I present you the program reception OP-32.

Author: RN3AUS / Alex

1. General Provisions.

Attention:

- The program does not claim to be a substitute for the "standard" Opera v1.5.x by EA5HVK;

- The program is not intended to exceed the quality of the detection of weak signals OPDS by DF6NM.

The author expresses his gratitude to Jose EA5HVK and G0NBD for the development and implementation of excellent regime Opera.

The author expresses his gratitude to Markus DF6NM for the development and provision of excellent open source program OPDS. Some ideas and software solutions borrowed from opds2h5c.bas

Author does not know how to construct a program Opera v1.5.x by EA5HVK. All possible algorithmic matching unintentional and quite natural.

The author made an attempt to implement a soft maximum likelihood list decoding, providing the program with additional convenient options. The author hopes that the program will be useful in cases where, for whatever reasons, difficult to use Opera v1.5.x by EA5HVK or OPDS by DF6NM.

The program is offered on an "as is" without warranty of any kind. Source code is open and the decoder can be used by everyone.

2. Basic information about the program.

The program is available in two versions:

- Console Op32RxC.exe (useful for grabbers)

- GUI Op32Rx.exe (for observations and experiments)

Signal processing occurs in the two versions of the same.

The graphical (GUI) version additionally supports saving screenshots.

What makes the program.

The program processes the signal samples from the sound card or wav-file, demodulates them and attempts to decode the signal in a format Opera-32. If the decoder is to find a match the checksum, decode written to a text file, and send a report on pskreporter. The program has a built-in automatic impulse noise blanker (NB).

3. The Op32Rx differs from the "standard" Opera v1.5.x by EA5HVK:

- Has an open source code in C ++ (CBuilder);

- Works with any Dial 500-4500 Hz;

- Than sending spots on site stores the decode a text file (as in detected.txt opds);

- Uses a list of frequencies for deep search signal, this list is automatically updated each time a successful decoding;

- You can set list of ignore frequency (for example, lines Loran);

- Correctly decode can be checked on the list of known correspondents (as callsloc.txt in opds);

- Can decode signals from wav-file;

- Version of the decoder is implemented with the list decoding (with a short and a long list, respectively, deep and ultradeep search). This allows you to correctly decode very weak signals, or the dying, is not always possible to make a standard decoder. (However, the author does not know how to implement the decoder in the original program Opera v1.5.x by EA5HVK; perhaps it is done the better.)

- Displays long average spectrum;

- It allows you to automatically save a screenshot of the waterfall spectrum when decoding with signal trace;

- Consumes little memory (5 - 9 MB);

- Do not use any additional information, do not come off the air; use only outgoing internet connections to sending report to pskreporter.

- A certain number of settings that can optimize the use of the system resources (CPU);

- May be able to detect weaker signals.

4. The Op32Rx differs from the OPDS by DF6NM:

- Works independently, it does not need a third-party program signal processing with a sound card (SpectrumLab);

- Can be an offline process prerecorded wav -files;

- It carries out decoding of the present rather than the correlation choose the most similar copies known signal;

- Can decode any call;

- Sends the spots on the Internet;

- You can assign a list of frequencies for deep signal search, the list is automatically updated so forth, and each successful decoding;

- In the case of a two-fold decode callsign missing sheet callsloc.txt, automatically stores it in a file known_list.txt;

- Monitors the slow drift and wander frequency transmitter / receiver without compromising the quality of decoding. There were times when the frequency of the receiver is experiencing temperature drift of 0.2 Hz during transmission. The signal was decoded without losing quality.

- Track difference of clock frequencies of the transmitting and receiving sides.

5. How it works (technical details).

The signal is taken from the sound card. If several such devices, it is possible to choose the right.

The sampling frequency of 12 kHz, 16 bits.

In the settings you can select the center frequency of the low frequency signal corresponding to the center of the strip 137500 Hz. For example, when Dial = 136 kHz center frequency spectrum of OP-32 segment will be equal to Fc = 1500 Hz. You can set the Fc from 500 Hz to 4500 Hz. This is done in the ini-file. (Next we will look at all the settings in this file in detail.)

Determined by the average signal level. If there are pulses in excess of three times the value of the average level of the pulse is suppressed (NoiceBlanker).

Further purified by impulse noise signal is filtered by a digital FIR filter with a bandwidth of 200 Hz at -3 dB.

For the three ratings of "standard" Dial (135,000 Hz, 135,500 Hz and 136,000 Hz, with the possibility of fine-tuning within the plus or minus 10 Hz) applied precalculated 6-order 255-tap Butterworth filters having a good uniformity in the passband and a large suppression beyond.

