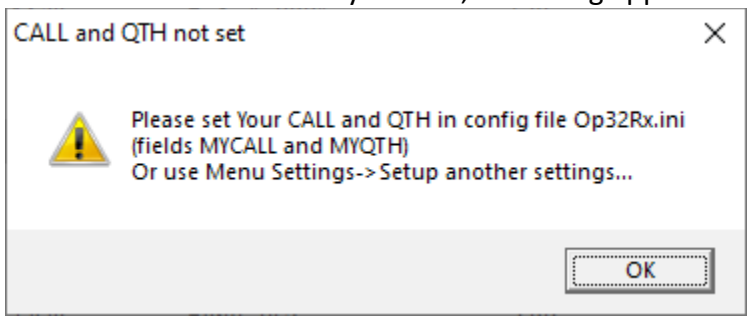
Op32Rx
v.1.8.6

Quick Start Guide

RN3AUS/Alex
January 2020

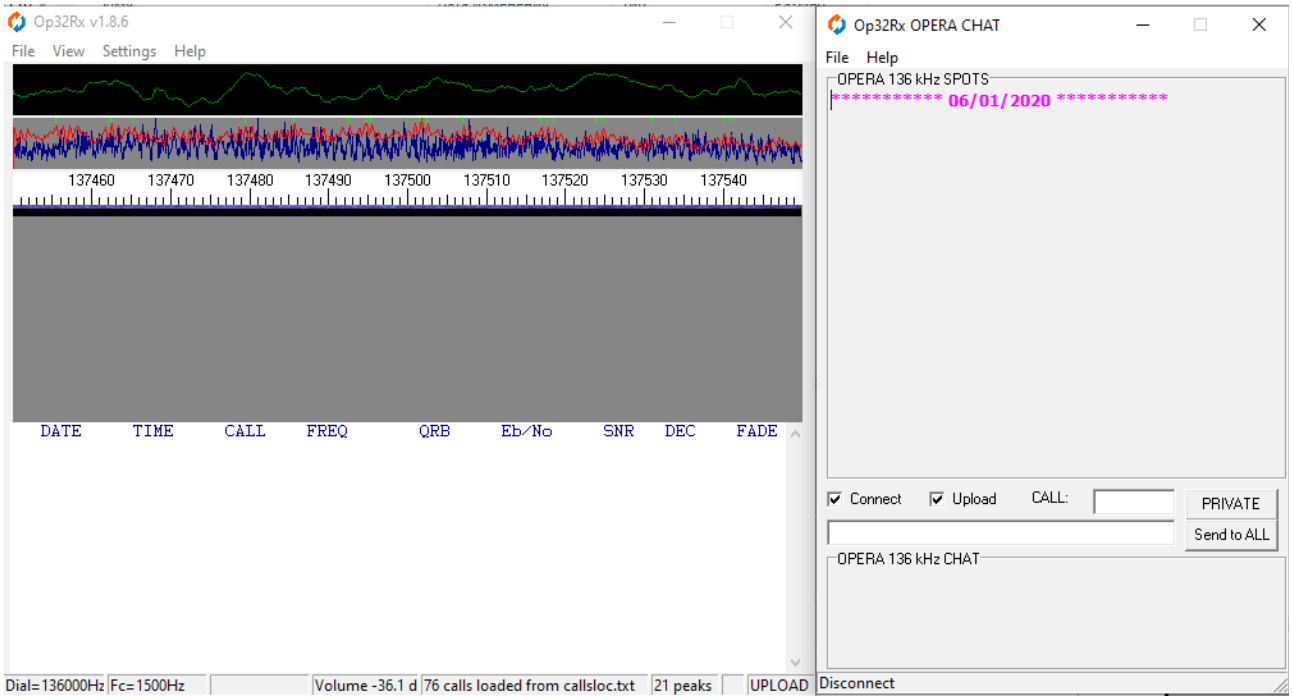
1. Op32Rx is a program for receiving OPERA signals on the amateur band 136 kHz (2200 m). Installing and configuring this program is very easy.
2. Download the latest version of the program from the http://rn3aus.136.su/Op32Rx/index.html site
   (backup site http://rn3aus.narod.ru/Op32Rx/index.html)
   At the time of writing of this management it is v.1.8.6:
   http://rn3aus.136.su/Op32Rx/Op32Rx_v.1.8.6_release.zip

3. Unzip it to an arbitrary directory on your computer 's hard drive. No installation is required. Start the Op32Rx.exe file. The first time you start, a warning appears:



   Click OK, we will enter the required data later.

