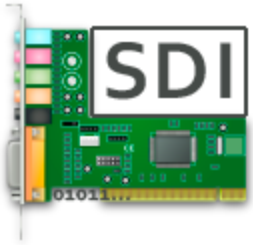


SoundCard Data Interface

by Yurii Faraday

Soundcard Data Interface is source-closed project based on idea to transfer data using sound card

Project's official application technical information:

Application programming platform:
- C# .NET 4.0;

Application CPU architecture:
- x86-64 (AMD64);

Custom libraries, used in project:
- NAudio;
- Magick.NET-Q8-AnyCPU;

Recommended system requirements:
- CPU: 4-core+, 2 GHz+;
- RAM: 2 Gb+, DDR3+;

Minimal system requirements:
- CPU: 2-core+, 1.6 GHz+;
- RAM: 2Gb+, DDR4+;

OS requirements:
- Windows Vista+;
- Windows 10 (recommended);

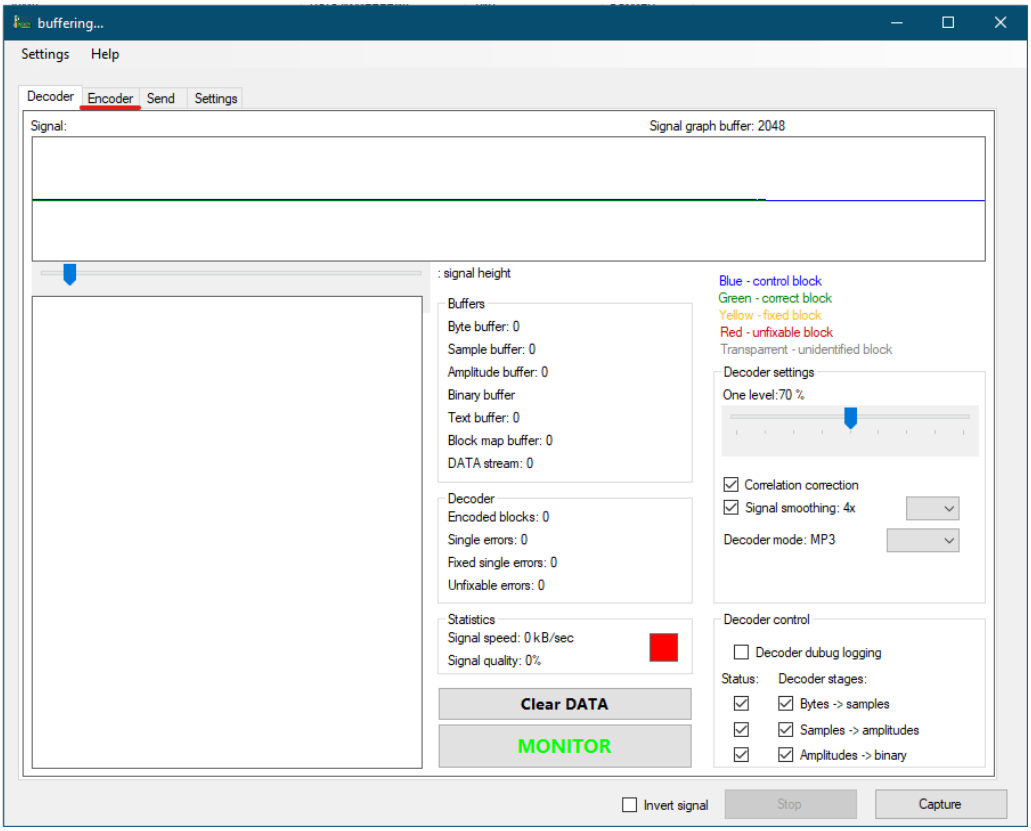
Software requirements:
- .NET Framework 4.0+;

Application languages:

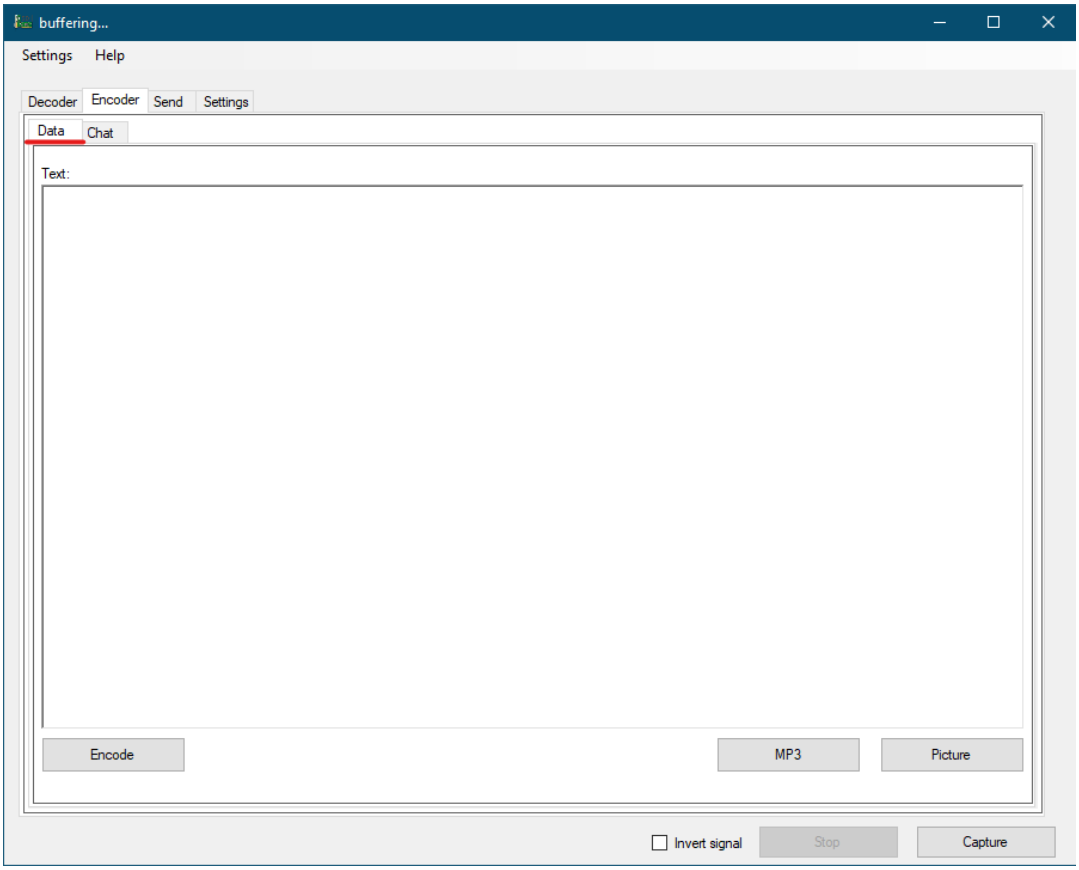
Language	**Author**
Russian	Yurii Faraday
English	Yurii Faraday

