IQConverter
A simple and extensible Tool for conversion and manipulation of IQ-Files

User manual

Bundesnetzagentur

Dipl.-Ing. (FH) Felix Schad

Betzenweg 32

81247 Munich

Germany

felix.schad@bnetza.de

Date: March 2015
Program version: 0.95

Index

1 Introduction

IQ-Files are computer files containing IQ-data. In most cases, they represent a digitized baseband representation of a signal that was recorded in the radio frequency domain. It also can carry data intended to be played back by a suitable signal generator.

Nowadays, IQ-Files are often used for measurement purposes and many other use cases. They allow a high degree of flexibility and have thus gained a wide popularity.

Depending on the use case and recording devices' manufacturer, those files often differ by their organization and data structure.

Usually, IQ-Files contain – apart from the obvious sampled and digitized representation of the waveform (or: signal) itself – other information, for example how to level the signal generator for optimal dynamics or the receivers recording settings.

The organization of the files often differs from device to device, depending on things like the digital-to-analogue or analogue-to-digital converter of the target device. Some formats were designed to store data, while others are organized to stream the packetized data over network connections.

Due to those exemplary matters an uncontrolled growth of file formats set in. Sometimes one can find software by device manufacturers to convert several formats to other ones or to use those files in a certain program for a specific task. But: one can be sure to be very limited when it comes to use files across the devices of different manufacturers.

Furthermore, the possibility to modify those files on top of the conversion is often limited (if existent, after all!), and no one can expect to get the most wanted basic features seen in one software package offered by another one.

The IQConverter is the answer to those issues. It's main task is to convert IQ-files from one format to another.

Its main focus is:

- A simple and fast handling of typical working tasks
- A directed supply of usual processing options
- A simple extension mechanism to supply
    - More data formats
    - Additional signal generators
    - Additional analysis routines
- Manufacturer Independence
- Mobile use
- Handling of arbitrary large files

<u>Note:</u>

During this manual, words written in capitals denote control elements of the program, for example

‚CONFIRM SELECTION‘.

2 Disclaimer

This software is provided as it is. Neither the author nor the German Bundesnetzagentur grant any form of warranty or obligation for the product described in this document and anything that's derived from working with that software.

3 Program features (selection)

- Possible ouput formats:
  - TIQ (Tektronix)
  - WV (Rohde&Schwarz)
  - WAV (Perseus-Receiver; Modified form of Microsoft Wave files)
  - RIQ (Rohde&Schwarz Receiver PR100, IF-Datastream as IQ samples)
  - CSV (Comma separated values, cf. section 8.4)
  - BIN (simple, self-made binary format, cf. section 8.5)
- Possible output formats:
  - WV (Rohde&Schwarz)
  - BIN (simple, self-made binary format, cf. section 8.5)
  - CSV (Comma separated values, cf. 8.4)
- Signal analysis mode for file's beginning and end
- Trunctation of file length
- Adjustable bandwidth limiter
- Power calculations across selected samples
- Automatic dynamic matching by normalization and scaling of the samples‘ power
- Spectrum inversion
- Language selection (currently: german and english)
- Ready to run from USB-Sticks
- Several options can be configured and carried along on a stick
- No installation neccesary
- Processing of arbitrary file sizes; almost independent of the computer's memory availability

4 Quick start (also: program start and shutdown)

This section describes a typical workflow with the program. The task is to convert a file recorded by a Tektronix RSA6114A to a format comprehensible to an R&S SMU200A. Additional requirements or modifications to the signal are not requested.

1. **Program start:** launch IQConverter.exe
2. **Choose the input file's type:** use the input field to set the type to *.TIQ: TEKTRONIX RSA6114A
3. **Load source file:** navigate to the source file after pressing the button ‚CHOOSE INPUT FILE‘
4. **Wait while the file is analyzed.** The progress is shown in the status bar at the program window's bottom.
5. **Choose the output file's type:** use the output field to select the output file's type to *.WV: R&S WV-FILE.

6. **Start conversion:** Click on the button CONVERT! in the bottom part of the window. The button is clickable, after a source file has successfully been opened. This will start the conversion. The progress is shown in the status bar at the window's bottom.
7. **The output file is located in the same directory as the input file.** The file's ending has changed according to the output file's type. The path shown if the mouse is being hovered over the field OUTPUT FILE.
8. **Close the program** by clicking on the 'X', using <ALT-F4> or the Menu ‚Tools'.
9. **Ready!**

5 Basics and clarification of terms

An I/Q-File (signal file) contains a number of I/Q-values in an arbitrary format according the the underlying file format.

A single I/Q-value represents a value $\underline{z}$ on the complex plane with $\underline{z} = I + jQ$.

A single I/Q-Value represents a sampling step with a sampling frequency F_s and a step interval $T = \frac{1}{F_s}$. The unit of the sampling frequency is Hertz [Hz]. Sometimes one can find the unit samples/second [samples/s], to emphasize the sampling frequencies nature.

The usable, aliasfree bandwidth of a sampled signal according to Shannons sampling theorem is half of the sampling frequency.
The so called complex-valued baseband signals here at hand are twosided. This means that the signals cover the frequency range of ${}^{-F_s}/_2 \leq f \leq {}^{+F_s}/_2$. This frequency range contains the frequency $f = 0\ Hz$, which is also called the direct voltage term or simply DC-Term. If one considers only the frequency range of $0 \leq f \leq {}^{F_s}/_2$, then the samples are considered to have twice the power as compared to the two-sided case.

The border frequency of ${}^{F_s}/_2$ has such a significant meaning, so that this limit is also termed as Nyquistlimit or Nyquist frequency.

In the digital signal processing domain, certain things are easier to comprehend when one thinks in multiples of the angular frequency ω or even π as the unit's circle circumference. The aliasfree frequency range is thus normalized with ${}^{F_s}/_2 = \pi$ to map it to $-\pi \leq f \leq \pi$.

A whole file contains N complex values.
All those values form the signal vector $\underline{\vec{x}}(\eta T)$ with $\eta = 0 \dots N - 1$.

In the following, the abbreviation $I_n \ = real\{\underline{\vec{x}}(\eta T)_{|\eta=n}\}$ and $Q_n \ = imag\{\underline{\vec{x}}(\eta T)_{|\eta=n}\}$ is applied.

The pseudo unit dBFS (DeziBel FullScale) is normalized to the full scale of a digital-to-analogue resp. an analogue-to-digital converter.
Example: a D/A-converter yields its peak voltage at a digital input value of 32767. When this converter is given that input value, then it is said to be leveled with 0 dBFS.

The RMS-level of a signal vector is given by:

$$ {}^{L_{RMS}}/_{dBFS} = 20 \cdot log_{10}\left(\sqrt{\frac{1}{N} \cdot \sum_{n=0}^{N-1} I_n^2 + Q_n^2}\right) $$

The peak-level of a signal vector is given by:

$$ {}^{L_{Peak}}/_{dBFS} = 20 \cdot log_{10}\left(\max\left(\sqrt{I_n^2 + Q_n^2}\right)\right) $$

The peak-to-average power ratio (or: crest-factor) is given by:

$$ {}^{PAPR}/_{dB} = {}^{L_{Peak}}/_{dBFS} - {}^{L_{RMS}}/_{dBFS} $$

6 Main window

The main window (cf.figure 1) is divided into three areas:

- Main area, cf. section 6.1
- Main menu, cf. section 6.2
- Area for status messages and progress displays at the window’s bottom.