4. The main window of the Op32Rx program appears and next to it the Opera Chat window.



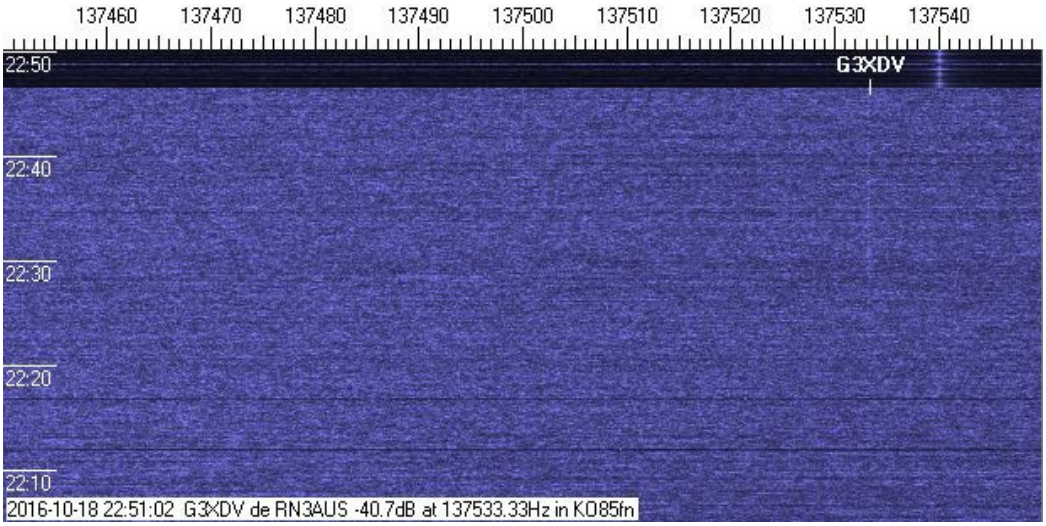
5. Enter the Settings menu → Setup another settings (CALL, QTH, thresholds)... The settings window will open.
   Fill in the fields My CALL, My QTH-lc, My City, My Name, My Ant.

   Attention! If the call sign has a prefix, such as AA1AA/1, it is recommended that you enter it as AA1AA-1. Otherwise, you will not be able to connect to the EA4RCH-5 cluster by which the spots are exchanged.

6. Check your radio settings. The standard tuning frequency is typically 136,000 Hz, with the center of the band allocated for OPERA 137500 Hz mode matching the 1500 Hz audio frequency. These settings are set by default. However, the Op32Rx program also allows you to work with non-standard Dial. For example, your receiver has a Dial of 134400 Hz. In this case, the center of the OPERA window will have to be on the audio frequency 137500 - 134400 = 3100 Hz. We enter: Fdial [Hz]=134400, Fcenter [Hz]=3100. The central audio part can range from 400 Hz to 23000 Hz, allowing any type of home-made receivers and converters to be used.

You can complete the initial settings and click **Apply**. However, to become more familiar with the program, consider the rest of the settings. You may never need to change them.

7. The **Capture path** field defines the path to the folder where screenshots of decoded signals will be stored. Example of such screenshot:



   Such screenshot gives a clear understanding of the signal reception progress, from which it is easy to determine whether the decoding was correct or false. Each screenshot has a unique name, for example G3XDV_20161018_2251.jpg - call sign, date and time of decoding. Thus, in the Capture folder you will have a collection of screenshots of all correspondents you have received.

8. A unique file name is not very convenient if it is desirable to post a screenshot on the Internet. However, the program also creates a screenshot with the op32rx_capture.jpg name each time (this name can be specified in the Capture file field). The file will be placed in the working folder of the program Op32Rx.exe.

9. The **Exec command if decoded** field allows you to run a script every time you decode, such as sending a spot to the Internet, etc. You can use the Test button to verify that the script is running. The checkbox allows execution of the command. Several space-separated parameters are passed as command line parameters to the run file:
   DATE TIME CALL FREQ QRB SNR FADE
   For example:
   2020-01-04 20:01:23 RN3AUS 137500.00 61 -15.8 12
   QRB [km] - distance, if unknown, -1
   SNR [dB] - signal-to-noise ratio
   FADE is the fading percentage of the signal. If the signal is received not by the decoder but by the correlation detector, FADE = 200.

10. The most subtle settings are **threshold** values. The correct operation of the program depends on their values. If the thresholds are reduced, on the one hand the sensitivity to weak signals will increase, but at the same time the probability of false decoding will increase too. By default, thresholds are set close to optimal values. With these settings, my grabber has been working for several years.

11. The **decoder threshold** should be selected to be slightly higher than the occasionally occurring false decodings displayed in the main window status bar, for example, "DEC: 5O0UAV 1. 3dB 41 137505.3." Here, the false decoding level is 1.3 dB. Many

experiments require a decoder threshold of 2 to 4 dB.

12. The **threshold of the correlation detector "opds"** may be slightly lower, from 1.5 to 2.5 dB. By setting the desired threshold, it is necessary to check the quality of work within 24 hours and make sure that there are no false detections.

13. The **OPDS Max1/Max2 threshold** makes the following sense. When comparing several signal variants corresponding to a callsign in the callsloc.txt list with the received signal, two are selected that give the largest correlation sums. The values of these correlation maxima (Max1 and Max2) are compared to each other. Further processing will occur if the Max1/Max2 ratio exceeds the threshold. The experiment provides a basis for setting this threshold on the order of 2.5 dB.

