshold will be 7 + ((40x7)/100)= 9.8 dBfs.

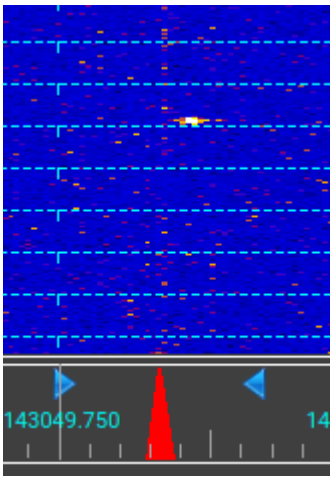
Whatever reference is chosen, it applies on both thresholds.

Averaged scans for N calculation

The **N** value could slightly vary between consecutive scans. This control allows to specify a number of scans to be averaged to calculate a flatter *N* . The bigger is this number, the flatter will be the cyan noise line on the *power history* graph. This is also the number of scans averaged for *(average S-N)* and *(average S)* calculation.

Peak detection range

In order to limit the detection of false positives, the *S* value is calculated considering a frequency range narrower than the entire scan portion displayed on waterfall. This range is expressed as *percentage of the waterfall's width* and is represented on *frequency ruler* with two blue triangular arrows.



Join events closer than

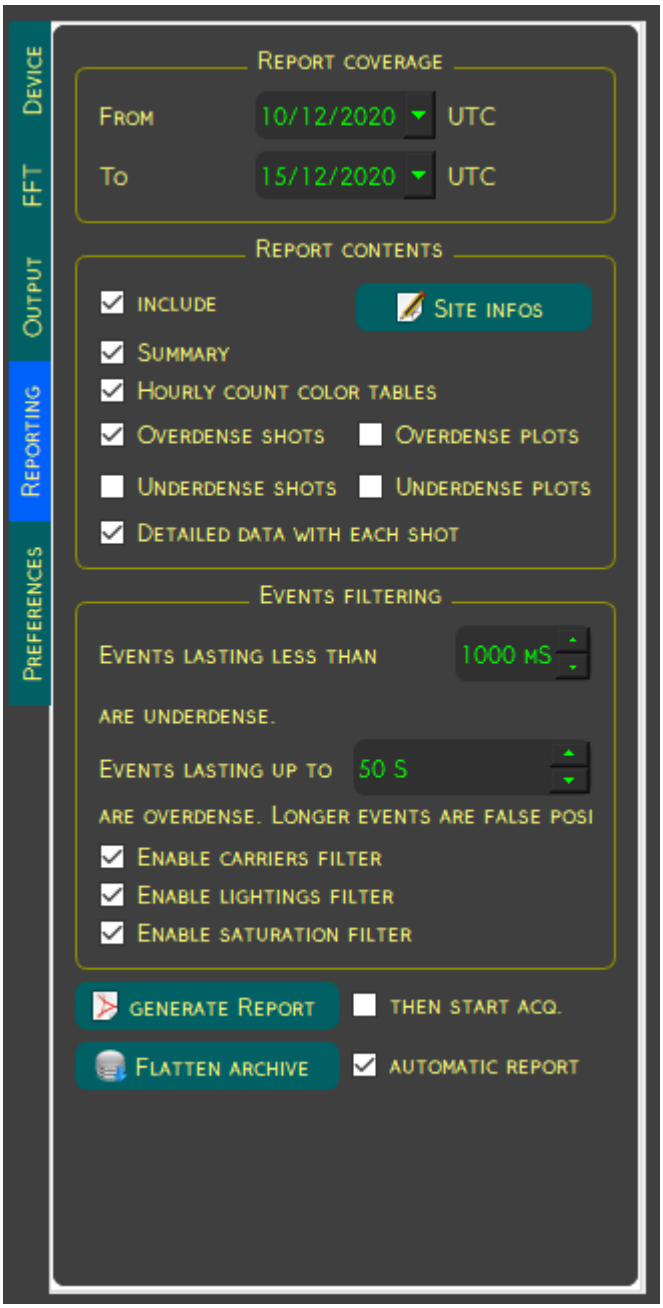
This control specifies the minimum time distance between consecutive events in order to be considered the same event, the default is 1000 mS, a zero value means no joins.

Reserved storage

This is the space in gigabytes that must be left free on the disk hosting the working directory. When the space occupied by the working directory grows out, the acquisition is stopped automatically. The default value is zero (no limits).

Reporting tab

When the program runs in *Automatic mode*, every night at 00:00 UTC the acquisition is stopped and the *postprocessing thread* starts. This thread moves all the data files generated during the expired day (screenshots and dumps) into an *archive subdirectory hierarchy*, classifying them by day and kind of events: *overdense*, *underdense* or *fake* (false positives).



The postprocessing thread generates and maintains a *daily report* file. This is a CSV file containing, for each day archived, the count - divided by hour - for each type of event (*Underdense*, *Overdense* or *Fake*) and the daily totals. If the file already exists in the archive, the daily counts are appended to that file.

***Report coverage* and *Report contents* groups**

Besides the daily report, that is simply a table, *Echoes* can generate a **full report**. It is a single *HTML* page divided in 7 sections that can be included or not in the report by checking the appropriate boxes:

1. **Include site infos**: The *Site infos* button opens a form where information about your station can be entered. It's also possible to replace the *Echoes* logo with a customized one. With box checked, information and logo will be inserted in report.

2. 

3. **Summary**: inserts a table where the events numbers are split by kind (underdense, overdense, fake) in columns and by days on rows.

Summary

This document has been generated by *Echoes - Meteor scatter with RTL-SDR v.00.00* in date 01/01/2021, on a x86_64-little_endian-llp64 machine (hostname: ASTROLABIO)
collecting data produced in the period between 27/12/2020 and 31/12/2020 UTC, running in automatic acquisition mode.
The configuration file used is *AUTO_GRAVES.rts*. The following table summarizes the event numbers, divided in 3 categories:

Underdense (events lasting less than 1000 milliseconds),
Overdense (events lasting longer than 1000 milliseconds),
and *Fakes* (events lasting longer than 50 seconds or captured by postprocessing filters)

Events	Underdense event numbers	Overdense event numbers	Fake events numbers
27/12/2020	2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 20, 21, 22, 23, 26, 27, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 51, 52, 53, 54, 55, 58, 59, 61, 62, 64, 65, 67, 69, 70, 71, 72, 73, 76, 77, 78, 79, 81, 82, 84, 85, 86, 87, 88, 90, 93, 96, 97, 99, 101, 102, 103, 104, 105, 106, 107, 111, 112, 113, 114, 115, 116, 119, 121, 122, 124, 125, 126, 127, 128, 131, 138, 139, 141, 144, 146, 147, 148, 150, 151, 152, 153, 154, 155, 158, 159, 160, 161, 168, 169, 172, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 193, 194, 195, 196, 198, 199, 200, 201, 202, 203, 204, 206, 207, 210, 211, 212, 213, 217, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 234, 235, 237, 238, 239, 241, 242, 243, 246, 247, 250, 251, 255, 256, 259, 260, 261, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 275, 276, 277, 278, 279, 280, 282, 285, 290, 292, 293, 294, 295, 296, 298, 299, 300, 301, 302, 303, 306, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 327, 329, 331, 332, 333, 334, 335, 336, 337, 340, 341, 342, 343, 344, 345, 346	1, 6, 13, 19, 24, 25, 35, 48, 50, 56, 57, 60, 63, 66, 68, 74, 75, 80, 83, 89, 91, 92, 94, 95, 98, 100, 108, 109, 110, 117, 118, 120, 123, 129, 130, 132, 133, 137, 140, 142, 143, 145, 149, 156, 157, 162, 163, 164, 165, 166, 167, 170, 171, 173, 174, 175, 188, 189, 190, 191, 197, 205, 208, 209, 214, 215, 216, 218, 233, 236, 240, 244, 245, 248, 253, 254, 257, 258, 262, 281, 283, 284, 286, 289, 291, 297, 304, 307, 308, 309, 310, 312, 325, 326, 328, 330, 338, 339	28, 134, 135, 136, 192, 249, 252, 274, 287, 288, 305, 311, 313
28/12/2020	2, 3, 4, 5, 6, 7, 8, 9, 11, 13,	1, 10, 18, 23, 24, 29, 32, 33, 41, 55,	12, 34, 48, 49, 50, 51, 80, 83, 89, 90,

4. **Hourly count color tables**: when checked, generates one or more RMOB's Colorgramme-like diagrams, where the daily counts are split by hour and expressed with a color scale where black means zero and red is the highest value in diagram. The figure below shows the *Total without fakes* where all the valid events recorded are counted; this diagram is always present if the box is checked. Then if *overdense shots/plots* has been checked, a second diagram is inserted where only overdense events are counted. Finally, a third diagram is inserted if *underdense shots/plots* has been checked.

Hourly data tables:

Hour / Day	27	28	29
00h	11	9	11
01h	13	8	14
02h	18	18	18
03h	26	10	21
04h	21	8	15
05h	13	18	17
06h	14	5	22
07h	15	13	14
08h	24	17	3
09h	14	10	4
10h	7	10	7
11h	11	20	9
12h	19	19	8
13h	15	23	6
14h	12	25	6
15h	10	19	2
16h	10	16	6
17h	13	14	1
18h	13	17	4
19h	13	16	2
20h	12	8	4
21h	13	11	9
22h	6	17	4
23h	10	16	2
Daily totals	333	347	209

Fig.1 - Total without fakes

5. **Overdense shots**: when checked, all the overdense events screenshots will be inserted in the report.
6. **Underdense shots**: when checked, all the overdense events screenshots will be inserted in the report.
7. **Overdense plots**: when checked, all the overdense events plots will be plotted in an image then inserted in the report.
8. **Underdense plots**: when checked, all the overdense events plots will be plotted in an image then inserted in the report.
9. **Detailed data with each shot**: when checked, all the information about each event will be included in tabular form. These are the information coming from the *statistic CSV file* where each event is recorded on three rows.

Event nr.	1		
Date	27/12/2020		
UTC time	00:07:43.110		
Unix timestamp (seconds since 01/01/1970)	1609027663.206		
Tune [Hz]	143040000		
Bandwidth [Hz]	60000		
Central frequency [Hz]	143050000		
Lowest frequency displayed[Hz]	143048000		
Highest frequency displayed[Hz]	143052000		
Zoomed bandwidth displayed[Hz]	4000		
Resolution step [Hz]	14,6484		
Threshold behavior	automatic		
Upper threshold [%]	60		
Lower threshold [%]	40		
Upper threshold [dBfs]	15,4122		
Lower threshold [dBfs]	13,4856		
Range from [Hz]	143049800		
to [Hz]	143050200		
Event state	Raising front	Event peak	Falling front
Maximum peak (S) [dBfs]	-0,928467	-0,928467	-9,80562
Average peak [dBfs]	-6,46195	-6,46195	-5,91167
Noise (N) [dBfs]	-15,6346	-15,6346	-15,5443
(S-N) [dBfs]	14,7062	14,7062	5,73866
Average(S-N) [dBfs]	9,1727	9,1727	9,63261
Maximum peak frequency [Hz]	143050007	143050007	143050200
Standard deviation	0,444712	0,444712	0,345324
Lasting [mS]	0	0	3071
Frequency shift [Hz]	-200		
Area covered by echo[FFT points]	2		
Area interval [FFT points]	108		
Total number of peaks	1		
LOS Speed [m/s]	1		

Events filtering group

The controls in this group configure the events classification functionality. This argument is developed in more detail in Recording section.

- In *Events lasting less than* can be set the minimum lasting for an event to be classified *overdense*. Event lasting less are classified *underdense*. The value is expressed in mS, the default is 1000.
- The maximum lasting for an *overdense* event is set in *Event lasting up to*. The value ix expressed in seconds. Events lasting more than this value, are classified as *fake*, false positives.

Besides this lasting-based discrimination, there are also three **experimental** filters aimed to discriminate three different causes for false positives. These filters work by applying simple rules on the event data collected in statistic CSV and are executed *before* the check of event lasting.