Most elements supply tool tips. Thus, hovering the mouse pointer for a short time over a graphic element will show a small help window.

6.1 The main area

Most part of the main window is occupied by the so called main area.

The main area is divided into four sections. The sections are dived in such a way to be worked from up to down, step by step during the course of a workflow.
The four sections are:

1. the input area
2. the property view are of the input file
3. the output options
4. the output area

Each of those four parts is marked by a red boundary in the following figure 1.

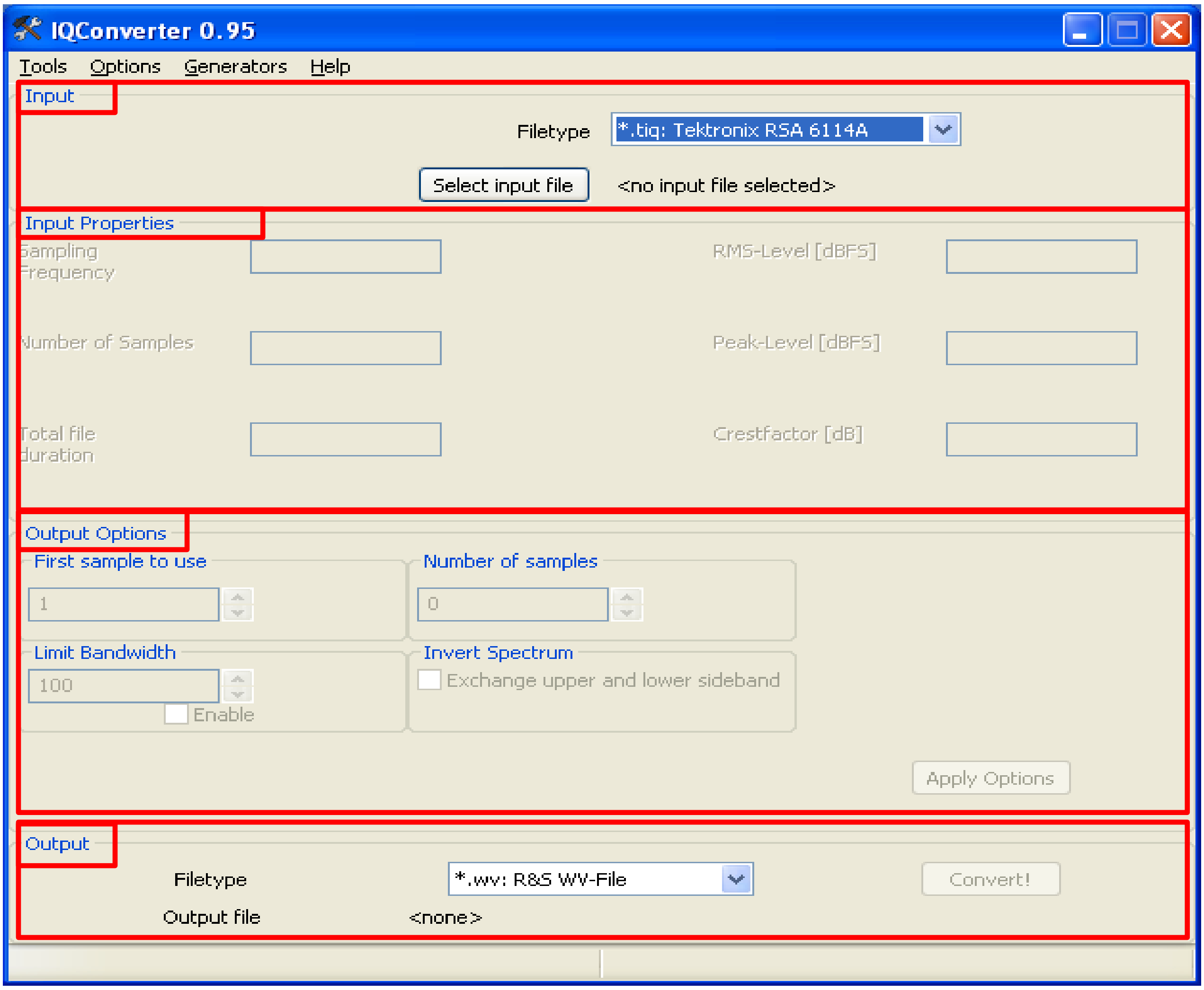

figure 1: the main window. The red boxes mark the individual parts of the main working area

The following section describes the user elements of the main area.

6.1.1 Input area

The input area allows the selection of the input files type and its format.

The file type's selection can be made by use of the Drop-Down-Box FILETYPE. The inputfile can be chosen by clicking the button SELECT INPUT FILE.

The filename is shown to the right of the button SELECT INPUT FILE. Hover the mouse pointer over that area if the name is too long to be shown completely. Selection of a different file type invalidates a previous file selection. Aborting the file selection dialogue by using the Escape-key or clicking Abort not only aborts the file selection dialogue, it also invalidates the currently selected file.

It is also possible to select the input file by dragging and dropping it from a file manager (example: the file manager of Windows). In that case, the IQConverter tries to guess the file type by the file's extension.

The IQConverter starts reading the source file right after its selection. It checks if the file is readable and if all information for further processing is available. It calculates the normalization factor, which is the maximum absolute value found in the file. Furthermore, the RMS-level and PAPR is calculated. The user can watch the progress of that operation by the progress bar in the status area, as shown in figure 2.