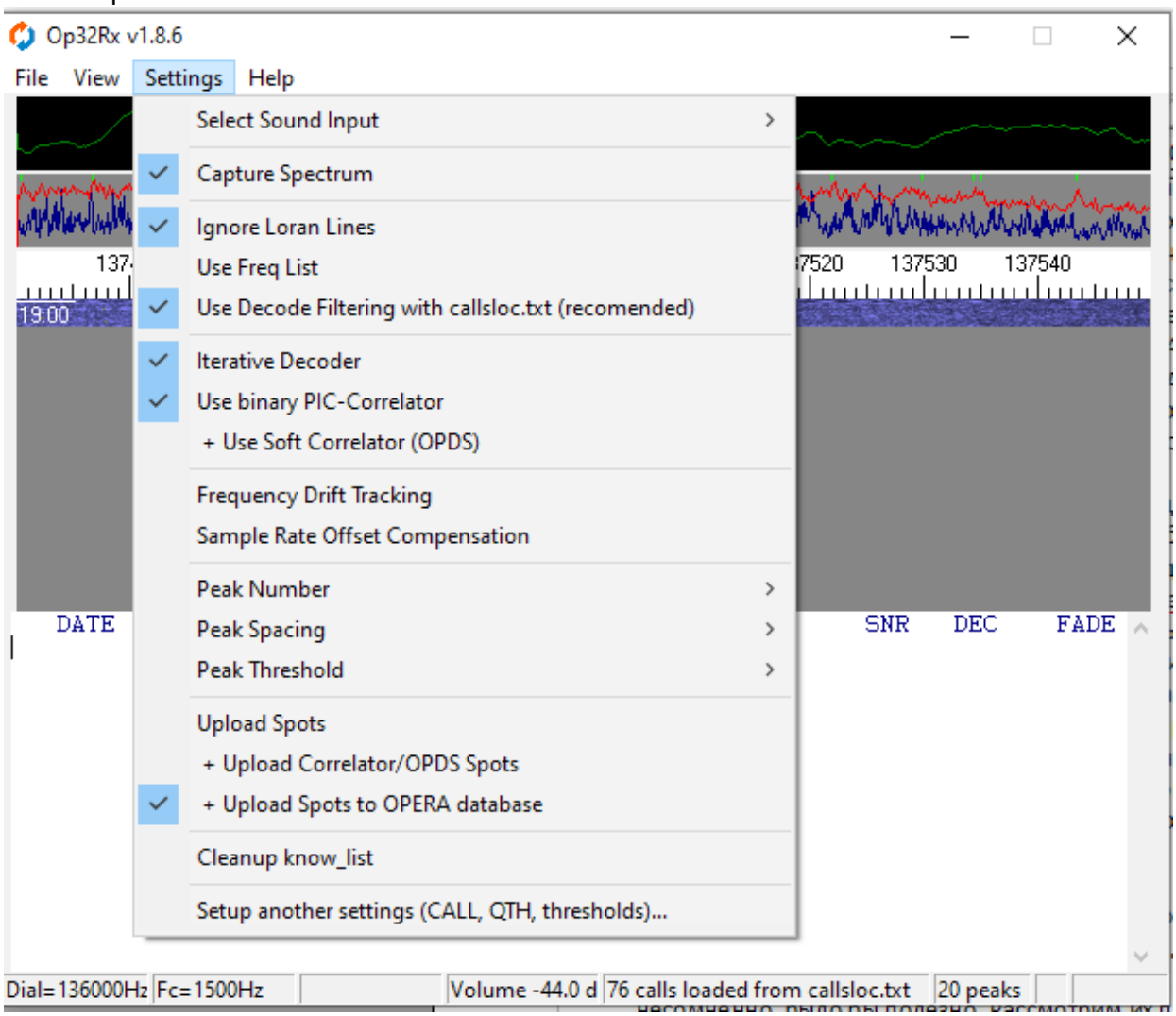
14. The program implements a rather interesting version of the correlation detector - a binary (hard) correlator. Its advantage is the independence of characteristics from the type of noise. The threshold specifies the minimum number of matching bits. For example, a threshold of 164 (of 239) provides extremely rare occurrence of false detections. Somewhat greater sensitivity is provided by thresholds from 161 to 163. The estimate of the probability of false detection of the signal is given to the right of the threshold input field. For example, for threshold 164, false detection (in the absence of powerful interference, by white noise) will occur no more than once every 5 months.

15. The **Opera dB offset** parameter is used to ensure that SNR estimates of received signals using Op32Rx and Opera1.6.5 are the same under equal conditions. In Op32Rx, SNR measurement occurs in the 100 Hz band. In another program, the band is probably 1 kHz. An amendment of -10dB was experimentally determined.