Text Encoding

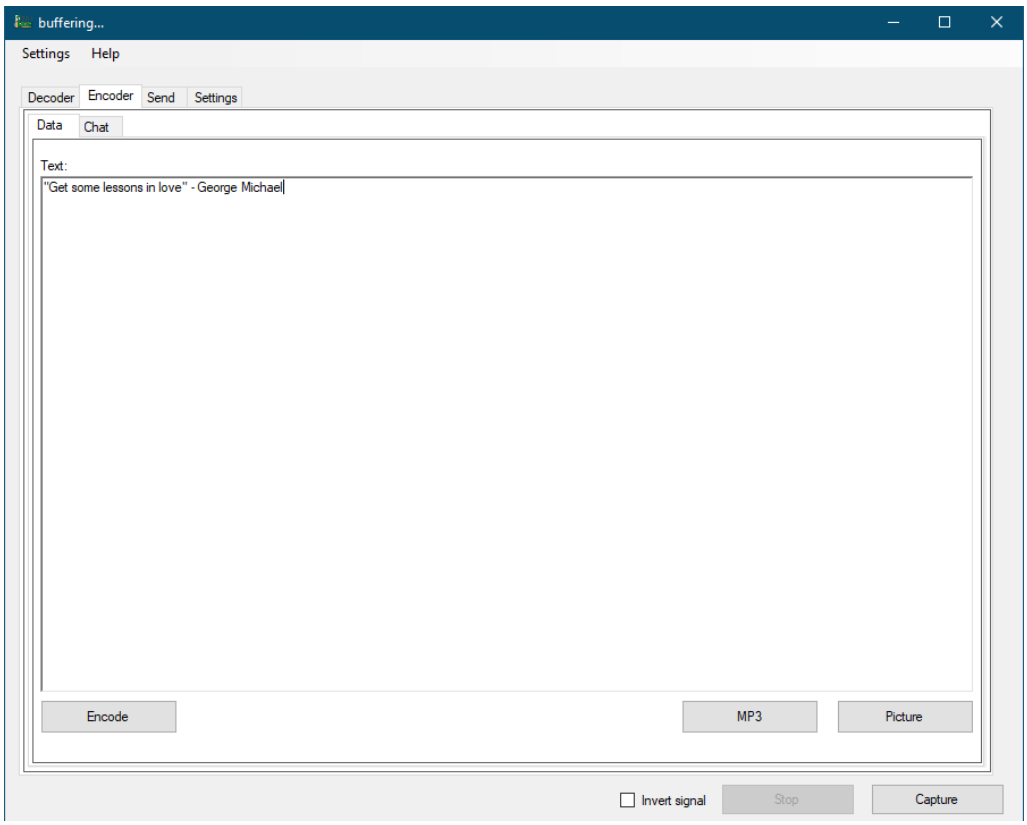
- Open the app.
- Go to “Encoder” menu



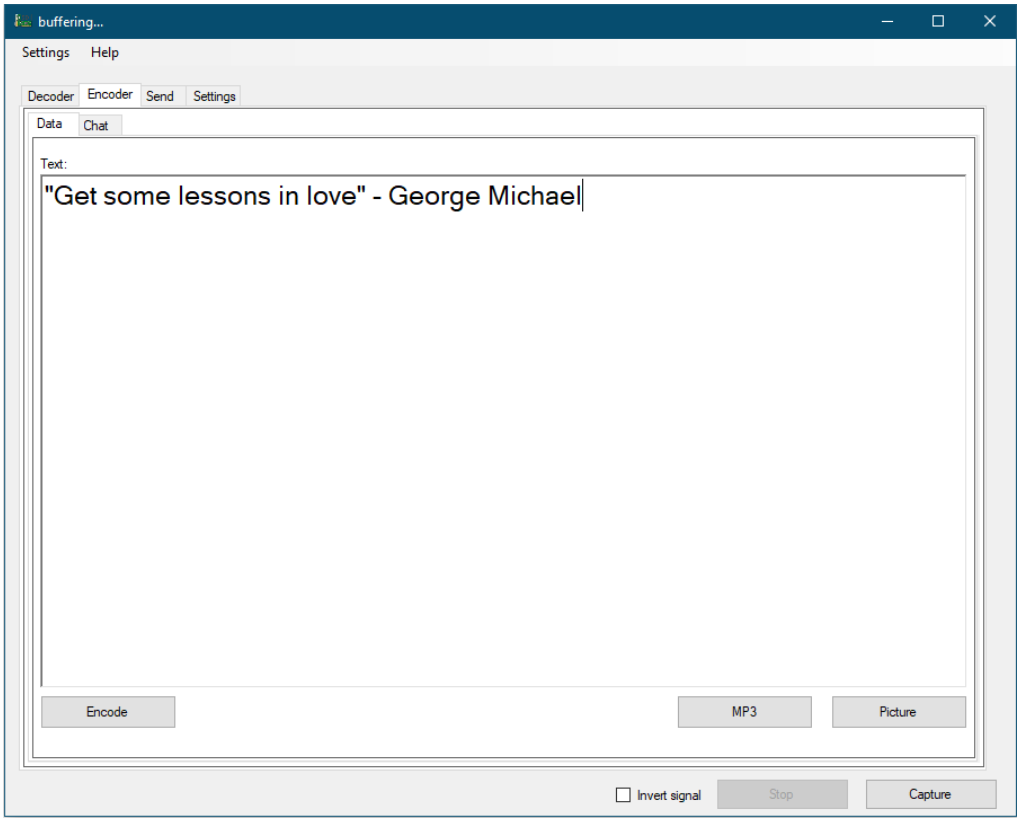
- Make sure, that you’re in “Data” submenu



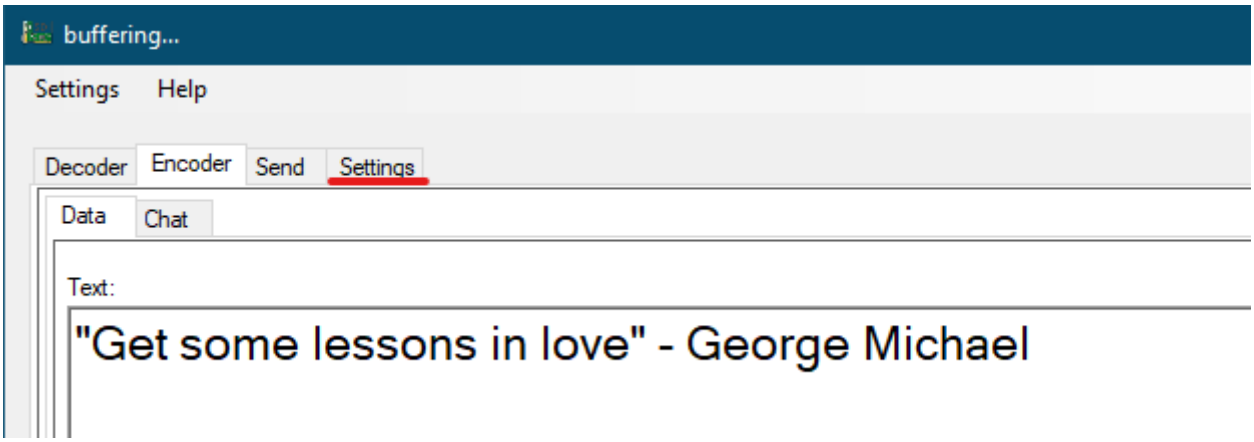
- Put some text into textbox



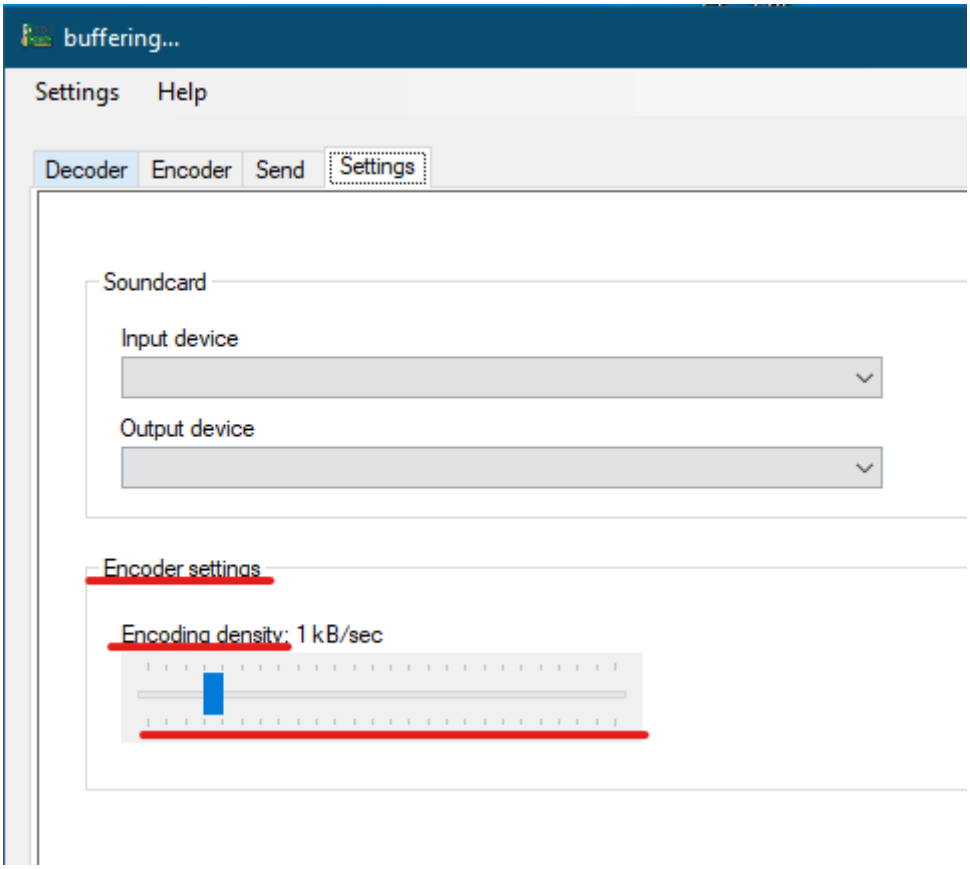
- Put cursor into textbox and use mouse wheel to change text size for more comfortable looking, if you need