- *Enable carriers filter* : continuous carriers can be easily discriminated because they trig a very long event that can be later discarded as fake on a lasting basis only. This filter instead aims to discard fake events caused by non-continuous but recurrent disturbances falling into the *peak detection range*.
- *Enable lightings filter* : electrostatic discharges and lightings produce a strong, short and wideband signal that appears as an horizontal white strike on the waterfall. This filters aims to discard such events as fakes.
- *Enable saturation filter* : if nearby our *peak detection range* there is a station sending something is very likely that this signal interfere with our dongle. The evidence of this problem is on the *Power history* graph, where the red *S* line and blue *N* line fall down suddenly for a while to the same value (maybe due to a dongle's self-protection feature, a sort of drastic AGC). This filter aims to discard these fake events.

> **Note:** being experimental, all the filter enabling checkboxes are checked by default but currently they depends of some hardcoded parameters that have been calculated under a determinate configuration (bandwidth, peak detection range, refresh interval, thresholds...); therefore it is very likely that changing the configuration will affect the functionality of these filters, in better or in worse.

Generate report

This button triggers the *full report* generation. The data included in the report are taken from the archive and cover the days range specified in *Report coverage* group. The report is generated if at least one of the days in the specified range are included in the archive. If the acquisition was active when the button was pressed, after some seconds of freezing the acquisition continues once the report has been generated.

...then start acq.

This checkbox has been disabled because obsolete. It has been marked as reserved for future uses.

Flatten

This button opens a dialog where the user is asked to specify an empty folder where the archive will be cloned flat, without hierarchy, with all the files in the same folder. If not existing, the dialog allows to create a brand new folder before cloning.

This feature has been added initially for development purposes, in order to test the behavior of archiving features (by restoring all the shots in the working directory and deleting the archive hierarchy before repeating the *midnight swap*).

Besides this, I found it useful also to compose a video by putting together events caught on several days; it's much easier to do when all the involved images fit in the same directory. The following video has been produced under Windows with the batch file *make_video.bat* present in the *Echoes* program folder:

Perseids 2020

It uses ImageMagick to compose the video, that must be installed on your system. Follow the instructions in the batch file to customize and use it.

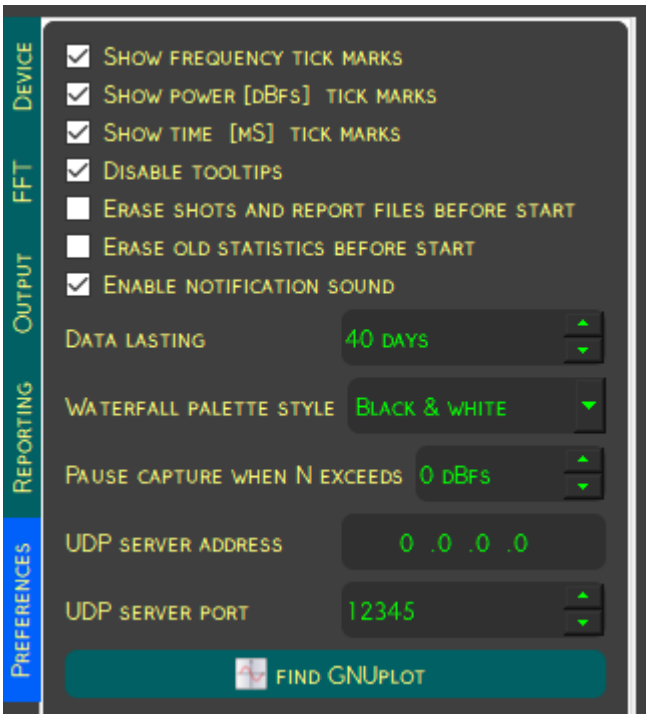
Automatic report

This checkbox enables the automatic *full report* generation. If checked, every night after the *midnight swap* the *Report coverage* dates are checked. If the *To* date matches the day just terminated, the report generation starts.

If generation ends successfully, the coverage dates are incremented both by the same number of days. This means - for instance - that if *From* is set on a Monday and *To* on a Sunday, the report will be generated once a week the night between Sunday and Monday, and will cover the entire week.

Preferences tab

The last main window's tab is about user's preferences, a miscellaneous of controls that covers different aspects of the program that can be customized.



Show... checkboxes

Some people dislikes to see the measuring grids above the waterfall and the side graphs, so this tab allows to hide them by unchecking the related box. By default, all these box are checked.

Disable tooltips

This box is unchecked by default, allowing the display of the help messages that appear on every control when the mouse cursor hovers them. Once the user has become familiar with *Echoes* controls, they could result annoying, so they can be hidden by checking this box.

Erase shots and report files before start

When checked, all the screenshots, the plots and report files *produced with the configuration file currently loaded* and present in the working directory will be deleted at acquisition's start. This option is useful while experimenting with the parameters setup, but **remember to uncheck it before saving the final setup on a new configuration file!** By default is unchecked.

Erase old statistics before start

When checked, all statistic CSV files *produced with the configuration file currently loaded* and present in the working directory will be deleted at acquisition's start. This option is useful while experimenting with the parameters setup, but **remember to uncheck it before saving the final setup on a new configuration file!** By default is unchecked.

Enable notification sound

Checked by default, it can be unchecked to avoid playing the notification sound when a new event is detected.

Data lasting

Here can be specified the maximum age of the files stored in the archive to preserve disk space. Older files will be automatically deleted at acquisition's start. *This applies only on the files generated with the configuration file currently loaded*, leaving others untouched. The default value is 40 days and - depending if you decide to capture screenshots only or also dump files - this could mean a disk space between 20 to 100 GB indicatively.

Waterfall palette style

In rel. 0.27 a new B/W palette has been added to waterfall, in alternative to the traditional color scale. You can choose here the preferred one. The default is B/W.

Pause captures if N exceeds...

In this control can be specified a threshold in dBfs. If the *average noise N* exceeds this threshold, the event captures are inhibited and on the waterfall's status bar appears the *TOO NOISY* message. The acquisition remains active to detect when *N* falls under the threshold but no events neither screenshots will be recorded. This is to prevent recording of false positives caused by unforeseen and strong disturbances.

UDP server address

Here can be specified the IP address of another machine running *Echoes* to recognize it as a *dongle server* device. It can still be the same machine, by specifying the *localhost* address (127.0.0.1). In this case, two *Echoes* instances must be running, one as *dongle client* (where *localhost* has been set) and the other as a *dongle server* connected to a physical dongle, where the *UDP server address* has been set to 0.0.0.0 that is the default value.

UDP server port

Port number where a *dongle server* accepts connections from clients. The default is 1234.

Find GNUplot

This buttons appears when Echoes doesn't know where to find *GNUplot*. When pressing the button, an *Open File* dialog opens to select the correct GNUplot's executable file (generally *wgnuplot.exe* under Windows, or simply *gnuplot* under Linux).

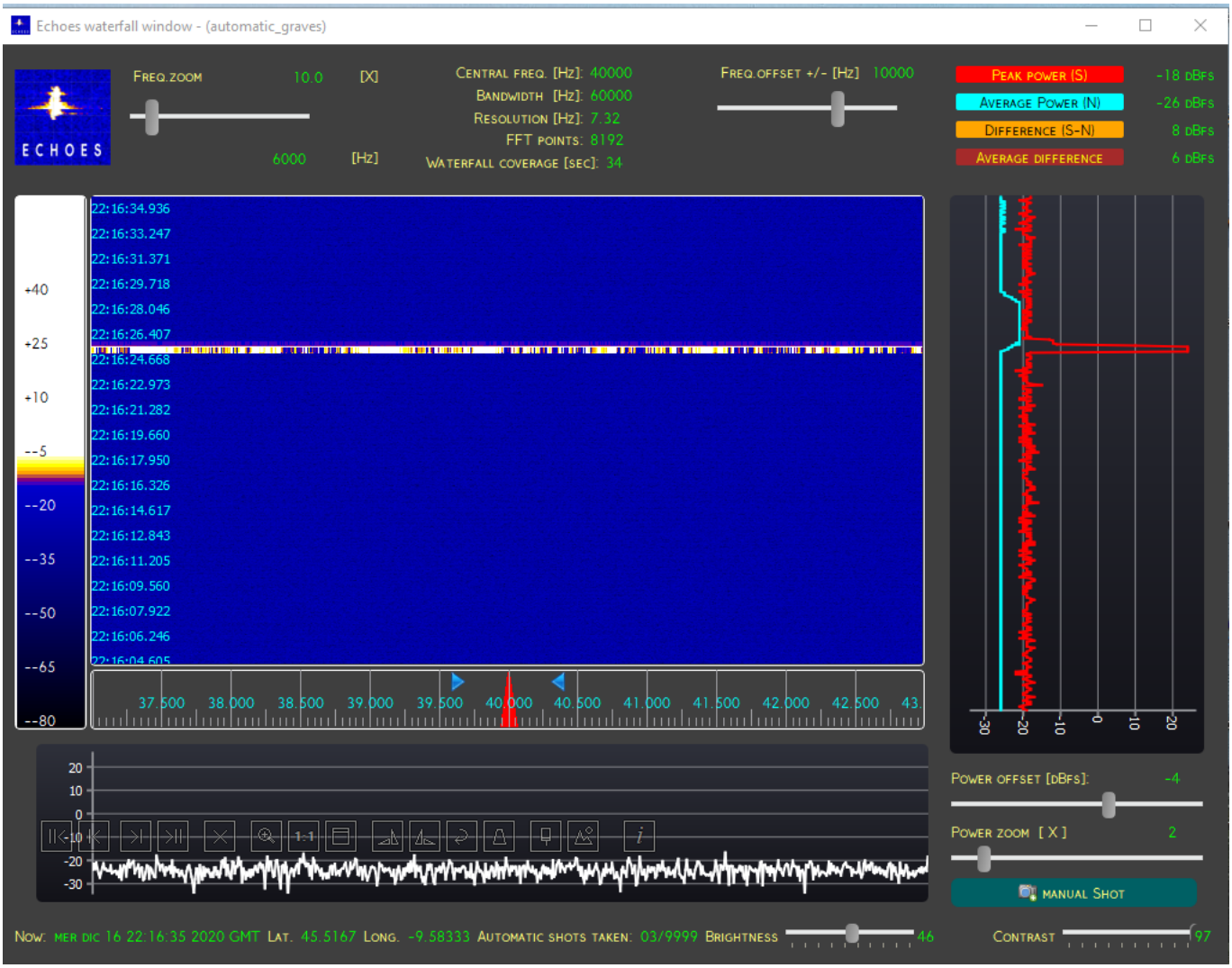
Once the executable has been specified, it is stored in a local configuration file (under Windows is the registry, under Linux is the *~/.config* folder). This won't be saved in *rts* configuration because it's a machine-specific setting. Since the next program restart, this button won't be showed anymore.

Status bar

It's a small place at bottom of main window where status messages are displayed.

The waterfall window

This window shows the output of FFT processing, split in three diagrams.



Waterfall spectrogram

At the center is the *Waterfall spectrogram*, with the time axis in Y and frequency in X, scrolling downwards.

If the *refresh interval* set is relatively high, the *radio thread* will be able to extract many scans per time interval, but only one can be displayed in spectrogram. For this reason, this scan will be the result of an integration of all the scans available in memory at the time of viewing, *keeping the maximum values of each*.

This *integrated scan* is good for recording on dump files and for statistic purposes but isn't yet ready for displaying on the waterfall, because the waterfall pane's horizontal width almost never matches the FFT resolution set; often it's smaller but could even be bigger, depending of the combination of *Bandwidth*, *FFT resolution*, *Freq. zoom* set.

So, the power represented by each pixel could be the result of the averaging of many FFT points, or could be the direct FFT point value and even replied for more pixels.

For instance, if the waterfall pane is 1024 pixel wide and FFT resolution is set to 4096 points, each pixel shows the average of four points.

This compaction algorithm is for visualization purposes only. Values saved in statistical CSV and GNUplot dump files only deal with integrated scans values.

While scrolling, horizontal time tick marks are plotted in green (color not present in color scale) with UTC time printed at the left side. These ticks can be hidden by unchecking the relative boxes in main window's *preferences* tab.

Brightness and contrast

At waterfall's left is the *color power scale* that shows the correspondence between waterfall color and dBfs; it can be adjusted with the *Brightness* and *Contrast* sliders placed at window's bottom. The waterfall is in fact a flat representation of 3D data, since the dBfs values are represented by colors instead of points on a Z axis.

System information

The window's bottom shows some system information: the *Latitude* and *Longitude* entered in *Site info* dialog, the *Automatic shots* captured vs. the total required and the *Current date/time* UTC.