If Dial custom, the FIR filter is calculated by means of the program (weighted by the window Blackman-Harris). The number of "taps" is determined automatically and does not exceed 1024 (in reality it turns out about 270).

The filtered signal is downconverted to a central frequency of 150 Hz (the entire spectrum is in the range from 100 to 200 Hz).

To suppress side products is another FIR filter with 64 taps.

Further decimation is carried out the first 24 times, and now counts followed with a frequency of 500 times per second.

To narrowband matched filtering is used in a program line of digital recursive IIR filters of 2nd order (Digital oscillation circuit) with a bandwidth of 0.122 Hz at -3 dB (equivalent to 1 / 8.192 with - the inverse of the duration of a parcel). The number of filters in the line 1230, that several redundant: in principle, it is sufficient to have a 100 / 0.122 = 820 filters to cover the entire range of 100 Hz. However, experiments have shown that the reception sensitivity is increased if filters "stand" somewhat denser; It was chosen as a result of one and a half the number of filters.

It may seem strange to use "digital oscillatory circuits" where usually use Fast Fourier Transform. Perhaps computational efficiency would be somewhat higher, but since the filters act on samples decimated, the win was not great. On the other hand, the visibility is obtained by using filters above, one can experiment with detection filters and the amount of what we would be denied when FFT. In addition, FFT has «crammed on edge” (bothered); much more interesting to a non-standard way.

After the narrowband filters are detectors. This can be either a simple quadratic amplitude detector (when the input sample filter multiplies at himself) or coherent (quadrature) detector. Experiments have shown that both detectors about are equivalent in

terms of sensitivity, however, coherent detector consumes more computing resources. By default, the coherent detector is turned off. Its advantage is beginning to be noticeable when extremely weak signals abroad detecting ability of the decoder; In this situation, the use of coherent detection increases the probability of finding a signal. The signal is then filtered by low-pass filter (IIR filter equivalent to a complex RC-circuit) with a cutoff frequency 0.123 Hz.

Further there is a decimation signal output from the detectors even 1024 times (1 sample about 2 seconds, 4 samples to 8.192 seconds).

After filtering and detection signals are sent to the buffer memory having 1230 channels and a length 240 (the number of parcels Opera plus one) * 4 (number samples per parcel) = 960 cells. It is this buffer and takes main place in the memory 1230 * 960 * 4 (size of the float in bytes) = 5 MB.

Further, only processed samples of this buffer storing a data for the last 32 minutes.

For each channel (the term refers to the channel output of the matched filter) is calculated current amount, which corresponds to energy of the signal transmission Op-32 (32 minutes). These values are then used to display the amplitude spectrum and the waterfall.

To reduce the amount of computation does not undergo decode all channels, but only those which have a maximum accumulated signal. The maxima are searched as follows. Each value is compared with several neighboring values (their number is specified by MIN_PEAK_SPACE). If neighboring values are less than the current - that is the maximum. Such peaks in the spectrum are many. To reduce their number, use the following algorithm. Given a desired number of peaks for processing (parameter N_MAX). If more highs than required, then the "decimation": the whole spectrum is divided into several sections, each of which removes a peak having the smallest amplitude. This is repeated until there is the desired number of maxima. A breakdown of the spectrum is performed on portions of the following considerations. If the input filter has a large non-uniformity, or there is a strong signal, the peaks of the lying close to the "hump" of the filter will always have a greater amplitude than the rest (that is itself "noise Regiment" will have a "hump"). If simply choose a certain number of peaks of the highest amplitude, they will be grouped around a powerful signal either near the "hump" filter. This is not good, because the most interesting faint signals from distant correspondents remain unaddressed. Application of the algorithm described above "thinning" highs allows for selection when they are about equal.

Usually it is sufficient to set the parameter N_MAX = 30 ... 40. If the computing power is not very critical, it is easier to specify the N_MAX = -1, and the program will process all results (even small) highs.

Because of the number of channels allocated for further processing are excluded those whose frequencies coincide with a list of

frequencies from the file loran_list.txt (if the parameter IGNORE_LORAN_LINES = 1).

If you specify USE_FREQUENCY_LIST = 1, then taken to the processing channels corresponding to the frequencies of the file freq_list.txt, regardless of whether there is a spectral maxima at them or not. Furthermore, these channels include the list decoding mode with a long list (Ultra Deep Decoder).