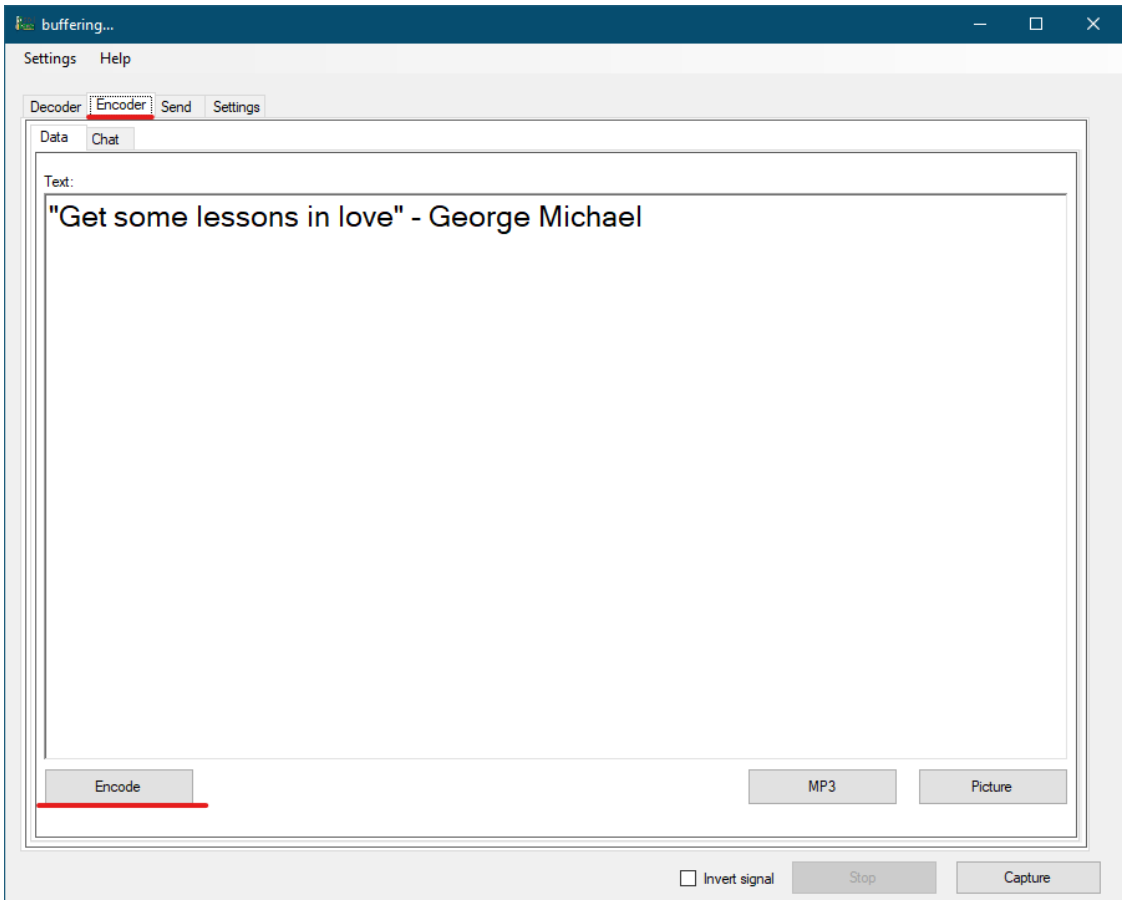
- Go to “Settings” menu



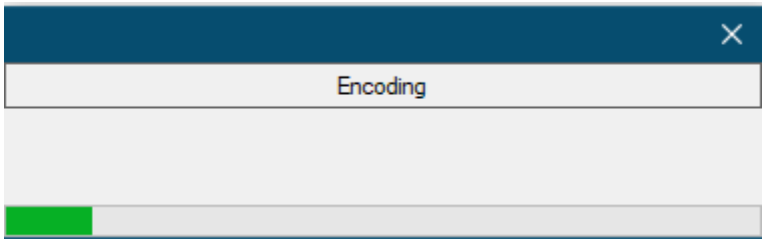
- Set “Encoding density” what (if) you want



- Then go back to “Encoder” menu and push “Encode” button

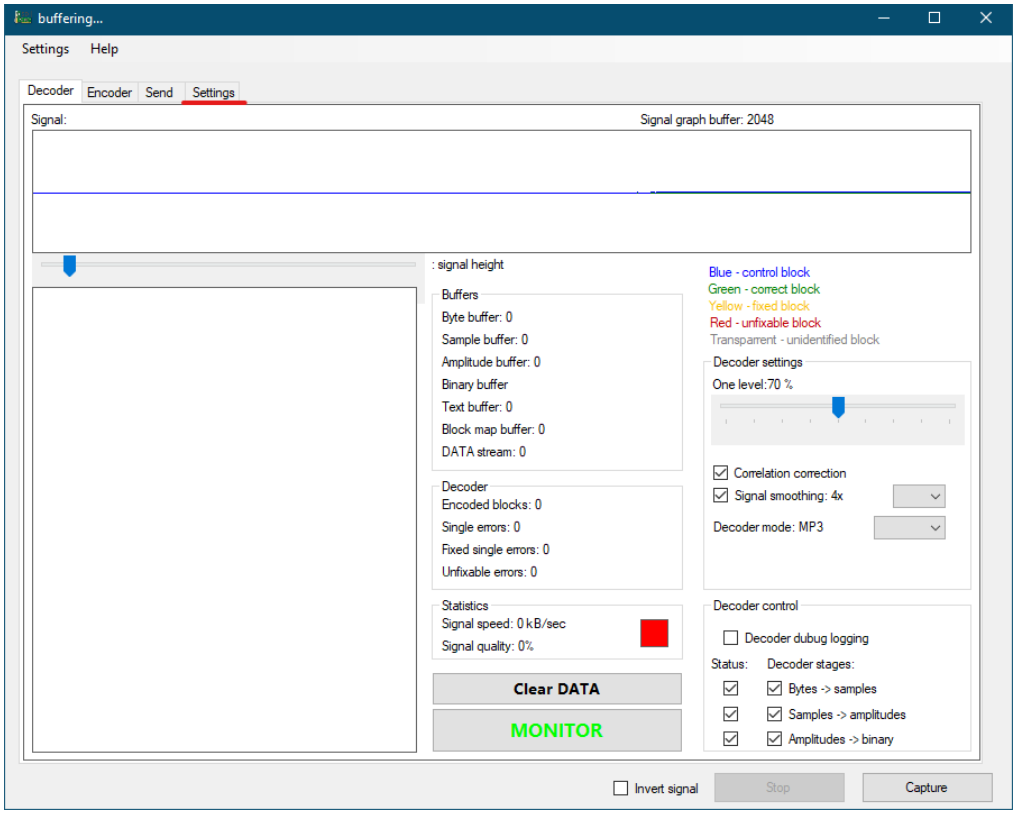


  - Encoding process will be shown like that:

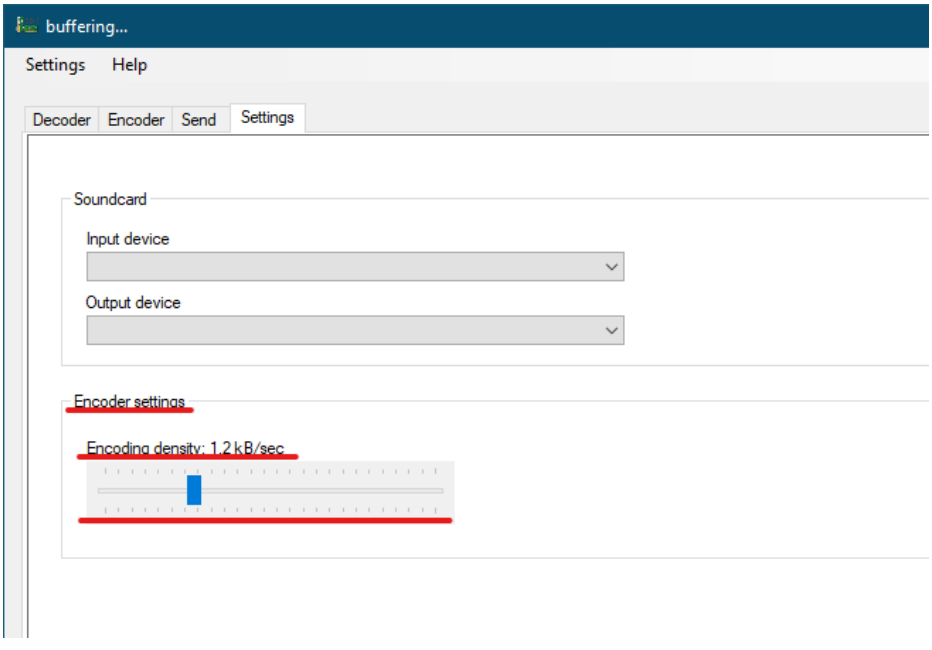


MP3 Encoding

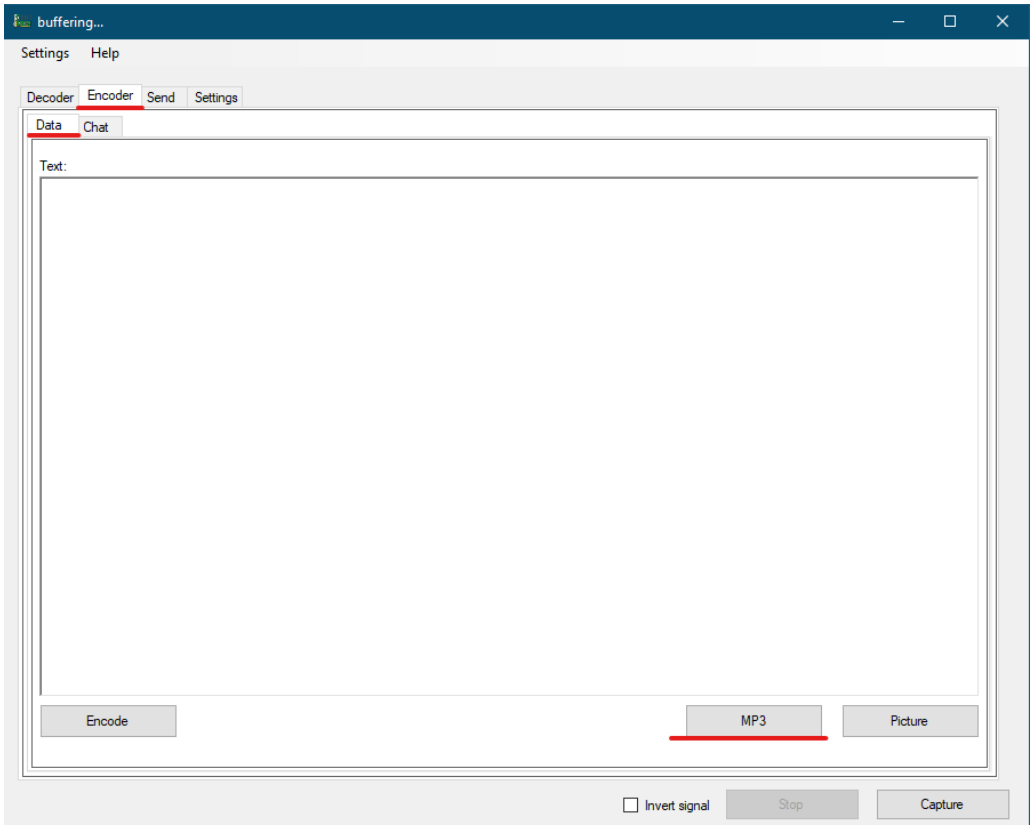
- Open the app
- Go to “Settings” menu



- Now you have to set “Encoding density” to 1.2 kB/sec or higher



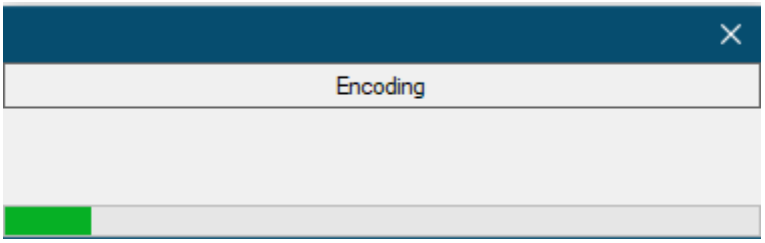
- Go to “Encoder” menu, make sure, that you’re in “Data” submenu, then push “MP3” button



- Now select MP3 file:
    - Be attention! Use only supported mp3 types for encoding from grid below

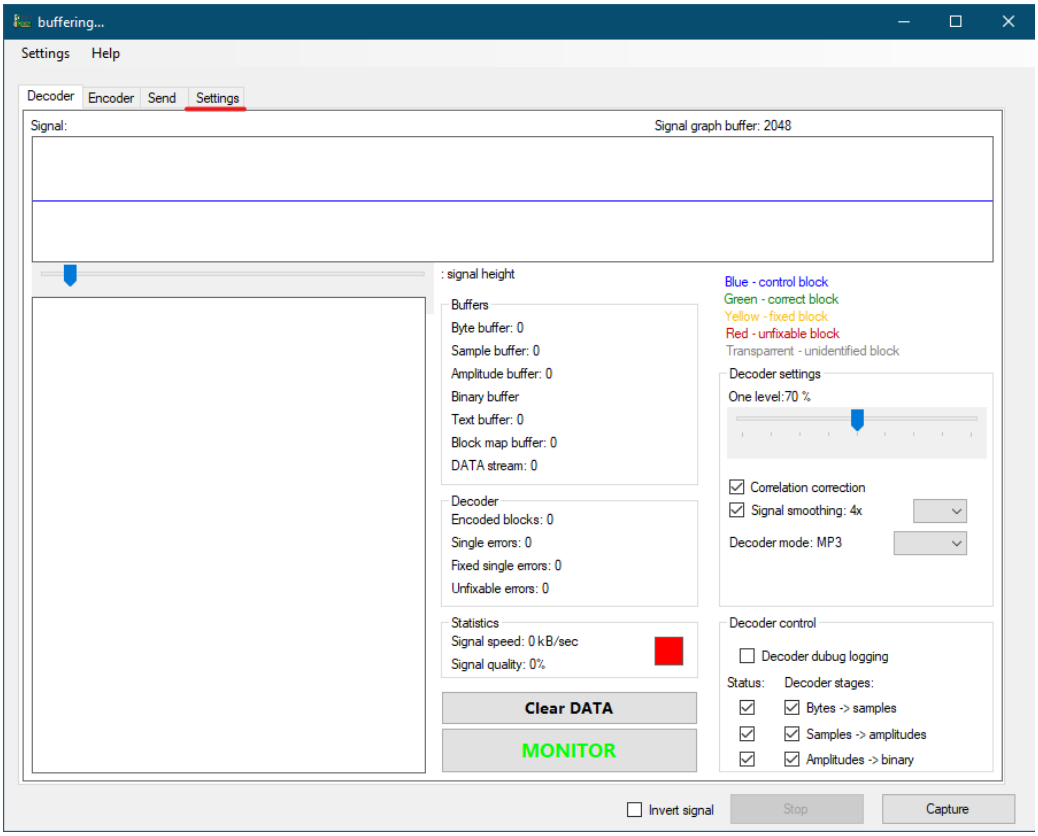
Format: MpegLayer3 (MP3); Sample type: • Sample rate: up to 8000 Hz; • Bit rate: up to 8 kbit/s • Bit depth: up to 16 bit; • Channels: 1 (mono) * You may use Adobe Audition or Audicity, for example, to convert any audio into these specials

    - Encoding process will be shown like that:

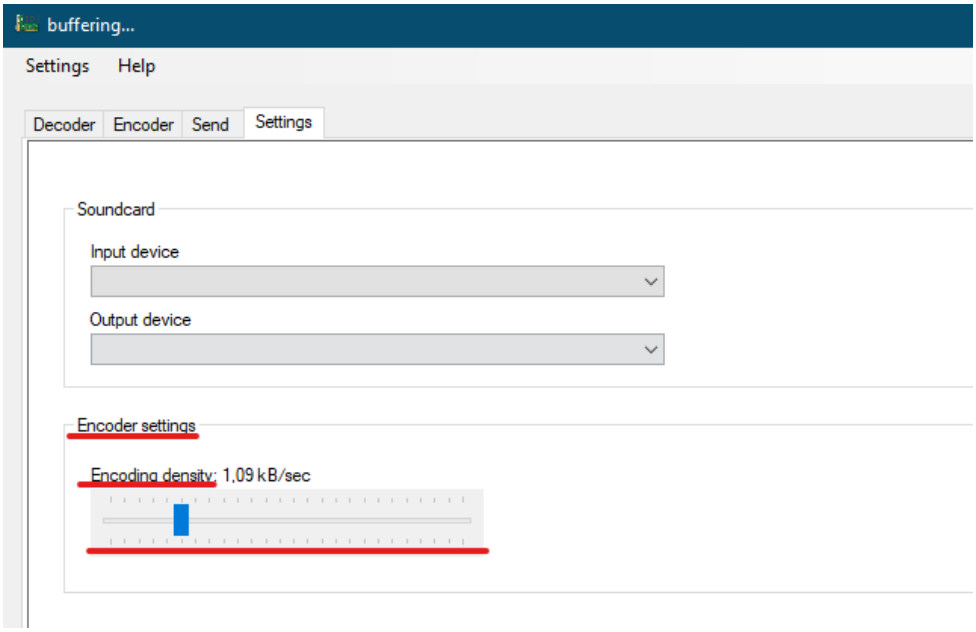


Picture Encoding

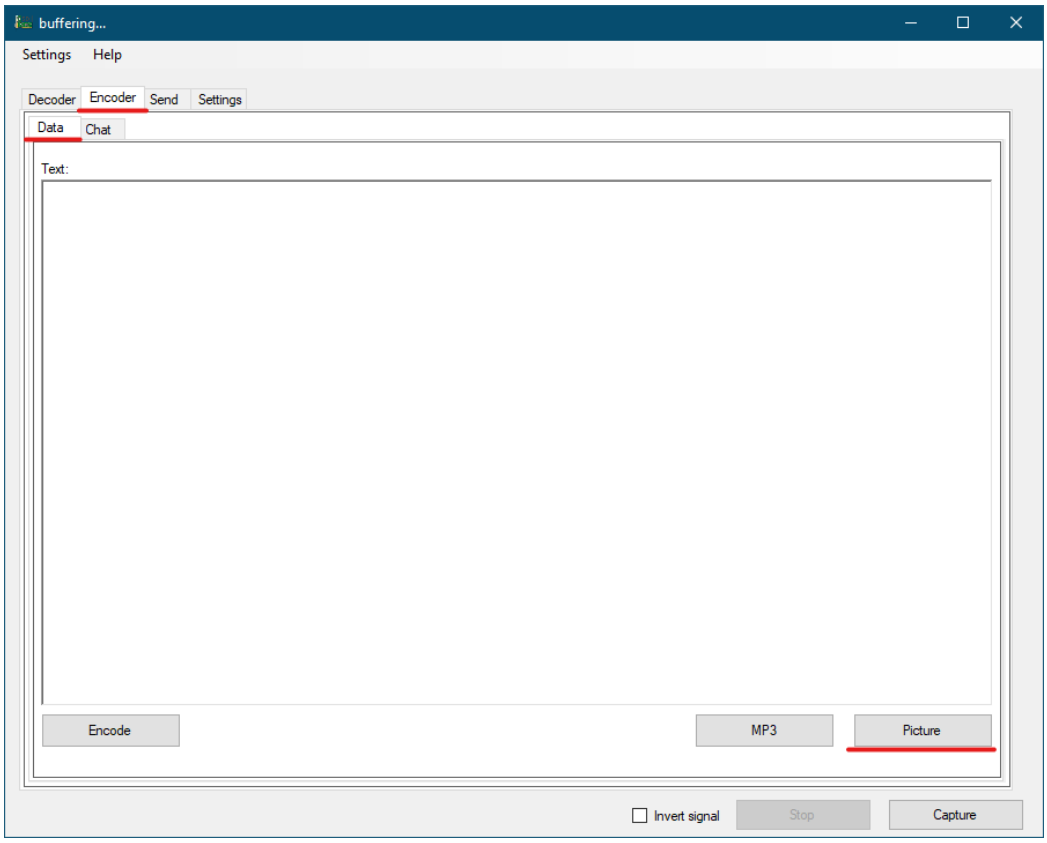
- Open the app
- Go to “Settings” menu



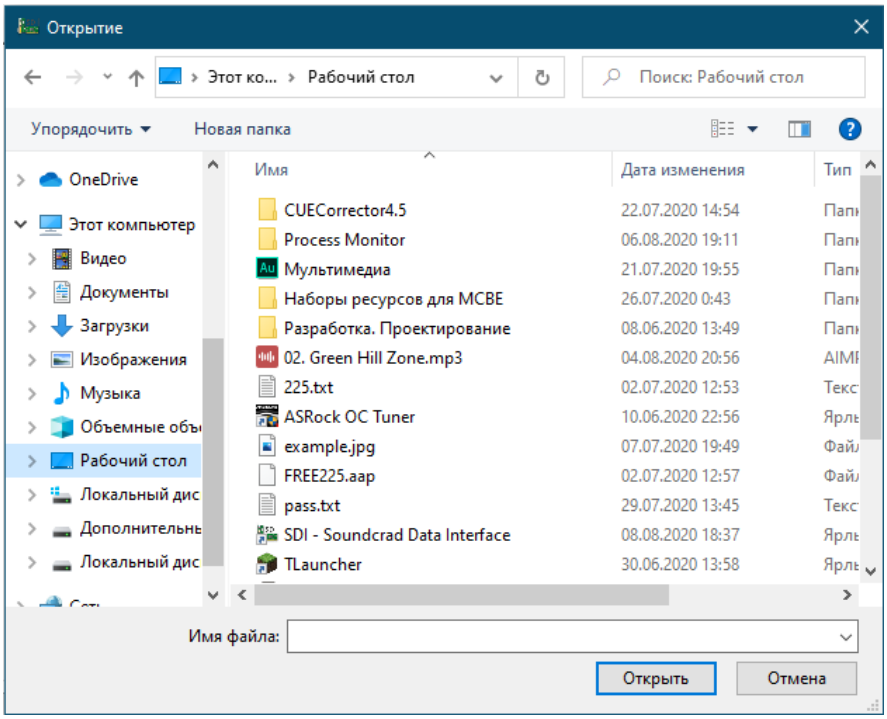
- Set “Encoding density” what (if) you want



- Go to “Encoder” menu, make sure, that you’re in “Data” submenu, then push “Picture” button



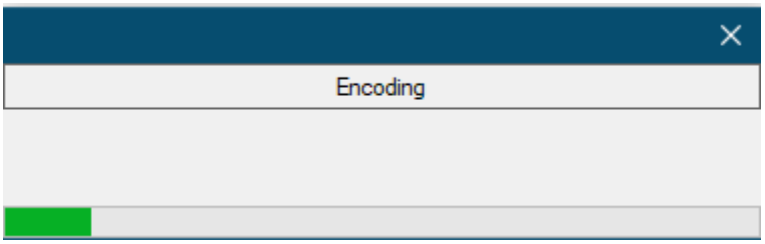
- Now you may select *any picture file*



- Selected file will be ***automatically*** converted into special format from grid below

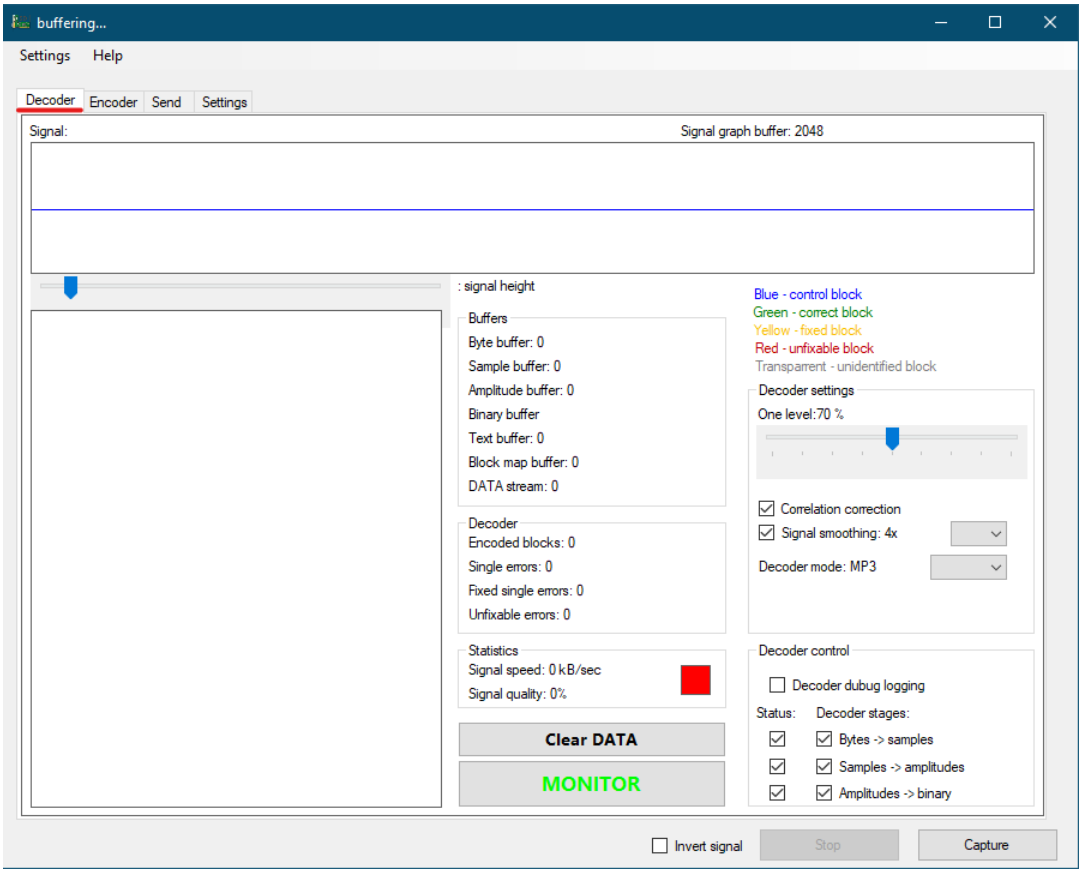
Format: jpg; Resolution: 800x600; Encoding type: progressive;

- Encoding process will be shown like that:

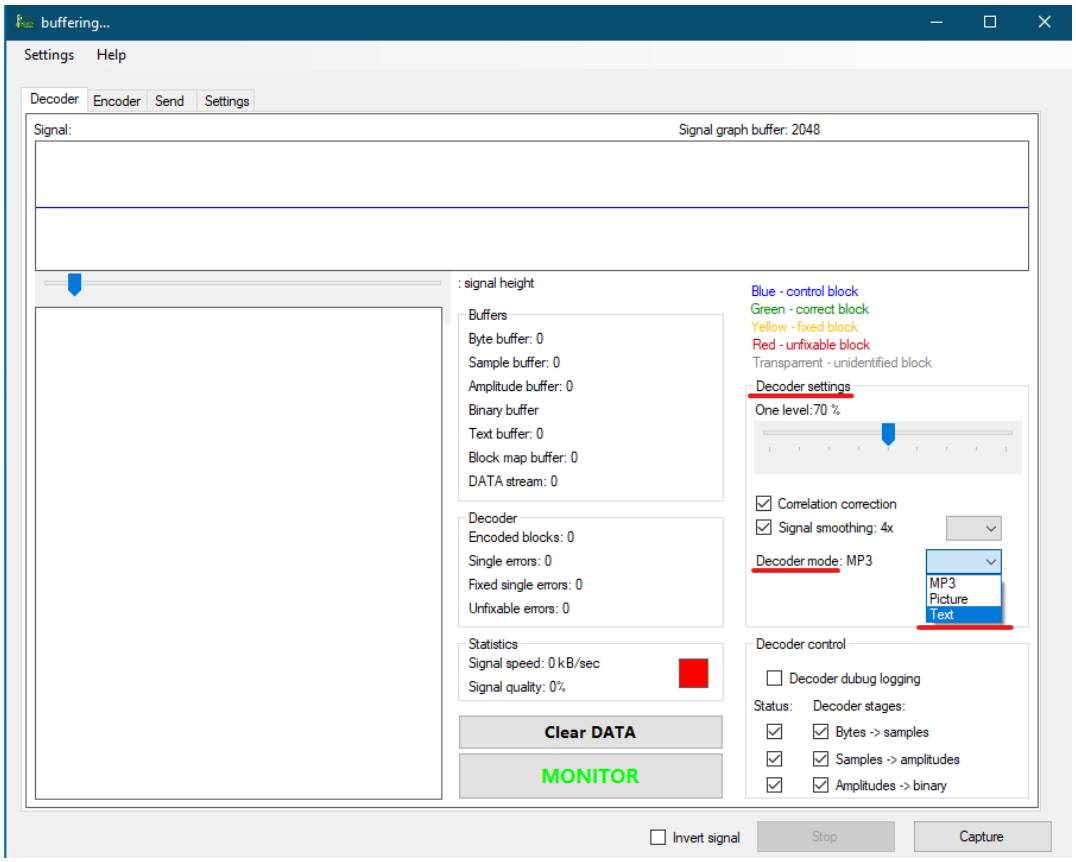


Data Decoding

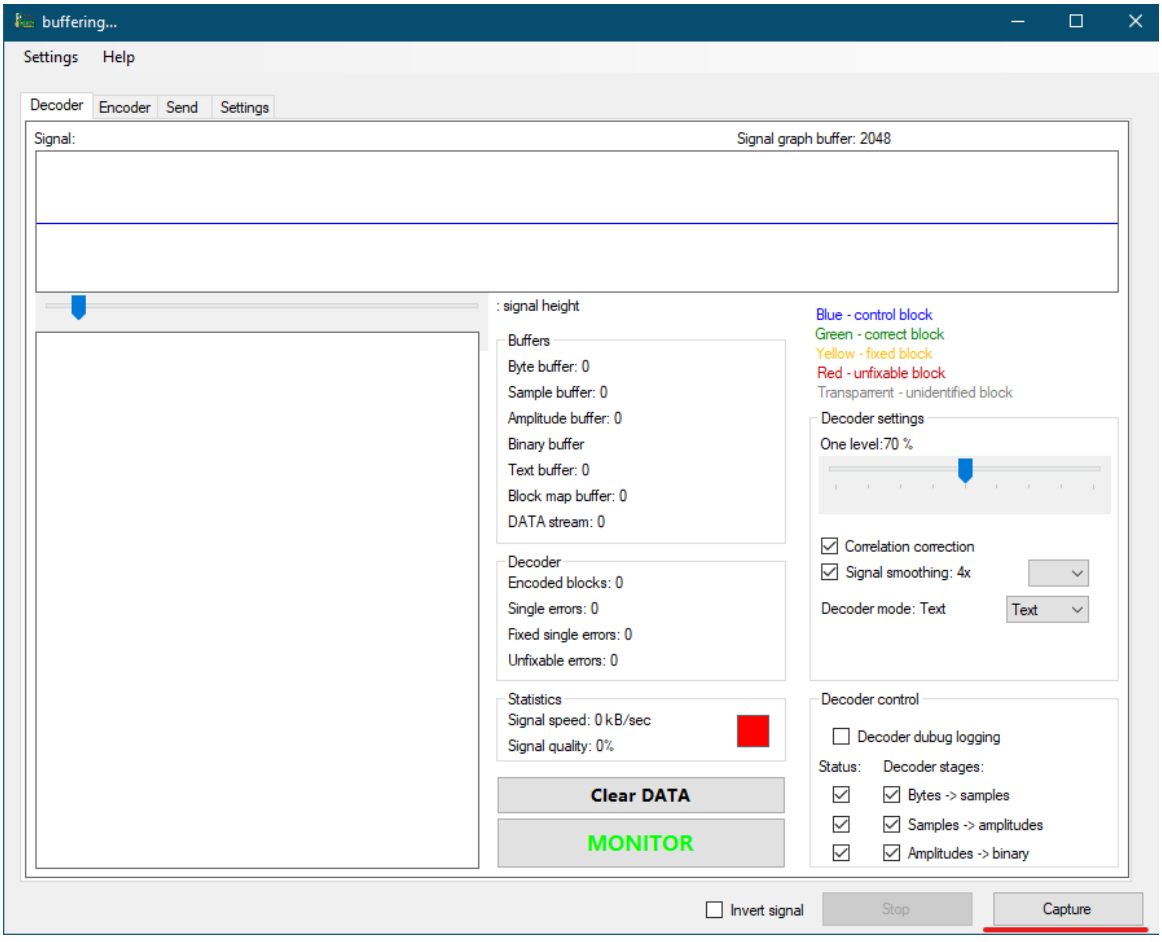
- Make sure, that you're in "Decoder" menu



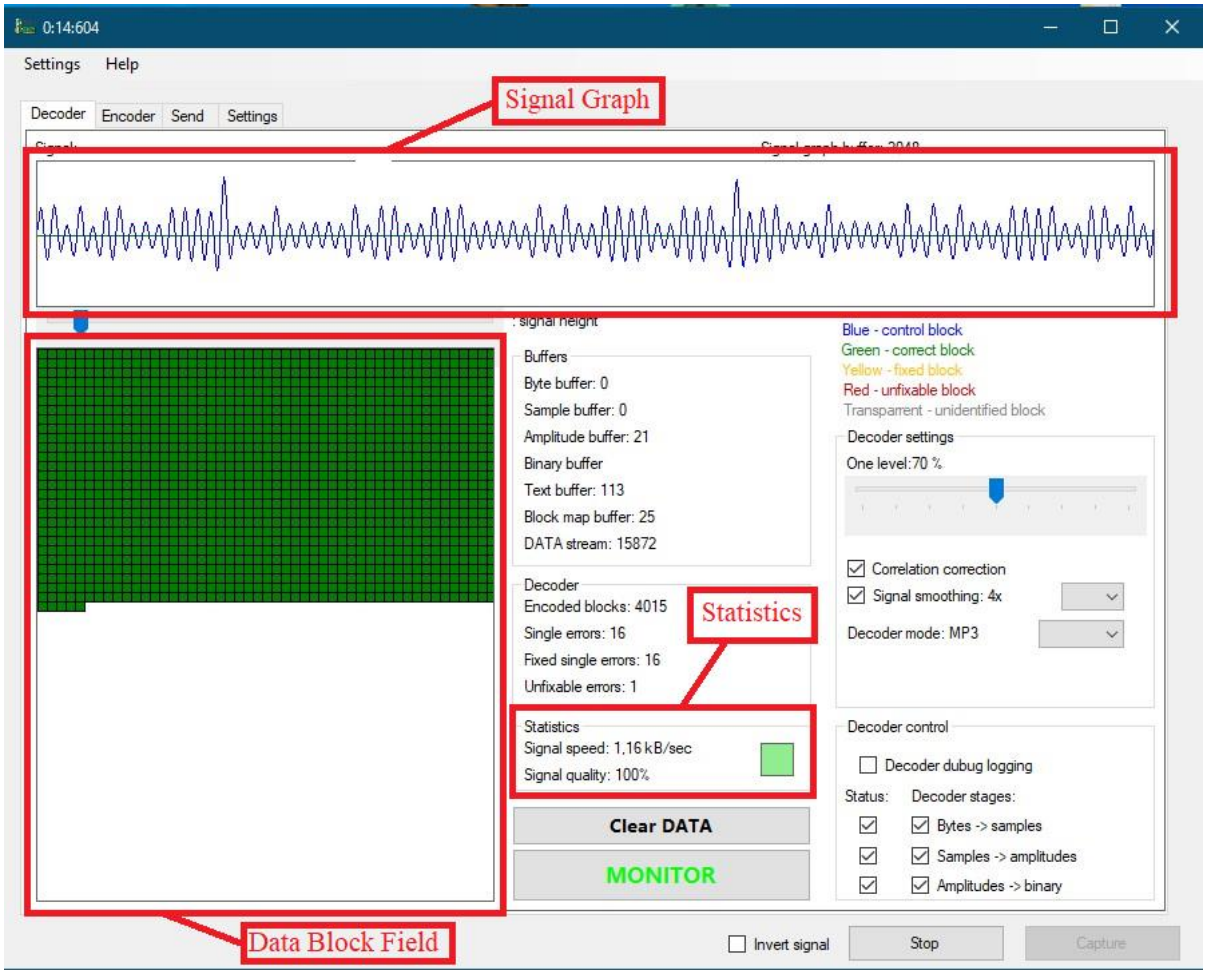
- Select “Decoder mode”, into “Decoder settings” group box, what you need