> **Note:** due to a bug in Qt5, the date shows *GMT* instead of *UTC*, but it's always UTC.

There are also two labels that appear in particular circumstances:

- a *TOO NOISY* label appears when the average noise *N* raises above the limit set in *Preferences* tab.
- a *CAPTURING* label appears when an event has been detected and remains shown until that event is terminated and all the relative data (screenshots and/or dumps) have been closed and saved to disk.

Power history

At the right side, a scrolling graph called *Power history* represents the average power of each scan line (in cyan) is called *noise value*(abbreviated *N*) and the maximum power (in red) detected within the *peak detection range* is called *peak power* (abbreviated *S*). The graph scrolls at the same rate and direction of waterfall.

Instantaneous power

Below the waterfall is the *Instantaneous power* graph, where the power values are represented in Y and their frequencies in X. The dBfs scale is normalized taking the DC spike value - the artifact always present at the tuned frequency - as the maximum value.

Power zoom and offset sliders

The two sliders *Power zoom* and *Power offset* are placed at window's bottom right side. The first reduces/enlarges the power scale range while the second shifts the position of the traces along the power axis.

Frequency zoom and offset sliders

The other two sliders *Freq. zoom* and *Freq. offset*, placed at window's top, define the frequency portion of bandwidth displayed in the waterfall and its side graphs.

By default *Freq. zoom* is set to 1.0X, so the waterfall shows the entire bandwidth set in main window's *FFT* tab. With this setting, the *Freq. offset* slider is locked in zero position. This slider gets unlocked when *Freq. zoom* is

moved to any value different to 1.0X and in this way, the frequency range monitored for events can be shifted away from the tuned frequency that is affected by the DC component of the spectra.

After any adjustment of the *tune* or *bandwidth* or *sample rate* controls, the *Freq. zoom* and *Freq. offset* are restored to their default values.

Radio parameters

The waterfall window shows also some parameters related to radio receiving:

- the *Central frequency*, the value in Hz corresponding to the position of the red arrow on *frequency ruler*, at waterfall's center.
- the *Full bandwidth* in Hz after downsampling.
- the *FFT resolution* in Hz.
- the FFT resolution as *FFT points*.
- the *Waterfall coverage* measured in mS.

The latter becomes available after some seconds of acquisition running and tells how much time needs a scan from waterfall's top to reach the last line at bottom. For performance reasons, it's advisable to set the *refresh interval* in *Output* tab just a little smaller than this value to minimize the CPU time spent in acquisition.

Finally, in the up right corner, there are four instantaneous values displayed:

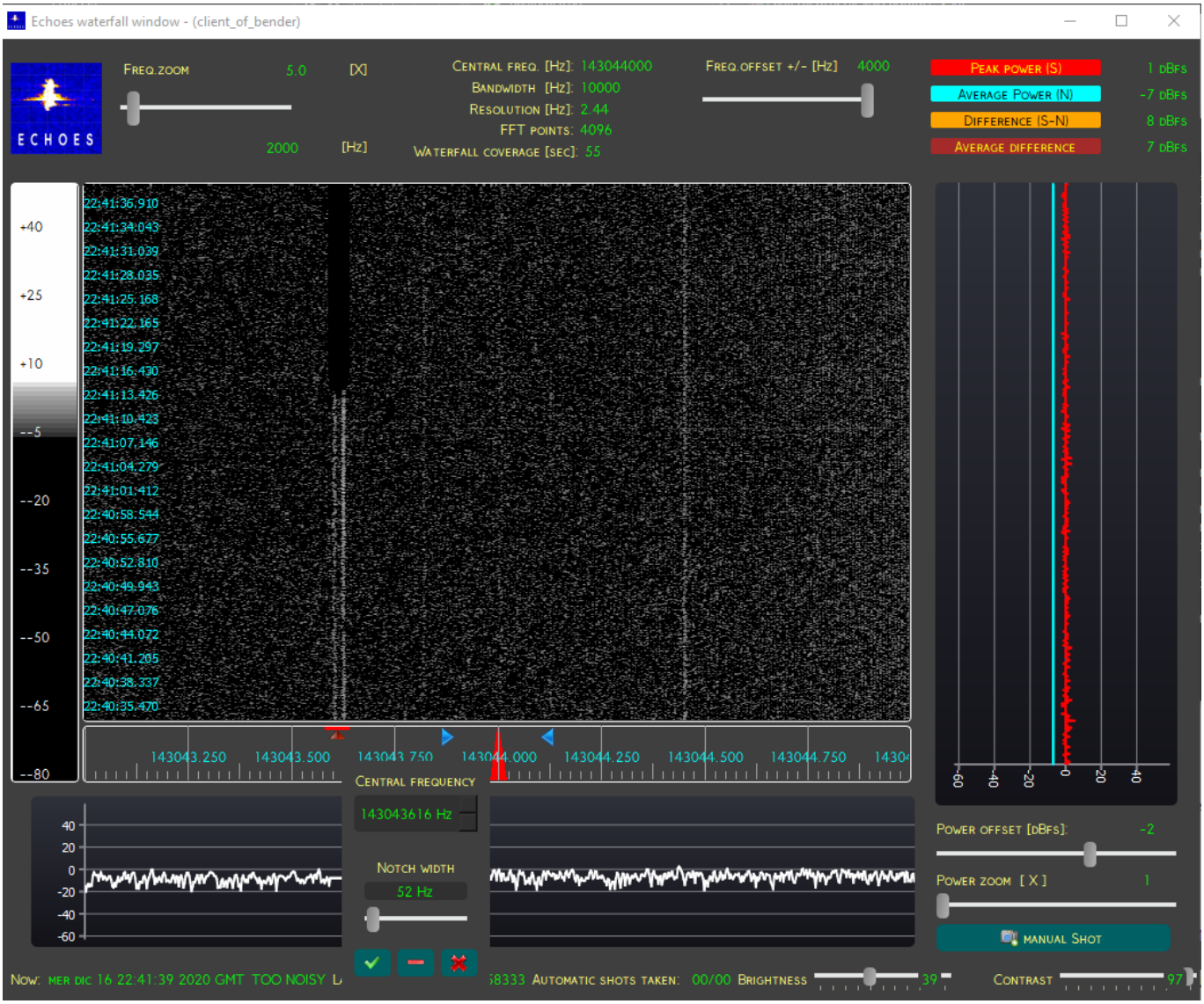
- the *Peak power (S)* on red background, that matches the red line on *power history* graph.
- the *Average power (N)* on cyan background, that matches the cyan line on *power history* graph.
- the *Difference (S-N)* on orange background.
- the *Average difference* on brown background.

Manual shot

Immediately below the power sliders, there is the *Manual shot* pushbutton that triggers a screenshot when pressed. Manual screenshots are counted separately from automatic screenshots and are named with a different naming convention. Finally they aren't accompanied by any GNUplot data.

Frequency ruler and notches

By right-clicking on the frequency ruler, it's possible to place a notch filter in that frequency. A small pop-up window appears, to set the width of the blocked band in Hz. The value can be roughly set by the means of a slider or edited with the keyboard. Once confirmed the value, a red triangle appears, with a horizontal red line above it, to visualize the blocked band width.



It's possible to add many filters, the unique condition is that they cannot overlap each other. The effect of notch filter is that, in the frequency range it covers, the waterfall will display the N value calculated in the last scan, in place of the real value produced by FFT. This effect applies also to all the data and plot files produced. The filter can be edited by right-clicking on the red arrow, and deleted by clicking the minus icon on the pop-up.

Recording

Generated files contents and their naming conventions

Configuration files

The configuration files are text files that can be created by pressing the *SaveAs* button and giving a new name for the file. These files contain setup data in format *key=value* one per row, split in *[groups]*. If needed, they can be patched by hand with a text editor. In case of missing fields, *Echoes* takes a silently a default hardcoded value.

Log file

The file *echoes.log* is a system log created in the working directory, overwriting a previous one if present. It could be interesting only for debugging purposes, to track what the program were doing just before a crash. With *-n* parameter at command line, its verbosity can be incremented when more details are needed. The defined levels are:

n	Mnemonic	Meaning
0	NONE	The log file is never created
1	FATAL	Logs only fatal errors that cause immediate crash
2	CRITICAL	Logs also critical errors causing loss of functionalities
3	WARNING	Logs also warnings about anomalous situations that could cause future loss of functionalities
4	INFO	Logs also some details about the internal state of the program that are not presented on GUI
5	DEBUG	Logs also cryptic and verbose messages for development/debugging purposes

On Linux platforms, *Echoes* uses the system logging daemon facility *syslog* for logging purposes, so echoes.log won't be created to avoid redundancies.

The text sent to the logger can be sent to the standard output too by specifying the parameter *-v* in the command line. In this way, the log messages can be watched even in real time in a console or terminal window while the program runs.

Automatic screenshots

Automatic screenshots are *.png* images captured on waterfall window while acquisition runs in *automatic* or *periodic* acquisition modes. Their file names are coded as follows:

```
autoshot_<configuration name>_<acquisition mode>_<date/time in ISO-8601
format>_<nnnnn>.png
```

for instance:

```
autoshot_AUTO_GRAVES_automatic_2021-01-04_00067.png
```

tells us that it's an image captured automatically on *Jan 04, 2021* when the program was running in *automatic acquisition mode* after loading a configuration file named *AUTO_GRAVES.rts* and it is the *67th* image captured since 00:00 UTC.

The last 5 digits are the *event/screenshot reference number* (column 1 in statistic CSV).

The screenshots caught while running in *periodic acquisition mode* differ only for the presence of the UTC Hour besides the date and progressive number:

```
autoshot_default_periodic_2021-01-04T14-37-06Z_00003.png
```

The hour is needed because in *automatic mode* the event numbering is zeroed by postprocessing thread while archiving the data at midnight UTC, while this doesn't apply to files generated in other acquisition modes; the progressive number is zeroed only by stopping the acquisition, and if periodic acquisition runs for many days, the numbering covers the entire period.

Manual screenshots

The screenshots can be generated even manually in any moment, regardless the acquisition mode selected, by pressing the button *Manual shot* at bottom/right corner of the waterfall.

Manual shots numbering follow a different progression (*mmmmm*) than automatic ones (*nnnnn*) and are named as follows:

```
manshot_<configuration name>_<acquisition mode>_<datetime in ISO-8601
format>_<mmmmm>.png
```

Manual shots are never considered for statistic purposes.

Dump files

The files containing data to be plotted are called *dump files*. There are two types of dump files: *power dumps* and *spectral dumps*.

Power dumps

These ASCII files contains all the information needed to plot a 2D power vs. time diagram, without information about frequencies. They are produced when *Plot type* is *power 2D* regardless the acquisition mode chosen.