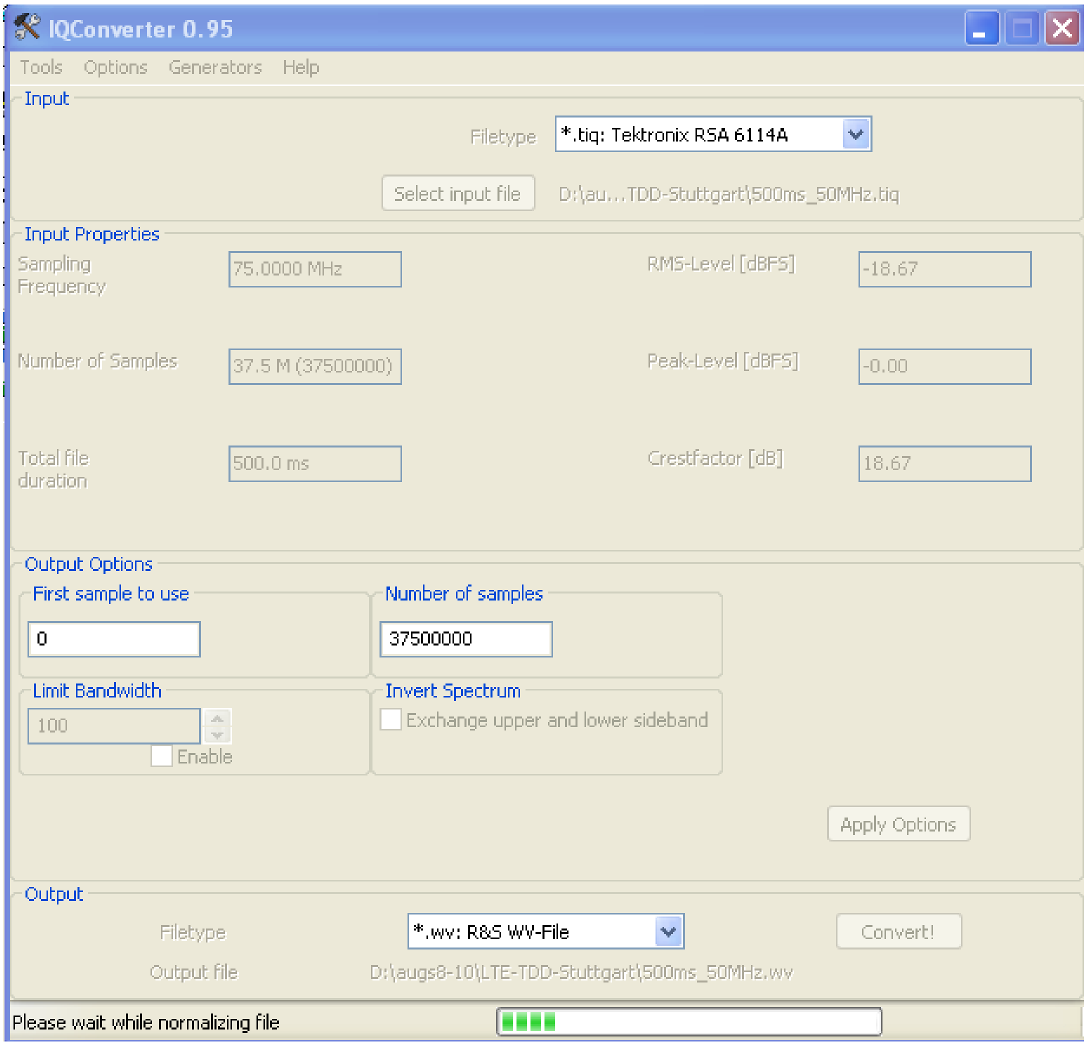

figure 2: main screen with progress bar

6.1.2 Input file property area

Several properties of the input file are shown once the reading process has finished. Those values can never be changed by the user.

The fields shown are:

- SAMPLING FREQUENCY F_s including its unit (kHz, MHz, …)
- NUMBER OF SAMPLES contained in the file. The value indicates a rounded value for faster orientation, while the number in braces gives the exact value of N.
- TOTAL FILE DURATION shows the total signal duration in terms of time. This value is given by: $t_{total} = N/F_s$.
- The values for the RMS-LEVEL, PEAK-LEVEL and CRESTFACTOR in dBFS resp. dB according to the equations given in section 5. Note: the peak level should always be 0 dBFS due to normalization.

6.1.3 Output options area

This area offers several signal processing options prior to the conversion.

The options are:

- The value given in the field FIRST SAMPLES TO USE sets the number of the first sample to use in the output file. Any other sample before this one is skipped. Please not that this will also change the NUMBER OF SAMPLES, since there are less samples available to put into the output file. The default value is 0 (zero). The number shown in the field NUMBER OF SAMPLES in the input file's property area always reflect the total number of samples in the input file.
  Example: assume an input file with 1000 samples. Setting the first sample number to use to 1 automatically sets the number of samples available for the output file to 999.
- The NUMBER OF SAMPLES states how many samples are to be written into the output file.
- The field bandwidth limitation is activated by marking the checkbox marked 'Enable' next to it. This inserts a lowpass into the conversion chain. Its border frequency is adjusted according to the percentage value in the bandwidth limitation field. This is a percentage value relative to the nyquist frequency. This means, that a value of 100% sets the border frequency to $F_s/_2$.
  The applied filter is a FIR-filter tapered with a Kaiser-window. The design parameters are:
  - Insertion loss: 0 dB
  - Typical stopband attenuation: 80 dB
  - Ripples in the passband: -80 dB
  - Ripples in the stopband: 20 dB
  - Width of the transition region (corner frequency → stop frequency): factor 1.02 relative to the filter's corner frequency.
    Example: assume that the corner frequency has been set to 25 kHz. This will make the stopband start at $1.02 \cdot 25\ kHz = 25.5\ kHz$.
  - Maximum number of filter taps: 1023. The actual value depends on the relation of sampling frequency and corner frequency. The filter design algorithm tries to reach a minimum number of taps in an iterative manner. If the maximum number of taps does not suffice, then the algorithm increases the transition width in small steps until the design parameters (ripples and stopband attenuation) are met.
  - The filter is always designed with an uneven number of taps. This ensures that the group delay always matches to an integer value of sampling steps. Furthermore, the phase transfer function is always linear in the passthrough area.
  - The filter is always normalized in such a way that the program's internal value range of $-1 \cdots + 1$ is never exceeded (BIBO-criterion). This is reached by normalizing the vector of filter coefficients $\vec{h}(\nu)$ with its length K by the factor M which is given by:

    $$M = \sum_{\nu=0}^{K-1} |h(\nu)|$$

    Finally, the filter coefficients are scaled by $1/\sqrt{2}$ because of I/Q-processing.
  - Further details (also for documentary use) of the filter can be seen by calling an option in the tools menu, see figure 6.
  - **<u>Important notice</u>**: the filtered output data stream is always shorter than the input data stream by a certain number of samples. This number is always half of the size of the filters group delay, to give the device enough time to settle.

If one uses a manual change of the sample numbers in combination with a filter, then it's easier write the filtered data stream first into an temporary file in an intermediate format (for example, the .bin-format). This file is then used to trim the number of samples. Of course, one can safely activate the filter if the number of samples in the output and the exact first / last sample in the file doesn't matter.

- The option INVERT SPECTRUM swaps the upper and lower sideband of the output file. The intention of that option is to correct a spectrum inversion if for example external mixers were applied prior to the sampling device. Internally, this process is nothing else than a swap of the I/Q-components, so I gets Q and vice versa.
- The button APPLY OPTIONS gets activated automatically if needed. This is the case if, for example, the number of samples has been changed. This requires a recalculation of the signal levels and the normalization factor. If the user has finished adjusting the number of samples, then this button has to be clicked. The button CONVERT is not useable before this has happened.

6.1.4 The output area

This area controls the output file format, shows the output files name and the starts the conversion process. The following elements are offered to the user:

- Drop-Down box FILETYPE: this drop-down box is used to set the output file type.
- Textfield OUTPUT FILE: this area shows the output files path and name. If the name is too long to be displayed (indicated by '…' marks in the filename string), then the user can hover the mouse pointer above the area. A tooltip will open up and show the full name.
  Important notice: the output file always gets the same name as the input file, only the extension changes. This makes it easier to see which files belong together. If the user likes to add some characters to the output files name, the he can use the option described in section 6.2.1.3.
- Button CONVERT: starts the conversion.

6.2 The menu bar

The menu bar offers several options and tools. They are described in this section.

6.2.1 Menu item „Tools“

The tools-menu contains things which are of interest during a conversion process.

6.2.1.1 Menu item „Run WrapAlyzer“

The WRAPALYZER[1] is a special, unique analysis tool to examine the start- and end-region of a source file. Its use is to identify and eliminate the spectral splatter that sometimes occurs, when a signal files end is reached and the generator starts to replay the file from its very beginning.
Spectral splatter means that the generator produces a wideband noise increase across its bandwidth. If this splatter is not wanted (to create a spectral clean signal and / or to protect measurement values from ambiguities due to this effect), then the WrapAlyzer offers the possibility to find a suitable sample combination between the file's start and end to suppress that effect as good as possible.
It should be obvious that filtering or other signal processing options does not lead to an optimal solution, since the nature of the wanted signal is generally not known and usually should not be

[1] Made-up-word consisting of the term ‚Wrap-around‘ and ‚Analyzer‘. The Wrap-around names the transistion from a file‘s ending to its beginning.

altered in any way. Furthermore, different signal types have to be treated in different ways: pulsed signals need a different treatment that continuous ones.