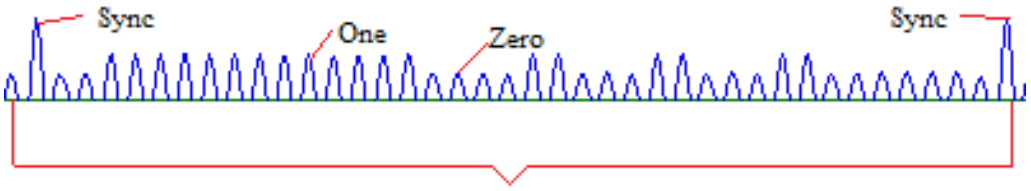


- Now push “Capture” button



Signal Monitoring



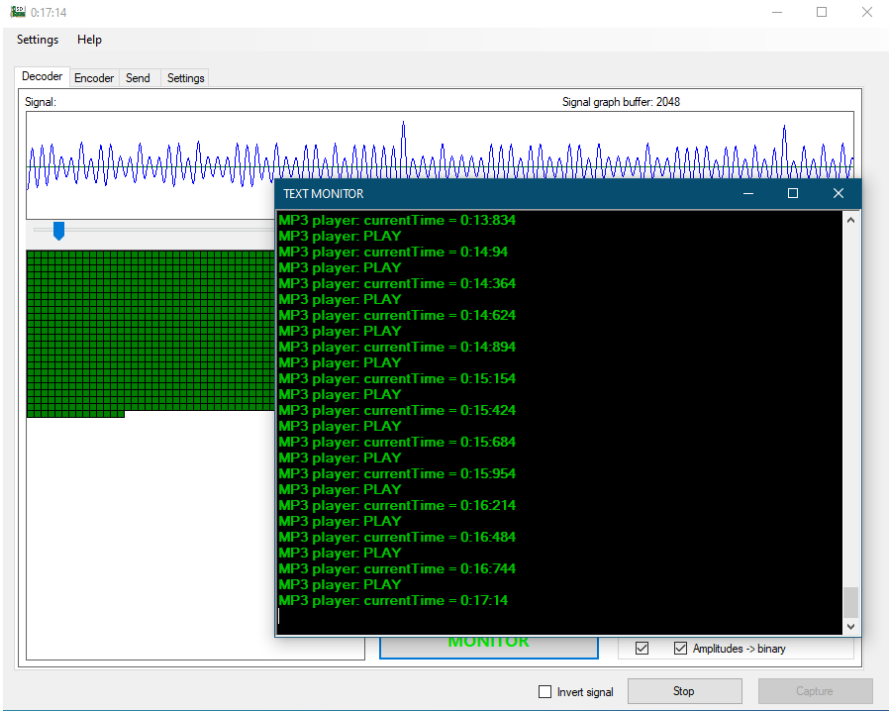


38-bit MP3 data block

Contains 4 bytes of useful data

12-bit data block
Contains 1 byte of useful data

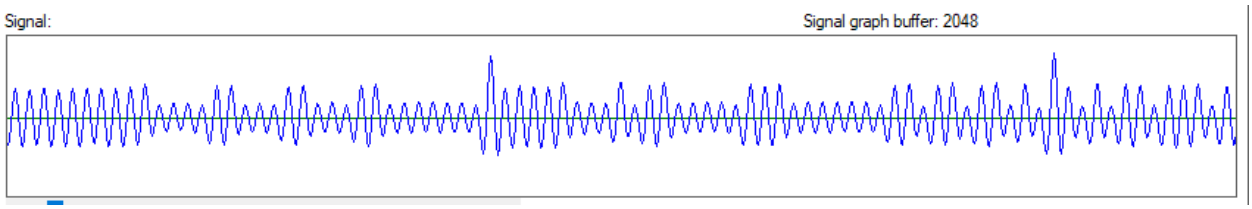
- Press “MONITOR” button to open decoded data monitor window (text, mp3 log, picture)



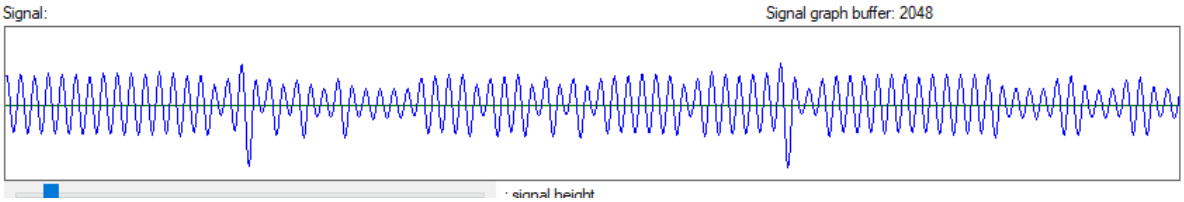
- Press “Clear DATA” button, if you want to clear data buffer (mp3, picture)

Decoder Settings

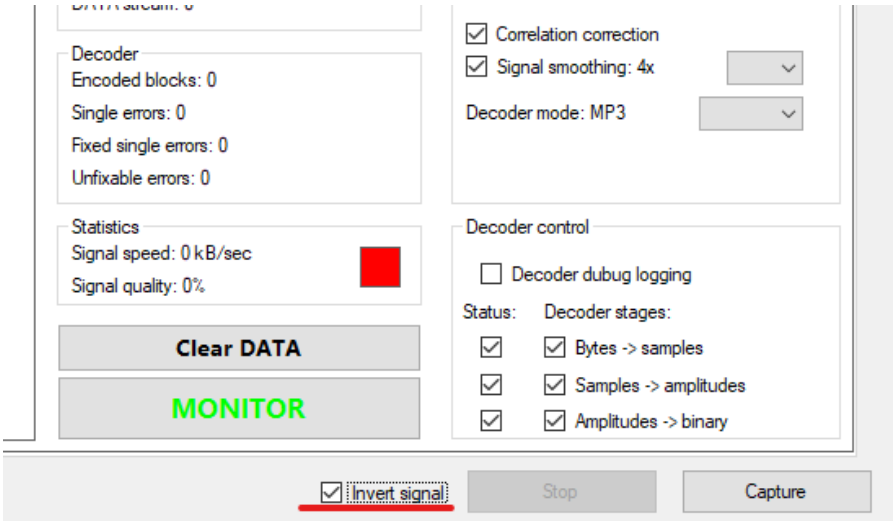
- If input signal has useful data, signal graph will be like that:



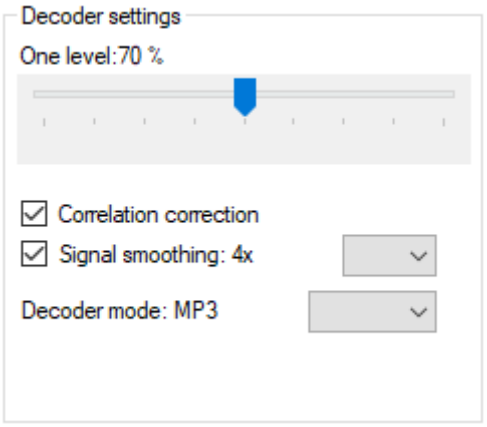
  - Be attention! Signal may be inverted around X axis and looks like:



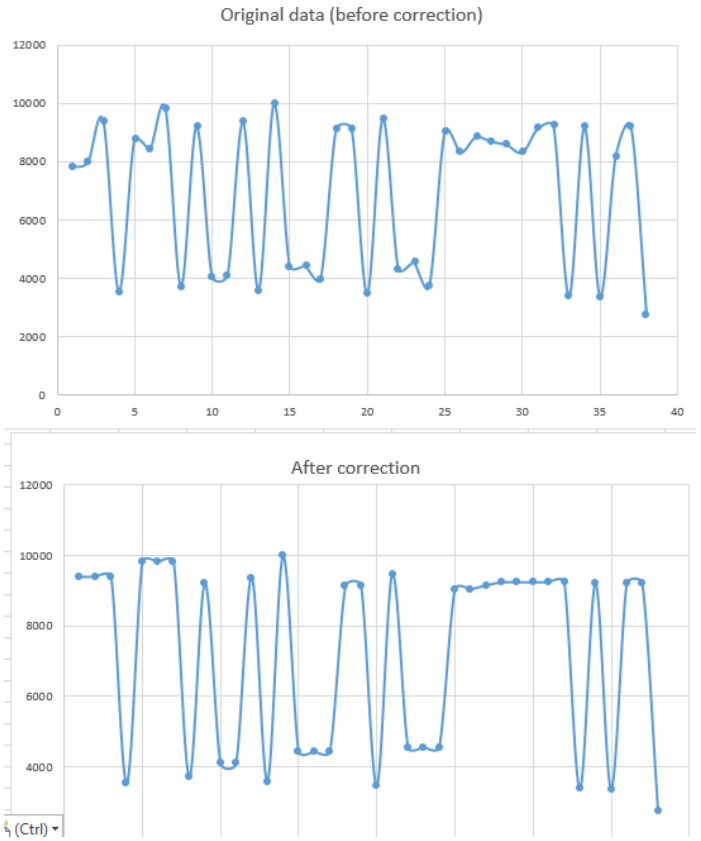
In this case, you have to enable “Invert signal” checkbox



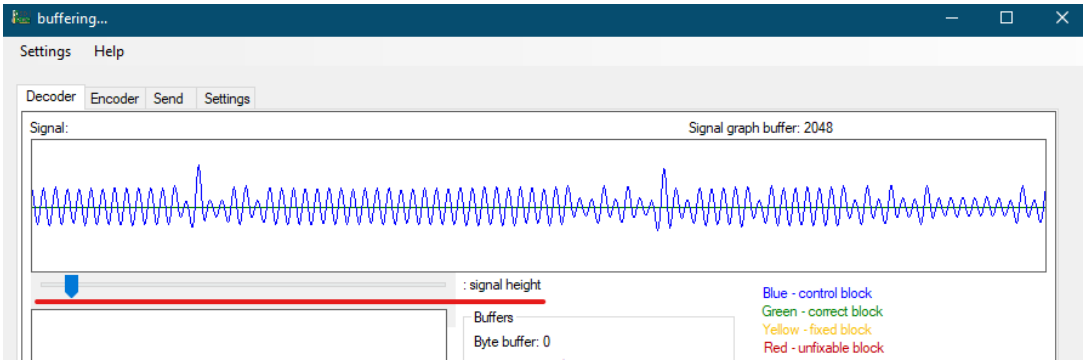
- General decoder settings:



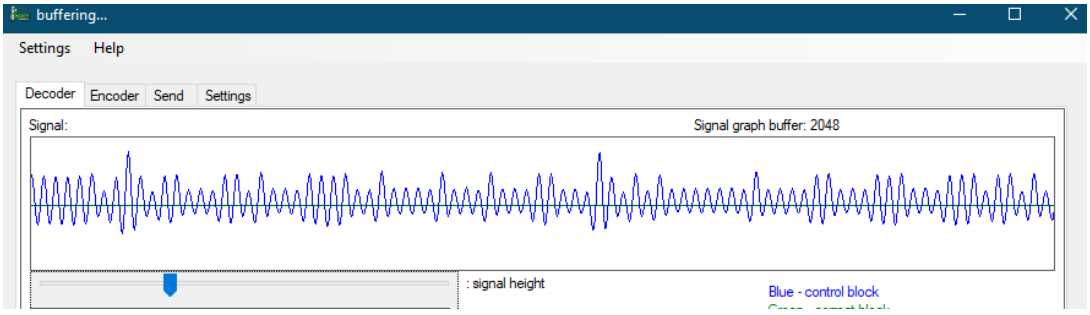
  - "One level" – part of bit detecting system, very important setting, that lets decoder work normally. You may change it to take more decoding stability and take less errors. Optimal value may variate depending on signal quality and signal source (cassette tape etc.)
  - Signal postprocessing options:
    - "Correlation correction": developed for signal linearization with low-stable sources, like cassette tapes.
    - "Signal smoothing": developed special for low-quality sources, like cassette tapes, that has non constant head azimuth:



- “Signal height” – setting, that lets you to change signal position around X axis. It may be need when “zero” amplitude is very small:

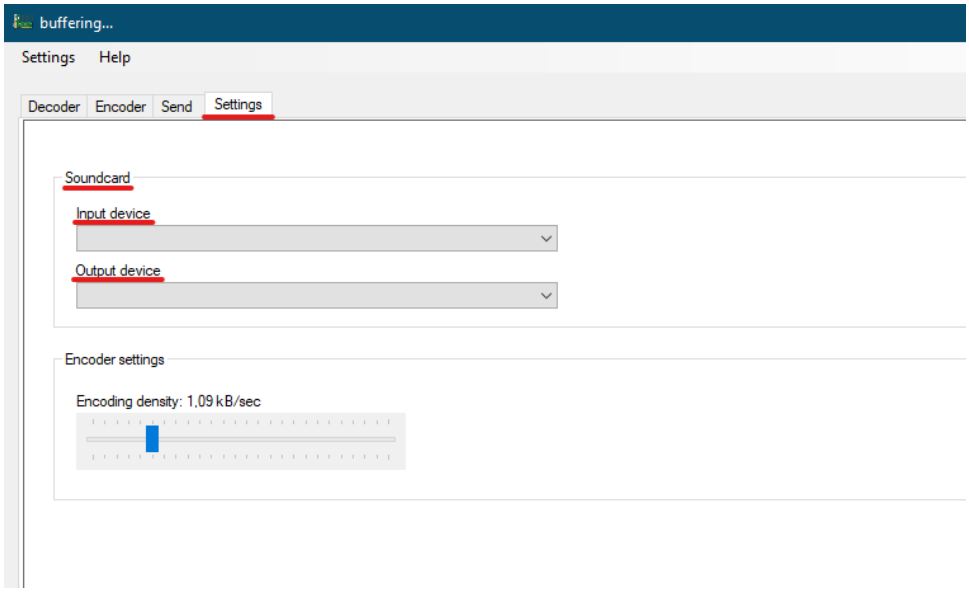


It’s much better:



Soundcard Settings

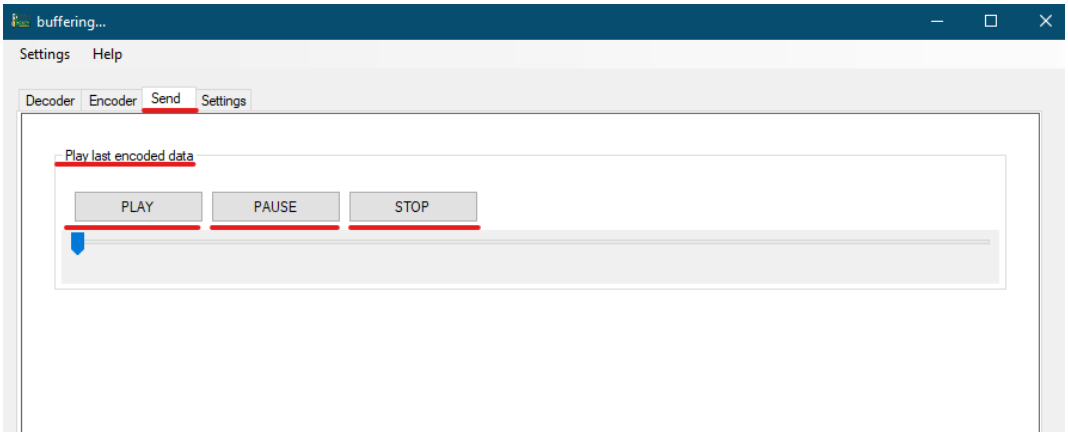
- You may change input/output audio device, using special setting



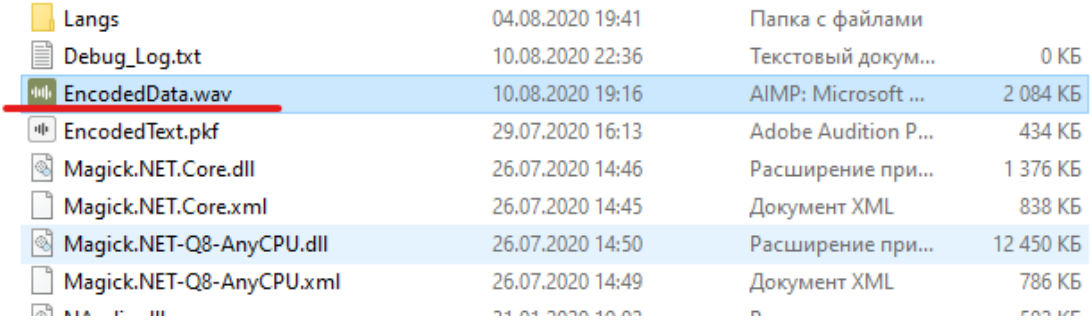
  - Application won’t be run without at least one input or output audio device!

Encoded Data Sending

- If you want to play encoded data, you should use “Play last encoded data” feature:

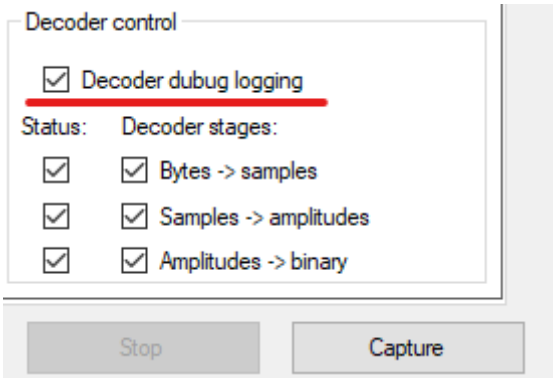


- However, you may play encoded data using external application with “EncodedData.wav” in app parent folder:



Debug Logging

- You may save all decoded data statistics onto the log file “Debug_Log.txt” (app parent folder)
    - Log file contains amplitude values of signal and signal analyzer data statistics, so you may analyze them to understand why you get errors



Encoded Signal Basics

- Text Signal Structure:

Block Type	**Size**	**Description**
Text Symbol	12-bit	Contains one text symbol (1-byte symbol)
Control Data Block	8-bit	Controls symbol libraries changing

- MP3/Picture Signal Structure:

Block Type	**Size**	**Description**
MP3 Data	38-bit	Contains 4 bytes of raw mp3
File Marker	32-bit	Lets decoder understand where useful data begins

- Decoder algorithm scheme:

Raw Audio Bytes from soundcard

Raw Audio Bytes to Samples conversion

Samples to Amplitudes conversion

Amplitudes to Postprocess Algorithm to Binary

Binary Error Analyzer (Hamming Decoder)

Final Data Block

General Remarks

- If you want to record some data on cassette tape, set recording level to -10 dB or less (for up to 1.3 kB/sec encoding density). To record data with 1.4+ kB/s encoding density, use -20 dB rec level or less.
- You may align play head azimuth on cassette deck, playing cassette tape, recorded on another cassette deck with digital data. Change play head azimuth until decoding process be stable.