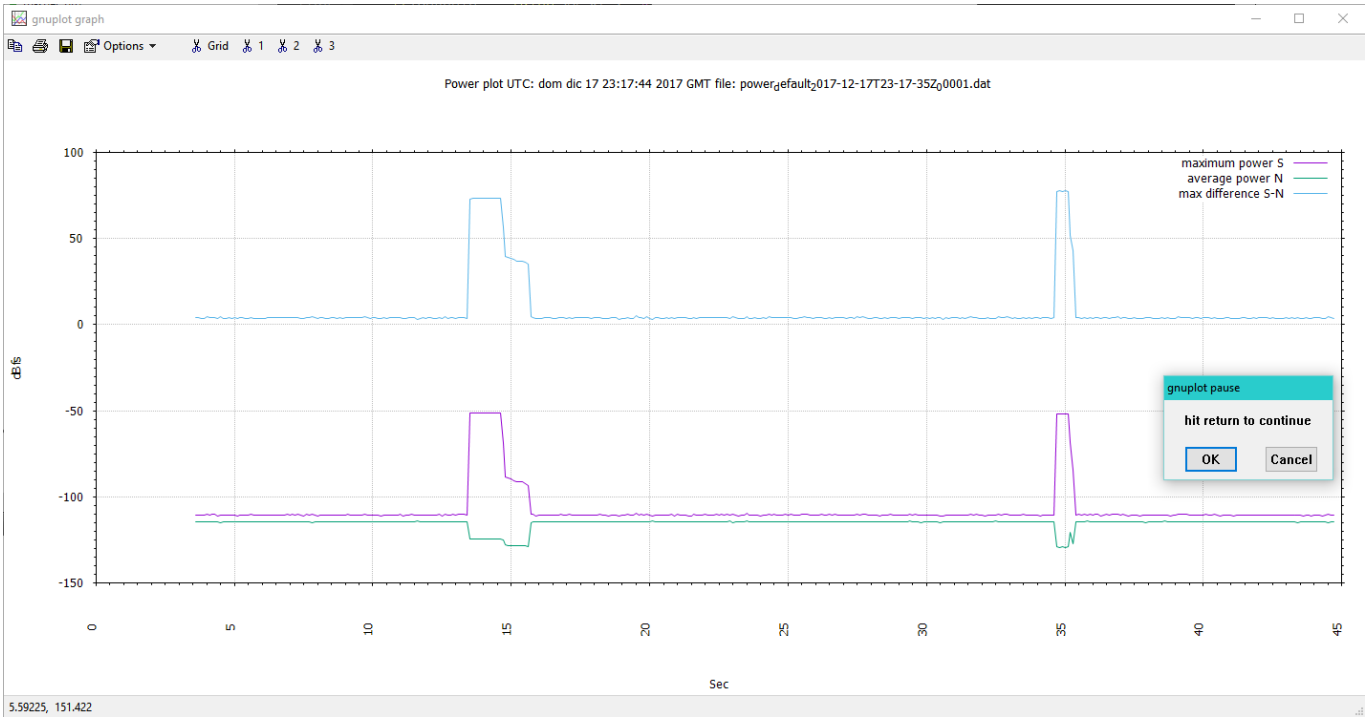
Each line of the file contains four numbers:

1. Unix timestamp (seconds elapsed since midnight UTC, Jan 1, 1970)
2. average signal level in the scan (N).
3. maximum signal level in the scan (S).
4. difference (S-N) in the scan.

for instance:

```
1609774415.473    -1.03   -16.35   -15.32
1609774425.472    -4.36   -16.35   -11.98
1609774435.474    -7.75   -16.35   -8.59
1609774445.474    -11.12   -16.35   -5.23
1609774455.474    -14.49   -16.35   -1.85
1609774465.473    -17.85   -16.35   1.50
1609774475.472    -21.18   -16.35   4.84
1609774485.474    -24.54   -16.35   8.19
...
```

The resulting plot looks similar to the *power history* graph at waterfall's right side.



When running in *Periodic* or *Automatic* modes, the power dumps files comply with the following naming convention:

```
power_<configuration name>_<acquisition mode>_<date in ISO-8601
format>_<nnnnn>.dat
```

like

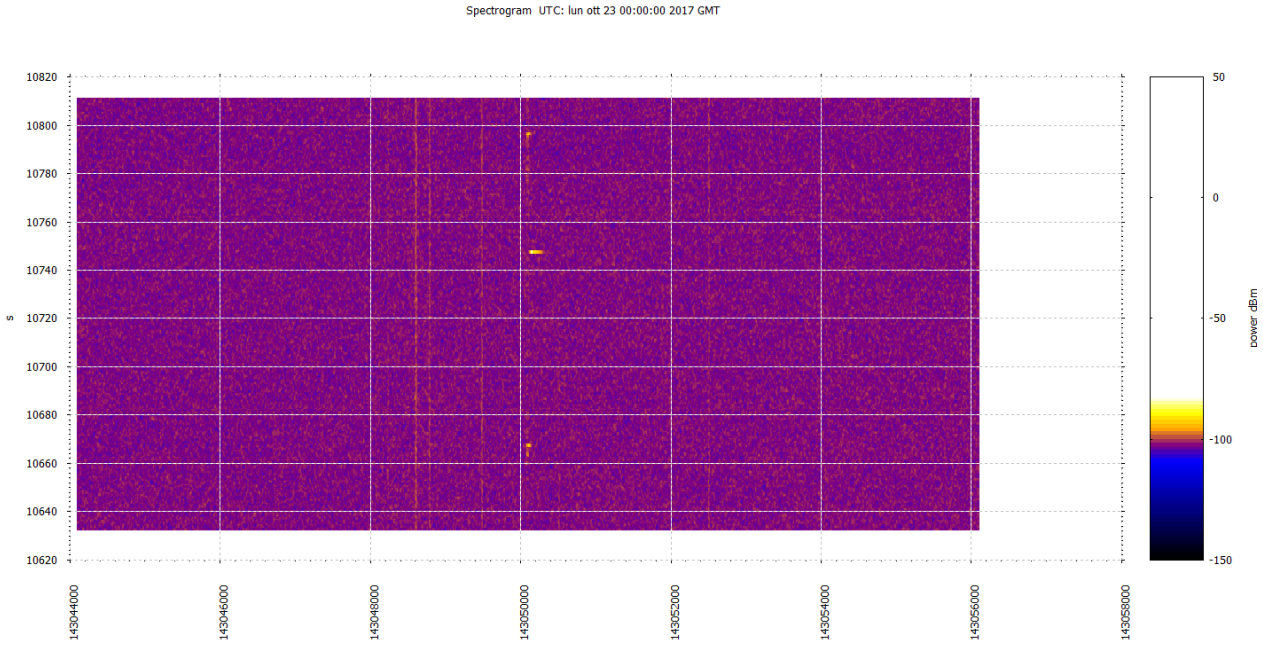
```
power_default_periodic_2021-01-04T21-42-15Z_00001.dat
```

The power dumps can be generated even in *Continuous mode*, that produces a continuous dump file that grows until acquisition stops. The power dumps follow the same naming convention except for the progressive number that is not needed:

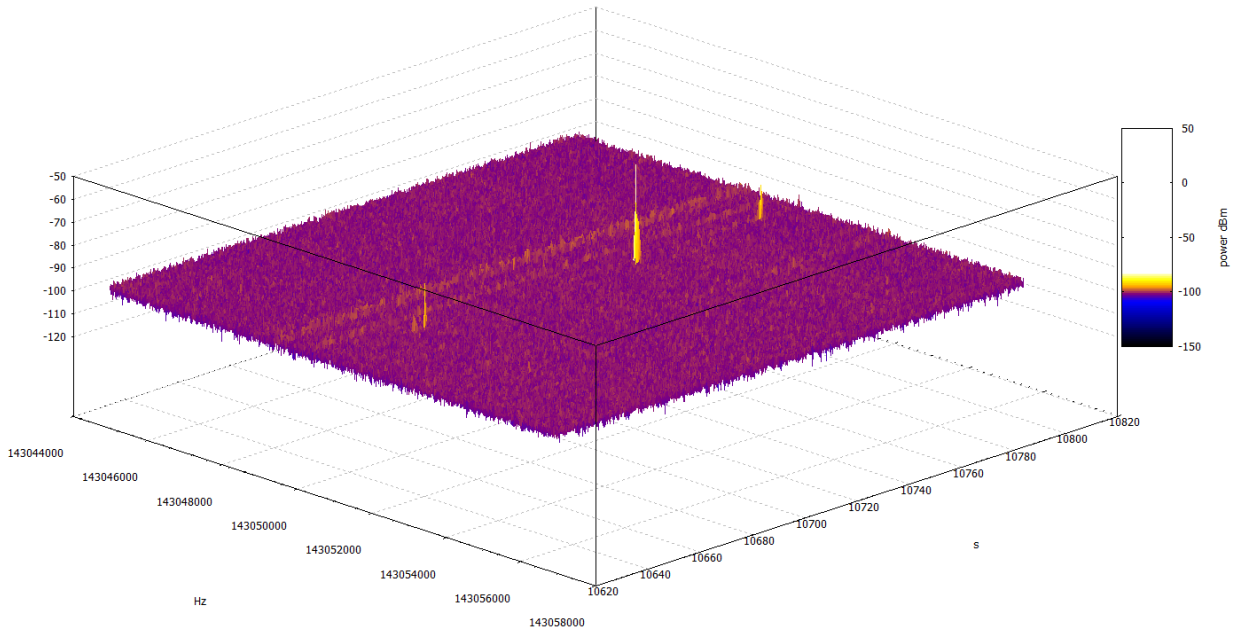
```
power_default_periodic_2021-01-04T15-32-29Z.dat
```

Spectral dumps

These ASCII files contains all the information needed to plot a waterfall-equivalent diagram. The spectral dump files can be plotted on a 2D surface like a waterfall (*Plot type = color mapped 2D*)



or in a 3D perspective (*Plot type = perspective 3D*)



the format and contents of these dump files is the same, only the *GNUplot command file* (described later) differs.

Each line contains three numbers:

1. Unix timestamp (seconds elapsed since midnight UTC, Jan 1, 1970)
2. FFT point frequency [Hz]
3. FFT point power [dBfs]

for instance:

```
1609794049.307    2158   -22.12
1609794049.307    4316   -20.67
1609794049.307    6474   -20.66
1609794049.307    8633   -21.07
1609794049.307    10791   -20.77
1609794049.307    12949   -21.47
1609794049.307    15107   -20.59
1609794049.307    17266   -20.63
1609794049.307    19424   -21.01
1609794049.307    21582   -20.64
1609794049.307    23741   -21.01
1609794049.307    25899   -20.72
...
```

When a scan line is completed, the last row contains further 4 numbers:

4. a repetition of first number: Unix timestamp in seconds.
5. average signal level in the scan (N).
6. maximum signal level in the scan (S).
7. difference (S-N) in the scan.

like this:

```
1609794049.307    1500000   -20.96    1609794049.307   -22.67   -16.35   6.32
```

the purpose of these additional numbers is to make still possible to plot a *Power 2D* graph from a spectral dump, ignoring the three-numbers rows.

After the last row, an empty row marks the end of a scan line. This is required by GNUplot to plot surfaces.

The number of scan lines present in a file depends of the *Refresh interval measured* value and the *Data dump coverage* set. For this reason the spectral dumps are bulky to manage. The dump generation should be enabled only on machines having plenty of free disk space to store them.

When running in *Periodic* or *Automatic* modes, the spectral dumps files comply with the following naming convention:

```
spectra_<configuration name>_<acquisition mode>_<date in ISO-8601
format>_<nnnnn>.dat
```

like

```
spectra_default_periodic_2021-01-04T21-42-15Z_00003.dat
```

The spectral dumps can be generated even in *Continuous mode*, that produces a continuous dump file that grows until acquisition stops. The spectral dumps follow the same naming convention except for the progressive number that is not needed:

```
spectra_default_continuous_2021-01-04T21-01-55Z.dat
```

When no more space is available on the drive (except the space reserved with *Reserved storage* in *Output* tab) the acquisition stops automatically.

GNUplot command file

This is an ASCII file containing commands that GNUplot executes to plot all the dump files produced in the acquisition session just ended. It is created when the *GNUplot output* checkbox is set, regardless the acquisition mode. Depending of the *Plot type* requested, the commands in the file change, in order to plot the data in the required format.

The naming convention for these files is:

```
<plot type>_<configuration name>_<acquisition mode>_<date in ISO-8601 format>.plt
```

where <plot_type> can be:

- *power_* when the requested *Plot type* is *Power 2D*
- *spectra_* when the requested *Plot type* is *Color mapped 2D*
- *spec3D_* when the requested *Plot type* is *Perspective 3D*

for instance:

```
power_default_continuous_2021-01-04T15-32-29Z.plt

spectra_default_periodic_2021-01-04T21-42-15Z.plt

spec3D_default_automatic_2021-01-04T23-07-35Z.plt
```

The command files are simple ASCII text files, so they can be edited later, to better suit user's needs. There are many examples at gnuplot.info site to learn how to personalize the plots. It's an universe in itself.

When opening the command file with GNUplot, it starts a sequential presentation of all the plots generated during the session. A small dialog box (*gnuplot pause*) appears to display the next plot, till the end. The box can be dragged in a corner with the mouse if covers the plot. The plotting window supports some keyboard/mouse commands to zoom / shift / rotate the plot. Plots can also be saved as image files.

Statistic CSV table file

These files are always generated regardless the acquisition mode set; anyway their content changes depending of it. The number of columns is always the same, unused columns are kept empty. In *automatic* acquisition mode, all the columns are filled.