16. The **CALL blocking time** parameter specifies the time when the re-detection of the same call sign is blocked. Since one Opera-32 transmission cycle takes 32 minutes, it is obvious that re-detection cannot take place before the transmission is complete. However, it makes sense to set this parameter from 10 to 20 minutes in case for some reason the correspondent was forced to suspend the incomplete transmission and start it again.

17. So, we reviewed all the settings. Now click **Apply** to save them. Additionally, callsloc.txt will be reloaded into memory. This is useful when callsloc.txt has been edited, but it is not desirable to reboot the program by interrupting reception. Just go to additional settings and click Apply.

18. When you start the Op32Rx.exe, you set a number of settings in the **Settings** menu. By default, the settings that are appropriate for most cases are enabled. You may not even need to change anything. But knowing what these settings mean would undoubtedly be helpful. Let 's look at them in order.



a) **Select sound input** - If your system has more than one sound device, you can select any of them from the list.

b) **Capture spectrum** - save screenshot of received signal (we have already mentioned it in more detail - see para. 7)

c) **Ignore Loran lines** - In the working directory of the program there is a **loran_lines.txt** file containing a list of frequencies that will be ignored. You can also put here any frequencies on which you have constant interference. You can do this either by directly editing the file or by hovering over the spectrogram and right-clicking **Add this frequency to Ignore list.**

d) **Use freq list** - There is a freq_list.txt file in the working directory that contains a list of frequencies that will always be processed, even if there is no spectral maximum. This list is added automatically during each new decoding - it contains the frequency at which the correspondent worked. Many of us have our "favorite" frequencies, so the program will monitor these frequencies more closely. In particular, the decoder will perform more iterations at these frequencies in the hope of detecting a weak signal.

e) **Use decode filtering with callsloc.txt** is the most recommended option. **Please use this option to avoid false decodes!** The point of filtering is that the call sign detected by the

decoder is compared to the list of known call signs (callsloc.txt). If there is a call sign in the list, it is further processed. If it is a new or strange call sign, it is highly likely to be false decoding. However, this call sign is not discarded, but remembered. If it is re-decoded at the same frequency, the program assumes that a new correspondent is on the air, which has not yet been included in the callsloc.txt file. Its call sign is written to the known_list.txt file and further processing is allowed. So, the new correspondent has to transmit twice on the same frequency. The first time it will be ignored, the second time it is successfully decoded and will be listed. Now each transmission will bring a successful decode.

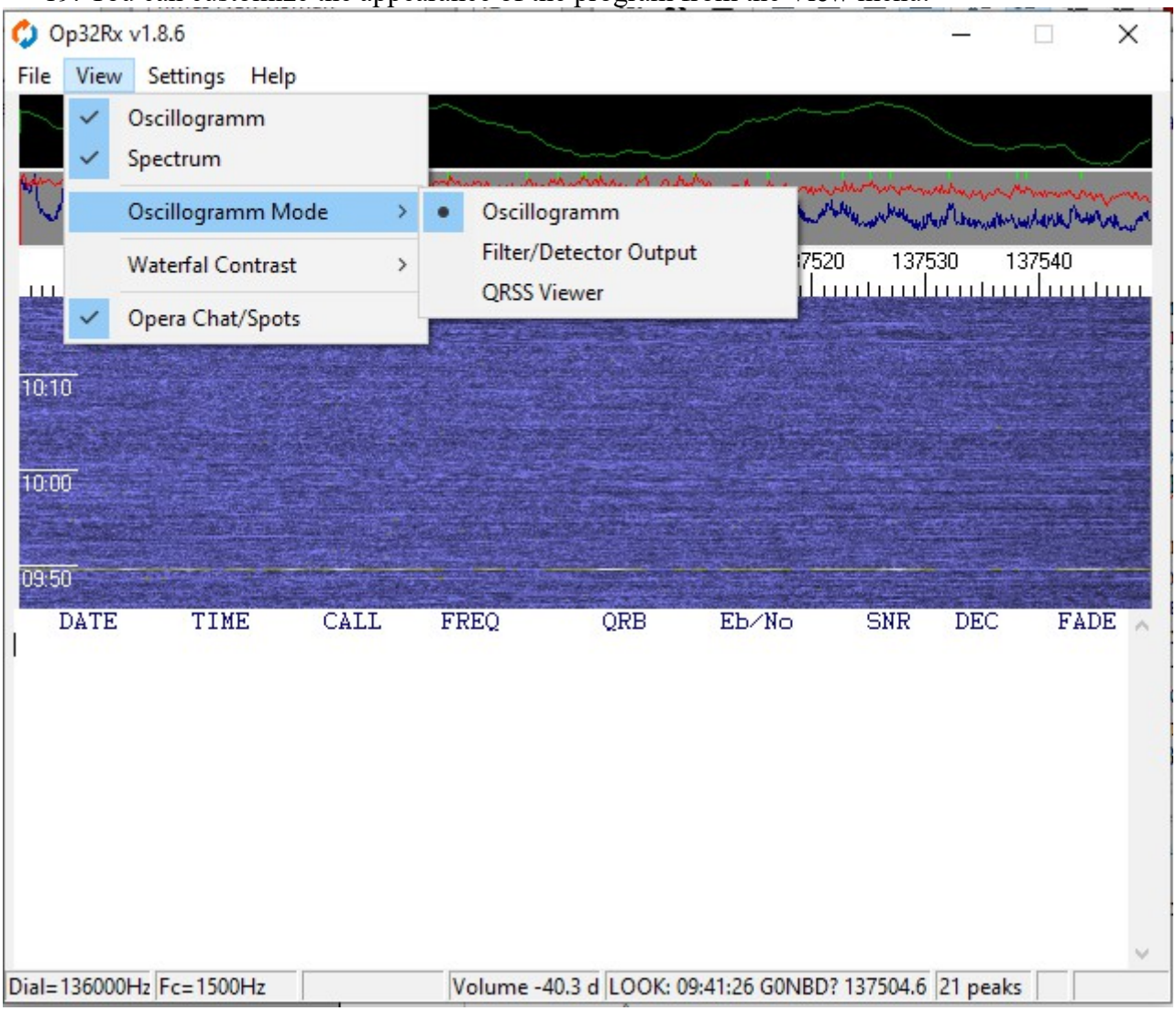
Note: sometimes false decoding is possible against the background of a powerful signal of another correspondent, a strange false call sign can be decoded several times on the same frequency. It will be written to the **known_list.txt** file, resulting in repeated false reports. Therefore, a good rule would be to periodically check the contents of the known_list.txt file and remove false call signs from it, and to transfer the found correct to callsloc.txt, complementing them with QTH locators.