Usage:

After an input file has been loaded, the menu item ‚Run WrapAlyzer' gets activated. The file must contain at least 2x4096 samples. The window shown in figure 3 appears.

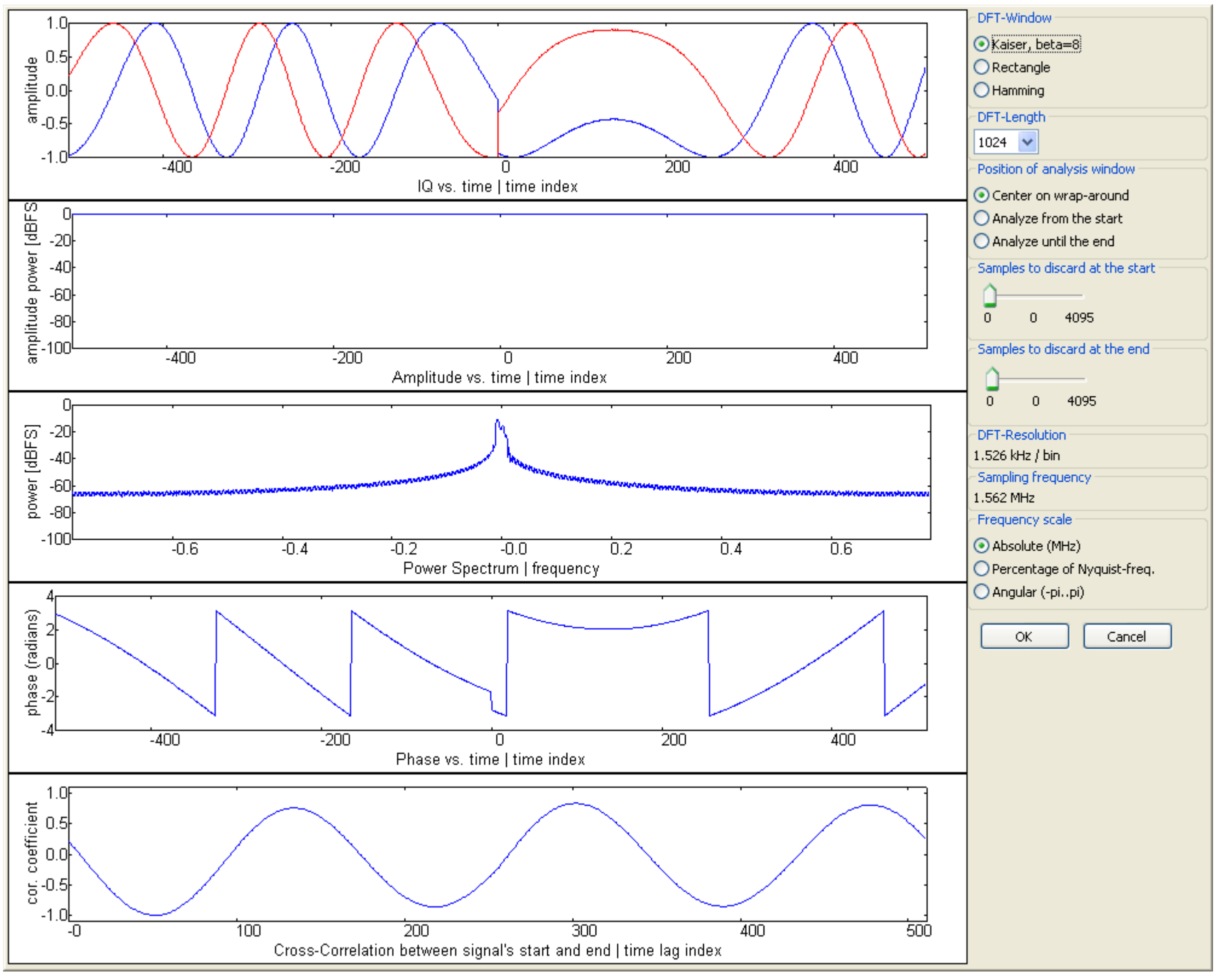

figure 3: the WrapAlyzer

The left part of the window shows several traces; the right window section offers several options of valuable settings to aid the analysis.

The options are:

- DFT-Length: offers a range of different DFT-windows which are applied to the spectrum analysis window. Possible choices are:
  - Kaiser, β=8
  - Rectangle
  - Hamming
- DFT-Length: sets the number of samples to analyze.
  Attention: all analysis traces get only as much samples as indicated by the DFT-length! There is no zero-padding involved.

Possible choices are: 32, 64, 128, 256, 512, 1024, 2048 points.

- POSITION OF ANALYSIS WINDOW: this choice determines what part of the signal goes to the analysis windows. Possible choices are:
  - CENTER ON WRAP-AROUND: the timeindex 0 marks the files start. Indices below zero relate to samples backward into the file (so, -1 is the files very last sample). Indices from zero on indicate samples from the files start. In this mode, DFT-Length/2 number of samples come from the last samples in the file and DFT/2 samples come from the first samples in the file.
  - ANALYZE FROM THE START: the analysis windows are fed with the very first sample of the file and end at sample number DFT-Length-1.
  - ANALYZE UNTIL THE END: as above, but seen from the files end and backwards.
- SAMPLES TO DISCARD AT THE START / AT THE END: Those two sliders allow to skip a certain number of samples from the files start and end. The left value below the sliders show the minumum value to set, the rightmost value shows the maximum adjustable value. The value centered below the slider shows the current setting.
  Those two sliders allow to adjust the start / end of the analysis window until a certain design goal – minimum spurious emissions at the wrap around point between the files ending and start, for example – is met.
  The sliders can be adjusted by the mouse or the keyboard. One can shift the sliders by one sample by using the left/right key on the keyboard. The page up/page down keys will shift them by 10% of the sliders end value. If the keys keep pressed, then the slider keeps moving until they are released. The analysis graph will be updated as soon as the slider doesn't move any longer.
- DFT-RESOLUTION: this option shows the frequency resolution of one DFT-Bin. The higher the DFT-Length, the smaller the resolution and the larger is the number of samples supplied into the analysis windows.
- SAMPLING FREQUENCY: displays the source files sampling frequency.
- FREQUENCY SCALE: determines the label style of the spectrums frequency axis. Possible options are:
  - ABSOLUTE (MHZ): the frequency axis is scaled according to the sampling frequency.
  - PERCANTAGE OF NYQUIST FREQUENCY: the frequency axis is scaled in a percentage relative to the Nyquist frequency.
  - ANGULAR: the negative / positive Nyquist frequency is shown as $-\pi$ bzw. $+\pi$.
- Button OK: quits the WrapAlyzer. The slider settings are transferred to the fields FIRST SAMPLE TO USE / NUMBER OF SAMPLES in the main window.
- Button CANCEL: quits the WrapAlyzer without effect of the slider settings.