> **Note:** in 0.27 there is a column containing zeros reserved for future purposes

The file naming convention is similar to that defined for screenshots:

```
scan_<configuration name>_<acquisition mode>_<date/time in ISO-8601 format>.csv
```

except the prefix *scan_*, the other fields have the same meanings. In *automatic* acquisition mode one files covers exactly one day while in other modes the file is created when the acquisition starts so the date and hour embedded in file name are related to the moment when acquisition started.

The statistic CSV is a textual semicolon-separated values table, 34 columns wide. The table below describes the content of each column. The letter refers to the column ID when the file is opened in a spreadsheet :

Column nr.	Column ID	Title	Description	Automatic mode only
1	A	Event/scan number	While working in *continuous* or *periodic* mode, this is a progressive *scan number* and each row gets a different number. Numbering starts from 1 at each acquisition session. Under *automatic* mode instead, the program gets the capability of events detection so this column keeps the progressive *event number*. Since always 3 lines are produced for each event (the raising front, the peak and the falling front) these lines will be marked with the same event number.	
2	B	Date	Date UTC of scan/event in local short format.	
3	C	UTC time	UTC time including milliseconds	

Column nr.	Column ID	Title	Description	Automatic mode only
4	D	Unix timestamp	Unix timestamp in seconds from January 1, 1970 midnight UTC	
5	E	Tune [Hz]	Dongle tuning frequency	
6	F	Bandwidth [Hz]	Subsampled bandwidth. If Downsampler bypass has been checked, it matches the sample rate.	
7	G	Central Hz	Frequency centered in waterfall, it's also the center of the peak detection range	
8	H	Lowest Hz	Lowest frequency covered by waterfall at full bandwidth.	
9	I	Highest Hz	Highest frequency covered by waterfall at full bandwidth.	
10	J	Zoomed BW Hz	Bandwidth displayed on waterfall after applying *Freq. Zoom*. If zoom is set to 1.0X, it matches the full bandwidth.	
11	K	Step Hz	Resolution in Hz of a single FFT point	
12	L	Threshold mode	Threshold mode: *Absolute*, *Differential* and *Automatic*	x
13	M	Up threshold [dBfs]	An event peak happens when the reference data value exceeds this threshold. This value matches the one entered in GUI, except if *Threshold mode* (column L) is *automatic*; in that case, the value is calculated dynamically at each scan on the basis of the given percentage (column P)	x
14	N	Dn threshold [dBfs]	An event starts when the reference data value exceeds this threshold and terminates when the value falls under it. This value matches the one entered in GUI, except if *Threshold mode* (column L) is *automatic*; in that case, the value is calculated dynamically at each scan on the basis of the given percentage (column P)	x
15	O	Auto up threshold [%]	Upper threshold percentage in *automatic mode*, filled only if column (L) is *Automatic*	x
16	P	Auto down threshold [%]	Lower threshold percentage in *automatic mode*, filled only if column (L) is *Automatic*	x

Column nr.	Column ID	Title	Description	Automatic mode only
17	Q	Range low Hz	Lowest frequency in *event detection range*	
18	R	Range high Hz	Highest frequency in *event detection range*	
19	S	Top peak (S) [dBfs]	Power of the highest signal got in the *event detection range*.	
20	T	Average peak [dBfs]	Average of the last *n* peaks detected. *n* is the number entered in *Averaged scans for N calculation*, *Output* tab.	
21	U	Average noise (N) [dBfs]	Filtered average value in the zoomed BW. This is in fact an average of average: the average values of the last *n* scans are averaged to obtain this value. *n* is the number entered in *Averaged scans for N calculation*, *Output* tab.	
22	V	Difference (S-N) [dBfs]	Instantaneous difference between columns (S) and (U). *S-N*	
23	W	Average difference (S-N) [dBfs]	Averaged difference of the latest n S-N values. *n* is the number entered in *Averaged scans for N calculation*, *Output* tab.	
24	X	Top peak [Hz]	Frequency of the peak signal (S)	
25	Y	reserved for future use	always zero	
26	Z	Standard deviation	Standard deviation of *S* versus *N*	
27	AA	Lasting [mS]	Time difference between the falling and raising fronts of the event. This value is nonzero only in falling front rows.	x
28	AB	Frequency shift [Hz]	Difference between the *peak frequency* (X) and the *central frequency* (G)	x
29	AC	Echo area [FFT points]	Number of contiguous FFT points which power value exceeds the column (N) forming the shape of the echo. This data is available only on falling edges.	x

Column nr.	Column ID	Title	Description	Automatic mode only
30	AD	Interval area [FFT points]	Number of all the FFT points included in *event detection range* multiplied for the number of scans covered by the event, forming a rectangle including the shape of the echo. This data is available only on falling edges.	x
31	AE	Peaks count	Number of the FFT points included in *event detection range* that exceed the upper threshold (M) counted on all the scans covering the event. This data is available only on falling edges.	x
32	AF	LOS speed [m/s]	Apparent speed (Line Of Sight). This data is available only on falling edges.	x
33	AG	Event status	This column marks the event fronts: raise, peak, fall.	x
34	AH	Shot name	File name of the screenshot representing this scan or event.	

A note about *LOS* column: that speed is calculated as follows:

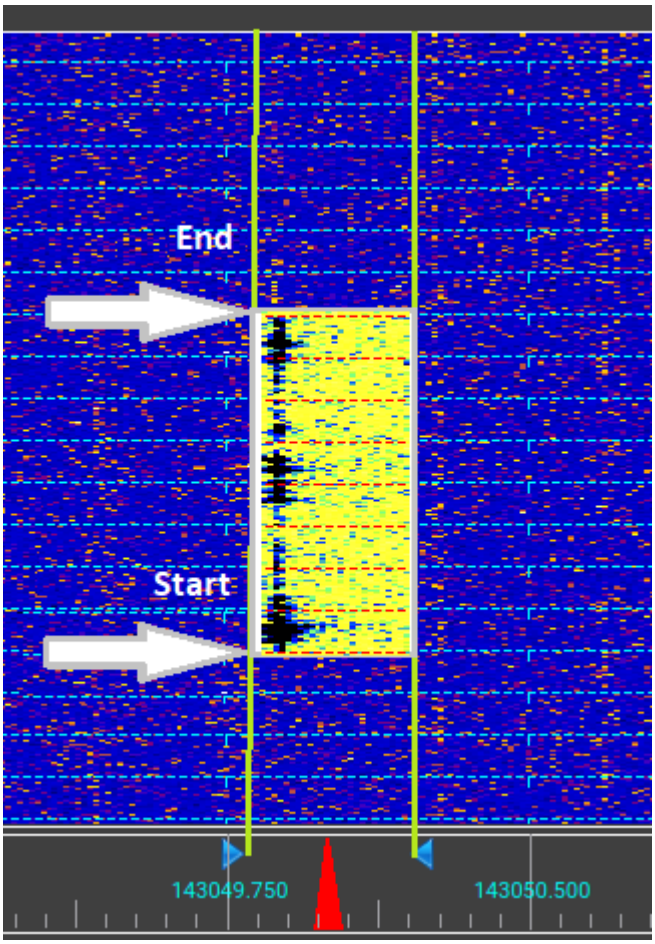
```
frequency shift at raising edge  x  Speed of light
-----------------------------------------------------
                central frequency
```

It should be plausible on head echoes only, but they're still hard to grasp for now. So in most of cases the content of this column will be unreliable.

Another note is about the *areas* columns:

- *Echo area* - column 29 / AC
- *Interval area* column 30 / AD

The image below shows what is intended for *areas*: the *Interval area* is the yellow box including the echo and covering the echo lasting and the event detection interval, while *Echo area* is the area covered by the echo itself, here in black.



Areas are expressed in *FFT points*, not pixels, so resizing the waterfall window doesn't affect areas. These areas are used (with other columns too) to discriminate false positives (*fakes*) in post-processing.

In order to avoid problems while importing the statistic CSV with office suites, the decimal point used in files could be a point or a comma, depending of the operating system localization (the language selected with *-l* option at startup is irrelevant).

Daily report

The data contained in report files are related to archived data (the archive is explained in Archiving section), that means only data recorded in *automatic mode*. One daily report file is created for each configuration *rts* archived. It's a CSV table file that resumes the counts of all the captured events contained in the statistic CSV files found in working directory and related to the same configuration.

There is one row per each day covered by those files; the events found are counted separately hour by hour and discriminated in underdense, overdense and fakes. Once opened in a spreadsheet, the file looks as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	Echoes Daily Report														//														
2	Date:	00h							01h						//		23h						Daily totals						
3		Max S [dBm]	Avg N [dBm]	Total	overdense	underdense	fakes		Max S [dBm]	Avg N [dBm]	Total	overdense	underdense	fakes	//	Max S [dBm]	Avg N [dBm]	Total	overdense	underdense	fakes		Max S [dBm]	Avg N [dBm]	Total	overdense	underdense	fakes	
4	dom dic 27 2020	0,25	0	11	2	9	0		3,54	-15,6	13	3	10	0	//	2,16	-12,84	10	2	8	0		27,41	-14,12	346	98	235	13	
5	lun dic 28 2020	1,01	0	9	1	8	0		5,9	-14,82	9	2	6	1	//	0,59	-13,77	16	0	16	0		19,49	-13,35	361	79	268	14	
6	mar dic 29 2020	1,97	0	11	2	9	0		9,47	-16,7	15	3	11	1	//	2,57	-15,4	2	0	2	0		15,61	-14,46	213	47	162	4	
7	mer dic 30 2020	5,9	0	3	0	3	0		-0,87	-15,46	3	1	2	0	//	-1,27	-16,06	3	2	1	0		10,9	-14,98	83	16	60	7	
8	gio dic 31 2020	-	-	0	0	0	0		0	0	1	0	1	0	//	-2,01	-16,51	1	0	1	0		3,06	-14,75	44	8	31	5	
9	ven gen 1 2021	-	-	0	0	0	0		0	0	1	0	1	0	//	-1,81	-16,75	1	0	1	0		6,14	-13,51	62	13	33	16	
10	sab gen 2 2021	0	0	1	1	0	0		-0,22	-15,66	3	0	3	0	//	3,98	-13,12	3	1	2	0		15,04	-14,19	116	25	42	49	
11	dom gen 3 2021	21,51	0	10	1	8	1		11,46	-15,62	11	3	7	1	//	15,01	-13,63	2	0	2	0		33,78	-14,45	472	152	301	19	
12	lun gen 4 2021	12,59	0	4	1	3	0		8,24	-16,63	6	2	4	0	//	-	-	0	0	0	0		26,89	-14,9	110	19	90	1	

For representation reasons (the complete table has 176 columns) the columns related to hours 02h to 22h have been cut away.

The file is created (if not yet existing) in the archive, under the folder named as the active configuration and, every midnight UTC (*midnight swap*, see Archiving), a new row with the data related to the day just terminated is appended to it.

Besides the counts, the table includes the values of *S* and *N* by hours. These values plotted in a graph can give an idea of the variation of background noise along the days.

The report files follow this naming convention:

```
daily_<configuration name>.csv
```

for instance:

```
daily_AUTO_GRAVES.csv
```

Full report

The *full report* is a single *HTML* page divided in 7 sections that can be included or not in the report by checking the appropriate boxes in the *Report contents* group, *Reporting* tab. It can be created by request when pressing the *Generate report* button, or automatically after *midnight swap* if the box *Automatic report* is checked and if the *To* date in *Report coverage* group matches the day just terminated.

The file is created in the archive, in the subfolder named as the *to* date, under the *root folder* of the active configuration (see also the *Archiving* section below).

Since it contains relative links to the external images (hourly counts, screenshots and dump images) when copying the report to another support the images present in the root folder and those in subfolders should be copied too maintaining the same subfolders structure.

As alternative, you can print a *pdf* document from the browser to get everything bundled together.

> **Note:** When generating a *full report* any previously existing *html* file present in the destination folder will be erased.

It is possible to generate a *full report* even without having screenshots available, but only dump files. While generating the report, the dumps are plotted then saved on *png* files making the same function of a screenshot, with the difference that there are three different *plot types*.

However, these automatic conversions take time, so the time it takes to generate the entire report, if you decide to include plots instead of including screenshots, could grow by a factor of 4.