The decoder implements the algorithm of maximum likelihood, performing search of all variants of codewords used by Opera. 17 such combinations, each combination has 7 options. (For more details on the structure and method of forming the signal Opera can be found in the file opera_protocol.pdf). Correlations sums are calculated amount of them selects the largest, and the second thereafter. Forming received information vector with the highest correlation sum and its list of options, replacing part of bits less likely (from the second correlation codewords). The first list consists of 19 options (1 + 17 the most likely options for the replacement of one codeword less likely +1 option with replacement of code combinations). The second list is made up of substitutes has two code combinations at the same time and consists of 136 variants. We could build and even longer lists, but it begins to increase sharply the number of false decoding.

Note that the decoder allows for the possibility of a divergence of the clock frequency of the transmitting and receiving sides, so that the total duration of the signal may be somewhat longer or shorter than expected. Search the duration within plus or minus one parcel (with 8.192), which corresponds to the divergence of the clock frequency 1/239 = 0.42% = 4184 ppm (this is quite a lot).

In addition, the decoder searches for the maximum likelihood in two adjacent frequency channels (adjacent filters) than partially offset drift of the frequency of the receiver. Decoding quality does not deteriorate when the frequency drift (one way) will be for the duration of a transmission (32 min), no more 0.2 Hz (and more acceptable, if a smooth drift); also allowed wandering around the nominal frequency in both directions by the same amount. If the drift will be of great value, the resulting signal to noise ratio will start to decline. In general, at LF this requirement for stability is quite easy to implement.

Decoder iterates until the list is ended, or checksums decoded correctly (call sign transmitted in the Opera, is protected by CRC-16 (16 parity bits) and the second CRC-16 checksum, which is transmitted from 3 bits). If the two checksums coincide, the decoder returns the decoded call. If only one coincided (internal 16-bit) checksums, the decoder marks the call as an invalid (question mark). This decode is taken into account, provided that the ratio of signal to noise ratio at the output of the filter exceeds a certain threshold (OP_THRESHOLD = 4 - the threshold is 4 dB).

What is this signal to noise ratio. For the most matching option

signal has the correlation sum - it is the energy of our signal. At the same time we have the energy value signal in the frequency channel - it is the energy of the signal + twice the energy of the noise (noise is and when parcels and in the pauses between them). About find the signal to noise ratio Eb / No = 10 * log (2.0 * Psn / (Psn - Ps)), where Psn - power mixture of signal and noise, Ps - signal strength (correlation sum decoder). The numerical values of Eb / No Exper are in range from +4.0 dB and above; for example, a signal satisfactorily visible on the spectrum and giving decode a "standard» Opera -36 dB (Opds - 37.1 dbOp) is Eb/No= + 4.8 dB. Weak signals that can be decoded already on the verge of the most profound features of the decoder and which is not always decode smiling Opera, I have tons of Eb/No lo ok of 4 dB. Sometimes, when properly decode and Eb/No below +4 dB, but it is a lot of false decode (but nevertheless with the correct checksum!). Therefore, the threshold for "suspicious" Decode wisely set about 4 dB).

Separately need to mention that, perhaps, it is not legitimate to use the term «Eb / No» - valued bits th signal to noise ratio. However, it seems, within the meaning of the measured value is close to us, the term «Eb / No»; moreover, the term is short and easy enough to be displayed on the screen, so I decided to use that designation in the future.

Thus, the output of the decoder, we have decoded the call, with the correct checksum. However, this does not mean that he is really present in the air. Practice shows that the false detection of noise random call signs with the same CRC occurs quite frequently, about once every half hour / hour. It might be protection against false decode typing compared to some threshold or register decode if Eb / No below this threshold . However, not uncommon for the Eb / No proper decode weak signals to be lower Eb / No! We are interested in receiving the weakest signals, do not want to lose them by setting the threshold so as to reliably decode pass false. There is a classic dilemma: either to pass a signal or a false alarm. In addition, in practice, often get the following situation: if the air there is someone strong signal, then it at ultra deep search easily "recruited" false decode with a very good level of Eb / No, so that the comparison with the threshold here, unfortunately, does not help.
The program features a method for protecting against false alarms. We must accept that on LF operates very few correspondents, their call signs are known, the appearance of each new correspondent does not pass unnoticed. For the program OPDS by DF6NM used file callsloc.txt, which shows all the call signs and locators really working or once worked correspondents. Therefore, to "screen out" false decode (which is always a very distinctive call sign!) Is used to search the callsign decoded file callsloc.txt. If he is there, then decode will most likely correct. At the same time and determine the distance to the correspondent.
But suddenly in the air there is a new correspondent, which is not yet in callsloc.txt? It would be a pity to drop the correct decode only because of the lack of a call sign in the file. In these cases a mechanism that twice adopted an arbitrary call is