Meaning of the analysis displays (from up to down):

- IQ vs. time: Shows the voltage of the I/Q-values along the time axis as on a two channel oscilloscope. The I-channel is shown in blue; the Q-channel in red.
- Amplitude vs. time: Shows the magnitude of the I/Q-pairs in dBFS.
- Power spectrum: DFT of the analyzed samples accordings to the settings for the DFT-window and frequency scale.
- Phase vs. time: Shows the phase of the I/Q-Pairs across the time axis in radians. Values are limited to the range $-\pi \ldots + \pi$. Values outside this range are automatically unwrapped.
- Cross-correlation: shows the similarity between the samples at the end and at the start of the analysis window. The correlation coefficient is shown on the y-axis.

Usage example:

A file containing an FM-modulated signal is to be tamerepd with the WrapAlyzer in such a way that the spurious emissions at the point where the file restarts are minimized.

After loading the 1-seconds long file and starting the WrapAlyzer, the window shown in figure 3 appears. One can see from the IQ-over-time diagram, that there's a sharp kinck where the files start and end come together (at sample number 0). The same can be seen from the phase diagram. Furthermore, the phase diagram shows that the phase is continuously decreasing in a linear fashion towards the files end. At the files start, the phase starts to increase. One could conclude that the file contains an FM-Chirp with a triangular shape. Furthermore, one can see from the correlation diagram, that there is kind of a periodicity in the file. The noise floor during the wrap-around point is approx. at -70 dB in the spectrum window.

First, switch through the option in the section POSITION OF ANALYSIS WINDOW . One can see, that the signal usually has spurious emissions in the range of -80 dB. This value is the target we'll want to reach in this optimization step.

The phase diagram shows, that in the region of sample number 250…400 the edge rises again in a similar way as on the file's end (but it's mirrored). Using the mouse, we'll set the slider SAMPLES TO DISCARD AT THE START into the region of approx. 300. Using the arrow-right key to step forward one can see, that the kinck (not only) in the phase diagram start to soften up. In the region of 320 one can see, that the spurious emission drop down to -100 dB. A few slight variations later, one could decide to set the slider to the value of 319, as shown in figure 4.The figure shows a significant lower level of spurious emissions. The kinck in the IQ vs. time diagram might look confusing, but it isn't: during that time, the Q-channel is at its peak, while the I-channel has almost no voltage at all. So, it won't contribute to the spectrum.

This means that a good point to cut the file has been found. A Click on the OK-Button confirms the selection. Back in the main window, one has to click on APPLY OPTIONS to set the new parameters for the conversion process. After the levels have been recalculated, the conversion process is started using the button CONVERT!

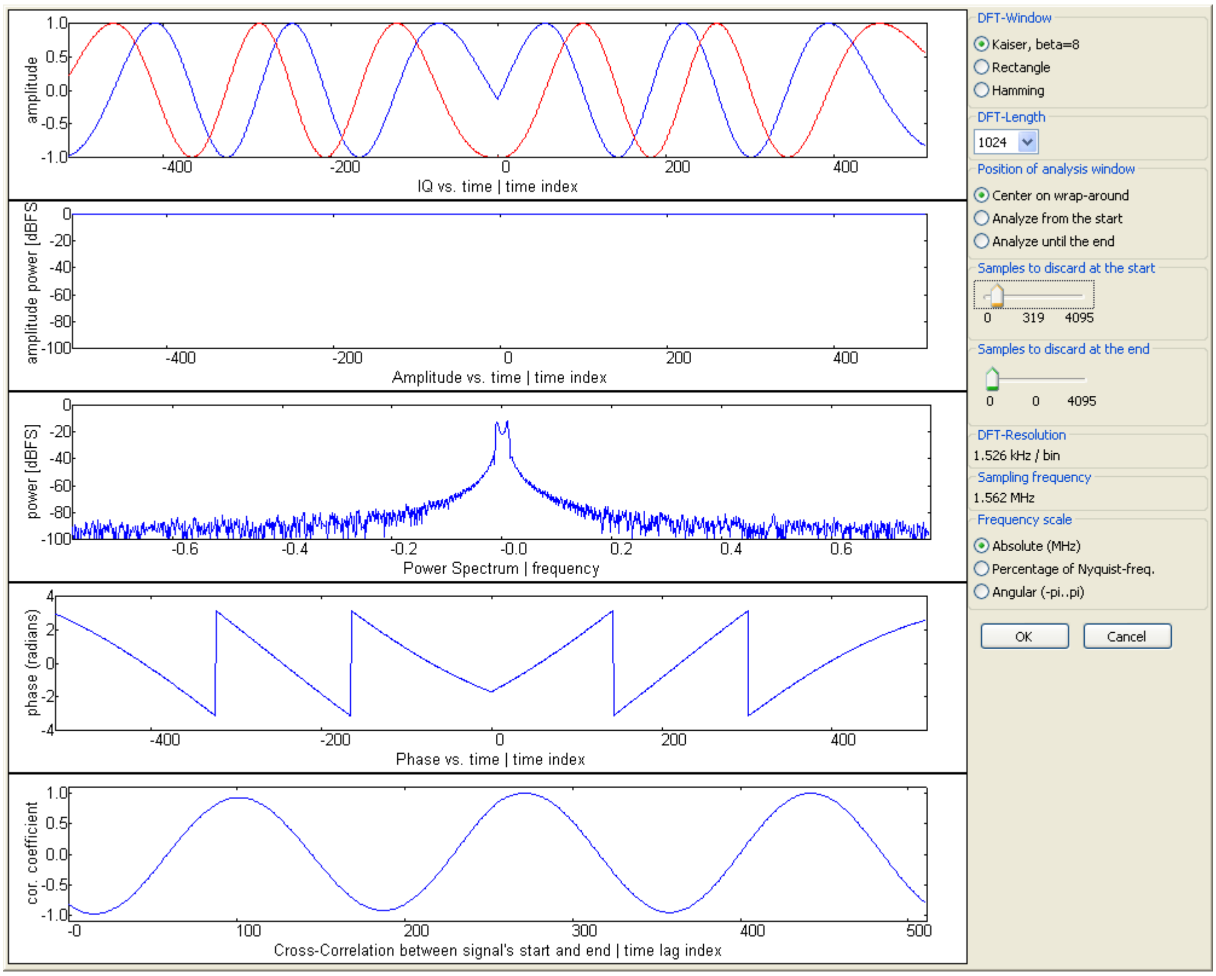

figure 4: FM-Chirp after optimization with the WrapAlyzer

After the conversion, a generator was used to replay the file. The generator was connected to a realtime spectrum analyzer. The orange spectrum in figure 5 shows the generator's signal during the time of the file's restart when the file is not optimized. The yellow spectrum shows the signal for the optimized file. One can see that the sideband emissions have dropped by roughly 30 dB – as indicated by the WrapAlyzer. This shows that the optimization's target has been reached.

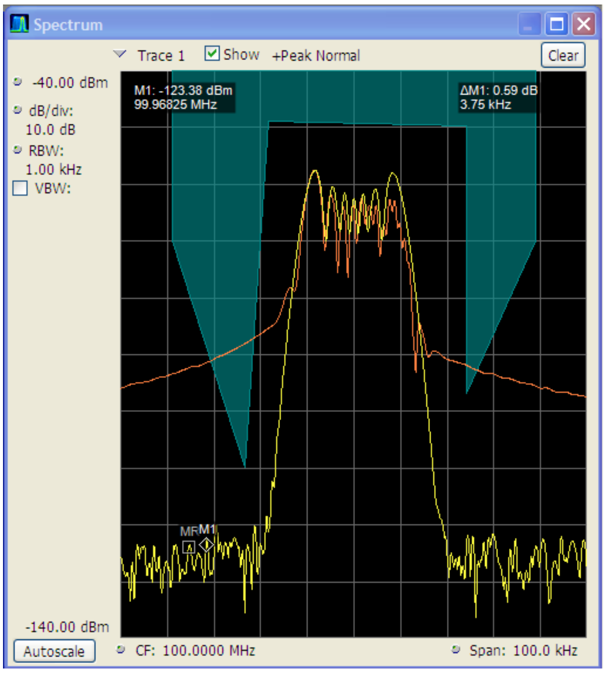

figure 5: wrap-around point of the edited signal shown on a spectrum analyzer

6.2.1.2 Menu item 'show filter'

This menu item allows viewing the filters response, if the output option for filtering has been activated (see section 6.1.3). The filter analyzer will appears as shown in figure 6.