The full report file name follows this naming convention:

```
<configuration name>-<start date>_to_<end date>.html
```

for instance:

```
AUTO_GRAVES_2020-12-18_to_2020-12-27.html
```

Please keep present that the start and ending dates embedded in filename are the dates entered in *report coverage* controls, that could not match with the dates of the data reported. This happens, for instance in case the *report coverage* interval includes a *To* date currently in the future and the report is generated now by pressing the pushbutton. The report name will include the future date even if it didn't come yet and the data and images included in the report are necessarily in the past.

You can find the report files in the *date subfolder* related to the last day covered by the report. The links to images and css are relative, so the report can be copied to any other place if you copy the covered *date subfolders* too.

Archiving

Midnight swap

If not yet existing, at midnight UTC, a 5-levels folders hierarchy is created under the working directory, where all the files produced in the past days are classified then moved. When this happens, the acquisition stops and is restarted automatically when the moving terminates. This operation is called *midnight swap*.

Generally speaking, the *midnight swap* takes a couple of minutes or even less if no plots are involved and there is only the past day to archive. Nevertheless, if the files in working directory have been produced in different days and never archived before (because the acquisition never ran on midnight UTC) *Echoes* will archive everyone in a proper day subfolder, increasing the total time.

The last operation executed in the swap is the deletion of expired data. This depends of the number of days specified in *Data lasting*, in *Preferences* tab; once created and filled up the folder of the past day, the oldest folders are deleted.

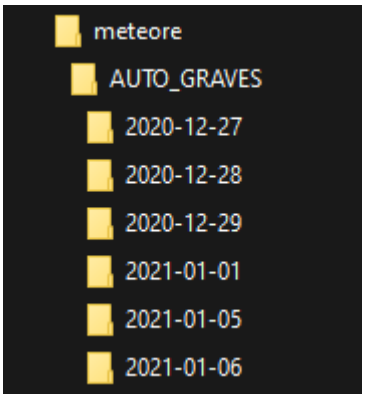
Archive folders hierarchy

Level I

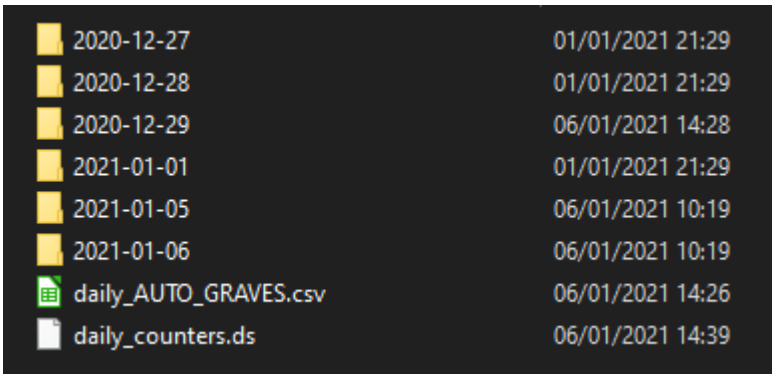
At first level is a folder named as the configuration file active; it is called the *root folder*. This folder contains:

- the *daily report*
- a file called *daily_counters.ds*, that is the binary counterpart of the daily_report.

For instance, this is the archive subfolders structure on my Windows10 laptop, below the working directory, here called *meteore* that is placed on an external USB HDD drive:



and these are the files in *AUTO_GRAVES*, the *root folder* created to store the data collected when *Echoes* was running after loading the configuration *AUTO_GRAVES.rts*.



Level II

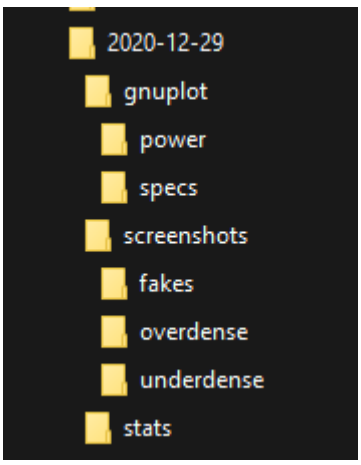
Under the root folder, at second level, there are several folders related each one to a day of acquisition called *daily folders*. The maximum number of daily folders that can be created is defined by *Data lasting*.

Daily folders could contain:

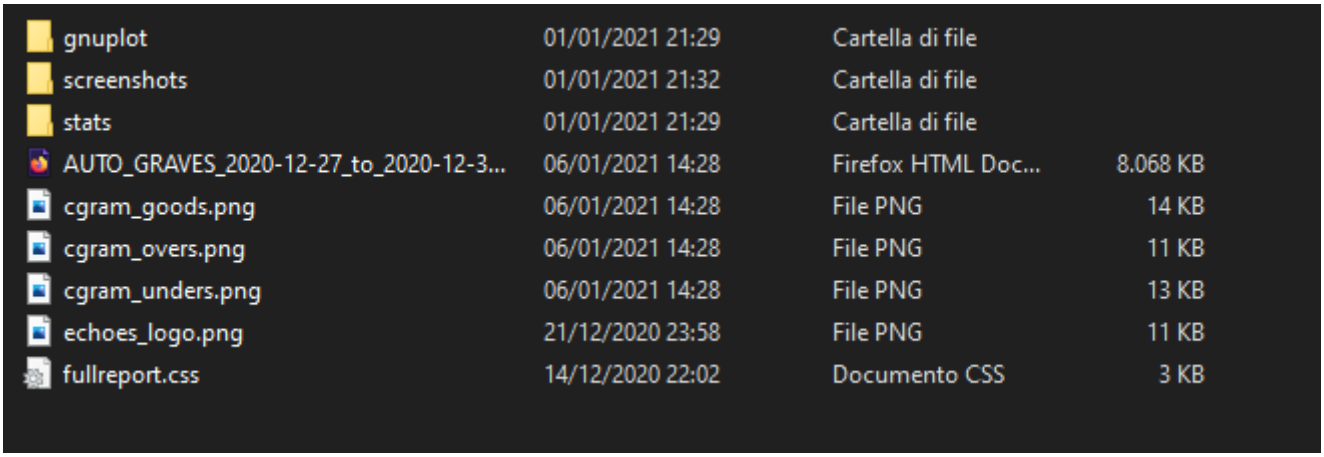
- A *full report* file
- images referenced by the *full report*

As example, this is the daily subfolders structure related to December 29, 2020 on my laptop:

For instance, this is the archive subfolders structure on my Windows10 laptop, below the working directory, here called *meteore* that is placed on an external USB HDD drive:



and these are the files in *AUTO_GRAVES*, the *root folder* created to store the data collected when *Echoes* was running after loading the configuration *AUTO_GRAVES.rts*.



Level III

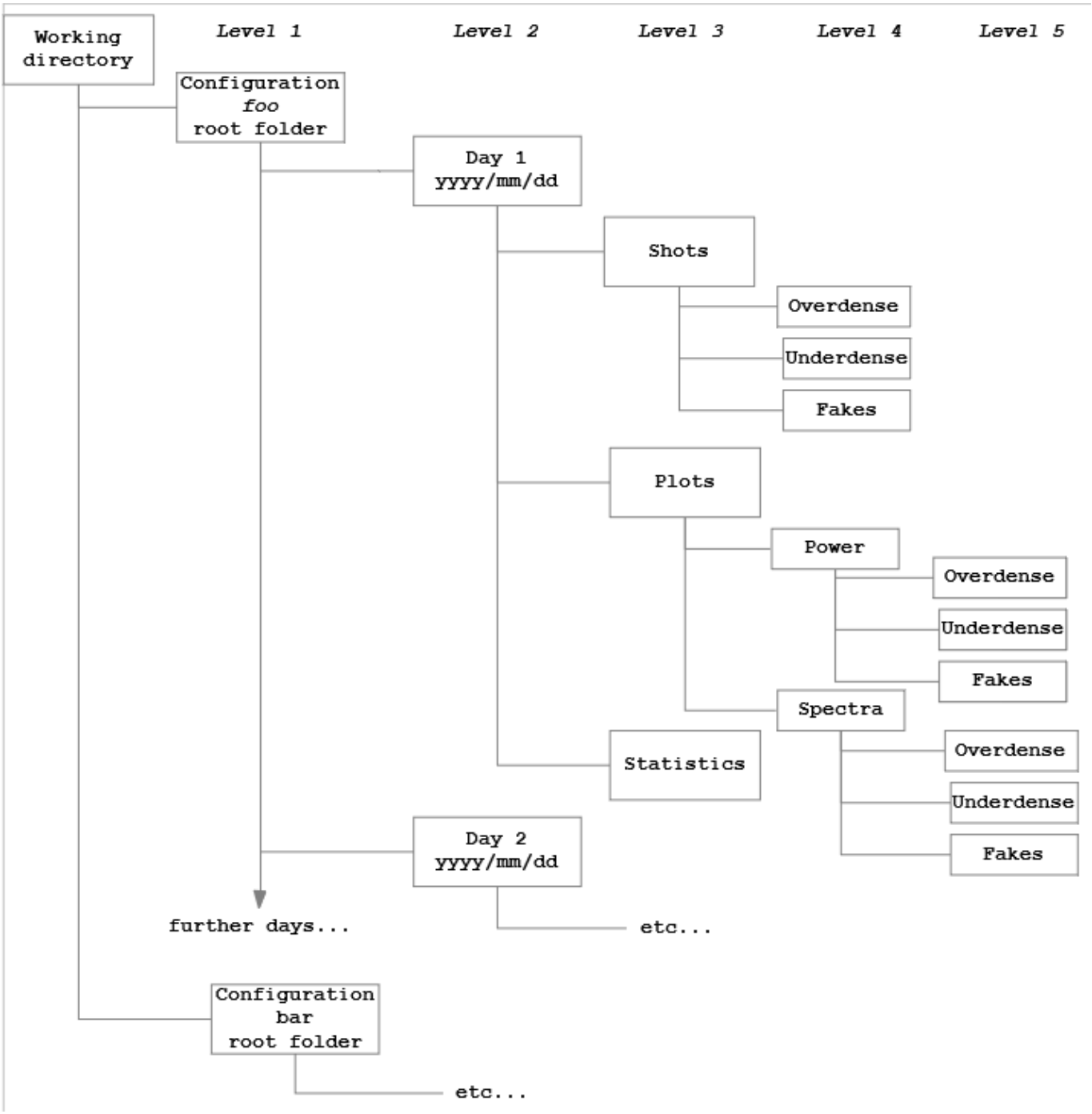
At the third level, under each day folder, three further subfolders are created. The first contains *screenshots*, the second contains the dumps and related *GNUplot* command files, while the third (*stats*) contains the statistic csv table of that day.

Level IV and V

The fourth level folders are created only for shots and plots, to subdivide the related files by category: underdense, overdense and false positives (fakes).
The plots have another subdivision by power and spectra (2D or 3D doesn't matter) before subdivision by category, that is at fifth level.

When the *midnight swap* terminates, the working directory contains only the configuration files, the manual shots and the program's log (unless other configurations have been running having the same working directory).