f) To clear (delete) a known_list.txt file, use the **Cleanup know_list** menu item. Use this command periodically (once a week or per month).
g) I**terative Decoder** - Allows the decoder to perform iterative decoding. The least reliable bits of the received signal will be sequentially inverted in various combinations until the CRC checksum of the message matches or the allowed number of iterations is exhausted. **This option is recommended**.
h) **Use binary PIC correlator (+ Use soft correlator OPDS)** – These options allow correlation search for call signs listed in the callsloc.txt file.
i) **Frequency drift tracking** – Enables the mode of slow frequency drift tracking (in case of insufficient stability of the reference generator of the transmitter or our receiver). However, there is a slight increase in processor utilization and possibly a slight decrease in noise immunity. Use this option if necessary.
j) **Sample rate offset compensation** – The sound card sampling rate may be different from the nominal value and instead of 12,000 samples per second, such as 12010. This results in a gradually increasing time shift, which degrades the interference immunity of the reception. Use this option when there are explicit problems with Sample rate inaccuracy. CPU utilization increases slightly.
k) **Peak number** - several spectral maxima are chosen to reduce computational costs for further processing, since even the weakest signal gives a sufficiently visible spectral peak, well visible on the spectrum with high averaging. It makes no sense to process the whole spectrum. It is possible to limit the number of such spectral peaks. A minimum of 10 peaks are needed. For any situation it will be enough to process 20 or 30 peaks. The more peaks are processed, the greater the CPU utilization.
l) **Peak spacing** - rarely when signals are transmitted very close to each other in frequency. Therefore, you can specify the minimum distance between spectral peaks. It is usually sufficient to set it to 0.2 - 0.4 Hz.
m) **Peak threshold** - for the spectral maximum to be taken for processing, it must exceed some small threshold relative to the averaged noise level. By default, this value is 0.4 dB.
n) **Upload spots** – Allow decoder spots to be sent to the site https://pskreporter.info/pskmap.html.
o) **Upload correlator/opds spots** – Allow transmission of correlation detector spots (to pskreporter site and database)
p) **Upload spots to Opera Database** – Allow spots to be written to the

http://spots.microwavers.es/index.php database and sent to Opera Chat on the EA4RCH-5 cluster.

It is recommended to check all these check boxes so that the results of your reception are available to all LF-radio fans.

19. You can customize the appearance of the program from the **View** menu.



The main window of the program consists of five zones:

- **Oscillogram** - displays the signal received from the air. It is automatically normalized by amplitude. If pulse interference is present, it is suppressed by Noise Blanker. If there are too many pulses, the oscillogram turns red. If you select any frequency in the spectrogram with the cursor (click with the left mouse button), instead of the oscillogram, you can observe the appearance of the signal after the filter and detector at that frequency, or you can see the scanning of the spectrogram in QRSS-viewer mode. The oscillogram can be hidden.
- **Spectrum** – Blue line shows instant spectrum, red - spectrum with large averaging. Spectral peaks are well visible on the red spectrum. Above these maximums are green strokes - this means that this frequency is processed by the decoder. If a black strip is present above the maximum, this frequency is temporarily blocked and is not processed. The spectrum can be hidden.
- **Spectrogram (waterfall)**. Here you can observe signal and tone interference traces. It is impossible to hide it. Time of waterfall scanning for the whole height roughly

corresponds to duration of one Opera-32 transmission. If the signal is decoded, a call sign at the appropriate frequency appears on the waterfall. If the signal trace is visible, the decoding is highly likely to be correct. You can set the contrast of the waterfall in View- > Waterfall contrast. Three gradations of contrast are available. The most pleasant looks waterfall with Low contrast.