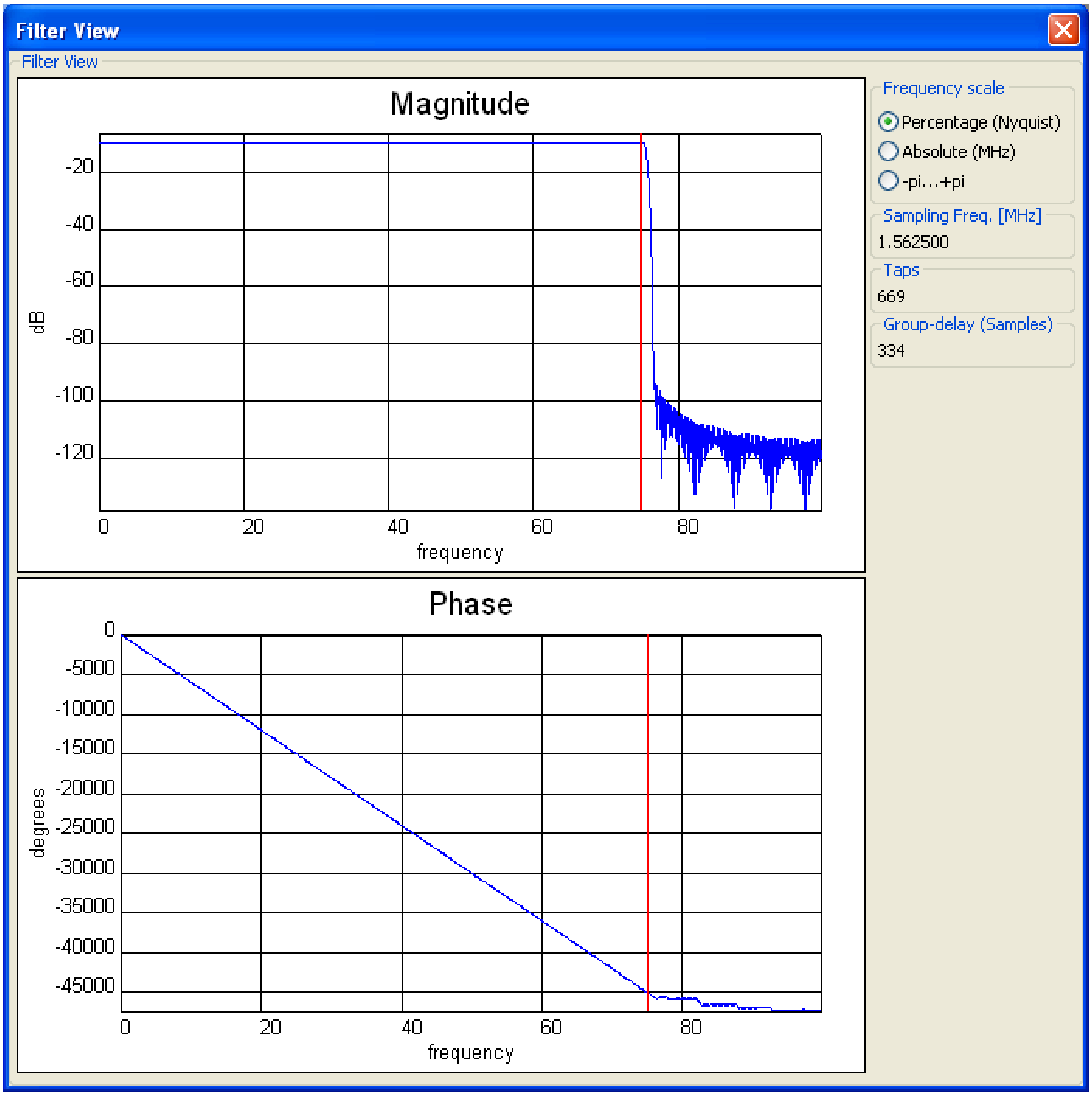

figure 6: the filter analysis window

The filter analyzer shows the magnitude and phase response of the filter along the frequency axis. Please note that the responses are the same for frequencies less than zero; they are just mirrored on the vertical axis. Furthermore, one can see that this design intentionally leads to a filter which is really close to the term 'brick wall filter'.

In the displays, the blue lines show the reponse of the filter. The red lines display the corner frequencies set in the main window. The phase response is scaled in degrees. In contrast to the WrapAlyzer, no phase wrapping is applied to keep the values in a range of $-180° \ldots +180°$ resp. $-\pi \ldots +\pi$, since the phase response is always linear for FIR-filters.

Further display elements are:

- FREQUENCY SCALE: see. Wrapalyzer , section 6.2.1.1
- SAMPLING FREQ. / MHZ: see. Wrapalyzer , section 6.2.1.1
- TAPS: number of filter taps. For this class of filters, the number of taps is always equal to the

length of the impulse response.

➢ GROUP-DELAY (SAMPLES): shows the delay in number of samples. It is also the number of samples required to move through the filter until it has reached a settled state from it's inital zero state on. The number is equal to the rounded down value of half the number of filter coefficients.

The window can be closed by clicking on the RED CROSSHAIRS at the upper right window corner.

6.2.1.3 Menu item 'set output-file appendix'

As stated in section 6.1.4, the output files name is always coupled to the name of the input file, except the files extension (e.g. .wv, .riq, …). Thus, the input file named ‚C:\Signals\Interferer.tiq' will be converted to ‚C:\Signals\Interferer.wv'. If one desired to alter that behavior (e.g.to create different variants of the same input), then one can use that menu item. The dialog window as shown in figure 7 is then shown on the screen.

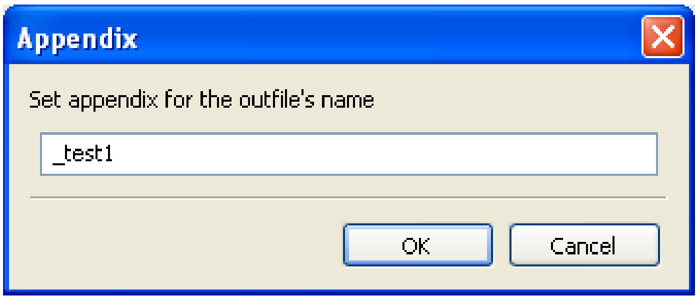

figure 7: editing the output file appendix

This dialog allows entering the characters which are appended at the end of the filename (but bevore the filenames extension). If one would enter '_test1', then the input file ‚C:\Signals\Interferer.tiq' will create an outputfile called ‚C:\Signals\Interferer_test1.tiq'.
If the window is closed by the means of the CANCEL-button, then no suffix is applied.