This is a condensed schema of the archive directories:



Advanced

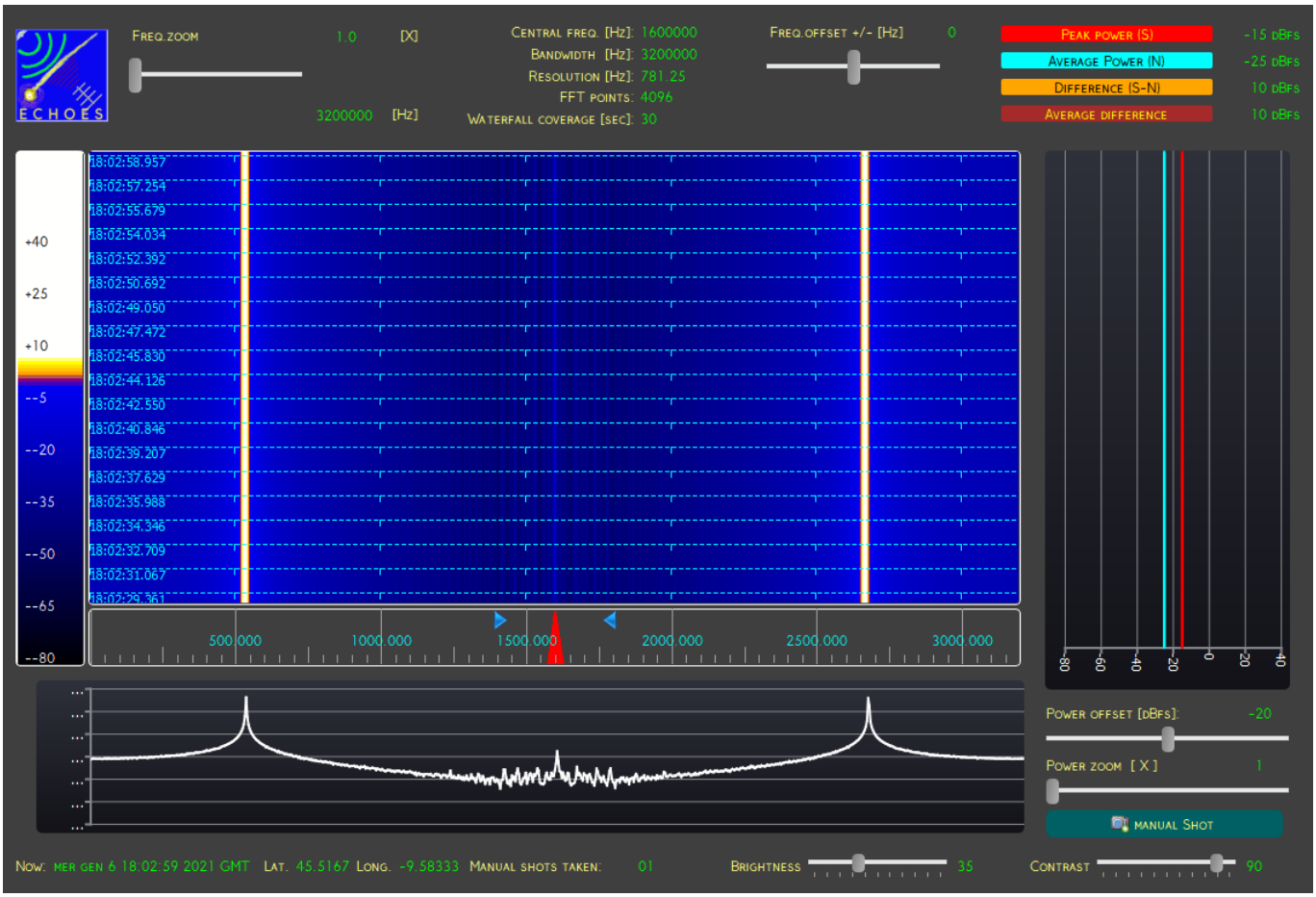
This section is a sort of appendix containing in-depth information on particular topics, some of them already mentioned in other sections, some not.

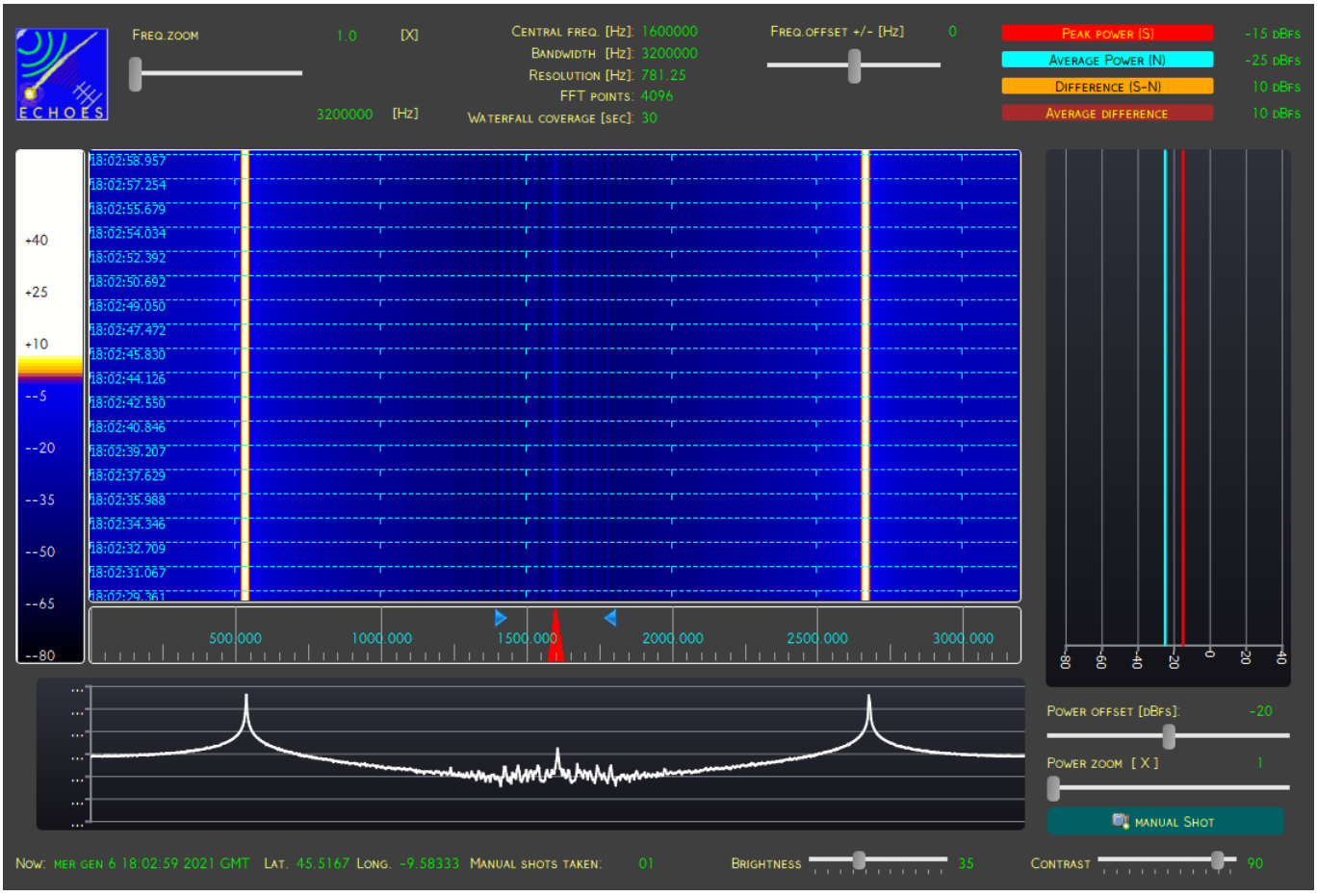
Test patterns

This is not a physical device but an internal functions generator. It has been created for development purposes, to generate *pseudo-echoes* to test the event detection, screenshots, dumps, reporting etc.

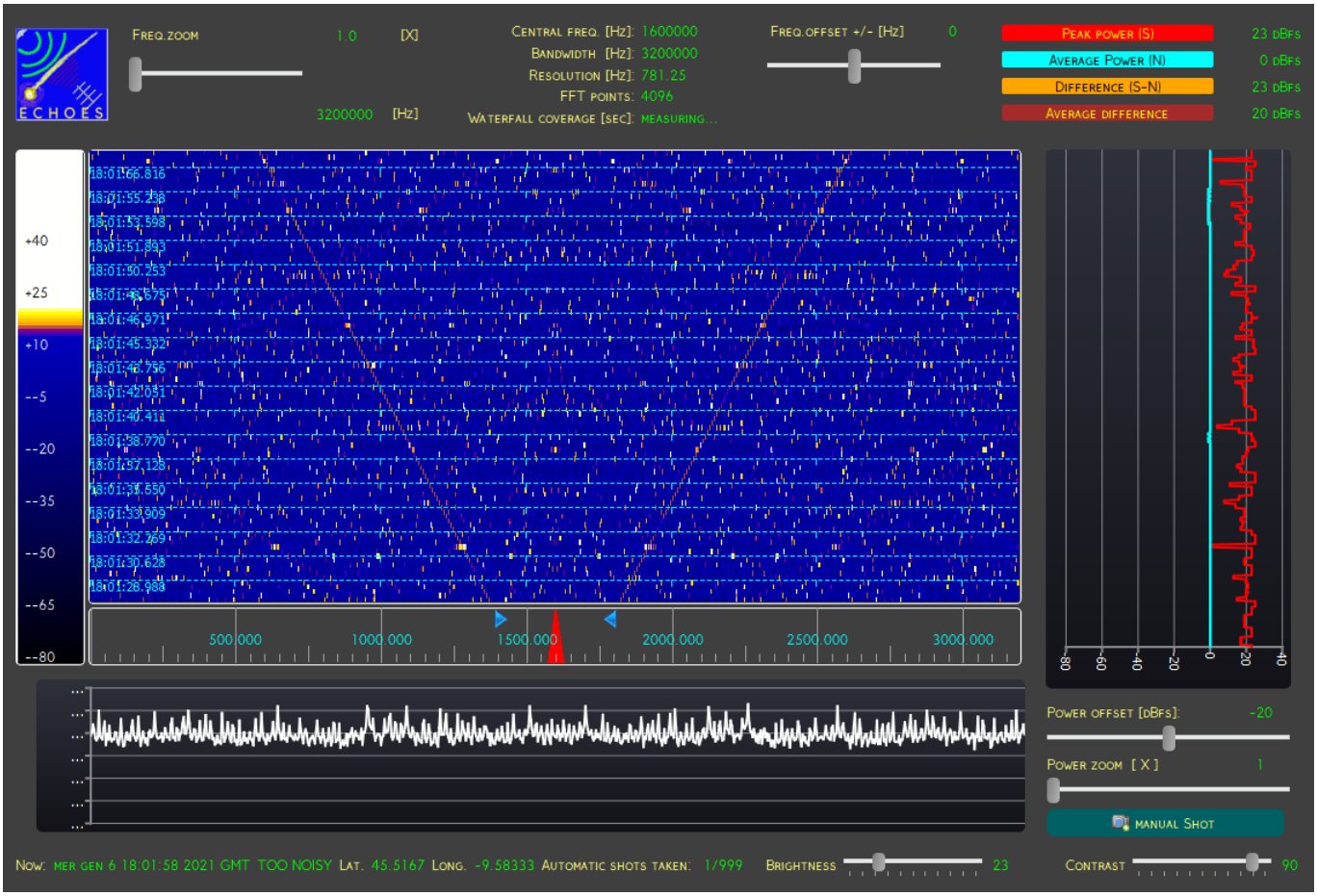
Depending of the *acquisition mode* selected in *Output* tab, this device generates different functions:

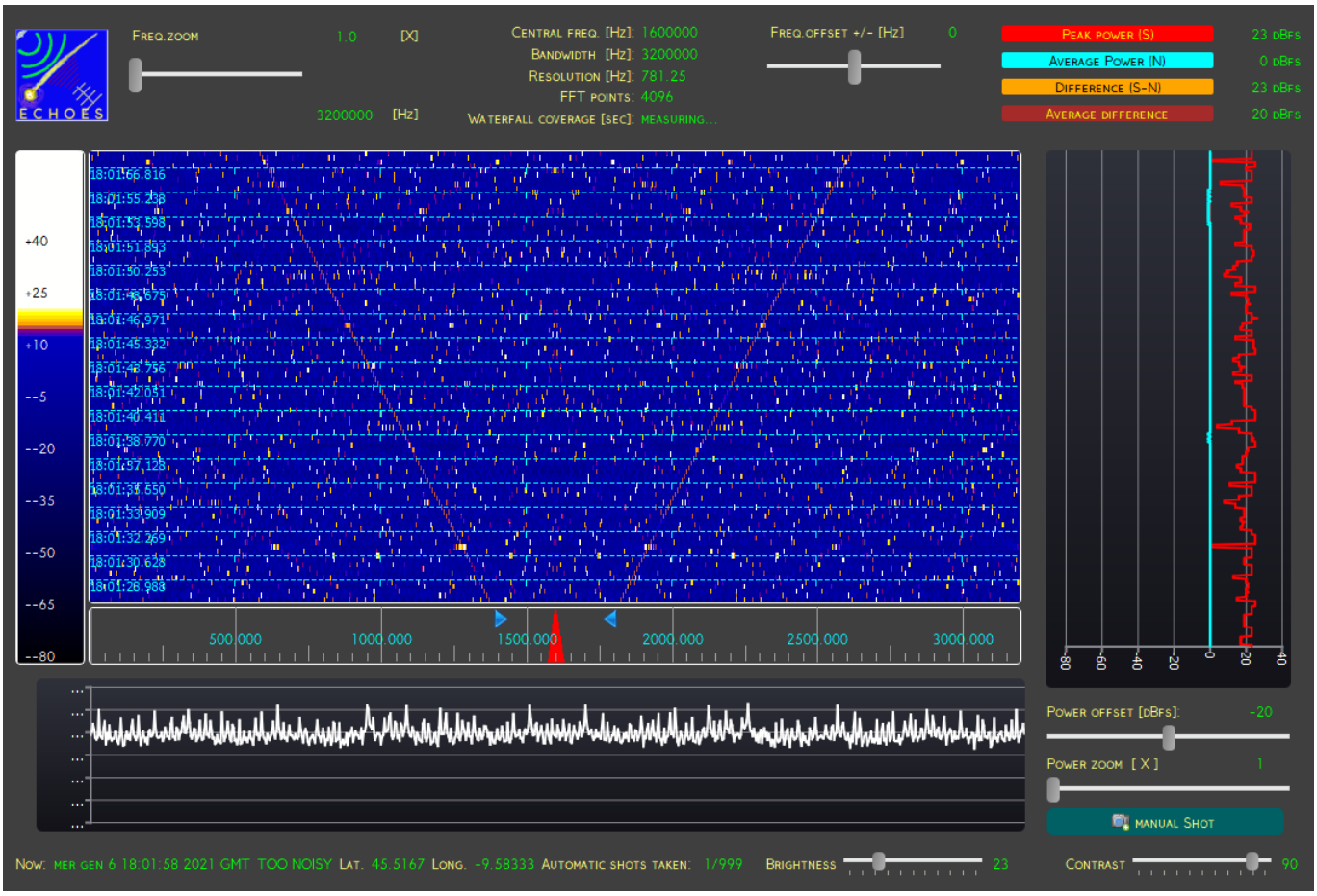
If *Continuous mode* is selected, this device generates a fixed frequency sinewave with frequency equal to a third of the sample rate (default 3,2 MHz) and amplitude equal to gain (default 127). Being the resolution 8+8 bits (I+Q) this means the *I* component varies from -127 to +127 while the *Q* component is always zero.