- Text zone of **spots**. Each row consists of the following fields:
  - DATE is date
  - TIME is time
  - CALL - call sign
  - FREQ is frequency
  - QRB - distance to correspondent (QTH locator is taken from callsloc.txt file)
  - Eb/No is an estimate of the signal-to-noise ratio to the detector output. This estimate is not strictly Eb/No and is called so conditional. This value is compared to the decoder and correlation detector thresholds.
  - SNR is the signal-to-noise ratio at the receiver input in the 1 kHz band. This value is passed in spots to the database, Opera cluster, and pskreporter.
  - DEC is the number of iterations of the decoder. If the correlation detector is activated, then the symbol 'c 'for the bit correlator and 'd' for opds. The most reliable result of reception is pure decoding, without the involvement of correlators.
  - FADE is the fading percentage of the signal.
- **Status bar.** There are several fields that display information about the signal strength at the input of the audio card (preferably within -20... -40 dB), decoder/correlator messages, number of spectral peaks processed, availability of connection to the pskreporter server to send spots.

It should be noted that depending on the computer model and operating system version, disabling the display of the oscillogram and spectrum can in some cases reduce CPU utilization by almost half.

20. The **File** menu contains a useful **Decode .wav-file** option. If there is a 16 bit, 12ksps, or 48ksps air record, this record can be processed by applying some decoder threshold settings, etc. While the record is being processed, spots are automatically blocked. Upon completion of wav-file processing, the program will resume operation with the sound card.

21. All program settings can be pre-saved File- > **Save settings to INI-file**. When the program window closes normally, all settings are also saved, but if the computer is turned off suddenly, the settings may be lost.