6.2.2 Menu 'Options'

This menu gives access to options that apply to program-wide behavior. The settings done here are read and written to the file 'IQConverter.ini'. The file is always in the IQConverter's current working directory (the directory, where the program is stored).
The presets – which apply when that configuration file is not found – are documented in the sections below.

6.2.2.1 Menu item 'Language'

The IQConverter fully supports internationalization. This means, by embedding more language files one can basically implement any desired language.

Languages supported so far:

➢ German [Preset]
➢ English

After a change to the language settings, the user is prompted to restart the program to apply the changes.

6.2.2.2 Menu item 'Preferred file format'

This menu item allows setting the file formats for the input / output filetype which are selected when the program is started. This menu item opens the dialogue shown in figure 8. Its settings can be applied by pressing the OK-Button. Pressing the Cancel-Button will discard the changes.

The changes will be applied as soon as the program is restarted.

The default settings are:

- *SOURCE FORMAT:* .tiq
- *TARGET FORMAT*: .wv

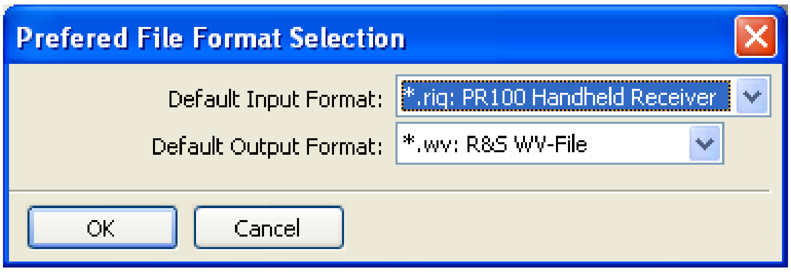

figure 8: dialogue window for setting the preset file formats

6.2.2.3 Menu item CSV-I/O-Options

This menu item allows setting the options for the output of csv-files, see figure 9. Further details about this topic are found in section 8.4.

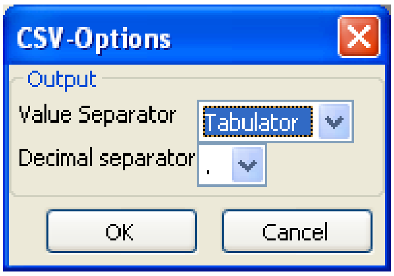

figure 9: options dialogue for CSV-files

6.2.3 Menu 'Generators'

This menu collects tools which can generate signal files independent of the actual file format.

Until now, that menu is rather a demonstration of possible further uses of the IQConverters software platform. Thus, there's only one menu item to be found here as a demonstrator.

6.2.3.1 Image2Signal

This menu item starts the dialogue window of the image to signal converter which is shown in figure 10. As stated above, this generator is to be understood rather as an example than a serious application.

The purpose of that generator is to convert an image into a signal using the following scheme:

1. The source image is loaded and scaled in its width to match the next larger FFT-width.
2. The source image is converter into a gray-scale image.
3. Each image line is interpreted as the real part of a FFT. The FFT-results imaginary part is set to zero. The columns within that image line are interpreted as a FFT-bin. This means that the brightness of one pixel determines the 'level' of the FFT-result.
4. In the style of the OFDM-principle, an IFFT is performed to create a time-domain signal for each line of the image.
5. That time signal is moved to the generator's output. Step 3-5 is repeated for every line of the image.
6. The sampling frequency of the signal is chosen to match one image line with the preset 'symbol' duration.

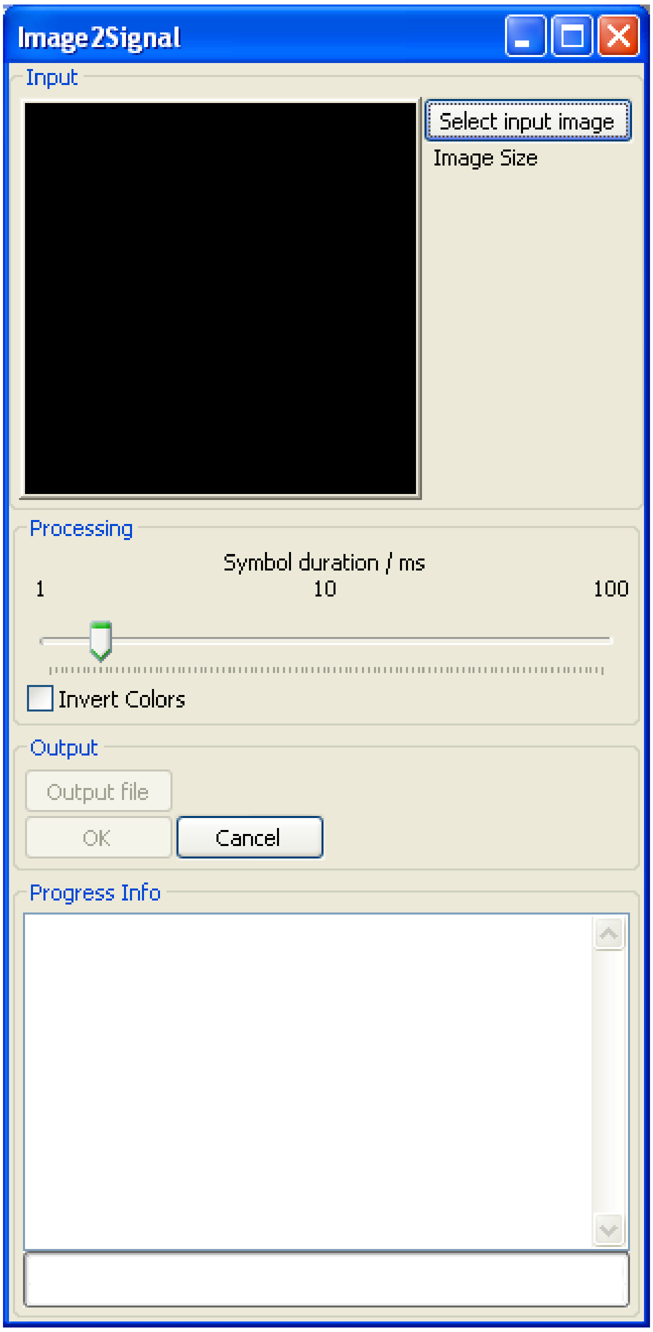

figure 10: the image2signal generator window

The generator's dialogue as shown above is divided into several parts: INPUT, PROCESSING and OUTPUT. The control elements have the following functions:

- Button SELECT INPUT IMAGE: opens a dialogue which requests the user to choose a JPG- or PNG-image to open. After that, a preview (not to scale) of the image is shown in the black image area. Furthermore, the image size is shown in the section IMAGE SIZE.
- Slider SYMBOL DURATION / MS: This sliders determines how long the time signal shall be valid for every single line of the image. The value in the middle of the three numbers reflects the current setting. The total adjustment range is shown as the leftmost / rightmost number above the slider.
- Checkbox INVERT COLORS: By default, bright pixels are mapped to high levels and vice versa. This

option allows to invert that behavior.

- Button OUTPUT FILE: opens a dialogue to select the output file.
- Button OK: starts processing.
- Button CANCEL: Cancels the operation and returns to the IQconverters main window.
- Textbox PROGRESS INFO: Shows messages from the generator about its current state.

6.2.4 Menu 'Help'

6.2.4.1 Menu item 'About'

Calling this menu item shows the programs version and contact information.