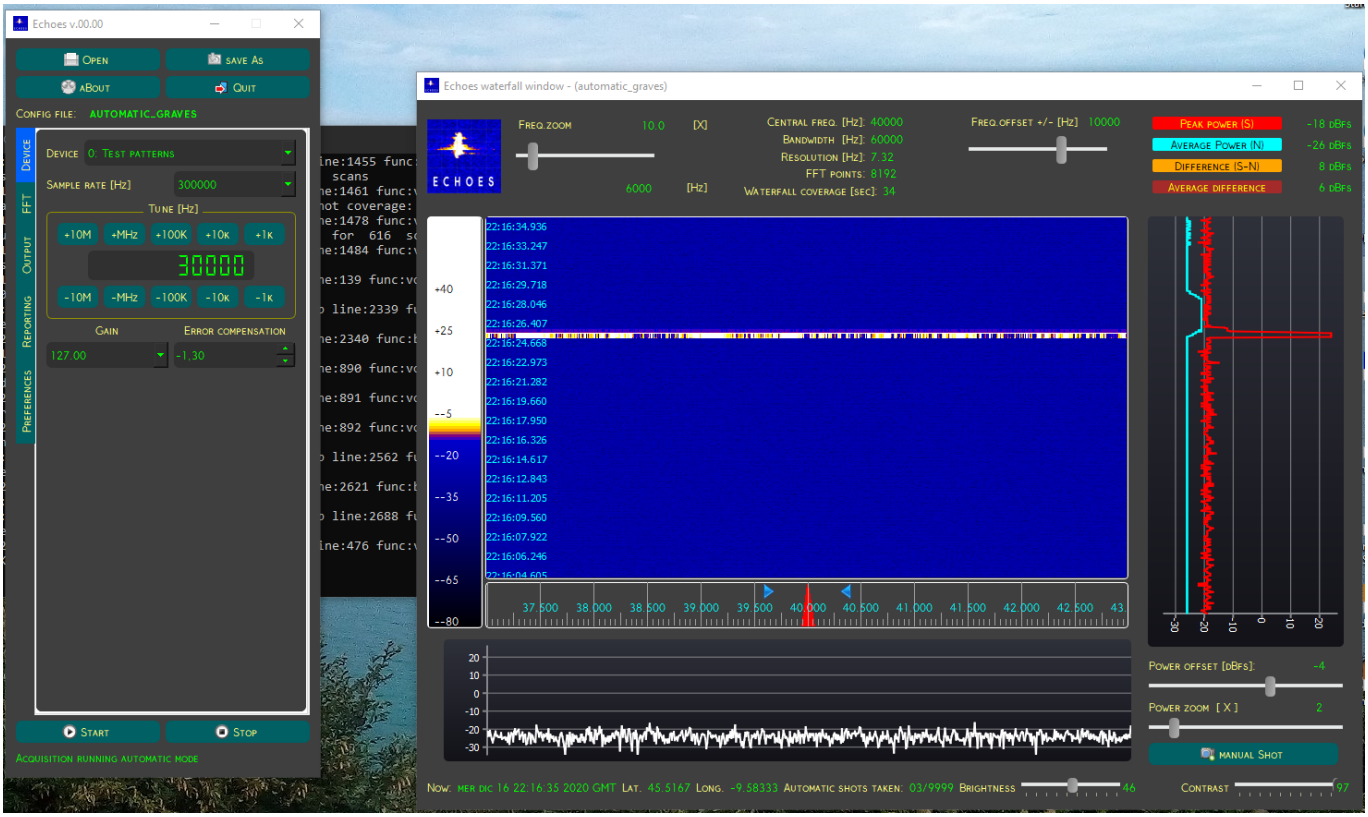


If *Periodic mode* is selected, this device generates a sweeping wave with frequency starting from zero to the tuned frequency increasing quickly.





If *Automatic mode* is selected, this device generates pulses *very vaguely* resembling a meteoric echo with a certain randomness in duration and size.



By default the tuning is set half the sample rate and cannot be changed, while on *waterfall window* the *Freq. zoom* slider is set to 1.0X (the default) and *Freq. offset* is set to zero by default but both can be adjusted.

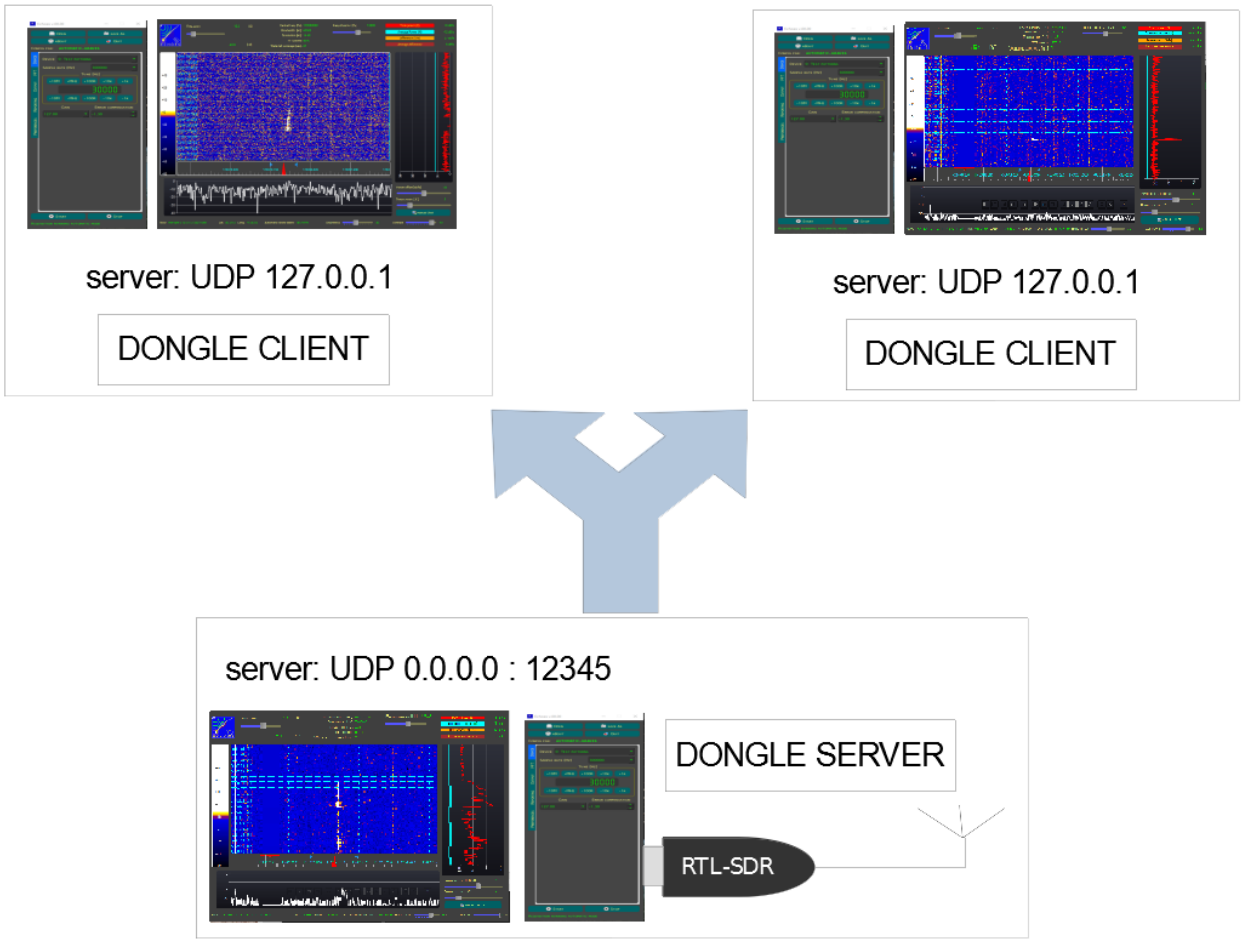
Dongle servers and clients

Besides physical dongles, in the devices list can now appear *dongle servers* too (*UDP server...*). This device appears if in *preferences* tab a server address has been specified. It can be the IP address of another machine running *Echoes* or the *localhost* address (127.0.0.1) in case the same machine runs more *Echoes* instances.

The purpose of running more instances on the same machine is to run simultaneously more waterfalls with different timing, power, bandwidth parameters. When a *Echoes* instance connects a dongle server, that instance becomes a *Dongle client*.

Dongle clients inherit the settings in *Device* tab from the server (except the connected device name): *sample rate*, *tune*, *gain* and *error compensation* will be set on the client from the server. If one or more of these controls are changed on the server, the client gets the modifications.

All the remaining settings in other main window's tabs can be freely changed.



Any *Echoes* instance connected with a physical dongle becomes implicitly a dongle server if the IP set in *preferences* tab is 0.0.0.0 (default value). A server won't send anything on the net until a client instance announces itself; this happens when the client starts the acquisition. After announcing, the client receives from the server the *Device* tab settings then the samples flux starts.

When the server stops the acquisition for any reason (manual stop or *midnight swap*) the client acquisition freezes waiting for samples without stopping. So when the server acquisition restarts, the client too resumes acquisition.

The *midnight swap* occurs on servers and clients. If the *Echoes* instances run on the same machine, it's advisable to define different working directories or different configuration files to avoid messing up the archive with file overlapping.

Finally, the following caveats should be considered when playing with multiple instances.

1. The server should start first, because the client at startup checks if the server is up. If not, the UDP device won't appear in device list.
2. The resolution set on client can be different than server's, but since the amount of data sent by server depends on its settings, the client cannot receive more data. This means that setting on the client an higher resolution than server's could produce evident artifacts on waterfall.
3. It's safer to perform *device* and *FFT* parameters change only when both server and client are stopped, to avoid possible crashes.

Automatic mode thresholds, first setup

A client and a server running on the same station can be very useful to set up the right thresholds to run a station in automatic mode.

First, let's prepare a dongle server configured in *periodic acquisition* then a dongle client connected to the server and configured in *automatic acquisition* and the same refresh interval set on the server.

The server will capture screenshots at regular cadence; the best is to set in *Output* tab, *data dump coverage* the same value displayed in waterfall as *Waterfall coverage* in order to get a long series of contiguous screenshots. The client should shot only on events. After some hours running, the periodic screenshots will show a certain number of echoes. The thresholds on the client should be adjusted until the echoes caught will match the ones seen on periodic screenshots.