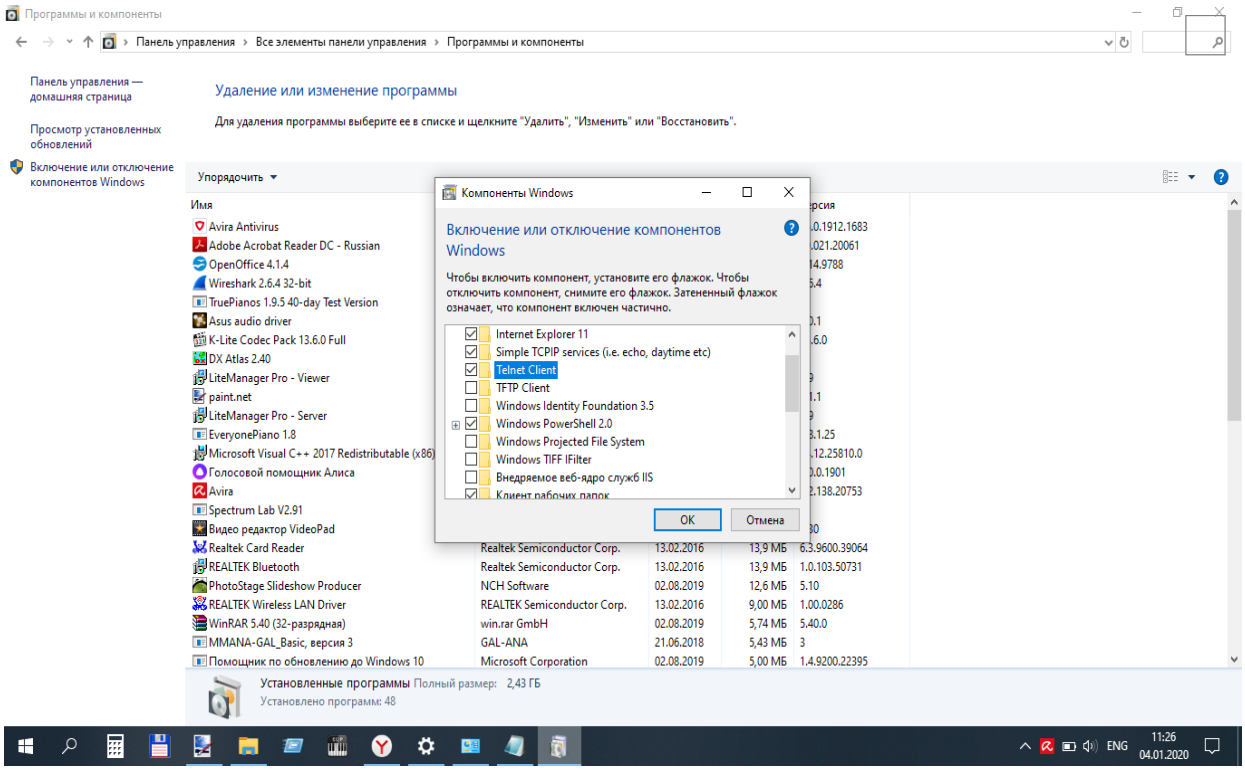
FAQ

Question: My computer is quite old and the CPU load of the Op32Rx program reaches 50%. How can I reduce CPU utilization?
Answer: Set minimum settings - Peak number = 10, Use freq list = OFF, Use binary PIC correlator (Use soft correlator OPDS) = OFF, Frequency drift tracking = Sample rate offset compensation = OFF. In the View menu, disable Oscillogramm and possibly Spectrum. All these will reduce CPU utilization by two to three times. In the most extreme case, disable the Iterative Decoder option.

Q. Is there no connection to the cluster, I don 't see spots, does the Opera Chat window status bar say Disconnect or Socket Error?
Answer:
Verify that cluster communication is enabled: Connect (connect to cluster) and Upload (send spots to cluster and database) are checked.

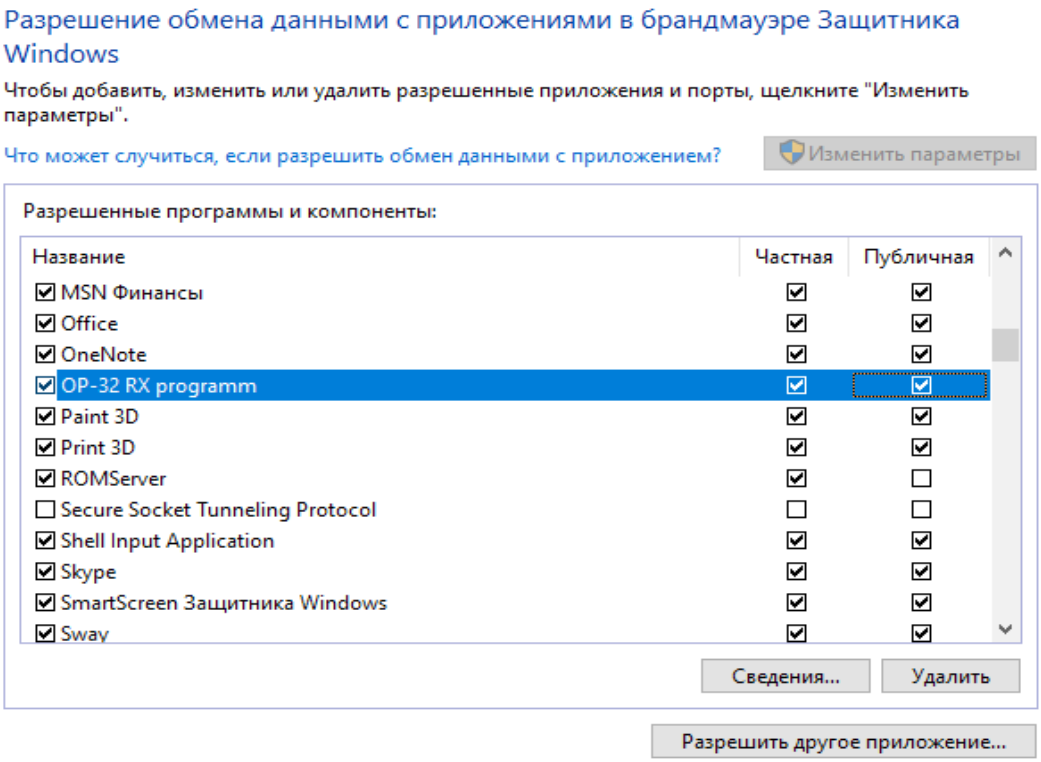
If all is correct, the Opera Chat window status bar will show cluster activity.

- Verify that your call sign is entered in the Settings - > Setup another settings window. Make sure that the call sign does not contain "/." It 's better to replace them with hyphen. Requirements of a cluster are that. You cannot use fractional call signs as a login.
- Please telnet your connection to the cluster. To do this, in win10 telnet, you must enable Control Panel\All Control Panel Items\Programs and Features\Enable or Disable Windows Features. The settings window appears, where you need to find the Telnet client and check the box.



Then, at the cmd prompt, type telnet 144.76.158.174 8000
The black screen should appear "Login:"

- Check your Windows firewall settings: "Control Panel\All Control Panel items\Windows Defender Firewall\Allowed Programs"



Question: experimenting with settings I seem to have ruined everything and I can 't remember what settings were at first. How do I return to my default settings?
Answer: Simply delete the Op32Rx.ini file. When you run the program again, this file will be recreated with the default settings. You will have to re-enter your call sign signal, locator, etc.