considered to be from now on "known" and it is placed in a file known_list.txt. Only the locator is not known to us. We can see decode from this correspondent, put his call from now on callsloc.txt, adding qth -lokator or, if this is an error, delete it from the file known_list.txt.
To increase the probability of correct decoding of weak signals is, as already mentioned, the list decoding mode with a long list. However, it loads the CPU. To reach a compromise intended mode of use of the file the preferred frequencies. This list can be made by us in advance, as many of us transmit on the same frequency. Also, if the frequency at which there was a valid decode, is not listed, it will be there automatically entered.
Row decode displayed on the screen and recorded in the file contains the date, time, callsign, frequency, QRB (if known qth), bit signal-to-noise ratio at the output of the decoder Eb / No, the signal-to-noise ratio SNR (calculated with respect to this spectral peak to the average amplitude of the spectrum of the entire band 100 Hz; the result is somehow correlated with the results of the Opera, to the obtained value added amendment set parameters OPERA_ DB_OFFSET = -10 - weak signal is about -10 dB). Displays the number of iterations of the decoder (1 - once decoded correctly, less than 20 - a short list, more than 19 - with a long list). Then, if matched only the first CRC , it displays a '?'. If the decoded call is not in the list of known callsigns '-' sign.
If allowed to send spots to the site pskreporter, in the event of success in the last field '+' sign and the failure of 'x' (in the console version; in the GUI version of the result is displayed in the right field Status Bar).
Conclusion decode on the screen and write to the file are made in the "reverse" order, the top will be the most recent decode. Please keep in mind that the insertion of a new line in the file is always carried out in the third line from the beginning of the file.

6. The following is the structure of ini -file with comments:

[OP-32]

my call

MYCALL = RN3AUS

My locator

MYQTH = KO85FN

header file

MYHEADER = RN3AUS KO85FN Opera-32 decoded (Op32RxC RN3AUS software)

The center frequency of the audio spectrum

F_CENTER = 1500

The frequency setting of the receiver

F_DIAL = 136000

```
The minimum distance between the spectral peaks
(in number of samples of the spectrum, one step 0. 08 Hz)
MIN_PEAK_SPACE = 3
The number of spectral peaks (-1 - auto)
N_MAX = -1
file for writing spots
FILE = my_detected.txt
ignore the frequency from the list loran_lines. txt (1-Yes 0-no)
IGNORE_LORAN_LINES = 0
always analyze the frequency from the list freq_list.txt (1-Yes 0-
no)
USE_FREQUENCY_LIST = 1
sift unknown callsign missing in callsloc.txt and known_list.txt
USE_ KNOWN_CALLSLIST = 1
use the decoder with a deep search (short list)
DEEP_DECODER = 1
use the decoder with a long list
ULTRA_DEEP_DECODER = 0
use a coherent detector (0-square amplitude)
USE_COHERENT_DETECTOR = 0
Amendment estimates SNR (dB) for consistency with Opera
OPERA_DB_OFFSET = -10
threshold (dB) to decode with mismatched external 3-bit CRC
OP_THRESHOLD = 4
allow to save screenshots spots (only for the GUI version)
CAPTURE_ENABLE = 1
a way to save screenshots (for GUI version)
CAPTURE_PATH = Capture\
[SOUND]
used sound card (-1 by default)
DEVICE = -1
[PSKREPORTER]
HOSTNAME = report.pskreporter.info
PORT = 4739
allow the server to send spots
UPLOAD_SPOT = 1
```

7. Installation and Getting Started.

The program requires no installation, simply place the following files to any location on the disk:

- Op32Rx.exe (Op32RxC.exe) - executable file

- Op32Rx.ini (Op32RxC.ini) - configuration ini -file. If he absence of the user, it will be created automatically with default values.

- Callsloc.txt - file with callsigns and locators is identical in structure and meaning of the same file for OPDS by DF6NM.

- Freq_list.txt - file with the frequencies to be constantly reviewed, regardless of the spectral peaks. If the file does not exist, it is created automatically. The file is filled automatically as they become decode. The frequency in Hz specified in floating point format, for example 137531.503

- Loran_lines.txt - file with the list of ignored frequency. If the frequency is specified at the same time freq_list.txt and loran_lines.txt, then the signal at that frequency will be decoded. The file is filled manually. Frequency in Hz specified in floating point format, for example 137531.503

- Known_list.txt - file with a list of at least two times the decoded callsigns missing in file callsloc.txt.

- PSKReporter.dll - dynamic library, needed to send the spots on the Internet. Without this library, the program will not start.