7 Data flow during conversion

This reference section shows the signal flow from the input to the output during a conversion process, see alsofigure 11.

The processing is organized in blocks. Thus, after the initial acquisition of signal parameters, data is read and written in a series of subsequent blocks.

During the acquisition step, the maximum absolute value in the source file is determined and taken as the base value for the normalization of all values. This ensures that the peak magnitude is always 1.0 (resp. 0 dBFS).

The processing blocks ‘swap IQ-components’ and ‘FIR-Filter’ are optional. They are created and added into the chain only on demand.

The delay block ensures that the filter does not write data into the output file when it hasn’t settled from its initial zeros state. Once the source file has reached its end, this block also ensures that the buffers are flushed to that point where the filter’s output ‘voltage’ starts to decrease due to the lack of input data.
The blocks ‘FIR-Filter’ and ‘Delay’ are two independent blocks, which are connected to the whole filter unit.

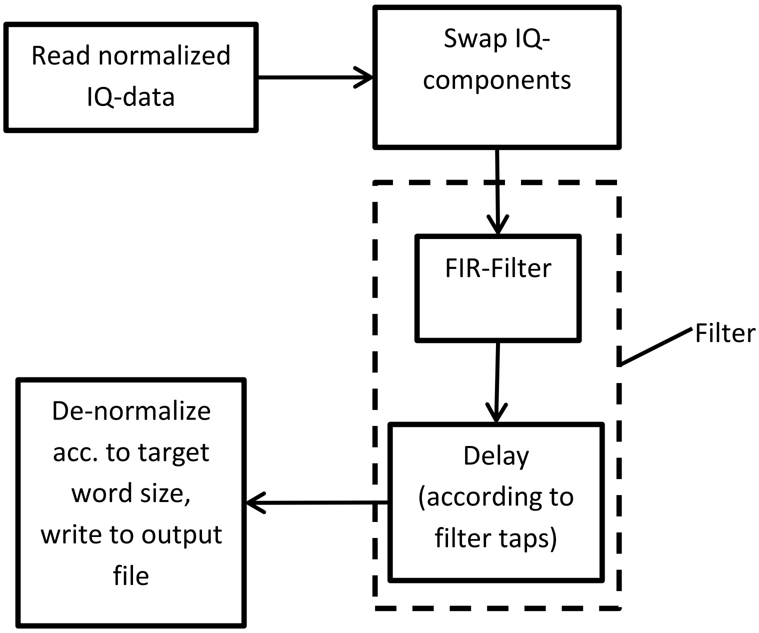

figure 11: block schematics of the converter engine

8 Known file formats

This section lists the parameters and particulars of the file formats implemented in this software.

8.1 WV (Rohde&Schwarz)

The .wv-format is used by generators of the series SMx, AMIx, … of Rohde&Schwarz (just to name a few). The format is documented in:

Röder, Thomas: Rohde&Schwarz Application Note 1GP62_2E: Importing Data in ARB, Custom Digital Modulation and RF List Mode, März 2007

Word width: 16 Bit, signed integer

Note: the IQConverter doesn't care about the output file's size! Sometimes, files can become too large for the generators memory.

Files which were generated with certain options for the R&S-generators are often marked with a so-called 'k-option'. Those files are encrypted. The IQConverter will not process those files.

8.2 RIQ (Rohde&Schwarz)

The .riq-format of Rohde&Schwarz is used (as an example) by the portable monitoring receiver PR100. It resembles the format of the EB200. This format is capable to transport several result types, depending on the receiver's operation mode (PScan, MScan, Trace, …). The IQConverter will process only files which contain the so called IF-data-blocks, for only these ones contain IQ-data.

No testing was done regarding to mixed data-streams (for example, a combination of IF- and PScan streams).

The IQConverter should be capable to cope with all IF-datasets which are marked with the so-called Major-Number 2.

The format is documentd in:

Rohde&Schwarz: R&S PR100 Portable Receiver – Manual, V3.1, English edition. Chapter 14: Data structure recorded files and chapter 13: UDP Data Streams

Word width: 16 Bit, signed integer, Little Endian

8.3 TIQ (Tektronix)

This format is used by (as an example) analyzers of the RSA-series.

The format is quite self-explanatory due to its xml-header.

Word width: 32 Bit (only 24 Bit are actually used), signed integer, Little Endian

Note: the IQConverter has only been tested for TIQ's which contain only one acquisition and no DPX-Spectra.

8.4 CSV (self defined)

This for internal or temporary use intended format is text-based. Its main use is the import / export of simple data sets.

Those files always start with a simple header. The header describes not only the signal's sampling frequency and it's number of samples, it also tries to solve some common problems assorted with csv-files.

The files are structured as shown in the example below, where lines beginning with'#' denote comments.

The entry 'DecimalSeparator' denotes the character used to separate the integer and decimal parts of a number. Possible characters are the point ('.') and the comma (',').

The character used to separate the I- and Q-values from each other is named 'ValueSeparator'. Possible characters are the tabulator and the space character.

After the header, the start of the data stream is marker with the line '#START'.

```
#IQ-Datafile
#SamplingFrequencyHz 195312
#Samplecount 1000
#DecimalSeparator .
#ValueSeparator
#START
-0.764472338964	0.638413121138
-0.764656436247	0.639640960894
-0.764312250967	0.639764671608
```

Users of that format can easily remove or insert that kind of header.
Furthermore, the options menu allows to change the settings for the decimal- and value separator.

If the IQConverter reads CSV-files, the it always takes the information stored in the header.

8.5 BIN (self defined)

The .bin-format is a simple format intended for fast import and export out of other formats. Also, one could use it as an intermediate format if a conversion process needs more than one run.

Format: the files header is written in ACSII. It contains two pointed brackets, denoting the sampling frequency in Hertz and the number of samples (IQ-pairs) contained in the file.

In the following example, the header marks a sampling frequency of 100 kHz and 12345 values:

<100000><12345>

Right after this header, the binary data stream starts. The samples are always given in the order 'IQIQIQIQIQ'.

Word width: 32 Bit, signed, Little Endian

8.6 WAV (microtelcom Perseus)

The files written by that receiver is built on top of the Microsoft© Wave format. Those files have an additional chunk in their header, the so called Rcvr-Chunk. It contains some information on the recorded file regarding the Perseus receiver, like the attenuator setting, center frequency and so on.

There's no official documentation known of that format. One can only find a comment about it in the Perseus Newsgroup on Yahoo.

Other types of WAV-files are not supported by the IQConverter's WAV-interpreter.

Word width: 24 Bit, signed integer, Little Endian

9 Internals and references to open source packages used

This program was developed using Python in combination with the wxWidgets-library for the GUI-elements. Basically, the software should be able platform independent (in the source version, the supplied compiled variant is for Microsoft Windows © only).

The IQConverter uses the following Open-Source parts:

- Python® (Language core)
  Python is a registered trademark of the Python Software Foundation.
- wxPython (GUI)
  wxPython is a Python-wrapper for wxWidgets and is published inder the wxWidgets-License.
- PIL (Python Imaging Library, for reading images)
- Numpy (fast numeric and arrays)
- GNU Gettext (Internationalization)
- Dispatcher (messaging between components in an observer-like fashion)
- pyInstaller (compilation of executable binary files). License model: GPL [GNU Public License])

The author is not aware of the necessity for mentioning other packages and their assorted licenses.

The licenses of the packages mentioned above are bundled with the provided software.