Question: There are too many false decodings coming from me. It 's decoding, not deepsearch.
Answer: Check the recommended option Settings- > Use decode filtering with callsloc.txt
Make sure the status bar does not contain the warning "Warning! Callsloc.txt is empty». Download a new file http://rn3aus.136.su/Op32Rx/callsloc.txt .
Please check and clear the known_list.txt file. This contains previously unknown call signs found twice on the same frequency. This may have occurred as a result of receiving against the background of a powerful signal. Just remove the strange call signs.
Maybe it 's time to change the decoder threshold a little bit. Increase it to start by 1 dB.

Question: I turned on the Use binary PIC-correlator option and there were false deepsearch spots, with the call signs real but not working at this time.
A. You need to raise the threshold Correlator239 threshold. Increase it by one. Quite a few false detections occur at a threshold of 164. And don 't forget to check the known_list.txt file again.

Question: Same when enabling Soft correlator (opds).
Answer: You need to increase the opds threshold by, say, 1 dB. Slightly increase the Opds Max1/Max2 threshold by 0.2 dB.

Question: I have already tried to raise thresholds, etc. In general, everything works normally. But when my neighbor starts working on the air with a powerful signal, I still see false spots sometimes. What could be the matter? Why is this even happening?
Answer: The reason is that Opera uses rather weak noise-immune coding with little "code distance." Information is protected by CRC-16. The decoder Op32Rx performs multiple iterations by inverting the least plausible bits whose soft estimates were obtained from the demodulator. This is done until the CRC matches or ends the allowed number of iterations. It turns out a CRC match can happen much more often than it seems! As a result, the combination of bits can be selected from the noise such that the CRC is correct and false decoding occurs with a period of about once every few hours. However, the measured Eb/No in this case usually does not exceed 1-2 dB. When receiving not only white noise, but also powerful signals, the level of false decoding Eb/No can be more than 6-8 dB! Therefore, in some cases it is not sufficient to increase the decoding threshold "Settings" - > Setup - > Decode threshold Therefore, I had to perform additional decoding filtering by comparing them with the list of known call signs (Settings - > Use decode filtering)
A similar situation occurs with the correlation detector. Unfortunately, Opera 's PIC code format has quite a high level of cross-correlation between different call signs. If you take the PIC code of one of the correspondents and calculate the number of his matches with the PIC code of another correspondent shifted by several positions, you found cases where up to 184 positions out of 239 matched, that is almost 77 percent! It 's a very strong crosscorrelation. It is natural that the correlation detector will give false detection with some shift (delay or advance relative to the correct moment in time) when receiving a powerful foreign signal.
The Op32Rx employs many mechanisms to protect against these negative phenomena. This is a frequency blocking after the decode for a time slightly less than the duration of the Opera-32 signal; a delay of 4 minutes in the decision of the correlator to wait for the possibly larger correlation maximum at that frequency. The correlator also looks for the position of the correlation

maximum with shifts relative to the current position. All these measures significantly reduce the number of false spots. But the false spots are still possible. However, Op32Rx produces fewer false spots than Opera1.6.5, which is easy to make sure by studying the database http://spots.microwavers.es/index.php.

Question: Why do the number and content of spots in Op32Rx and Opera1.6.5 differ? What are more correct?
Answer: the mechanisms of exchange of spots are different. Opera 1.6.5 has its own "secret" mechanism and shows spots only from itself. Op32Rx uses the old mechanism - through the cluster, as was the case in all Opera1.5.x before. The most relevant database is now the http://spots.microwavers.es/index.php database, where both Op32Rx 1.8.5(6) and Opera 1.6.5 are placed. The spots from the Op32Rx have a frequency indication. The share is also https://pskreporter.info/pskmap.html

Question: What information from the Internet does the Op32Rx program use for decoding? Can I work without connecting to an internet?
A. The Op32Rx program does not use any information from the Internet. You only need the local callsloc.txt file, which is used by the decoder for two purposes. First, to calculate the distance, since the file has QTH locators and call signs. Second, the decoder uses a list of known call signs to "filter" false decodes. Just as our brains and eyes do with "decoding" weak traces of QRSS. Thus, if the decoder decodes a call sign, it moves to the next processing layer. If an unknown call sign turns out, it will be rejected. However, the decoder has no information as to which call sign it is to detect. Everything is made honestly! Of course, the program will work without a connection to the Internet as well as with a connection to it.

Q. Why is the curl.exe utility in the program folder? Can I use the curl of another version?
A. The curl.exe is used to send spots to the http://spots.microwavers.es/ database. The version is suitable for any, including the rather old (and compact) that was used in older versions of wspr-x.

Question: I would like to save all the spots coming into the Opera Chat window.
A. This is the option and is enabled by default. The option is located in the menu of Opera Chat: File- > Save Opera spots to log-file window. The file name is specified in the Select log-file menu. By default, this is opera_log.txt

Question: I did not find information of interest to me in this manual.
Answer: View the opera_protocol.pdf files, Op32RxHelp .pdf. Ask a question in rsgb_lf_group or on the forum http://136.su/index.php/topic, 277.0.html.

73!