When you first start the program will offer (if ini -file absent) to specify your callsign and locator in the ini-files. Be sure to include them in the title MYHEADER instead of automatically inserted values of <null>.

In the process, sometimes open a file known_list.txt to remove from it the false call signs, if it is, or to transfer the call sign of a new stations to file callsloc.txt (on that I wanted to put on - alas, new correspondents in the range of 136 kHz It appears only once a year, :( ).

8. Recommended parameters.

a) If your computer is fast enough, it is best to set the following parameters:

MIN_PEAK_SPACE = 3

N_MAX = -1

USE_FREQUENCY_LIST = 1

USE_KNOWN_CALLSLIST = 1

DEEP_DECODER = 1

ULTRA_DEEP_DECODER = 1

USE _COHERENT_DETECTOR = 1

This will provide the most profound search for the signals.

b) If the computer is not very fast, saving CPU resources, you can set the parameters:

MIN_PEAK_SPACE = 3

N_MAX = 30

USE_FREQUENCY_LIST = 1

USE_KNOWN_CALLSLIST = 1

DEEP_DECODER = 1

ULTRA_DEEP_DECODER = 0

USE_COHERENT_DETECTOR = 0

These settings provide adequate close to the maximum sensitivity. At frequencies of freq_list.txt will be ultra-deep search. With these settings performance requirements CPU are the same as at the Opera v1.5.x by EA5HVK, memory and consumes many times less.

c) Finally, if you have an old computer and the performance is not enough, try

MIN_PEAK_SPACE = 6

N_MAX = 20 (or even 10)

IGNORE_LORAN_LINES = 1

USE_FREQUENCY_LIST = 0

USE_KNOWN_CALLSLIST = 1

DEEP_DECODER = 0

ULTRA_DEEP_DECODER = 0

USE_COHERENT_DETECTOR = 0

In this case the CPU and RAM are loaded minimally. Chance pass weak signal, if there are several strong signals and / or Loran lines. It is very convenient to ignore Loran peaks, making it to the list; Use a list of frequencies freq_list.txt (parameter USE _ FREQUENCY _ LIST = 1), if the work is expected to correspondents in their usual frequencies.
Note: about ain impact on the CPU have parameters N _ MAX, ULTRA _ DEEP _ DECODER and USE _ COHERENT _ DETECTOR.
Warning: Using the USE_KNOWN_CALLSLIST = 1 should be considered mandatory to ensure that no false decode (it is the primary means of protection against them!). Disable filtering decode the list of call can be known for experimentation, while not forget to disable (UPLOAD_SPOT = 0) and uploading spots to the Internet!

9. Features GUI-Version:

- Entering in the expected frequency or ignored lists can be done by moving the mouse pointer over the waterfall or spectrum and clicking the right mouse button.

- Click the left mouse button on the waterfall or spectrum (if you see something similar to the signal OP -32), includes a display of the output signal of the matched filter and enters the mandatory search / decoding at this frequency. To cancel, double click on the oscilloscop (top picture).

- On the spectrum and waterfall displays linear (not logarithmic)

scale amplitudes. Is automatically scaled to the maximum. This is done to simplify the interface while maintaining sufficient visibility. The author does not set himself the task to repeat, or at least come close to the quality of the excellent spectrum analyzer SpectrumLab by DL4YHF.

- This version 1.0.0 introduced yet experimental mode correlation detection signal (similar to the OPDS). It can produce many false decode, as a matter of choosing the correct threshold OP_THRESHOLD = 4 so far remains unclear. Until then, n eye on the launch of the program took less than 32 minutes, please do not check «Correlation OPDS»! As long as the internal buffer is not full, they can appear false decode a large Eb / No. After the expiration of the time, you can enable this feature. In future versions of the program I plan something to improve, or disable this mode, if he did not gain the efficiency of the search signal.

10. Features of the console version.

At the command prompt, you can specify a file for analysis:

Op32RxC.exe myfile.wav

The file must be written in the format of 12 ksps, 16 bps. The program opens and starts to analyze the file, which will indicate the inscription "processing file ...". After the analysis, the program automatically switches to the signal processing to the sound card, which will witness the disappearance of the above labels. Analysis time record of 35 minutes (48 MB) is about 5 minutes and is highly dependent on the settings of the decoder and the computer's performance.

The console version is intended for use on the grabber, where there is no need to use a graphical user interface for the current settings. In addition, understand the source code of the console version is somewhat easier as may port it to other platforms.

The identified weaknesses as well as your wishes, it is desirable to inform the author: rn3aus@mail.ru

Sincerely, RN3AUS / Alex

73!